



for a greener tomorrow



**MITSUBISHI  
ELECTRIC**

*Changes for the Better*

FACTORY AUTOMATION

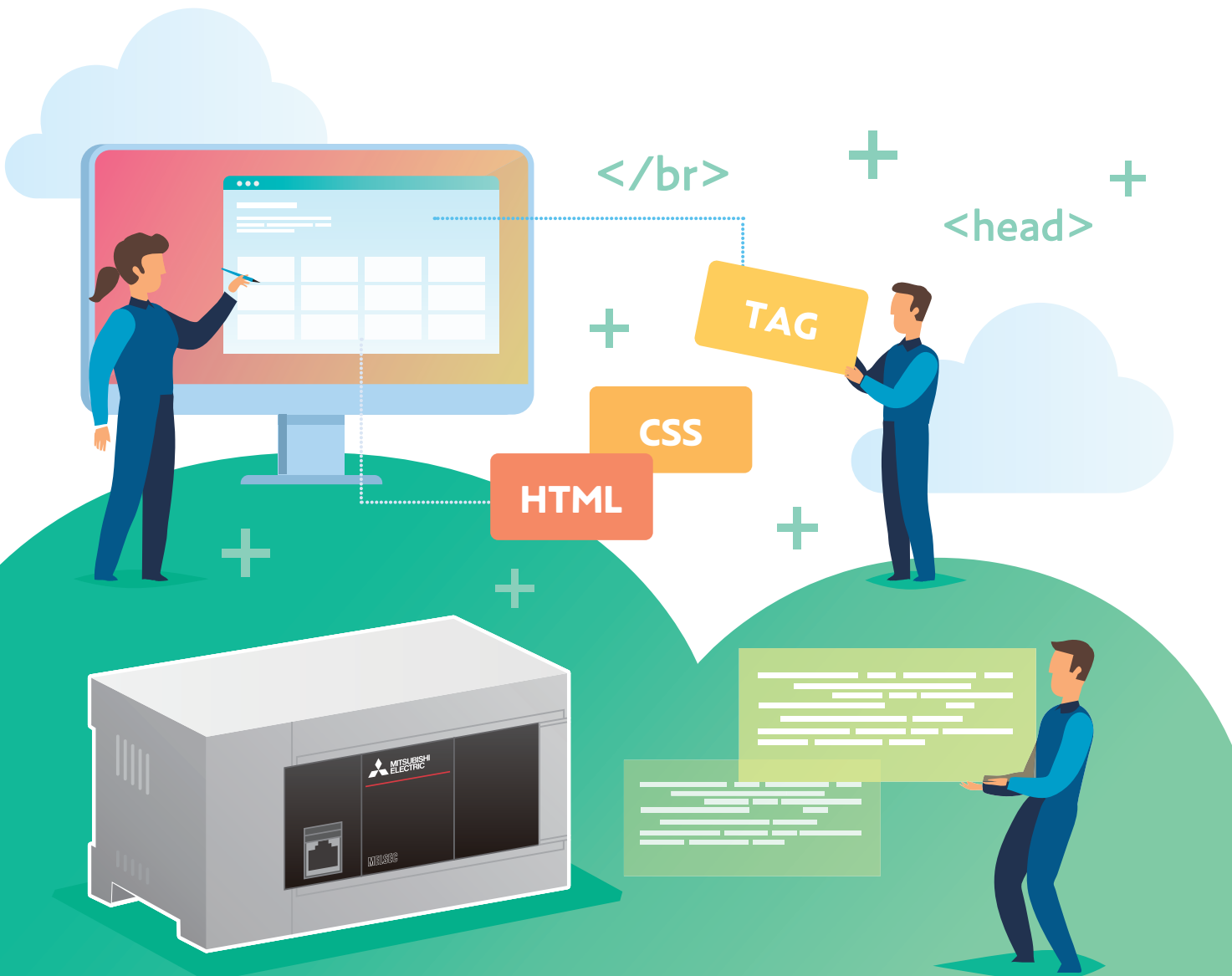
Programmable Controller  
MELSEC iQ-F Series

**MELSEC iQ-F**  
series

# Web Server Function Application Guide

## Using Web Page

### User Web Page HTML Creation



# INTRODUCTION

Thank you for purchasing the MELSEC iQ-F Series.

This guide describes the settings related to the Web server function of the FX5 CPU module.

It should be read and understood before attempting to install or use the module.

## Regarding use of this product

- This product has been manufactured as a general-purpose part for general industries, and has not been designed or manufactured to be incorporated in a device or system used for purposes related to human life.
- Before using the product for special purposes such as nuclear power, electric power, aerospace, medicine, or passenger movement vehicles, consult Mitsubishi Electric representative.
- This product has been manufactured under strict quality control. However, when installing the product where major accidents or losses could occur if the product fails, install appropriate backup or failsafe functions in the system.

## Note

- If in doubt at any stage during the installation of the product, always consult a professional electrical engineer who is qualified and trained in the local and national standards. If in doubt about the operation or use, please consult the nearest Mitsubishi Electric representative.
- Mitsubishi Electric will not accept responsibility for actual use of the product based on these illustrative examples.
- This guide content, specification, etc., may be changed, without a notice, for improvement.
- The information in this guide has been carefully checked and is believed to be accurate; however, if you notice a doubtful point, an error, etc., please contact the nearest Mitsubishi Electric representative. When doing so, please provide the document number given at the end of this guide.

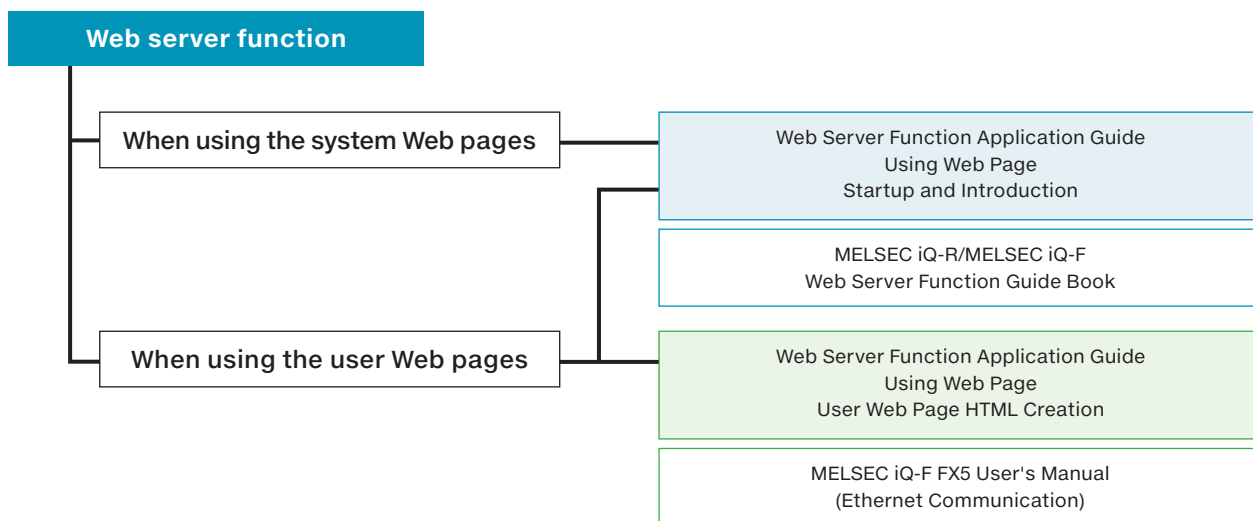
# TABLE OF CONTENTS

INTRODUCTION .....	2
RELEVANT CATALOGS/MANUALS .....	4
PRECAUTIONS .....	5
LIMITATIONS .....	5
<b>1. BASICS OF CREATING HTML &lt;TAGS&gt;</b> .....	<b>6</b>
1.1 About HTML .....	6
1.2 HTML Structure .....	7
1.3 Basic Rules About Tags .....	9
1.4 Tag Types .....	10
1.5 Naming Files .....	12
<b>2. CREATING AND DISPLAYING SIMPLE HTML DOCUMENTS ON A PERSONAL COMPUTER</b> .....	<b>13</b>
2.1 Preparation .....	13
2.2 HTML File Creation .....	13
2.3 Web Browser Display .....	15
2.4 Displaying HTML Source in the Web Browser .....	16
<b>3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE</b> .....	<b>17</b>
3.1 Before Starting .....	17
3.2 Preparation .....	19
3.3 Reuse .....	21
3.4 User Web Page Creation .....	26
3.5 File Save Destination .....	59
3.6 Troubleshooting .....	62
<b>4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)</b> .....	<b>64</b>
4.1 Function Introduction .....	64
4.2 Common Items .....	65
4.3 Overall Structure (UserWebStyle.css File) .....	67
4.4 Setting Details .....	67
4.5 Style Sheet Details .....	68
<b>5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)</b> .....	<b>75</b>
5.1 What Is JavaScript (JS)? .....	75
5.2 Common Items .....	77
5.3 Table .....	81
5.4 Graphs .....	87
5.5 Image/Figure .....	98
5.6 Operation Buttons .....	107
<b>6. CREATING A DEVICE MONITOR WINDOW WITH CGI</b> .....	<b>113</b>
6.1 What Is CGI? .....	113
6.2 Overview and Functions of CGI .....	113
6.3 Common Specifications .....	116
6.4 Device Reading Web Page Creation .....	118
6.5 Device Writing Web Page Creation .....	125
<b>7. REFERENCES</b> .....	<b>132</b>
7.1 Style Sheet References .....	132
7.2 HTML Tag References .....	137
7.3 Color Name/Color Code .....	143

# RELEVANT CATALOGS/MANUALS

## ► Configuration of catalogs and manuals

The following catalogs and manuals should be referred to before using the Web pages.



## ► Relevant catalogs/manuals

For the detailed specifications, precautions, and limitations of the product, refer to the following catalogs or manuals.

Name [Catalog/manual number]	Explanation	Available form
Web Server Function Application Guide Using Web Page Startup and Introduction [L(NA)08643ENG]	Preparation of necessary equipment for the use of the Web server function of the MELSEC iQ-F and the introduction procedure for displaying the Web page.	PDF
Web Server Function Application Guide Using Web Page User Web Page HTML Creation [L(NA)08645ENG] (This guide)	For users who create their own Web page from the user Web page. Details on Style Sheet-based designs, JavaScript objects, and CGI objects used during HTML creation.	PDF
MELSEC iQ-R/MELSEC iQ-F Web Server Function Guide Book [SH-081982ENG]	Specifications, procedures before operation, and troubleshooting of the Web server function.	e-Manual, PDF
MELSEC iQ-F FX5 User's Manual (Ethernet Communication) [JY997D56201]	Details on the Ethernet communication function. For the Web server function, details on CGI objects that can be used in the user Web page.	e-Manual, PDF, Print book

### Point

- e-Manual refers to the Mitsubishi Electric's FA electronic book manuals that can be browsed using a dedicated tool. e-Manual has the following features:
- Required information can be cross-searched in multiple manuals. (Cross Document Search)
  - Pages that users often browse can be bookmarked.

# PRECAUTIONS

## Illegal access from external devices

Incorporate other measures if the safety of the programmable controller system must be maintained against illegal access from an external device. Mitsubishi shall not be held liable for any system problems that may occur due to illegal access. The user authentication of the Web server function is one of the methods for preventing illegal access (such as program or data destruction) from an external device. It does not completely prevent illegal access.

Examples of measures against illegal access are given below.

- Install a firewall.
- Install a personal computer as a relay station, and control the relaying of sent/received data with an application program.
- Install an external device for which the access rights can be controlled as a relay station (contact the network provider or equipment dealer for details on the external devices for which access rights can be controlled).

## Firewall functions

### • Filtering function

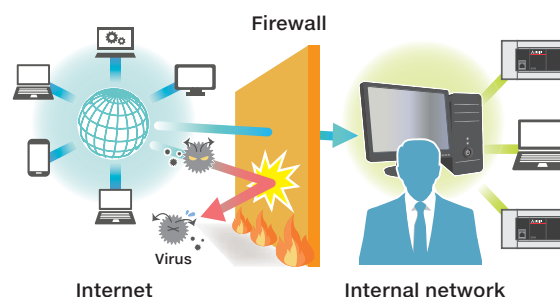
This function intercepts illegal packets and passes permitted packets only.

### • Address conversion function

This function reciprocally assigns IP addresses between the internal and external networks.

### • Remote control and monitoring function

This function can be used to set a firewall and check logs from another computer.



# LIMITATIONS

## Response performance of Web pages

The response performance of Web pages varies according to the following causes.

The response performance may degrade due to the network line status or the communication status of the Ethernet function of the FX5 CPU.

- Load factor of an Ethernet line (line congestion)
- Number of connections that are used simultaneously (refreshing other connections)
- Communication load status including the communication functions other than the Web server function set using the Ethernet function of the FX5 CPU
- Large scan time of the FX5 CPU when Web page update is requested

## Web server access

Up to four users can log in to the MELSEC iQ-F Series simultaneously.

If the file reading was not completed successfully or a message that requests to reload the page appears while the Web server function is used, read the file again.

## Communication timeout time

Since the communication timeout time depends on the browser specifications, it may not be the same as the timeout time of the Web server (five minutes). The design of the error window differs depending on the browser.

**Special Note** For the precautions and limitations, refer also to "FX5 User's Manual (Ethernet Communication) [JY997D56201]".

# 1. BASICS OF CREATING HTML <TAGS >

[About HTML](#)
[HTML Structure](#)
[Basic Rules About Tags](#)
[Tag Types](#)
[Naming Files](#)

## 1.1 About HTML

### What Is HTML?

HTML is an abbreviation for “Hyper Text Markup Language”. It is a language developed for creating Web pages. A great number of Web pages currently accessible on the Internet have been created with HTML and can be viewed with Web browsers (such as Internet Explorer® and Google Chrome®).

A Web browser reads HTML files and image files, organizes the contents into an easily readable form, and displays them as a Web page.

The source of the HTML file of the Mitsubishi Electric site can be displayed in Internet Explorer® with the following procedure.

Methods for displaying the source (operation example using Internet Explorer®)

Right-click the Internet Explorer® window and select [View Source].

Select [View] - [Source] on the toolbar.

Press the [F12] key.

Any of the methods listed on the left will display the source at the bottom of the Web browser window.

Mitsubishi Electric site

The screen shows a display example when Internet Explorer® is used.

The screenshot shows the Mitsubishi Electric website for the MELSEC IQ-F Series. The page content is highlighted with an orange box. Below the website screenshot, the Internet Explorer Developer Tools (F12) are open, showing the DOM Explorer with the HTML source code of the highlighted content. The source code is also highlighted with an orange box. A callout box points to the source code with the text "Source display of HTML file".

```

625 <div id="mel_fa_main_area">
626 <h1>
627 MELSEC IQ-F Series</h1>
628
629 <div class="clearfix">
630 <div class="imageHalf">
631 
632 <p>The Mitsubishi Electric MELSEC-F series has undergone many advancements, making way for the next
633 generation MELSEC IQ-F Series, with enhanced high speed bus, expanded built-in functions, advanced SSCNET III/H
634 support, and improved engineering environment with parameter settings in GX Works3 engineering software.</p>
635 <p>From standalone use to networked system application, MELSEC IQ-F takes your business to the next level.</p>
636 <p style="text-align: right;">
637 <a href="/fa/en/products/controllers/programmable-controllers-melsec/iqf_series/cpu-module/overview"
638 title="Open" class="readMore" style="white-space: pre;" target="_self"><strong>IQ-F Series Overview</strong></a></p>
639 <p style="text-align: right;">
640 <a href="https://www.youtube.com/embed/Lf1a1D0C56w?width=854&height=480;autoplay=1" class="readMore
641 fancybox"><strong>IQ-F Overview Video</strong></a></p>
642 </p>
643 </div>
644 </div><strong><strong><strong><strong><strong><strong></div>
  
```

HTML rules consist of a simple structure and character strings. HTML can be created easily.

# 1. BASICS OF CREATING HTML <TAGS>

About HTML

HTML Structure

Basic Rules About Tags

Tag Types

Naming Files

1

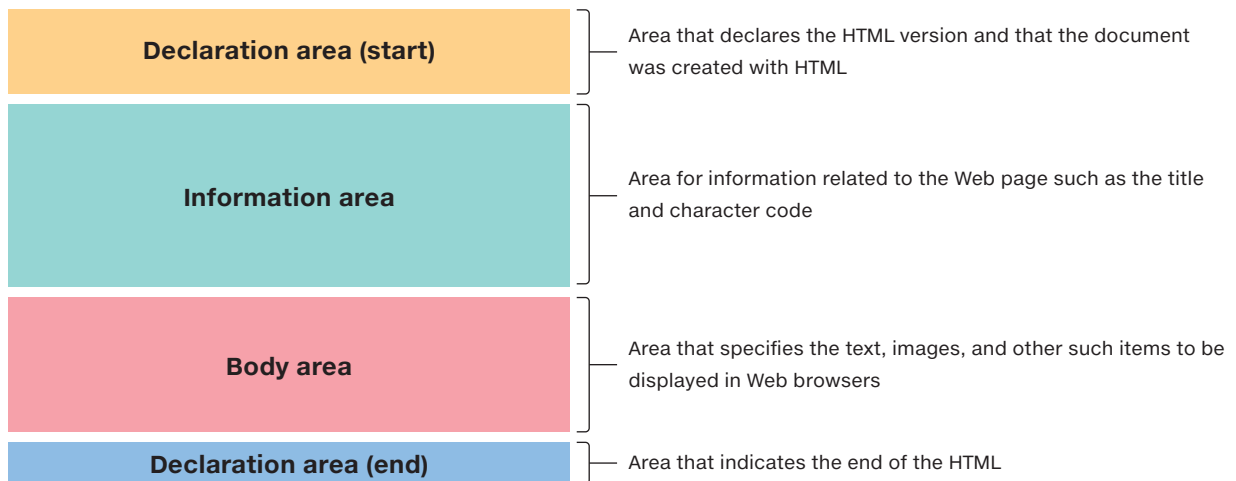
BASICS OF CREATING HTML <TAGS>

## 1.2 HTML Structure

Creating a Web page requires a language for writing the Web page, and there are various languages other than HTML such as XHM, SGML, and XHTML.

### Structure of HTML and meaning of each area

The figure below illustrates the basic structure of HTML and the meaning of each area.



### HTML example

This example is a configuration that contains the minimum number of tags to write in an HTML file. This is a legal configuration, so this file can be displayed in a Web browser.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Web Server Page</title>
  </head>
  <body>
    Images and document contents
  </body>
</html>
```

The screenshot shows a Notepad window with the following HTML code. The code is annotated with colored boxes and labels:

- Declaration area (start)**: A yellow box highlights the `<!DOCTYPE html>` and `<html lang="en">` lines.
- Information area**: A teal box highlights the `<head>` section, including `<meta charset="UTF-8">` and `<title>Web Server Page</title>`.
- Body area**: A pink box highlights the `<body>` section, containing the text "Images and document contents".
- Declaration area (end)**: A blue box highlights the closing `</html>` tag.

2

CREATING AND DISPLAYING SIMPLE HTML DOCUMENTS ON A PERSONAL COMPUTER

3

USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

4

LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

# 1. BASICS OF CREATING HTML <TAGS >

[About HTML](#)
[HTML Structure](#)
[Basic Rules About Tags](#)
[Tag Types](#)
[Naming Files](#)

## Elements of HTML documents

An HTML document consists of many elements. The following table lists some basic elements.

<code>&lt;!DOCTYPE html&gt;</code>	<p><b>DOCTYPE declaration</b></p> <p>This tag is written at the beginning of an HTML document to declare the version of HTML. <code>&lt;!DOCTYPE html&gt;</code> means that “This HTML is written in HTML5”.</p>
<code>&lt;html&gt;</code> : Start tag <code>&lt;/html&gt;</code> : End tag	<p><b>html element</b></p> <p>This tag pair indicates that the language of this document is HTML. All elements should be written within the range between <code>&lt;html&gt;</code> and <code>&lt;/html&gt;</code>.</p>
<code>&lt;head&gt;</code> : Start tag <code>&lt;/head&gt;</code> : End tag	<p><b>head element</b></p> <p>The information about the entire document, such as meta information and title information, is written within the range between <code>&lt;head&gt;</code> and <code>&lt;/head&gt;</code>.</p>
<code>&lt;meta&gt;</code>	<p><b>meta element</b></p> <p>Meta information (meta data) means “the information about information”. This tag indicates the information about this HTML document. <code>&lt;meta charset="UTF-8"&gt;</code> means that “This HTML document is written with the character code UTF-8”.</p>
<code>&lt;title&gt;</code> : Start tag <code>&lt;/title&gt;</code> : End tag	<p><b>title element</b></p> <p>This tag pair specifies the title to be displayed in the title bar of the Web browser.</p>
<code>&lt;body&gt;</code> : Start tag <code>&lt;/body&gt;</code> : End tag	<p><b>body element</b></p> <p>This tag pair specifies the text, images, and tables to be displayed on the Web page. Only the contents within the range between <code>&lt;body&gt;</code> and <code>&lt;/body&gt;</code> are displayed on the Web page.</p>

## Entering tags with alphabet characters

The characters used for HTML tags are not case-sensitive, so `<html>` and `<HTML>` mean the same. In this guide, tags are written in lowercase.

`<body>`      `<BODY>`



# 1. BASICS OF CREATING HTML <TAGS>

About HTML

HTML Structure

Basic Rules About Tags

Tag Types

Naming Files

## 1.3 Basic Rules About Tags

In HTML, all information is written with “tags”. There are tags specified for a wide variety of items including paragraphs, line breaks, and images.

This section explains the basic way of using tags.

### What is a tag?

A tag is a symbol or a mark such as <html> or <head> that is used for creating HTML. The way to use tags is determined by the rules of HTML. A document that is written according to the rules of HTML is called an “HTML document”. HTML is provided in several versions including “HTML1.0” and “HTML4.01”, each of which has different specifications for the writing format and the available tags.

### Structure of tags

Tags can broadly be separated into the two types shown below.

Classification	Format	Description
Nestable tag	<tag name> to </tag name>	The written details are nested in the tag. The <tag name> tag is called the start tag and the </tag name> tag is called the end tag.
Singular tag	<tag name>	Some tags are used in a stand-alone manner. (Examples: <meta> tag, <img> tag, and others)

It is also possible to create a layered structure by writing tags within the contents of other tags. Note that although HTML is basically written as a layered structure of tags, the deeper the layers, the more difficult the document is to read.

### ■ Nested tag structure

The screenshot shows a Notepad window with the following HTML code:

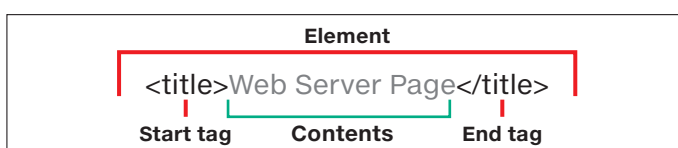
```
<html lang="ja">
  <head>
    <meta charset="UTF-8">
    <title>Web Server Page</title>
  </head>
  <body>
    Images and document contents
  </body>
</html>
```

Callouts in the image indicate the following ranges:

- Range of the title tags (around <title>Web Server Page</title>)
- Range of the head tags (around the entire <head>...</head> block)
- Range of the body tags (around the entire <body>...</body> block)
- Range of the html tags (around the entire <html>...</html> block)

### Tags and elements

An “element” refers to the entire content enclosed by the start tag and end tag.



### Point

Single-byte spaces after the start tag and before the end tag are ignored. Also, two or more consecutive single-byte spaces within the tag contents are handled as if they were just one single-byte space.

# 1. BASICS OF CREATING HTML <TAGS >

About HTML

HTML Structure

Basic Rules About Tags

Tag Types

Naming Files

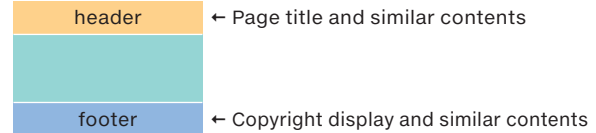
## 1.4 Tag Types

This section introduces tags that are commonly used with HTML.

### Tags for writing the page structure

<code>&lt;header&gt;</code> <b>Header</b>	Writes the contents that are displayed at the top of the Web page.
<code>&lt;footer&gt;</code> <b>Footer</b>	Writes the contents that are displayed at the bottom of the Web page.

Web page



### Tags for writing the document

<code>&lt;h1&gt; to &lt;h6&gt;</code> <b>Headings</b>	<p>The <code>&lt;h1&gt;</code> to <code>&lt;h6&gt;</code> tags are used to write the document headings. The larger the number in the tag, the smaller the font size of the heading.</p> <p><b>HTML document</b></p> <pre>&lt;h1&gt;h1 heading&lt;/h1&gt; &lt;h2&gt;h2 heading&lt;/h2&gt; &lt;h3&gt;h3 heading&lt;/h3&gt; &lt;h4&gt;h4 heading&lt;/h4&gt; &lt;h5&gt;h5 heading&lt;/h5&gt; &lt;h6&gt;h6 heading&lt;/h6&gt;</pre> <p><b>Web browser display</b></p> <pre>h1 heading h2 heading h3 heading h4 heading h5 heading h6 heading</pre>
<code>&lt;p&gt;</code> <b>Paragraph</b>	<p>The <code>&lt;p&gt;</code> to <code>&lt;/p&gt;</code> tags are used to write a paragraph in the document.</p> <p><b>HTML document</b></p> <pre>&lt;p&gt;Paragraph 1&lt;/p&gt; &lt;p&gt;Paragraph 2&lt;/p&gt;</pre> <p><b>Web browser display</b></p> <pre>Paragraph 1 Paragraph 2</pre>
<code>&lt;br&gt;</code> <b>Line break</b>	<p>Long sentences automatically breaks at the right edge of the display area, but this tag can be used to insert a line break at the specified location.</p> <p><b>HTML document</b></p> <pre>The line break position can be specified as desired with the &lt;br&gt; tag.</pre> <p><b>Web browser display</b></p> <pre>The line break position can be specified as desired with the tag.</pre>

### Point

Line breaks within an HTML file are ignored. For an easy-to-understand display, use the `<p>` and `<br>` tags effectively.

# 1. BASICS OF CREATING HTML <TAGS>

About HTML

HTML Structure

Basic Rules About Tags

Tag Types

Naming Files

1

BASICS OF CREATING HTML <TAGS>

2

CREATING AND DISPLAYING SIMPLE HTML DOCUMENTS ON A PERSONAL COMPUTER

3

USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

4

LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

11

Use the <table> tag to create a table. Use the <tr> and <td> tags to write the elements in the table, as shown in the following example. For tag details, refer to Table in 7.2 HTML Tag References.

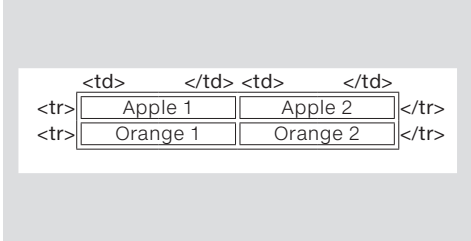
HTML document

<table>  
Table

```
<table>
<tr>
<td>Apple 1</td><td>Apple 2</td>
</tr>
<td>Orange 1</td><td>Orange 2</td>
</tr>
</table>
```

Web browser display

```
<td> </td> <td> </td>
<tr>
<tr>
```



## Other tags

<img>  
Displays  
an image

To display an image, use the <img> tag to specify the image file. Commonly used image file formats are PNG, JPEG, and GIF. (PNG is the recommended file format.)

Syntax: 

HTML document

```

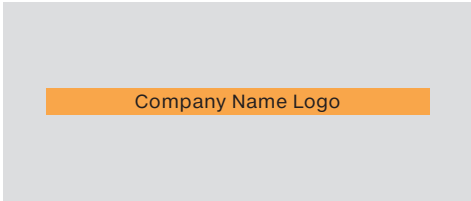
```



company-logo.png

Web browser display

Company Name Logo



<a>  
Anchor  
(for  
jumping  
to a link  
destination)

It is possible to jump to a different page by clicking part of a sentence or an image.

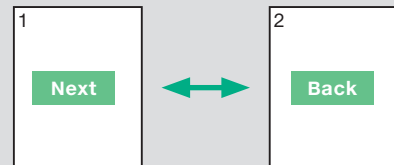
Syntax: <a href="link destination URL">text to click to jump to the link destination</a>

HTML document

```
<a href="2.html">Next</a>
```

```
<a href="1.html">Back</a>
```

Web browser display



Related Page 7.2 HTML Tag References

# 1. BASICS OF CREATING HTML <TAGS>

About HTML

HTML Structure

Basic Rules About Tags

Tag Types

Naming Files

## 1.5 Naming Files

### Characters that can be used for file names

The characters listed in the table below can be used.

Type	Character
Single-byte alphabet	a b c d e f g h i j k l m n o p q r s t u v w x y z
Single-byte number	0 1 2 3 4 5 6 7 8 9
Single-byte symbol (limited)	- (hyphen), _ (underscore)

### Naming Files

The name of the HTML file that is used as the top page of user Web pages must be "index.html". The HTML files used for other pages can be named as desired.

Ex. menu.html  
 Name Extension

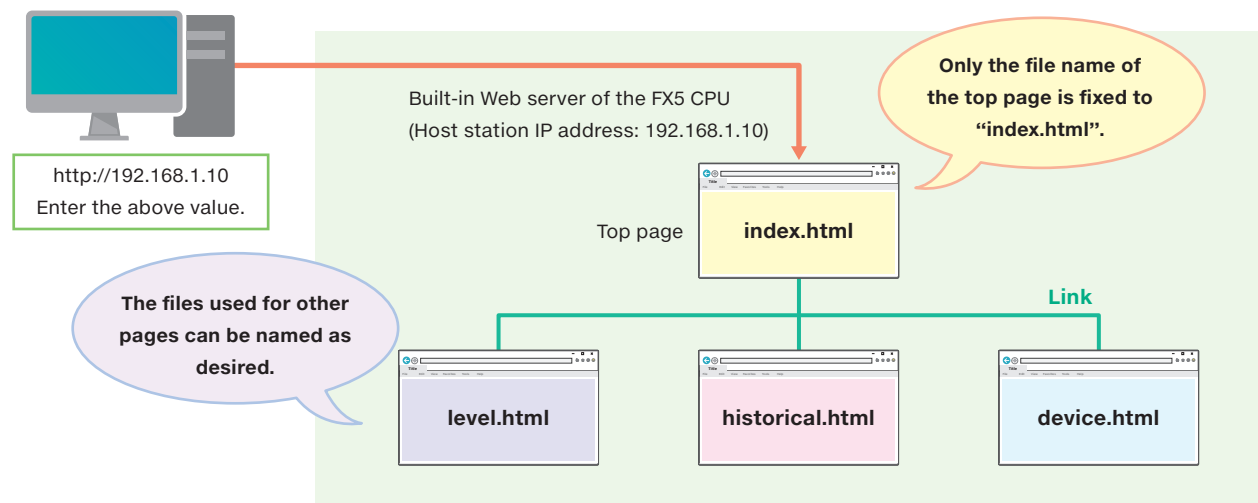
#### Further information

#### Why are there two file types: .html and .htm?

The normal extension for HTML files is ".html", but personal computers that have old operating systems (MS-DOS® and Windows® 3.1) could only handle extensions with up to three characters, so the extension ".htm" was used instead of ".html" and the extension ".jpg" was used instead of ".jpeg". This is why the .html and .htm extensions both exist. Both .html and .htm can be used, but only one of these extensions can be used according to the Web server setting specifications.

### ► File access mechanism

In the address bar of the Web browser, enter the IP address of the FX5 CPU module specified in the parameter to access the Web server and call "index.html" automatically.



#### Point

Use easy-to-understand file names that match the display contents. Complicated file names and serial numbers can be easily mistaken when entering URLs directly or setting links.

## 2. CREATING AND DISPLAYING SIMPLE HTML DOCUMENTS ON A PERSONAL COMPUTER

Preparation

HTML File Creation

Web Browser Display

HTML Source Display

Use Notepad in Windows® to create an HTML file and display it in a Web browser (Internet Explorer®). Moreover, learn the method of using a Web browser to check the HTML (source) of the displayed Web page.

### 2.1 Preparation

Prepare a personal computer running Windows® that complies with the following specifications.

Personal computer running Windows®	Microsoft® Windows® 10 Microsoft® Windows® 8 Microsoft® Windows® 7
Required software (Standard software included in Windows®)	Notepad ..... For creating HTML files Internet Explorer 11® Web browser ..... For displaying and checking created HTML files


### 2.2 HTML File Creation

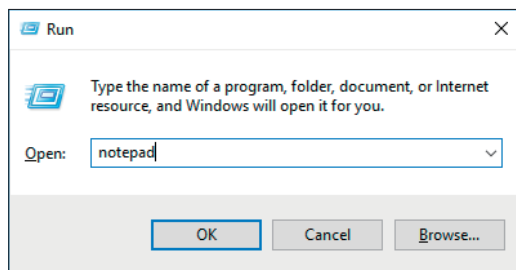
This section uses an example in Windows® 10 to describe the procedure of using Notepad, which is included in Windows® as standard, to actually create an HTML file and display it in a Web browser.

#### Opening Windows® Notepad

Open Notepad by any of the following methods.

##### ▶ Using a shortcut key

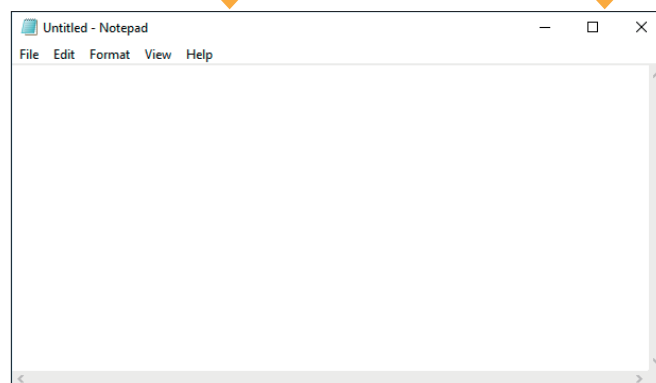
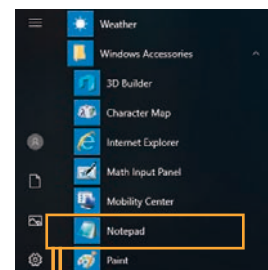
Press the [Windows® logo ] + [R] keys simultaneously. ▶  
Change “control” to “notepad”. ▶ Click [OK].



\*: The window examples are from Windows® 10.

##### ▶ From the Start menu

Click Notepad under Windows® Accessories.



Notepad window

## 2. CREATING AND DISPLAYING SIMPLE HTML DOCUMENTS ON A PERSONAL COMPUTER

Preparation

HTML File Creation

Web Browser Display

HTML Source Display

### Entering the HTML document

HTML is composed entirely of text, so Web pages can be created and edited through the use of a general-purpose text editor.

Enter the HTML document on the right in Notepad.

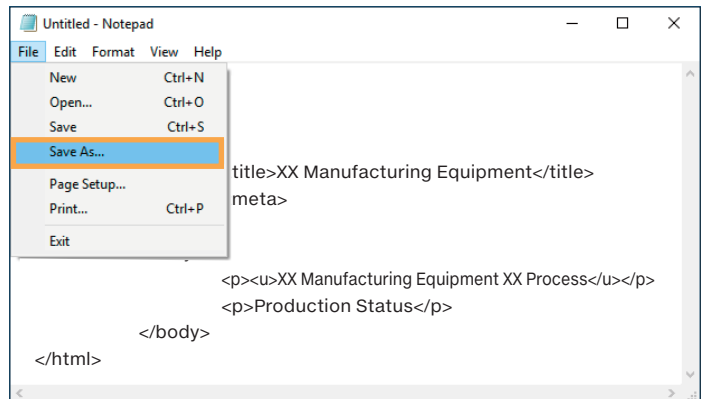
```

<!DOCTYPE html>
<html>
<head>
<title>XX Manufacturing Equipment</title>
<meta>
</head>
<body>
<p><u>XX Manufacturing Equipment XX Process</u></p>
<p>Production Status</p>
</body>
</html>

```

### Saving the created HTML document into a file

[Toolbar: File] ▶  
Click [Save As].

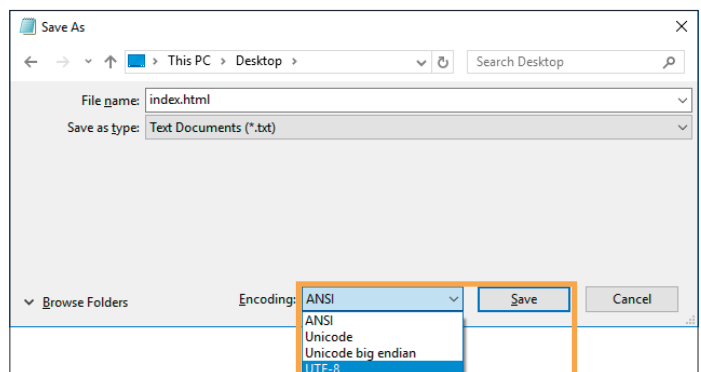


Set the name of the HTML file of the top page to "index.html". ▶  
Select [UTF-8] for the Encoding. ▶  
Click [Save].

**index . html**

↑            ↑  
File name    Extension

Save the top page with the file name "index.html". Pages other than the top page can be named as desired, but use the extension "html".



## 2. CREATING AND DISPLAYING SIMPLE HTML DOCUMENTS ON A PERSONAL COMPUTER

Preparation

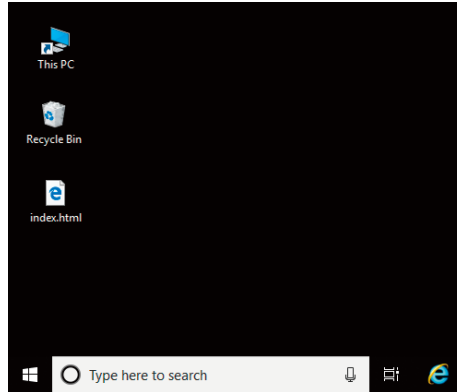
HTML File Creation

Web Browser Display

HTML Source Display

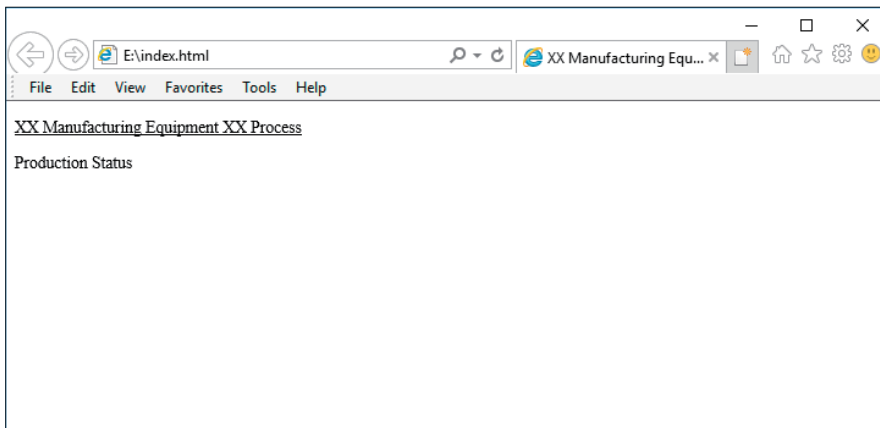
### 2.3 Web Browser Display

Double-click the “index.html” file saved in the previous step.



#### ► Web browser display window

Internet Explorer® starts automatically and displays the Web page.



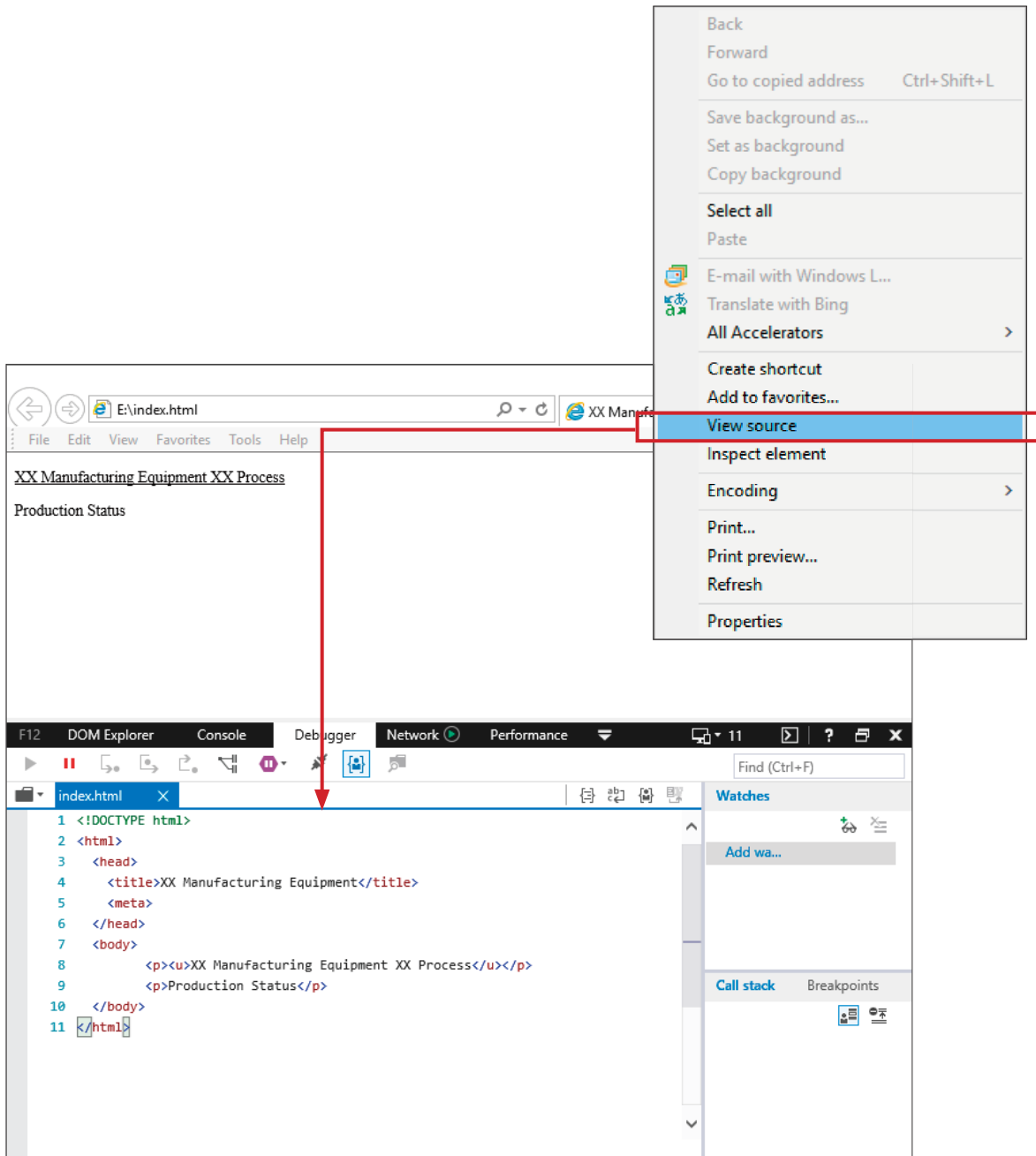
## 2. CREATING AND DISPLAYING SIMPLE HTML DOCUMENTS ON A PERSONAL COMPUTER

Preparation | HTML File Creation | Web Browser Display | **HTML Source Display**

### 2.4 Displaying HTML Source in the Web Browser

Use one of the following methods to display the line numbers and HTML document at the bottom of the Web browser window.

- Right-click the Internet Explorer® window and select [View Source].
- Select [View] - [Source] on the toolbar.
- Press the [F12] key.





# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

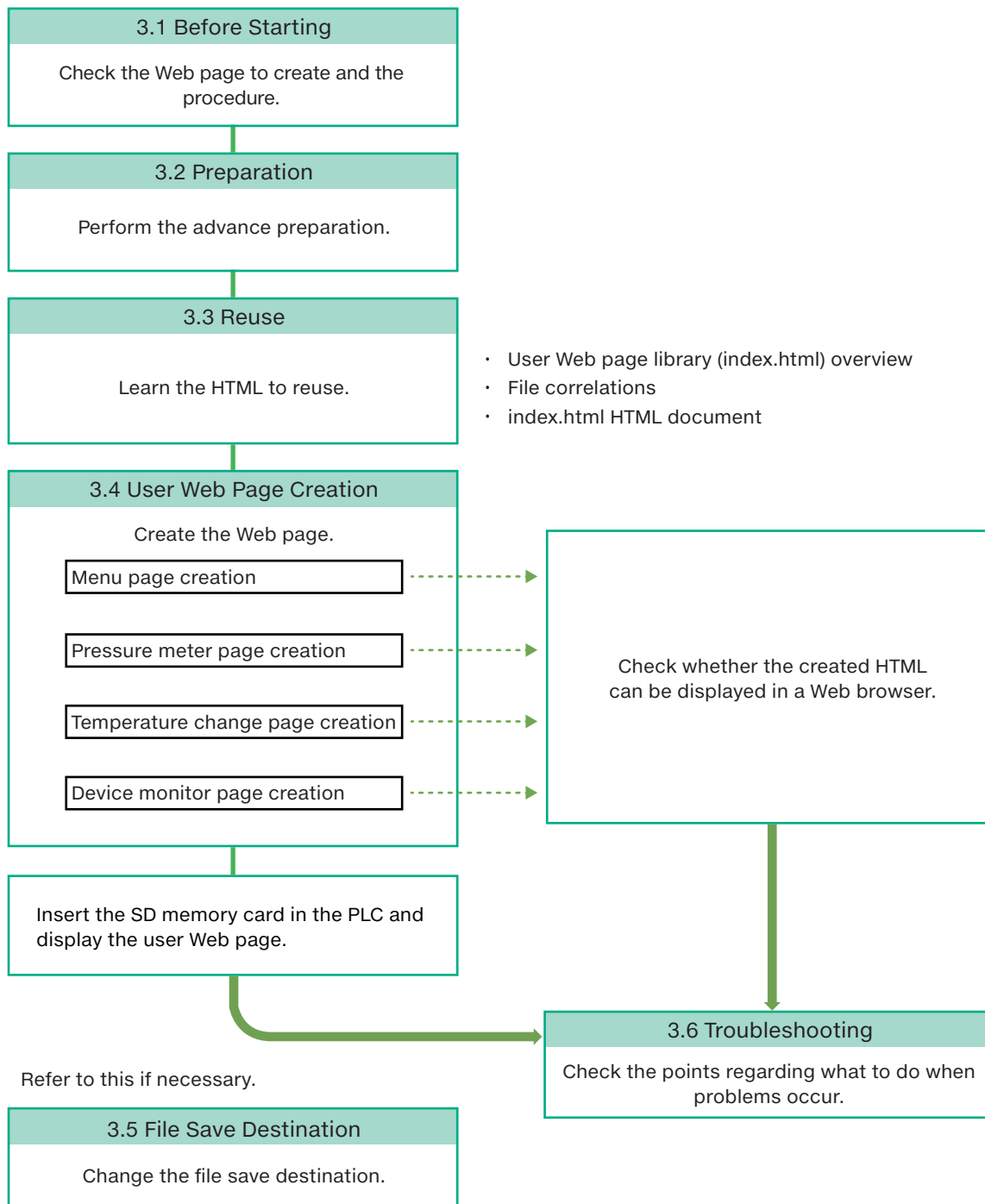
Troubleshooting

This chapter describes how to create a user Web page by using HTML files (the Web server function library) that can be obtained from the Mitsubishi Electric representative.

## 3.1 Before Starting

### Creation flow and related pages

This section explains the procedure for creating the Web page.



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting | Preparation | Reuse | User Web Page Creation | File Save Destination | Troubleshooting

## Overall user Web page to create

In this chapter, the JavaScript and CGI objects that can be used with user Web pages are used to create an HTML page like that shown below.

For details on and to change JavaScript and CGI objects, refer to their chapters.

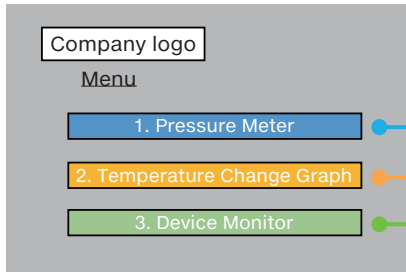
### Related Page

- JavaScript details ..... 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)
- CGI details ..... 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

### ► Menu: Page switching

Create a new user Web page.

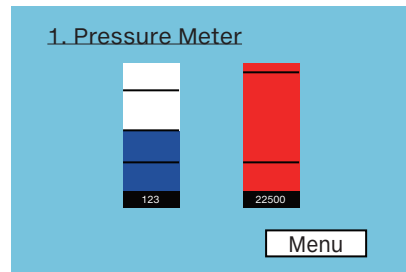
Refer to 3.4 User Web Page Creation  
- Menu page creation.



### ► Level display: Pressure status display

Reuse the user Web page library to create the component.

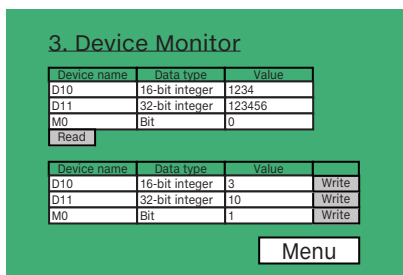
Refer to 3.4 User Web Page Creation  
- Pressure meter page creation.



### ► Device monitor: Device reading/writing

Reuse the CGI example in chapter 6 to create the component.

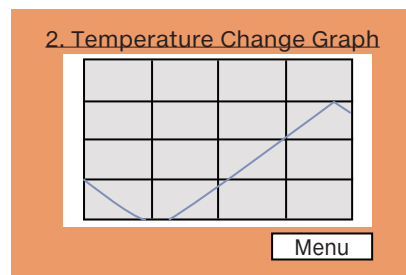
Refer to 3.4 User Web Page Creation  
- Device monitor page creation.



### ► Historical graph: Temperature change display

Reuse the user Web page library to create the component.

Refer to 3.4 User Web Page Creation  
- Temperature change page creation.



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

## 3.2 Preparation

- Relevant documents**
- Web Server Function Application Guide Using Web Page Startup and Introduction [manual number: L(NA)08643ENG]
- Related page**
- 6. USING USER WEB PAGE

### STEP 1. Prepare Web server HTML files.

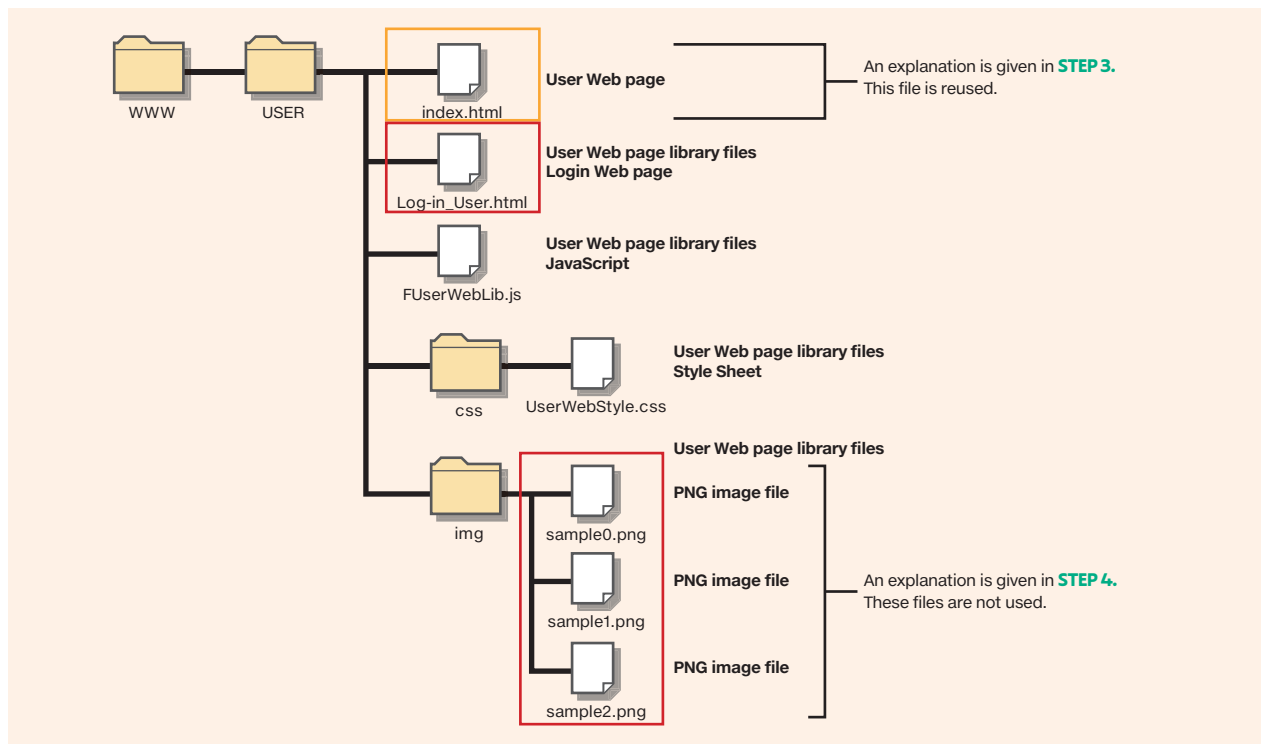
Obtain the user Web page library from the Mitsubishi Electric representative.  
Decompress the obtained file.

### STEP 2. Confirm files and folders.

Check that the structure of the obtained files and folders is as shown below.

[File and folder structure]

User Web page library files



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

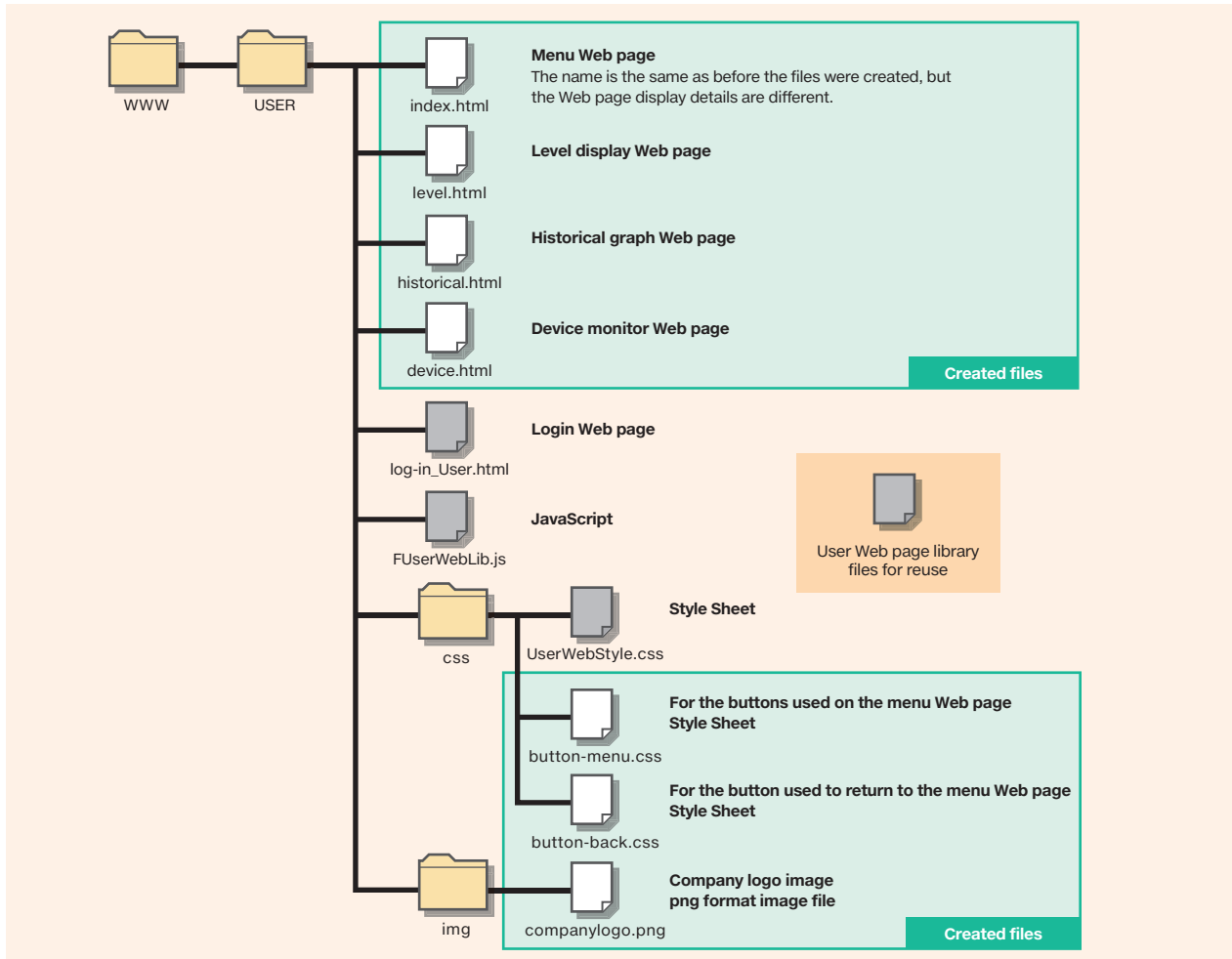
Reuse

User Web Page Creation

File Save Destination

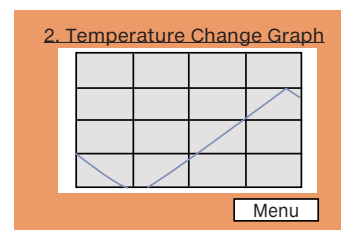
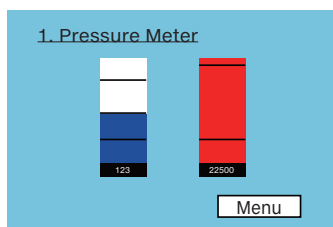
Troubleshooting

## Files after Web page creation



### STEP 3. Create HTML files.

Reuse index.html (an HTML file obtained from the Mitsubishi Electric representative) to create the level display Web page (level.html) and the historical graph Web page (historical.html). For the reuse method, refer to 3.3 Reuse.



### STEP 4. Delete unnecessary files.

The following image files in the img folder are not used in this guide. Please remove them.

Image file	sample0.png	sample1.png	sample2.png
------------	-------------	-------------	-------------

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting | Preparation | **Reuse** | User Web Page Creation | File Save Destination | Troubleshooting

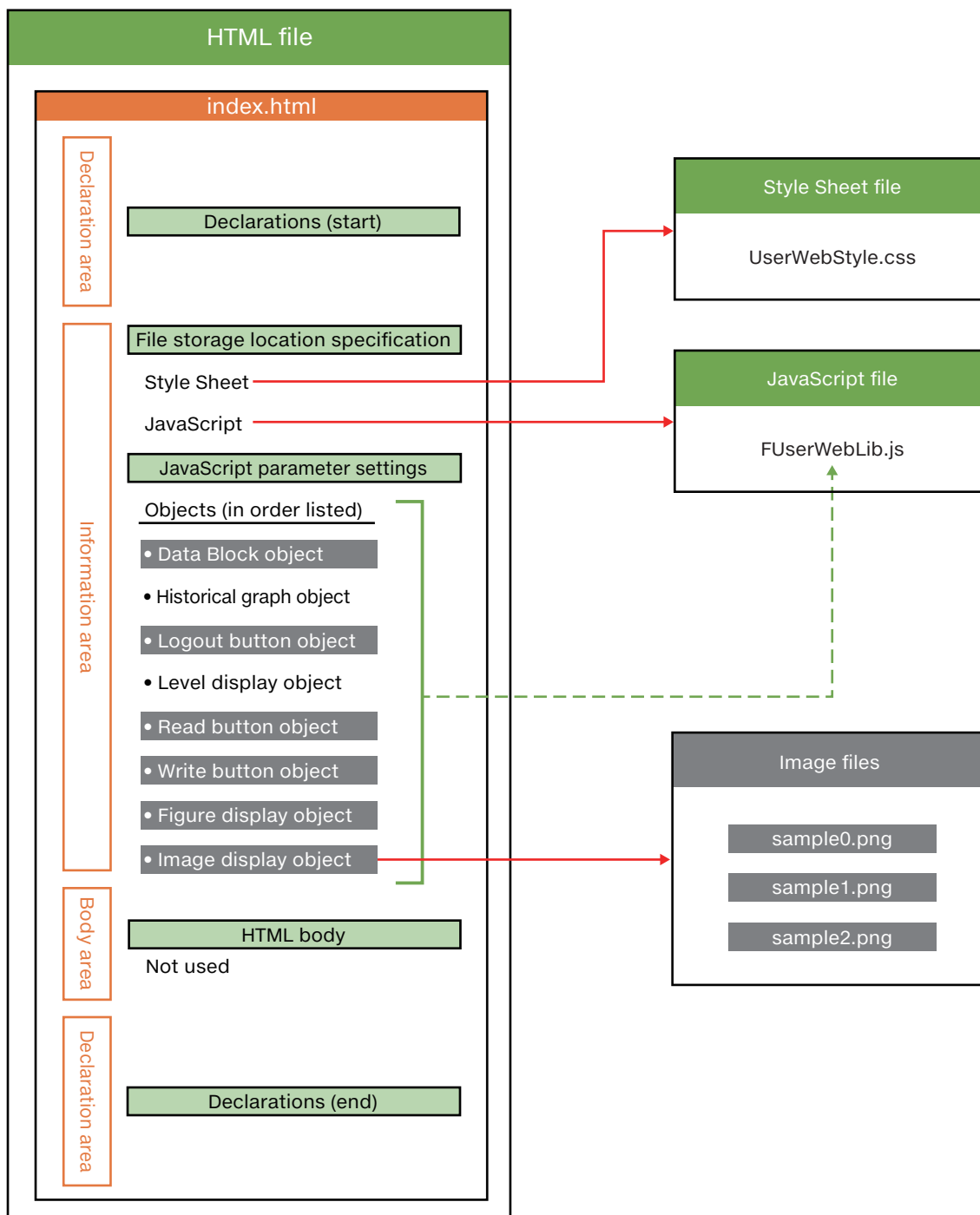
## 3.3 Reuse

### “index.html” overview

The relationships between and the display objects of the files contained in the User Web page library are shown in the following figure.

#### ► File correlations

The JavaScript parameter setting objects and image files written in white on a gray background are not used in the creation example of this guide.



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting | Preparation | **Reuse** | User Web Page Creation | File Save Destination | Troubleshooting

## index.html HTML document

The user Web page library HTML document (index.html) that is reused when creating the level display Web page (level.html) and the historical graph Web page (historical.html) is shown below.

Line No.	HTML
1	<!DOCTYPE html>
2	<html xmlns="http://www.w3.org/1999/xhtml">
3	<head>
4	<!-- charset setting *Because the Web server setting is UTF-8, specify UTF-8.) -->
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible" content="IE=edge"/>
7	<!-- Set the title. -->
8	<title>Sample</title>
9	<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />
10	<!-- Reading the library JavaScript (Change the path to match the storage location.) -->
11	<script src="/FUserWebLib.js"></script>
12	<!-- Write the proprietary JavaScript from this point. -->
13	<!-- Write the user JavaScript here. -->
14	<script>
15	var updateInterval = 5;
16	var dspLanguage = 'en-US';
17	
18	// Data Block object
19	temp = [];
20	for(var i = 0; i < 8; i++){
21	temp.push({
22	dsp:        'X' + i,
23	name:      'X' + i,
24	base:      'B',
25	format:    6
26	
27	});
28	}
29	for(var i = 0; i < 8; i++){
30	temp.push({
31	dsp:        'Y' + i,
32	name:      'Y' + i,
33	base:      'B',
34	format:    6
35	
36	});
37	}
38	dataBlockParam = {
39	dev:          temp,
40	direction:   1,
41	blkSize:     8,
42	devNamDisp:  1,
43	devNamCol:   'white',
44	devNamBkCol: '#808080',
45	devNamWidth: 100,
46	devNamHeight: 40,
47	devValCol:   'blue',
48	devValBkCol: 'white',
49	devValWidth: 80,
50	devValHeight: 50,
51	InCol:       'blue',
52	xPos:        20,
53	yPos:        40
54	}
55	WSDatblk(dataBlockParam);
56	

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

**Reuse**

User Web Page Creation

File Save Destination

Troubleshooting

Line No.	HTML
57	// Historical graph object
58	temp = [];
59	num = 2;
60	temp.push({
61	devName:    'D0',
62	lnCol:      'red',
63	});
64	temp.push({
65	devName:    'D1',
66	lnCol:      'blue',
67	});
68	hstGrpParam = {
69	xPos:       20,
70	yPos:       250,
71	grElmNum:   num,
72	devFormat:  0,
73	
74	grElm:      temp,
75	grBkCol:    '#F0F0F0',
76	dspCol:     'black',
77	pointNum:   20,
78	upper:      32767, //100,
79	lower:      -32768, //0,
80	xLine:      9,
81	yLine:      5,
82	grHeight:   380,
83	grWidth:    550,
84	upperMargin: 15,
85	leftMargin: 75,
86	lowerMargin: 55,
87	rightMargin: 25
88	}
89	WSHstgrp(hstGrpParam);
90	
91	// Logout button object
92	logoutBtnParam = {
93	xPos:       20,
94	yPos:       730,
95	btnHeigh:   26,
96	btnWidth:   100,
97	btnTxt:     'Log Out'
98	}
99	WSLogoutBtn(logoutBtnParam);
100	

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Line No.	HTML
101	// Level display object
102	WSLevel({
103	devName: 'D0',
104	direction: 0,
105	levCol: 'mediumblue',
106	upperCol: 'red',
107	lowerCol: '#00FF00',
108	bkCol: 'white',
109	upperVal: 32767,
110	lowerVal: -32768,
111	upperAlmV: 20000,
112	lowerAlmV: -20000,
113	dspAlmLn: 1,
114	almLnCol: 'black',
115	levLength: 400,
116	levWidth: 150,
117	dspVal: 1,
118	valFormat: 0,
119	
120	devValCol: 'white',
121	devValBkCol: 'black',
122	devValWidth: 150,
123	devValHeight: 50,
124	xPos: 700,
125	yPos: 250,
126	});
127	
128	// Write button object
129	WSWrtBtn({
130	devName: 'X0',
131	devBase: 'B',
132	devFormat: 6,
133	
134	wrVal: '1',
135	wrBtn: 'write_btn',
136	btnTxt: 'OK',
137	btnWidth: 150,
138	btnHeigh: 50,
139	wrComfirm: 1,
140	language: 1,
141	xPos: 700,
142	yPos: 40,
143	});
144	
145	// Write button object
146	WSWrtBtn({
147	devName: 'X0',
148	devBase: 'B',
149	devFormat: 6,
150	
151	wrVal: '0',
152	wrBtn: 'write_btn',
153	btnTxt: 'NG',
154	btnWidth: 150,
155	btnHeigh: 50,
156	wrComfirm: 1,
157	language: 1,
158	xPos: 700,
159	yPos: 130,
160	});
161	



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Line No.	HTML
162	// Figure display object
163	WSFigure({
164	devName: 'D0',
165	devFormat: 0,
166	
167	figType: 'tri',
168	figHeight: -30,
169	figWidth: 60,
170	defCol: 'red',
171	rangeNum: 2,
172	range:[
173	{
174	low: -5000,
175	high: 5000,
176	col: 'green',
177	},
178	{
179	low: -10000,
180	high: 10000,
181	col: 'blue',
182	},
183	],
184	xPos: 300,
185	yPos: 730,
186	});
187	
188	// Image display object
189	WSPicture({
190	devName: 'D0',
191	devFormat: 0,
192	
193	pictHeight: 30,
194	pictWidth: 30,
195	defPicture: './img/sample0.png',
196	rangeNum: 2,
197	range:[
198	{
199	low: -5000,
200	high: 5000,
201	picture: './img/sample1.png',
202	},
203	{
204	low: -10000,
205	high: 10000,
206	picture: './img/sample2.png',
207	},
208	],
209	xPos: 200,
210	yPos: 730,
211	}); </script>
212	</head>
213	<body>
214	</body>
215	</html>

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting	Preparation	Reuse	<b>User Web Page Creation</b>	File Save Destination	Troubleshooting	
			Menu page creation	Pressure meter page creation	Temperature change page creation	Device monitor page creation

## 3.4 User Web Page Creation

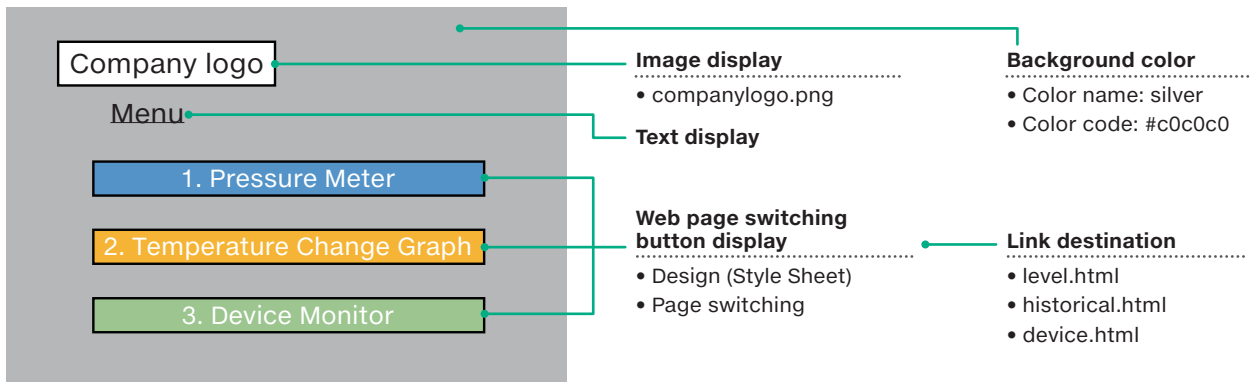
### Menu page creation

An image and buttons for switching the Web page are displayed on the page created with this example. Use a Style Sheet to design the buttons for switching the Web page.


**This Web page does not reuse index.html.**

### ► Specifications

[Design example]



### ► Functions to use

Function	Creation method	Example/special note	Reference section
Background color	Style Sheet	Write a Style Sheet inside the HTML file.	7.1
Image display	<img> tag	Use an HTML tag to display the image. 	7.2
Text display	Font size	<h3></h3> tags <h3>Heading 3</h3>	
	Underline	<u></u> tags <u>Text in this range is underlined.</u>	
Web page switching button display (menu selection)	Paragraph	<p></p> tags Use <p></p> to indicate paragraphs and insert line breaks.	7.1
	Design	Style Sheet When the button is clicked, its display changes to that shown on the right. 	
	Page switching	<a></a> tags <a href="link destination URL">Displayed text</a>	
Centering	Style Sheet	Even if the width of the Web page is changed, the buttons remain displayed in the center.	

### ► Files to create

File type	File name	Remarks
HTML	index.html	The file name of the initial user Web server page that is displayed is "index.html". Do not change this name.
Image	companylogo.png	This file can be named as desired. If you change it, also change the file name of the image files in the <img> tags in the link settings of the HTML file.
Style Sheet	button-menu.css	This file can be named as desired. If you change it, also change the file name of the Style Sheet file in the link settings of the HTML file.

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

## ► Creation procedure

### STEP 1. Create a new HTML file.

#### Related Page

For details on operations, also refer to the following.

2. CREATING AND DISPLAYING SIMPLE HTML DOCUMENTS ON A PERSONAL COMPUTER

1. Start Notepad in Windows®.
2. Create the parts of the following HTML file from <!DOCTYPE html> to </html>.
3. On the Notepad toolbar, select [File] - [Save As].
4. In the File name field, enter "index.html", and then click [Save]. Save the file in an easy-to-understand location.

[HTML]

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charset setting *Because the Web server setting is UTF-8, specify UTF-8.) -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- Set the title. -->
    <title>Menu</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta http-equiv="Content-Style-Type" content="text/css">
    <!-- Link settings of the Style Sheet file for the design of the Web page switching buttons -->
    <link href="/css/button-menu.css" rel="stylesheet" media="all"/>
    <!-- Web page background color setting (Create a Style Sheet inside the HTML file.) -->
    <style>
      body {
        background-color: #c0c0c0;
      }
    </style>

    <!-- For centering the Web page switching buttons (Create a Style Sheet inside the HTML file.) -->
    <style type="text/css">
    <!--
    .web-center {
      text-align:center; /* Internet Explorer centering */
      margin-left:auto; /* Centering for Web browsers other than Internet Explorer */
      margin-right:auto; /* Centering for Web browsers other than Internet Explorer */
      width:300px; /* Width */
      height:180px; /* Height */
      color:#ffffff; /* Character color */
    }
    -->
    </style>
  </head>
  <body>
    <!-- Company logo image display settings -->
    
    <!-- Web page title text display -->
    <h2><u>Menu</u></h2>
    <!-- Display of (three) Web page switching buttons -->
    <div class="web-center">
      <p>
        <a href="index1.html" class="button1">1. Pressure Meter</a>
      </p>
      <p>
        <a href="index2.html" class="button2">2. Temperature Change Graph</a>
      </p>
      <p>
        <a href="index3.html" class="button3">3. Device Monitor</a>
      </p>
    </div>
  </body>
</html>
```

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter  
page creationTemperature change  
page creationDevice monitor  
page creation

## STEP 2. Create Style Sheet files.

1. Start Notepad in Windows®.
2. In Notepad, enter all the Style Sheet details shown below.
3. On the Notepad toolbar, select [File] - [Save As].
4. In the File name field, enter "button-menu.css", and then click [Save].

For Style Sheet details, refer to 7.1 Style Sheet References.

[Design example] Use different colors to indicate the designs of the Web page switching buttons.

[Style Sheet]

class attributes: button3

```

a.button3{
  display: block;
  text-decoration: none;
  height: 35px;
  width: 250px;
  line-height: 37px;
  text-align: center;
  color: #ffffff;
  background: #3cb371;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
  -o-transition: 0.3s;
  -ms-transition: 0.3s;
  transition: 0.3s;
}
a.button3:hover{
  background: #fff;
  color: #3cb371;
  border:solid 1px #3cb371;
}

```

class attributes: button2

```

a.button2{
  display: block;
  text-decoration: none;
  height: 35px;
  width: 250px;
  line-height: 37px;
  text-align: center;
  color: #fff;
  background: #f4a460;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
  -o-transition: 0.3s;
  -ms-transition: 0.3s;
  transition: 0.3s;
}
a.button2:hover{
  background: #fff;
  color: #f4a460;
  border:solid 1px #f4a460;
}

```

class attributes: button1

```

a.button1{
  display: block;
  text-decoration: none;
  height: 37px;
  width: 250px;
  line-height: 37px;
  text-align: center;
  color: #fff;
  background: #4169e1;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
  -o-transition: 0.3s;
  -ms-transition: 0.3s;
  transition: 0.3s;
}
a.button1:hover{
  background: #fff;
  color: #4169e1;
  border:solid 1px #4169e1;
}

```

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

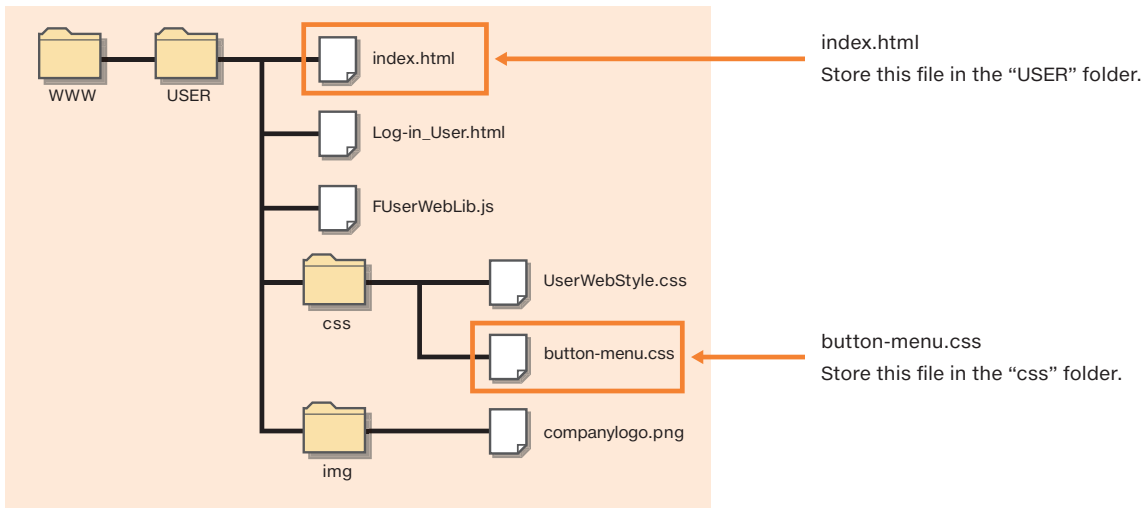
Temperature change page creation

Device monitor page creation

## STEP 3. File storage destination.

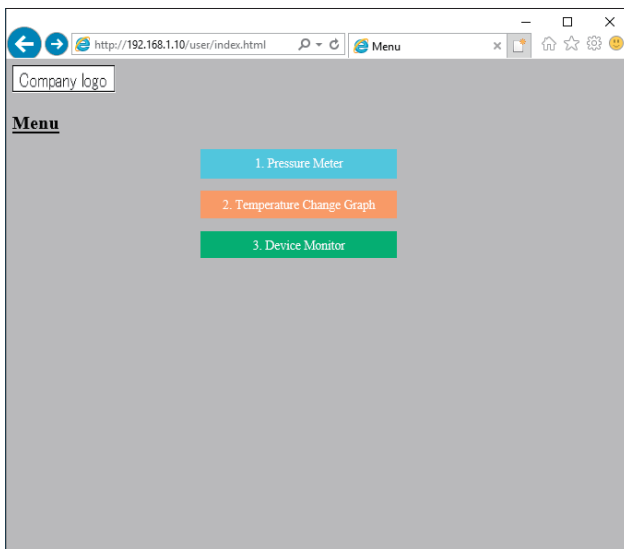
Store the created files so that the file structure is as shown below.

[File structure after organization]



## STEP 4. Check whether the page can be displayed correctly in a Web browser.

Double-click the "index.html" file.



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting	Preparation	Reuse	User Web Page Creation	File Save Destination	Troubleshooting
			Menu page creation	Pressure meter page creation	Temperature change page creation
					Device monitor page creation

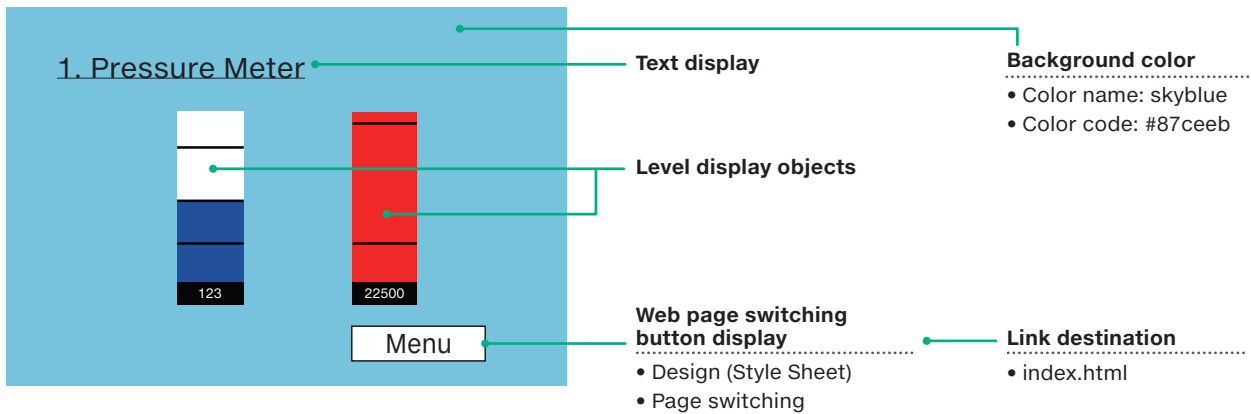
## Pressure meter page creation

This section explains the procedure for creating the pressure meter page.

Level display objects and a button for switching the Web page are displayed on the page created with this example. Reuse and edit “index.html” to create the level display objects and use a Style Sheet to design the button for switching the Web page.

### ► Specifications

[Design example]



### ► Functions to use

Function	Creation method	Example/special note	Reference section
Background color	Style Sheet	Write a Style Sheet inside the HTML file.	7.1
Image display	<img> tag	Use an HTML tag to display the image. 	
Text display	Font size	<h3></h3> tags	7.2
	Underline	<u></u> tags	
	Paragraph	<p></p> tags	
Level display objects (two)		Reuse the “level display object” in the HTML file in the user Web page library.	
	Devices		5.4
	Size	JavaScript	
	Display starting coordinates		
Web page switching button display (return to the menu)	Design	Style Sheet	7.1
	Page switching	<a></a> tags	
	Display starting coordinates	Style Sheet	

### ► Files to create

File type	File name	Remarks
HTML	level.html	Linked from the menu page. If the file name is changed, the menu page needs to be corrected.
Style Sheet	button-menu.css	This file can be named as desired. If you change it, also change the file name of the Style Sheet file in the link settings of the HTML file.

### ► Required files

File type	File name	Remarks
JavaScript	FUserWebLib.js	Required for displaying the level display objects.
Style Sheet	UserWebStyle.css	Use a <link> tag to set links to the files on the left.

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

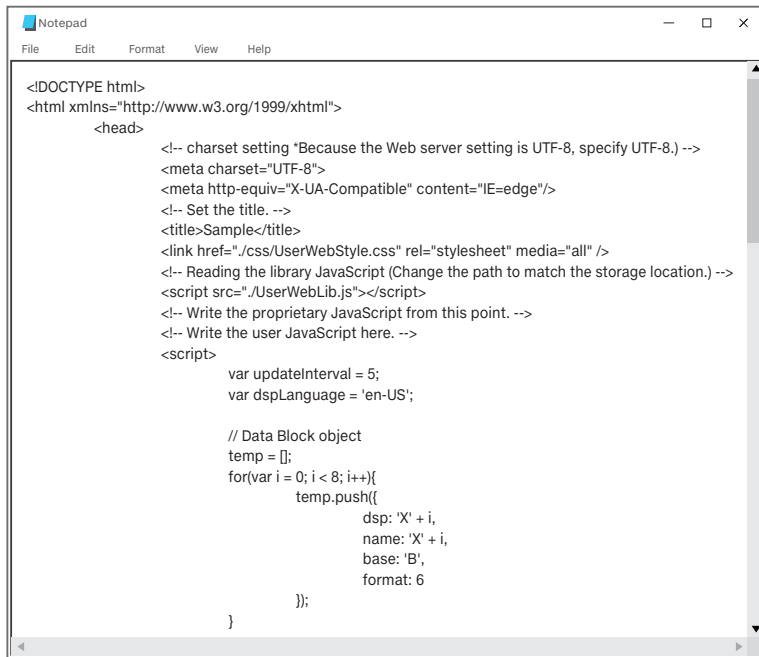
Device monitor page creation

## ► Creation procedure

Create this component by reusing the file “index.html” included in the user Web page library obtained from the Mitsubishi Electric representative.

### STEP 1. Open the file to reuse in Notepad.

In Notepad, open the “index.html” file from the user Web page library.



```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charset setting "Because the Web server setting is UTF-8, specify UTF-8." -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- Set the title. -->
    <title>Sample</title>
    <link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />
    <!-- Reading the library JavaScript (Change the path to match the storage location.) -->
    <script src="/UserWebLib.js"></script>
    <!-- Write the proprietary JavaScript from this point. -->
    <!-- Write the user JavaScript here. -->
    <script>
      var updateInterval = 5;
      var dspLanguage = 'en-US';

      // Data Block object
      temp = [];
      for(var i = 0; i < 8; i++){
        temp.push({
          dsp: 'X' + i,
          name: 'X' + i,
          base: 'B',
          format: 6
        });
      }
    </script>
  </head>
</html>
```

### STEP 2. Save the file under a different name: “level.html”.

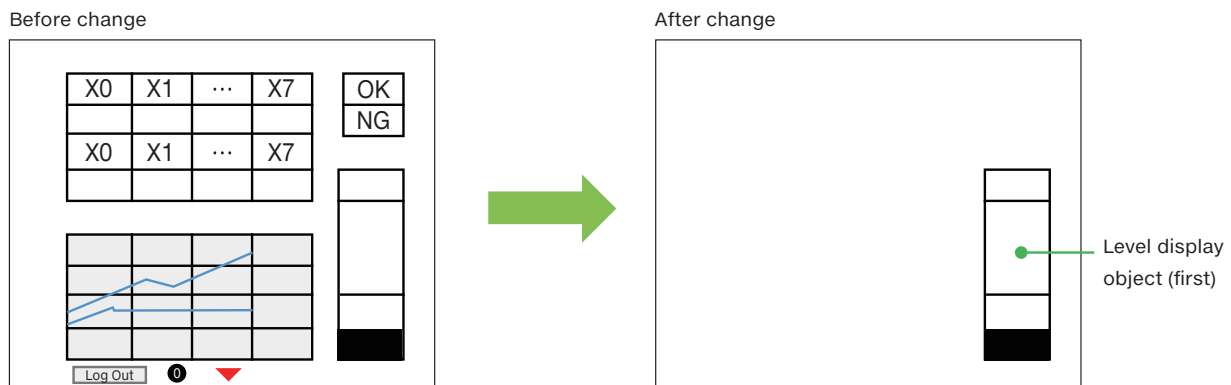
1. On the Notepad toolbar, select [File] - [Save As].
2. In the File name field, enter “level.html”, and then click [Save].

This prevents the original file from being overwritten incorrectly with the HTML file created by reusing the original.

### STEP 3. Delete the lines (sections) that are unnecessary for the Web page being created from the HTML.

Delete everything other than the sections for the level display objects.  
For details on the lines to delete, refer to the next page.

Starting with STEP 3, use “level.html”. Exercise caution to avoid correcting the wrong file.



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter  
page creationTemperature change  
page creationDevice monitor  
page creation

## Sections to reuse

To create a Web page that uses a pressure meter (level display objects), reuse the sections indicated by “Do not delete.” in the following table from the user Web page library (index.html) and delete the other sections indicated by “Delete”.

Line No.	HTML	Pressure meter
1	<!DOCTYPE html>	Do not delete.
2	<html xmlns="http://www.w3.org/1999/xhtml">	
3	<head>	
4	<!-- charset setting *Because the Web server setting is UTF-8, specify UTF-8.) -->	
5	<meta charset="UTF-8">	
6	<meta http-equiv="X-UA-Compatible" content="IE=edge"/>	
7	<!-- Set the title. -->	
8	<title>Sample</title>	
9	<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />	
10	<!-- Reading the library JavaScript (Change the path to match the storage location.) -->	
11	<script src="/UserWebLib.js"></script>	
12	<!-- Write the proprietary JavaScript from this point. -->	
13	<!-- Write the user JavaScript here. -->	
14	<script>	
15	var updateInterval = 5;	
16	var dspLanguage = 'en-US';	
17		
18	// Data Block object	
19	temp = [];	
20	for(var i = 0; i < 8; i++){	
21	temp.push({	
22	dsp:        'X' + i,	
23	name:      'X' + i,	
24	base:      'B',	
25	format:    6	
26		
27	});	
28	}	
29	for(var i = 0; i < 8; i++){	
30	temp.push({	
31	dsp:        'Y' + i,	
32	name:      'Y' + i,	
33	base:      'B',	
34	format:    6	
35		
36	});	
37	}	
38	dataBlockParam = {	
39	dev:        temp,	
40	direction:  1,	
41	blkSize:   8,	
42	devNamDisp: 1,	
43	devNamCol:  'white',	
44	devNamBkCol: '#808080',	
45	devNamWidth: 100,	
46	devNamHeight: 40,	
47	devValCol:  'blue',	
48	devValBkCol: 'white',	
49	devValWidth: 80,	
50	devValHeight: 50,	
51	InCol:      'blue',	
52	xPos:       20,	
53	yPos:       40	
54	}	
55	WSDatblk(dataBlockParam);	
56		



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

Line No.	HTML	Pressure meter
57	// Historical graph object	
58	temp = [];	
59	num = 2;	
60	temp.push({	
61	devName:    'D0',	
62	lnCol:      'red',	
63	});	
64	temp.push({	
65	devName:    'D1',	
66	lnCol:      'blue',	
67	});	
68	hstGrpParam = {	
69	xPos:       20,	
70	yPos:       250,	
71	grElmNum:   num,	
72	devFormat:  0,	
73		
74	grElm:      temp,	
75	grBkCol:    '#F0F0F0',	
76	dspCol:     'black',	
77	pointNum:   20,	
78	upper:      32767, //100,	
79	lower:      -32768, //0,	
80	xLine:      9,	
81	yLine:      5,	
82	grHeight:   380,	
83	grWidth:    550,	
84	upperMargin: 15,	
85	leftMargin: 75,	
86	lowerMargin: 55,	
87	rightMargin: 25,	
88	}	
89	WSHstgrp(hstGrpParam);	
90		
91	// Logout button object	
92	logoutBtnParam = {	
93	xPos:       20,	
94	yPos:       730,	
95	btnHeigh:   26,	
96	btnWidth:   100,	
97	btnTxt:     'Log Out'	
98	}	
99	WSLogoutBtn(logoutBtnParam);	
100		

Delete.

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting	Preparation	Reuse	User Web Page Creation	File Save Destination	Troubleshooting
			Menu page creation	Pressure meter page creation	Temperature change page creation
					Device monitor page creation

Line No.	HTML	Pressure meter
101	// Level display object	
102	WSLevel({	
103	devName: 'D0',	
104	direction: 0,	
105	levCol: 'mediumblue',	
106	upperCol: 'red',	
107	lowerCol: '#00FF00',	
108	bkCol: 'white',	
109	upperVal: 32767,	
110	lowerVal: -32768,	
111	upperAlmV: 20000,	
112	lowerAlmV: -20000,	
113	dspAlmLn: 1,	
114	almLnCol: 'black',	
115	levLength: 400,	
116	levWidth: 150,	
117	dspVal: 1,	
118	valFormat: 0,	
119		
120	devValCol: 'white',	
121	devValBkCol: 'black',	
122	devValWidth: 150,	
123	devValHeight: 50,	
124	xPos: 700,	
125	yPos: 250,	
126	});	
127		
128	// Write button object	
129	WSWrtBtn({	
130	devName: 'X0',	
131	devBase: 'B',	
132	devFormat: 6,	
133		
134	wrVal: '1',	
135	wrBtn: 'write_btn',	
136	btnTxt: 'OK',	
137	btnWidth: 150,	
138	btnHeigh: 50,	
139	wrComfirm: 1,	
140	language: 1,	
141	xPos: 700,	
142	yPos: 40,	
143	});	
144		
145	// Write button object	
146	WSWrtBtn({	
147	devName: 'X0',	
148	devBase: 'B',	
149	devFormat: 6,	
150		
151	wrVal: '0',	
152	wrBtn: 'write_btn',	
153	btnTxt: 'NG',	
154	btnWidth: 150,	
155	btnHeigh: 50,	
156	wrComfirm: 1,	
157	language: 1,	
158	xPos: 700,	
159	yPos: 130,	
160	});	
161		

Do not delete.

Delete.

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

Line No.	HTML	Pressure meter
162	// Figure display object	
163	WSFigure({	
164	devName: 'D0',	
165	devFormat: 0,	
166		
167	figType: 'tri',	
168	figHeight: -30,	
169	figWidth: 60,	
170	defCol: 'red',	
171	rangeNum: 2,	
172	range:[	
173	{	
174	low: -5000,	
175	high: 5000,	
176	col: 'green',	
177	},	
178	{	
179	low: -10000,	
180	high: 10000,	
181	col: 'blue',	
182	},	
183	],	
184	xPos: 300,	
185	yPos: 730,	
186	});	
187		
188	// Image display object	
189	WSPicture({	
190	devName: 'D0',	
191	devFormat: 0,	
192		
193	pictHeight: 30,	
194	pictWidth: 30,	
195	defPicture: './img/sample0.png',	
196	rangeNum: 2,	
197	range:[	
198	{	
199	low: -5000,	
200	high: 5000,	
201	picture: './img/sample1.png',	
202	},	
203	{	
204	low: -10000,	
205	high: 10000,	
206	picture: './img/sample2.png',	
207	},	
208	],	
209	xPos: 200,	
210	yPos: 730,	
211	}); </script>	
212	</head>	
213	<body>	
214	</body>	
215	</html>	

Delete.

Do not delete.

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

- Before Starting
- Preparation
- Reuse
- User Web Page Creation**
- File Save Destination
- Troubleshooting
- Menu page creation
- Pressure meter page creation
- Temperature change page creation
- Device monitor page creation

## STEP 4. Add a </script> tag.

Add a </script> tag on line 44 (blank).

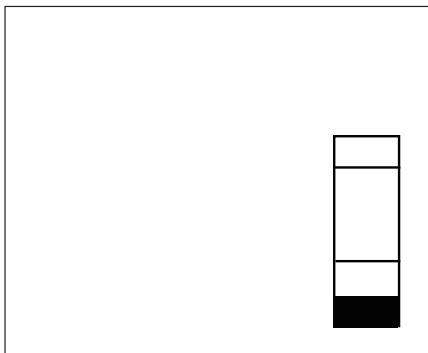
Line No.	HTML
...	...
18	// Level display object
...	...
41	xPos: 700,
42	yPos: 250,
43	});
44	</script> ← Addition
45	</head>
46	<body>
47	</body>
48	</html>

## STEP 5. Correct items such as the size and display coordinates of the (first) level display object.

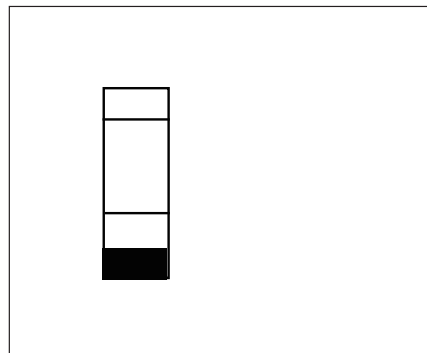
Correct items such as the size and display coordinates of the (first) level display object.

Line No.	HTML	
...	...	
18	// Level display object	
19	WSLevel({	
20	devName: 'D0',	Device specification
...	...	
32	levLength: 400,	← 200, Level size
33	levWidth: 150,	← 100, Level size
...	...	
39	devValWidth: 150,	← 100, Device value display size
40	devValHeight: 50,	← 50, Device value display size
41	xPos: 700,	← 100, Display starting coordinates
42	yPos: 250,	← 50, Display starting coordinates
43	});	

Before change



After change



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

## STEP 6. Add a (second) level display object.

Line No.	HTML
:	:
18	<code>// Level display object</code>
:	:
41	<code>  xPos: 100,</code>
42	<code>  yPos: 50,</code>
43	<code>});</code>
45	<code>&lt;/script&gt;</code>
46	<code>&lt;/head&gt;</code>
47	<code>&lt;body&gt;</code>
48	<code>&lt;/body&gt;</code>
49	<code>&lt;/html&gt;</code>

Copy lines 18 to 43 and paste these lines between lines 43 and 44.

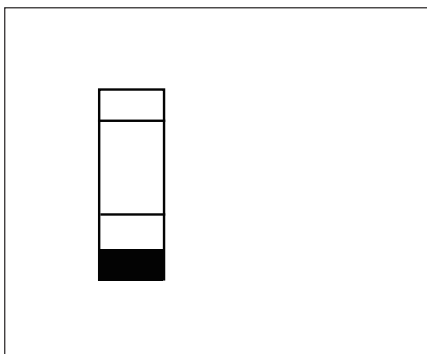
## STEP 7. Correct the display position and size of the (second) level display object.

Line No.	HTML
:	:
44	<code>// Level display object</code>
45	<code>WSLevel{</code>
46	<code>  devName: 'D0',</code>
:	:
66	<code>  devValHeight: 50,</code>
67	<code>  xPos: 100,</code>
68	<code>  yPos: 50,</code>
69	<code>});</code>
70	<code>&lt;/script&gt;</code>
71	<code>&lt;/head&gt;</code>
72	<code>&lt;body&gt;</code>
73	<code>&lt;/body&gt;</code>
74	<code>&lt;/html&gt;</code>

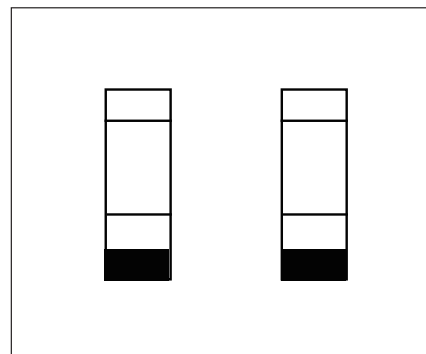
'D1', Device specification

300, Display starting coordinates

Before change



After change



### 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter  
page creationTemperature change  
page creationDevice monitor  
page creation

#### STEP 8. Set the background color.

Add the HTML for the background color between lines 13 and 14.

If the background color is white, there is no need to add this HTML.

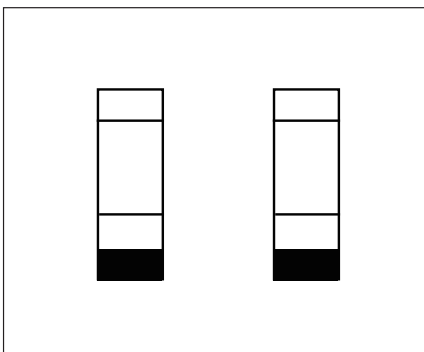
Line No.	HTML
:	:
10	<!-- Reading the library JavaScript (Change the path to match the storage location.) -->
11	<script src="/FUserWebLib.js"></script>
12	<!-- Write the proprietary JavaScript from this point. -->
13	<!-- Write the user JavaScript here. -->
	<pre> &lt;style&gt;   body {     background-color: #87ceeb;   } &lt;/style&gt; </pre>
14	<script>
15	var updateInterval = 5;
16	var dspLanguage = 'en-US';

#### STEP 9. Change the page title and add a Web page switching button.

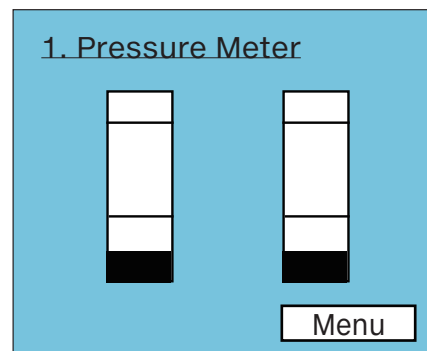
Change the page title on line 8, and then add the HTML for the Web page switching button between lines 77 and 78.

Line No.	HTML
:	:
8	<title>1. Pressure Meter</title> ← Page title text display
:	:
77	<body>
	<pre> &lt;!-- Web page title text display --&gt; &lt;h2&gt;&lt;u&gt;1. Pressure Meter&lt;/u&gt;&lt;/h2&gt; &lt;!-- Display of Web page switching button --&gt; &lt;a href="index.html" class="menu" style="position: absolute; left: 350px; top: 550px;" /&gt;Menu&lt;/a&gt; </pre>
78	</body>
79	</html>

Before change



After change



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

## STEP 10. Add the <link> setting to Style Sheet file for Web page switching button.

Add the HTML for the <link> settings of the Style Sheet file for the Web page switching button between lines 9 and 10.

Line No.	HTML
:	:
7	<!-- Set the title. -->
8	<title>1. Pressure Meter</title>
9	<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />
	<link href="/css/button-back.css" rel="stylesheet" media="all" />
10	<!-- Reading the library JavaScript (Change the path to match the storage location.) -->

## STEP 11. Create Style Sheet files for the Web page switching button.

1. Start Notepad in Windows®.
2. Create the Style Sheet shown below.
3. On the Notepad toolbar, select [File] - [Save As].
4. In the File name field, enter "button-back.css", and then click [Save].

**This button is used on the "pressure meter page", "temperature change page", and "device monitor page".**

[Design example] Indicate the design of the Web page switching button.

1. Pressure Meter

Menu

[Style Sheet]

class attributes: menu

```
a.menu{
display: block;
text-decoration: none;
height: 37px;
width: 250px;
line-height: 37px;
text-align: center;
color: #fff;
background: #4169e1;
-webkit-transition: 0.3s;
-moz-transition: 0.3s;
-o-transition: 0.3s;
-ms-transition: 0.3s;
transition: 0.3s;
}
a.menu:hover{
background: #fff;
color: #4169e1;
border:solid 1px #4169e1;
}
```

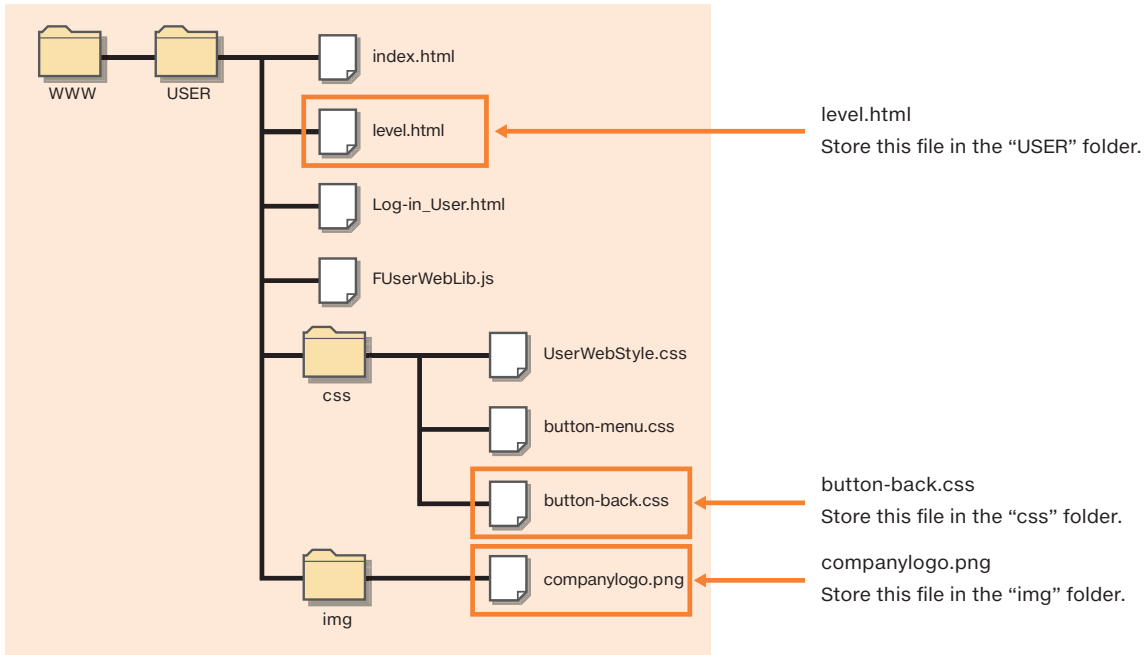
# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

- Before Starting
- Preparation
- Reuse
- User Web Page Creation**
- File Save Destination
- Troubleshooting
- Menu page creation
- Pressure meter page creation
- Temperature change page creation
- Device monitor page creation

## STEP 12. File storage destination.

Store the created files so that the file structure is as shown below.

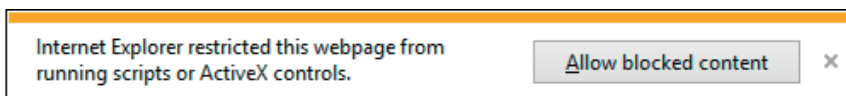
[File structure after organization]



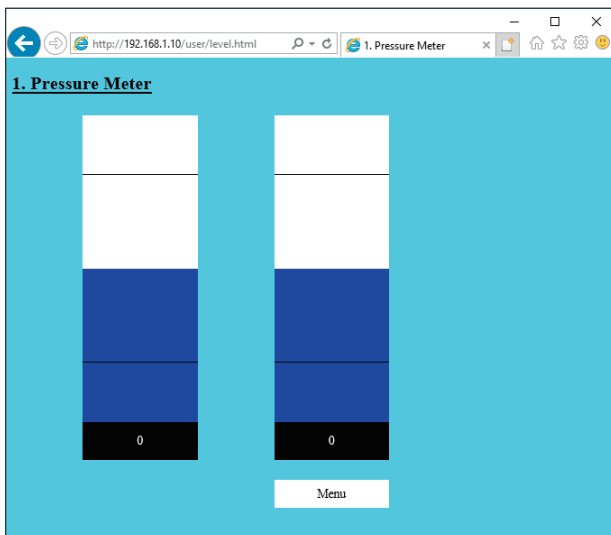
## STEP 13. Check whether the page can be displayed correctly in a Web browser.

Double-click the "level.html" file.

If the following dialog box is displayed, click "Allow blocked content".



Correctly displayed Web page





# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

## Temperature change page creation

This section explains the procedure for creating the temperature change page.

A historical graph object and a button for switching the Web page are displayed on the page created with this example. Reuse and edit "index.html" to create the historical graph object and use a Style Sheet to design the button for switching the Web page.

### ► Specifications

[Design example]

**Text display**

**Historical graph object**

**Web page switching button display**


**Background color**

- Color name: sandybrown
- Color code: #f4a460

**Link destination**

- Design (Style Sheet)
- Page switching
- index.html

### ► Functions to use

Function	Creation method	Example/special note	Reference section
Background color	Style Sheet	Write a Style Sheet inside the HTML file.	7.1
Text display	Font size	<h3></h3> tags <h3>Heading 3</h3>	7.2
	Underline	<u></u> tags <u>Text in this range is underlined.</u>	
	Paragraph	<p></p> tags Use <p></p> to indicate paragraphs and insert line breaks.	
Historical graph object	Reuse the "historical graph object" in the HTML file in the user Web page library		5.4
	Devices	D0 and D1	
	Size	Change the grHeight and grWidth parameters.	
Web page switching button display (return to the menu)	Design	Style Sheet When the button is clicked, its display changes to that shown on the right. 	7.1
	Page switching	<a></a> tags <a href="link destination URL">Displayed text</a>	
	Display starting coordinates	Style Sheet Embed a Style Sheet in the <a> tag to display the button in the lower-right corner.	

### ► Files to create

File type	File name	Remarks
HTML	historical.html	Linked from the menu page. If the file name is changed, the menu page needs to be corrected.
Style Sheet	button-back.css	This file can be named as desired. If you change it, also change the file name of the Style Sheet file in the link settings of the HTML file.

### ► Required files

Use the files in the user Web page library as-is.

File type	File name	Remarks
JavaScript	FUserWebLib.js	Required for displaying the historical graph object.
Style Sheet	UserWebStyle.css	Use a <link> tag to set links to the files on the left.

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

- Before Starting
- Preparation
- Reuse
- User Web Page Creation**
- File Save Destination
- Troubleshooting
- Menu page creation
- Pressure meter page creation
- Temperature change page creation**
- Device monitor page creation

## ► Creation procedure

Create this component by reusing the file “index.html” included in the user Web page library obtained from the Mitsubishi Electric representative.

### STEP 1. Open the file to reuse in Notepad.

In Notepad, open the “index.html” file from the user Web page library.

```

Notepad
File Edit Format View Help
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charset setting "Because the Web server setting is UTF-8, specify UTF-8." -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- Set the title. -->
    <title>Sample</title>
    <link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />
    <!-- Reading the library JavaScript (Change the path to match the storage location.) -->
    <script src="/UserWebLib.js"></script>
    <!-- Write the proprietary JavaScript from this point. -->
    <!-- Write the user JavaScript here. -->
    <script>
      var updateInterval = 5;
      var dspLanguage = 'en-US';

      // Data Block object
      temp = [];
      for(var i = 0; i < 8; i++){
        temp.push({
          dsp: 'X' + i,
          name: 'X' + i,
          base: 'B',
          format: 6
        });
      }
    </script>
  </head>
</html>
  
```

### STEP 2. Save the file under a different name: “historical.html”.

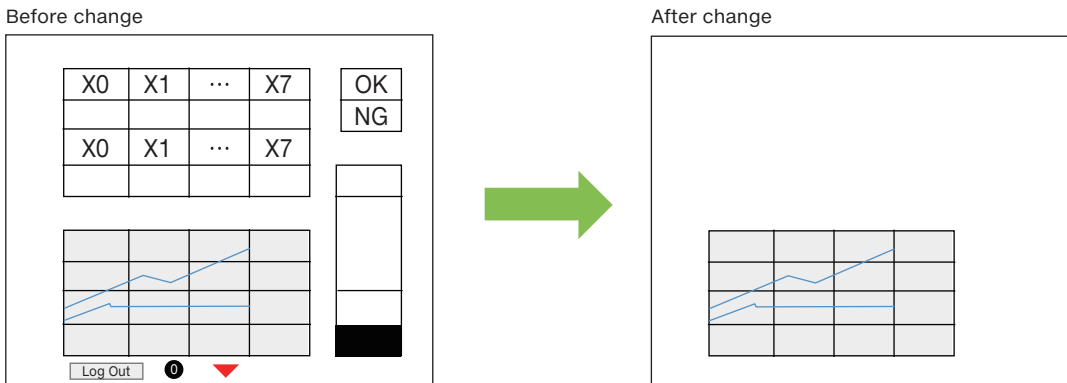
- On the Notepad toolbar, select [File] - [Save As].
- In the File name field, enter “historical.html”, and then click [Save].

**This prevents the original file from being overwritten incorrectly with the HTML file created by reusing the original.**

### STEP 3. Delete the lines (sections) that are unnecessary for the Web page being created from the HTML.

Delete everything other than the sections for the level display objects.

**Starting with STEP3, use “historical.html”. Exercise caution to avoid correcting the wrong file.**



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting	Preparation	Reuse	User Web Page Creation	File Save Destination	Troubleshooting
			Menu page creation	Pressure meter page creation	Temperature change page creation
					Device monitor page creation

## Sections to reuse

To create a Web page that uses a temperature change graph (historical graph object), reuse the sections indicated in the following table from the user Web page library (index.html) and delete all the other sections.

Line No.	HTML	Temperature change graph
1	<!DOCTYPE html>	Do not delete.
2	<html xmlns="http://www.w3.org/1999/xhtml">	
3	<head>	
4	<!-- charset setting *Because the Web server setting is UTF-8, specify UTF-8.) -->	
5	<meta charset="UTF-8">	
6	<meta http-equiv="X-UA-Compatible" content="IE=edge"/>	
7	<!-- Set the title. -->	
8	<title>Sample</title>	
9	<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />	
10	<!-- Reading the library JavaScript (Change the path to match the storage location.) -->	
11	<script src="/UserWebLib.js"></script>	
12	<!-- Write the proprietary JavaScript from this point. -->	
13	<!-- Write the user JavaScript here. -->	
14	<script>	Delete.
15	var updateInterval = 5;	
16	var dspLanguage = 'en-US';	
17		
18	// Data Block object	
19	temp = [];	
20	for(var i = 0; i < 8; i++){	
21	temp.push({	
22	dsp:        'X' + i,	
23	name:      'X' + i,	
24	base:      'B',	
25	format:    6	
26		
27	});	
28	}	
29	for(var i = 0; i < 8; i++){	
30	temp.push({	
31	dsp:        'Y' + i,	
32	name:      'Y' + i,	
33	base:      'B',	
34	format:    6	
35		
36	});	
37	}	
38	dataBlockParam = {	
39	dev:        temp,	
40	direction:  1,	
41	blkSize:    8,	
42	devNamDisp: 1,	
43	devNamCol:  'white',	
44	devNamBkCol: '#808080',	
45	devNamWidth: 100,	
46	devNamHeight: 40,	
47	devValCol:  'blue',	
48	devValBkCol: 'white',	
49	devValWidth: 80,	
50	devValHeight: 50,	
51	InCol:      'blue',	
52	xPos:       20,	
53	yPos:       40	
54	}	
55	WSDatblk(dataBlockParam);	
56		

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

Line No.	HTML	Temperature change graph
57	// Historical graph object	
58	temp = [];	
59	num = 2;	
60	temp.push({	
61	devName:    'D0',	
62	lnCol:      'red',	
63	});	
64	temp.push({	
65	devName:    'D1',	
66	lnCol:      'blue',	
67	});	
68	hstGrpParam = {	
69	xPos:       20,	
70	yPos:       250,	
71	grElmNum:   num,	
72	devFormat:  0,	
73		
74	grElm:      temp,	
75	grBkCol:    '#F0F0F0',	
76	dspCol:     'black',	
77	pointNum:   20,	
78	upper:      32767, //100,	
79	lower:      -32768, //0,	
80	xLine:      9,	
81	yLine:      5,	
82	grHeight:   380,	
83	grWidth:    550,	
84	upperMargin: 15,	
85	leftMargin: 75,	
86	lowerMargin: 55,	
87	rightMargin: 25	
88	}	
89	WSHstgrp(hstGrpParam);	
90		
91	// Logout button object	
92	logoutBtnParam = {	
93	xPos:       20,	
94	yPos:       730,	
95	btnHeigh:   26,	
96	btnWidth:   100,	
97	btnTxt:     'Log Out'	
98	}	
99	WSLogoutBtn(logoutBtnParam);	
100		

Do not delete.

Delete.

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

Line No.	HTML	Temperature change graph
101	// Level display object	
102	WSLevel({	
103	devName: 'D0',	
104	direction: 0,	
105	levCol: 'mediumblue',	
106	upperCol: 'red',	
107	lowerCol: '#00FF00',	
108	bkCol: 'white',	
109	upperVal: 32767,	
110	lowerVal: -32768,	
111	upperAlmV: 20000,	
112	lowerAlmV: -20000,	
113	dspAlmLn: 1,	
114	almLnCol: 'black',	
115	levLength: 400,	
116	levWidth: 150,	
117	dspVal: 1,	
118	valFormat: 0,	
119		
120	devValCol: 'white',	
121	devValBkCol: 'black',	
122	devValWidth: 150,	
123	devValHeight: 50,	
124	xPos: 700,	
125	yPos: 250,	
126	});	
127		
128	// Write button object	
129	WSWrtBtn({	
130	devName: 'X0',	
131	devBase: 'B',	
132	devFormat: 6,	
133		
134	wrVal: '1',	
135	wrBtn: 'write_btn',	
136	btnTxt: 'OK',	
137	btnWidth: 150,	
138	btnHeigh: 50,	
139	wrComfirm: 1,	
140	language: 1,	
141	xPos: 700,	
142	yPos: 40,	
143	});	
144		
145	// Write button object	
146	WSWrtBtn({	
147	devName: 'X0',	
148	devBase: 'B',	
149	devFormat: 6,	
150		
151	wrVal: '0',	
152	wrBtn: 'write_btn',	
153	btnTxt: 'NG',	
154	btnWidth: 150,	
155	btnHeigh: 50,	
156	wrComfirm: 1,	
157	language: 1,	
158	xPos: 700,	
159	yPos: 130,	
160	});	
161		

Delete.

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting	Preparation	Reuse	User Web Page Creation	File Save Destination	Troubleshooting
			Menu page creation	Pressure meter page creation	Temperature change page creation
					Device monitor page creation

Line No.	HTML	Temperature change graph
162	// Figure display object	
163	WSFigure({	
164	devName: 'D0',	
165	devFormat: 0,	
166		
167	figType: 'tri',	
168	figHeight: -30,	
169	figWidth: 60,	
170	defCol: 'red',	
171	rangeNum: 2,	
172	range:[	
173	{	
174	low: -5000,	
175	high: 5000,	
176	col: 'green',	
177	},	
178	{	
179	low: -10000,	
180	high: 10000,	
181	col: 'blue',	
182	},	
183	],	
184	xPos: 300,	
185	yPos: 730,	
186	});	
187		
188	// Image display object	
189	WSPicture({	
190	devName: 'D0',	
191	devFormat: 0,	
192		
193	pictHeight: 30,	
194	pictWidth: 30,	
195	defPicture: './img/sample0.png',	
196	rangeNum: 2,	
197	range:[	
198	{	
199	low: -5000,	
200	high: 5000,	
201	picture: './img/sample1.png',	
202	},	
203	{	
204	low: -10000,	
205	high: 10000,	
206	picture: './img/sample2.png',	
207	},	
208	],	
209	xPos: 200,	
210	yPos: 730,	
211	}); </script>	
212	</head>	
213	<body>	
214	</body>	
215	</html>	

Delete.

Do not delete.

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

- Before Starting
- Preparation
- Reuse
- User Web Page Creation**
- File Save Destination
- Troubleshooting
- Menu page creation
- Pressure meter page creation
- Temperature change page creation**
- Device monitor page creation

## STEP 4. Add a </script> tag.

Add a </script> tag on line 51 (blank).

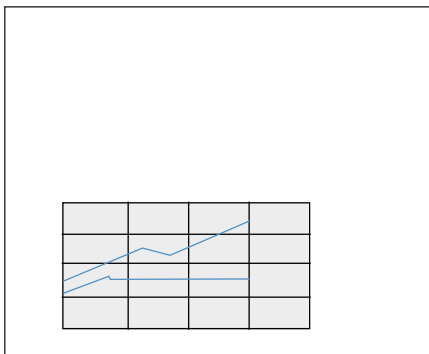
Line No.	HTML
...	...
18	// Historical graph object
19	temp = [];
20	num = 2;
...	...
47	lowerMargin: 55,
48	rightMargin: 25,
49	}
50	WSHstgrp(hstGrpParam);
51	</script> ← Addition
52	</head>
53	<body>
54	</body>
55	</html>

## STEP 5. Correct items such as historical graph object size and display coordinates.

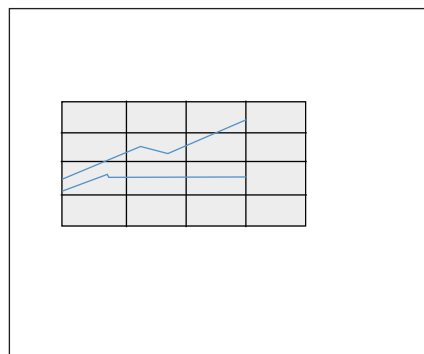
Correct items such as the size and display coordinates of the historical graph object.

Line No.	HTML
...	...
18	// Historical graph object
19	temp = [];
20	num = 2;
...	...
29	hstGrpParam = {
30	xPos: 20, ← 60,
31	yPos: 250, ← 250,
32	grElmNum: num, ← 450,
...	...
43	grHeight: 380, ← 380,
44	grWidth: 550, ← 550,
45	upperMargin: 15, ← 15,
46	leftMargin: 75, ← 75,
47	lowerMargin: 55, ← 55,
48	rightMargin: 25,
49	}
50	WSHstgrp(hstGrpParam);
51	</script>
52	</head>
53	<body>
54	</body>
55	</html>

Before change



After change



### 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter  
page creationTemperature change  
page creationDevice monitor  
page creation

#### STEP 6. Set the background color.

Add the HTML for the background color between lines 13 and 14.

If the background color is white, there is no need to add this HTML.

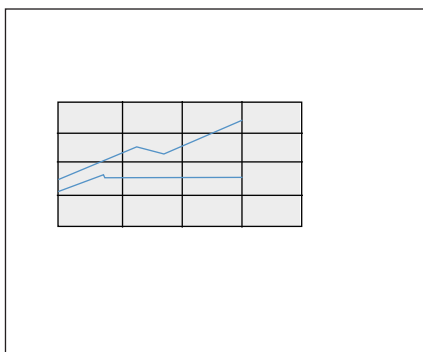
Line No.	HTML
:	:
10	<!-- Reading the library JavaScript (Change the path to match the storage location.) -->
11	<script src="/FUserWebLib.js"></script>
12	<!-- Write the proprietary JavaScript from this point. -->
13	<!-- Write the user JavaScript here. -->
	<pre> &lt;style&gt;   body {     background-color: sandybrown;   } &lt;/style&gt; </pre>
14	<script>
15	var updateInterval = 5;
16	var dspLanguage = 'en-US';

#### STEP 7. Change the page title and add a Web page switching button.

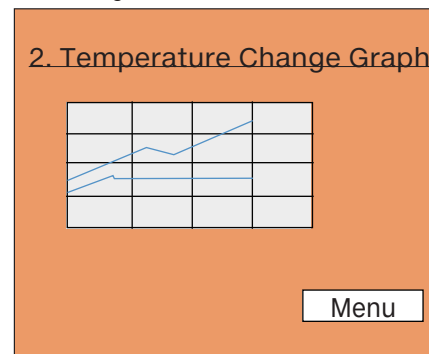
Change the page title on line 8, and then add the HTML for the Web page switching button between lines 58 and 59.

Line No.	HTML
:	:
8	<title>2. Temperature Change Graph</title> ← Page title text display
:	:
58	<pre> &lt;body&gt; &lt;!-- Web page title text display --&gt; &lt;h2&gt;&lt;u&gt;2. Temperature Change Graph&lt;/u&gt;&lt;/h2&gt; &lt;!-- Display of Web page switching button --&gt; &lt;a href="index.html" class="menu" style="position: absolute; left: 500px; top: 550px;" /&gt;Menu&lt;/a&gt; </pre>
59	</body>
60	</html>

Before change



After change





# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

## STEP 8. Add the <link> setting to Style Sheet file for Web page switching button.

Add the HTML for the <link> settings of the Style Sheet file for the Web page switching button between lines 9 and 10.

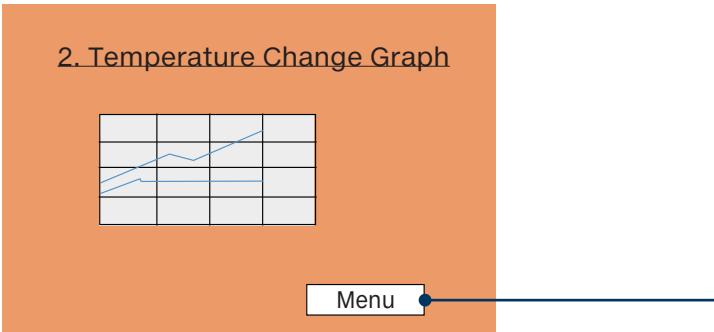
Line No.	HTML
:	:
7	<!-- Set the title. -->
8	<title>2. Temperature Change Graph</title>
9	<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />
	<link href="/css/button-back.css" rel="stylesheet" media="all" />
10	<!-- Reading the library JavaScript (Change the path to match the storage location.) -->

## STEP 9. Create Style Sheet files for the Web page switching button.

1. Start Notepad in Windows®.
2. Create the Style Sheet shown below.
3. On the Notepad toolbar, select [File] - [Save As].
4. In the File name field, enter "button-back.css", and then click [Save].

**This button is used on the "pressure meter page", "temperature change page", and "device monitor page".**

[Design example] Indicate the design of the Web page switching button.



The design example shows a web page with a title "2. Temperature Change Graph" and a line graph. Below the graph is a button labeled "Menu". A blue arrow points from the "Menu" button to the CSS code block below.

```
[Style Sheet]
class attributes: menu

a.menu{
display: block;
text-decoration: none;
height: 37px;
width: 250px;
line-height: 37px;
text-align: center;
color: #fff;
background: #4169e1;
-webkit-transition: 0.3s;
-moz-transition: 0.3s;
-o-transition: 0.3s;
-ms-transition: 0.3s;
transition: 0.3s;
}
a.menu:hover{
background: #fff;
color: #4169e1;
border:solid 1px #4169e1;
}
```

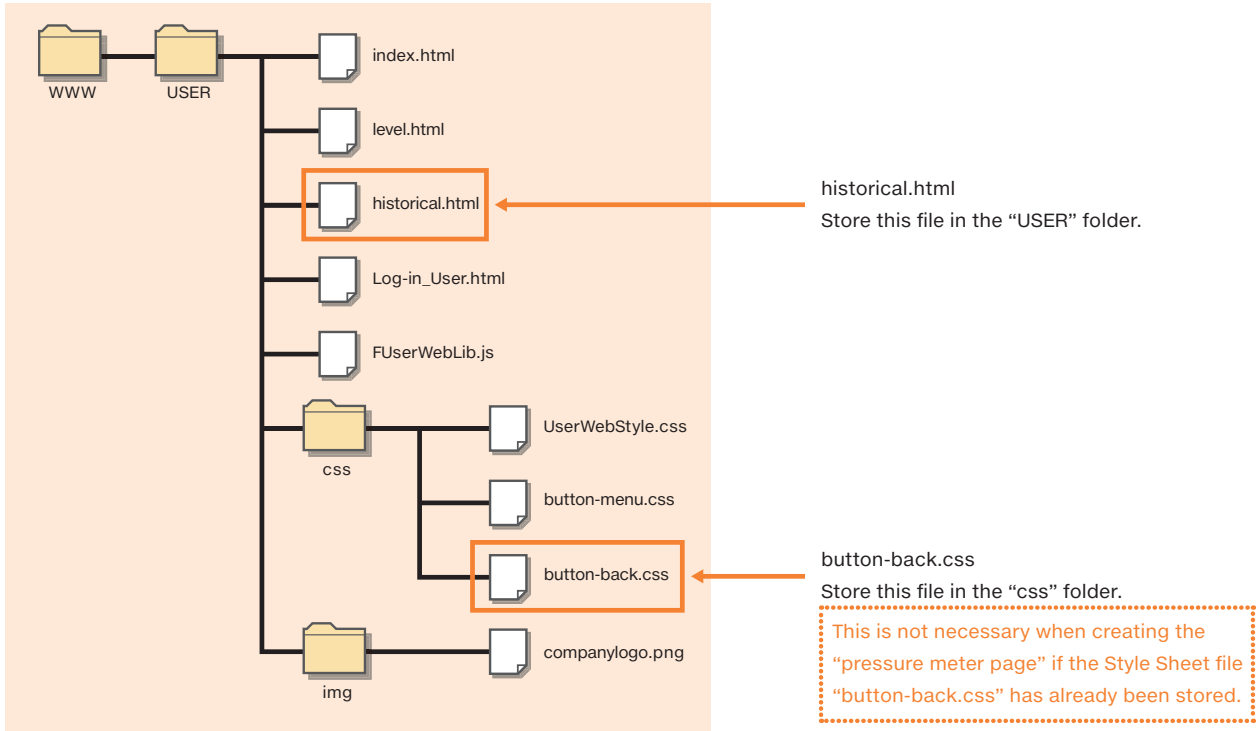
# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

- Before Starting
- Preparation
- Reuse
- User Web Page Creation**
- File Save Destination
- Troubleshooting
- Menu page creation
- Pressure meter page creation
- Temperature change page creation**
- Device monitor page creation

## STEP 10. File storage destination.

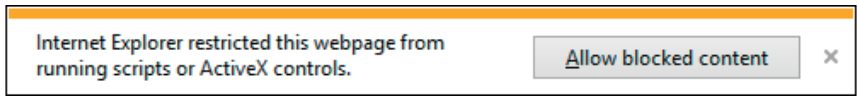
Store the created files so that the file structure is as shown below.

[File structure after organization]

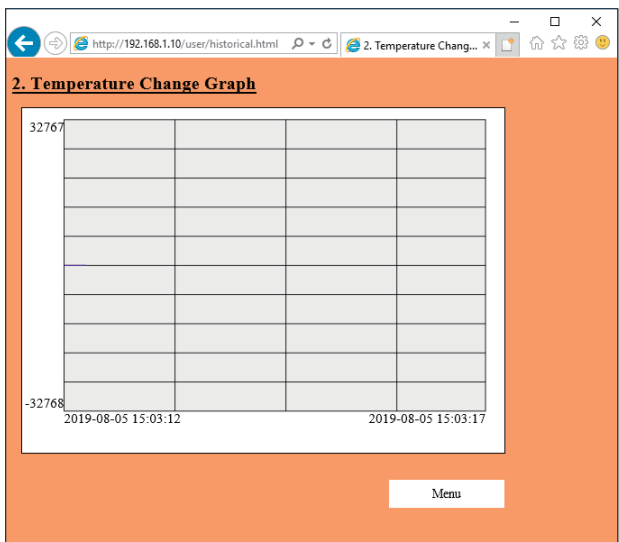


## STEP 11. Check whether the page can be displayed correctly in a Web browser.

Double-click the "historical.html" file.  
If the following dialog box is displayed, click "Allow blocked content".



Correctly displayed Web page



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

## Device monitor page creation

This section explains the procedure for creating the device monitor page.

Device reading CGI, device writing CGI, and a button for switching the Web page are displayed on the page created with this example. Use a Style Sheet to design the buttons for switching the Web page.

**This Web page does not reuse index.html. Refer to the following chapter to create the HTML.**

**→ 6. CREATING A DEVICE MONITOR WINDOW WITH CGI**

### Specifications

[Design example]

**Text display**

**Device reading CGI**

Device name	Data type	Value
D10	16-bit integer	1234
D11	32-bit integer	123456
M0	Bit	0

**Device writing CGI**

Device name	Data type	Value	Write
D10	16-bit integer	100	Write
D11	32-bit integer	600	Write
M0	Bit	1	Write

**Web page switching button display**

**Menu**

**Background color**

- Color name: mediumseagreen
- Color code: #3cb371

**Link destination**

- index.html

### Functions to use

Function	Creation method	Example/special note	Reference section
Background color	Style Sheet	Write a Style Sheet inside the HTML file.	7.1
Text display	Font size	<h3></h3> tags <h3>Heading 3</h3>	7.2
	Underline	<u></u> tags <u>Text in this range is underlined.</u>	
	Paragraph	<p></p> tags Use <p></p> to indicate paragraphs and insert line breaks.	
Device reading CGI		Reuse the CGI example (HTML) in this guide to create the component.	6.4
	Devices	D10, D11, and M0	
	Link file	RdDevRnd.cgi file (built into the PLC)	
Device writing CGI		Reuse the CGI example (HTML) in this guide to create the component.	6.5
	Devices	D10, D11, and M0	
	Link file	WrDev.cgi file (built into the PLC)	
Web page switching button display (return to the menu)	Design	Style Sheet When the button is clicked, its display changes to that shown on the right. <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px 5px;">Text</div> <div style="border: 1px solid black; padding: 2px 5px;">Text</div> </div> <p style="text-align: center;">Before being clicked                      After being clicked</p>	7.1
	Page switching	<a></a> tags <a href="link destination URL">Displayed text</a>	
	Display starting coordinates	Style Sheet Embed a Style Sheet in the <a> tag to display the button in the lower-right corner.	

### Files to create

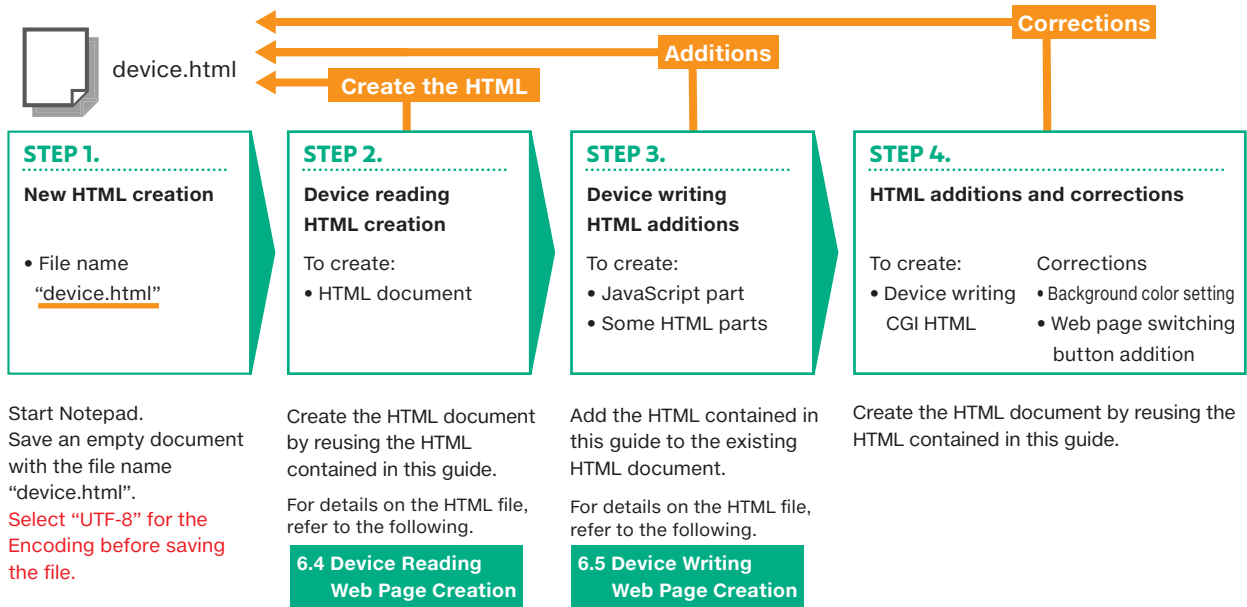
File type	File name	Remarks
HTML	device.html	Linked from the menu page. If the file name is changed, the menu page needs to be corrected.
Style Sheet	button-back.css	This file can be named as desired. If you change it, also change the file name of the Style Sheet file in the link settings of the HTML file.

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting	Preparation	Reuse	<b>User Web Page Creation</b>	File Save Destination	Troubleshooting
			Menu page creation	Pressure meter page creation	Temperature change page creation
					<b>Device monitor page creation</b>

## ► Procedure

This section explains an outline procedure for using CGI objects to create HTML for reading/writing from/to devices.



## STEP 1. Create a new HTML file for the device monitor page.

1. Start Notepad in Windows®.
2. On the Notepad toolbar, select [File] - [Save As].
3. In the File name field, enter "device.html", select "UTF-8" for the Encoding, and then click [Save]. Save the file in an easy-to-understand location.

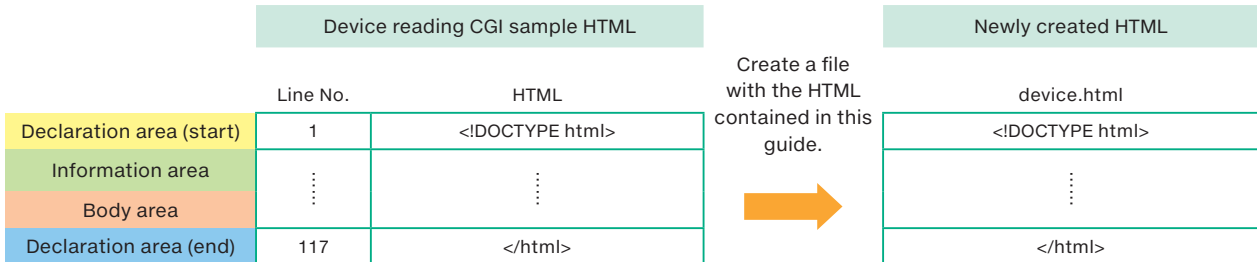
Starting with STEP2, use "device.html". Exercise caution to avoid correcting the wrong file.

## STEP 2. Create the device reading CGI HTML.

Reuse the entire HTML written on the following page (<DOCTYPE html> to </html>) to create the HTML file.

**Related Page** 6.4 Device Reading Web Page Creation - HTML creation example - HTML

[Created details]



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

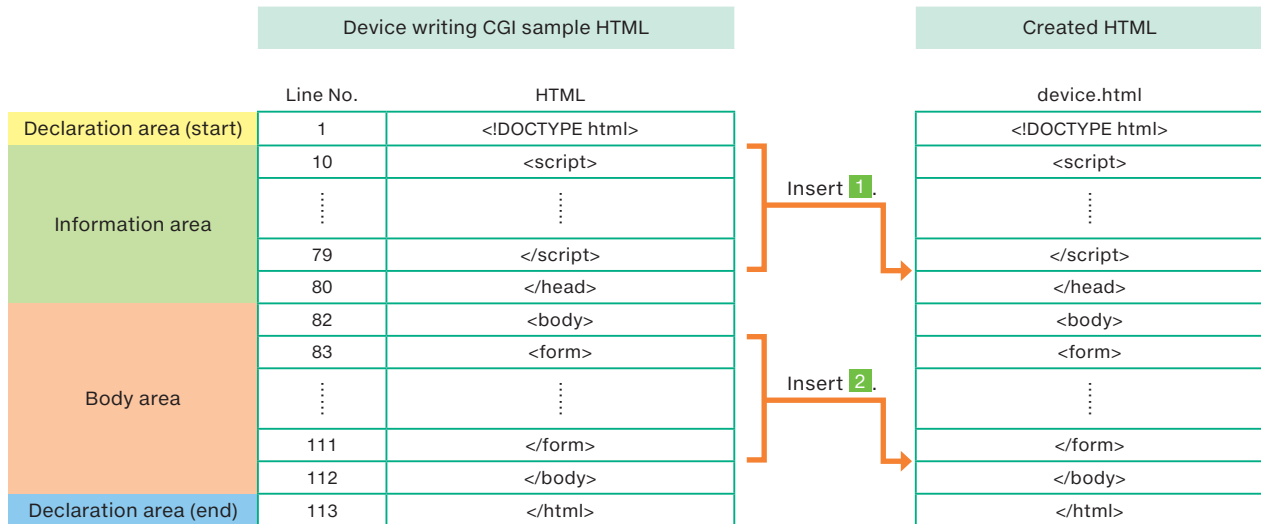
## STEP 3. Add the device writing CGI (sample HTML).

Add two sections from the device writing CGI written in this guide to the following locations.

**Related Page** 6.5 Device Writing Web Page Creation - HTML creation example - HTML

- 1 Insert lines 10 (<script>) to 79 (</script>) of the device writing CGI sample HTML in the space between </script> and </head> in device.html.
- 2 Insert lines 83 (<form>) to 111 (</form>) of the device writing CGI sample HTML in the space between </form> and </body> in device.html.

[Created details]



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

## STEP 4. Correct HTML for coexistence with the device writing CGI.

Make corrections to use both the device reading CGI and a device writing CGI on a single Web page.

In this example, the device writing CGI ID names are corrected to values between 10 and 19 to prevent them from duplicating the ID names used with the device reading CGI.

**If these corrections are not made, the PLC and device reading/writing cannot be performed correctly.**

[Correction details]

(1)			(2)			(3)		
Line No.	Before change	After change	Line No.	Before change	After change	Line No.	Before change	After change
210	id="DEV1"	id="DEV10"	210	id="DEV2"	id="DEV11"	222	id="DEV3"	id="DEV12"
211	id="TYP1"	id="TYP10"	211	id="TYP2"	id="TYP11"	223	id="TYP3"	id="TYP12"
212	id="DATA1"	id="DATA10"	212	id="DATA2"	id="DATA11"	224	id="DATA3"	id="DATA12"

[HTML after correction]

Line No.	HTML
201	<form>
202	<table id="devtbl2" class="devtbl" border="1">
203	<tr>
204	<th>Device name</th>
205	<th>Data type</th>
206	<th>Value</th>
207	</tr>
208	<tbody>
209	<tr> Before change After change
210	<td><input type="text" id="DEV10" name="DEV1" class="input" value="D10"/></td>
211	<td><input type="text" id="TYP10" name="TYP1" class="input" value="16-bit integer"/></td>
212	<td><input type="text" id="DATA10" name="DATA1" class="input" value="3"/></td>
213	<td><input type="button" value="Write" class="input" onclick="WriteDeviceBlockTbl('devtbl2',1)"/></td>
214	</tr>
215	<tr>
216	<td><input type="text" id="DEV11" name="DEV1" class="input" value="D11"/></td>
217	<td><input type="text" id="TYP11" name="TYP1" class="input" value="32-bit integer"/></td>
218	<td><input type="text" id="DATA11" name="DATA1" class="input" value="10"/></td>
219	<td><input type="button" value="Write" class="input" onclick="WriteDeviceBlockTbl('devtbl2',2)"/></td>
220	</tr>
221	<tr>
222	<td><input type="text" id="DEV12" name="DEV1" class="input" value="M0"/></td>
223	<td><input type="text" id="TYP12" name="TYP1" class="input" value="Bit"/></td>
224	<td><input type="text" id="DATA12" name="DATA1" class="input" value="1"/></td>
225	<td><input type="button" value="Write" class="input" onclick="WriteDeviceBlockTbl('devtbl2',3)"/></td>
226	</tr>
227	</tbody>
228	</table>
229	</form>

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting	Preparation	Reuse	User Web Page Creation	File Save Destination	Troubleshooting
			Menu page creation	Pressure meter page creation	Temperature change page creation
					Device monitor page creation

## STEP 5. Set the background color.

Add the HTML for the background color between lines 9 and 10.  
If the background color is white, there is no need to add this HTML.

Line No.	HTML
:	:
7	<!-- Set the title. -->
8	<title>Device Reading CGI Sample</title>
9	<!-- Write the user JavaScript here. -->
	<style>
	body {
	background-color: mediumseagreen;
	}
	</style>
10	<script>
11	// CGI request function
12	function ReadDeviceRandomTbl(devtblid) {

## STEP 6. Change the page title and add a Web page switching button.

Change the page title on line 8, and then add the HTML for the Web page switching button between lines 158 and 159.

Line No.	HTML
:	:
8	<title>3. Device Monitor</title> ← Page title text display
:	:
156	</script>
157	</head>
158	<body>
	<!-- Web page title text display -->
	<h2><u>3. Device Monitor</u></h2>
	<!-- Display of Web page switching button -->
	<a href="index.html" class="menu" style="position: absolute; left: 400px; top: 350px;" />Menu</a>
159	<form>
160	<table id="devtbl" class="devtbl" border="1">

Before change

Device name	Data type	Value
D10	16-bit integer	1234
D11	32-bit integer	123456
M0	Bit	0

Read

Device name	Data type	Value	
D10	16-bit integer	100	Write
D11	32-bit integer	600	Write
M0	Bit	1	Write

After change

Device name	Data type	Value
D10	16-bit integer	1234
D11	32-bit integer	123456
M0	Bit	0

Read

Device name	Data type	Value	
D10	16-bit integer	100	Write
D11	32-bit integer	600	Write
M0	Bit	1	Write

Menu

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting	Preparation	Reuse	User Web Page Creation	File Save Destination	Troubleshooting	
			Menu page creation	Pressure meter page creation	Temperature change page creation	Device monitor page creation

## STEP 7. Add the <link> setting to Style Sheet file for Web page switching button.

Add the HTML for the <link> settings of the Style Sheet file for the Web page switching button between lines 8 and 9.

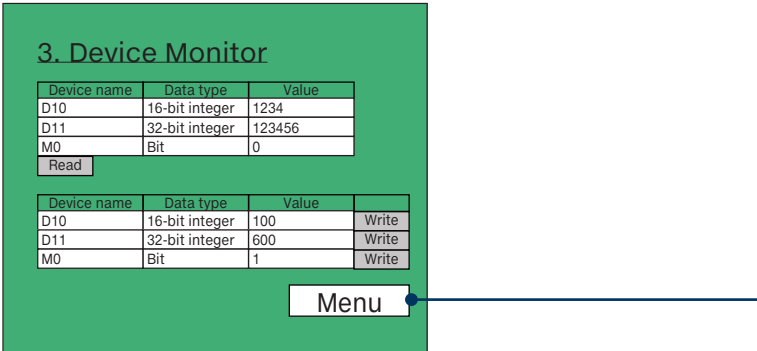
Line No.	HTML
:	:
7	<!-- Set the title. -->
8	<title>3. Device Monitor</title>
	<link href="/css/button-back.css" rel="stylesheet" media="all" />
9	<!-- Write the user JavaScript here. -->

## STEP 8. Create the Style Sheet file for Web page switching button.

1. Start Notepad in Windows®.
2. Create the Style Sheet shown below.
3. On the Notepad toolbar, select [File] - [Save As].
4. In the File name field, enter "button-back.css", and then click [Save].

This button is used on the "pressure meter page", "temperature change page", and "device monitor page".

[Design example] Indicate the design of the Web page switching button.



[Style Sheet]

class attributes: menu

```

a.menu{
display: block;
text-decoration: none;
height: 37px;
width: 250px;
line-height: 37px;
text-align: center;
color: #fff;
background: #4169e1;
-webkit-transition: 0.3s;
-moz-transition: 0.3s;
-o-transition: 0.3s;
-ms-transition: 0.3s;
transition: 0.3s;
}
a.menu:hover{
background: #fff;
color: #4169e1;
border:solid 1px #4169e1;
}
    
```



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

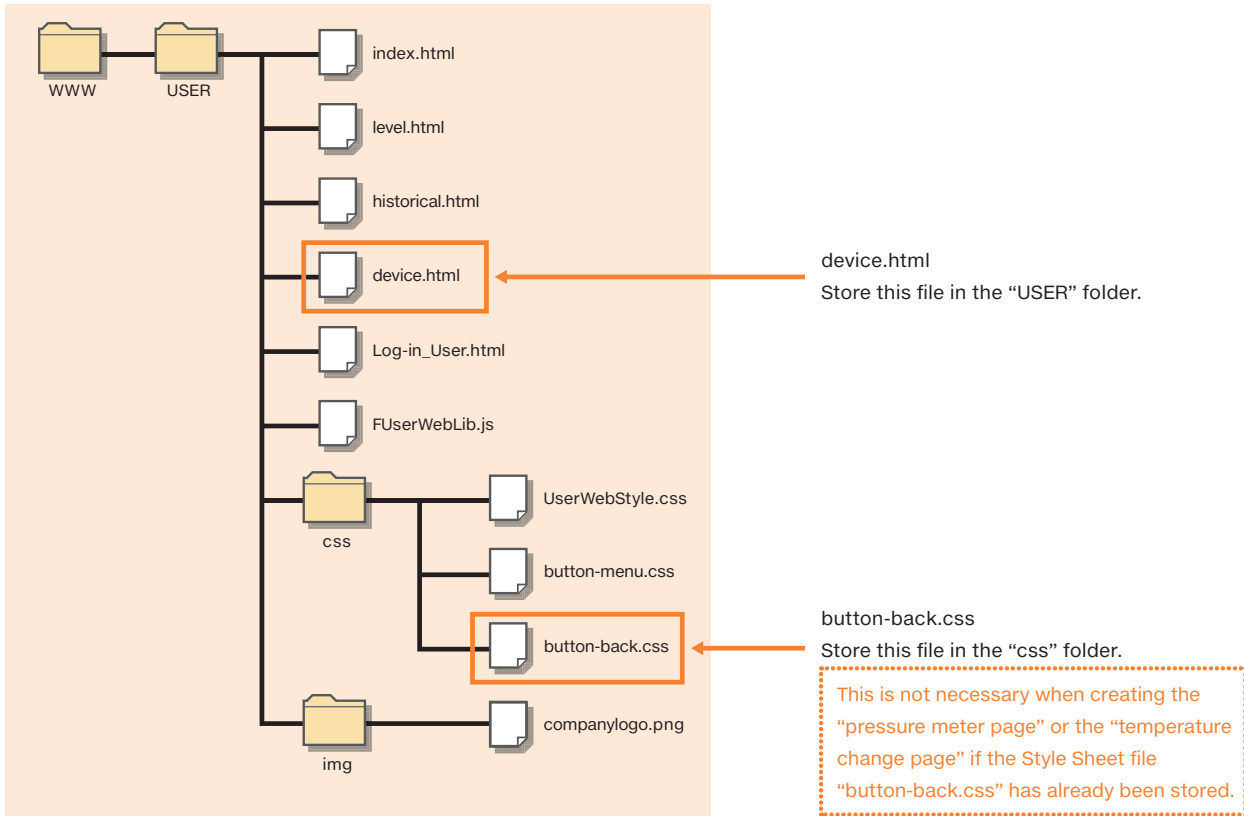
Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

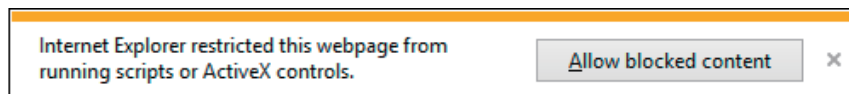
## STEP 9. File storage destination.



## STEP 10. Check whether the page can be displayed correctly in a Web browser.

Double-click the "device.html" file.

If the following dialog box is displayed, click "Allow blocked content".



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

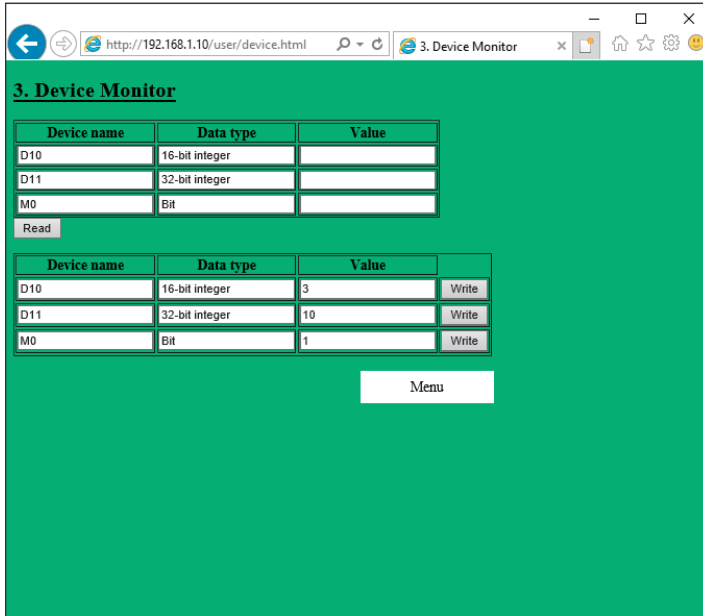
Menu page creation

Pressure meter page creation

Temperature change page creation

Device monitor page creation

Correctly displayed Web page



This completes the creation of all the user Web pages.

If Web pages are not displayed correctly, refer to 3.6 Troubleshooting.

For details on operations, refer to the following.

<b>Relevant documents</b>	Web Server Function Application Guide Using Web Page Startup and Introduction [manual number: L(NA)08643ENG]
---------------------------	---

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting | Preparation | Reuse | User Web Page Creation | **File Save Destination** | Troubleshooting

## 3.5 File Save Destination

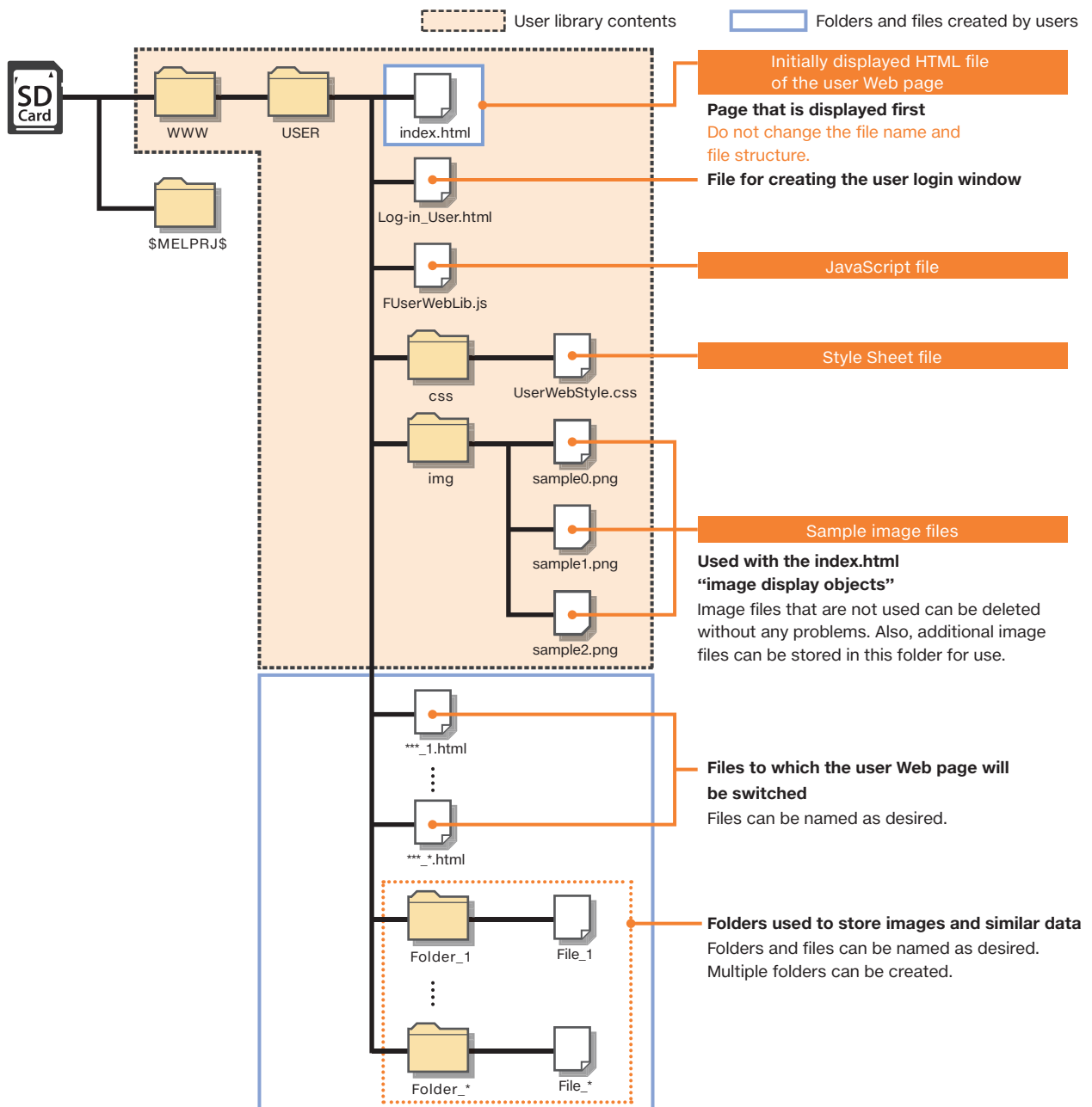
Store the created user Web server files on an SD memory card. This section explains the destination in which to save the user Web server files. If you want to change the file structure, consider this action by referring to the following explanation.

### User Web page library file structure

This section explains the file structure of the user Web page library.

The following figure shows the file structure stored on the SD memory card with the file group provided by the user Web page library and the user-created file group. Changing the folder structure or file names of the user library may prevent normal operation.

#### Web server file group



# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting

Preparation

Reuse

User Web Page Creation

File Save Destination

Troubleshooting

There is normally no need to change the folder structure or file names, but use the information here as a reference when sorting by folder and when storing other JavaScript files in the folders. The explanation here covers JavaScript and Style Sheet files.

## JavaScript files

### File name

File name	Function	Description	Default storage destination
FUserWebLib.js	Library of JavaScript objects	A file included in the user Web page library	./FUserWebLib.js

### HTML document

This section explains how to read JavaScript files from HTML files and how to change the path according to the file storage location.

#### When the “index.html” and “FUserWebLib.js” files are on the same level (default storage destination)

HTML document	Folder and file structure
<code>&lt;script src=“./FUserWebLib.js”&gt;&lt;/script&gt;</code>	

#### When the “FUserWebLib.js” file is stored in a user-created folder (example: js)

HTML document	Folder and file structure
<code>&lt;script src=“./js/FUserWebLib.js”&gt;&lt;/script&gt;</code>	

### How to write statements in the HTML file

Write the statements between the `<head>` and `</head>` tags. However, write the statements above the JavaScript starting line (`<script>`).

Line No.	HTML
1	<code>&lt;!DOCTYPE html&gt;</code>
2	<code>&lt;html xmlns=“http://www.w3.org/1999/xhtml”&gt;</code>
3	<code>&lt;head&gt;</code>
4	<code>&lt;!-- charset setting *Because the Web server setting is UTF-8, specify UTF-8. --&gt;</code>
5	<code>&lt;meta charset=“UTF-8”&gt;</code>
6	<code>&lt;meta http-equiv=“X-UA-Compatible” content=“IE=edge”/&gt;</code>
7	<code>&lt;!-- Set the title. --&gt;</code>
8	<code>&lt;title&gt;Sample&lt;/title&gt;</code>
9	<code>&lt;link href=“./css/UserWebStyle.css” rel=“stylesheet” media=“all” /&gt;</code>
10	<code>&lt;!-- Reading the library JavaScript (Change the path to match the storage location.) --&gt;</code>
11	<code>&lt;script src=“./FUserWebLib.js”&gt;&lt;/script&gt;</code>
12	<code>&lt;!-- Write the proprietary JavaScript from this point. --&gt;</code>
13	<code>&lt;!-- Write the user JavaScript here. --&gt;</code>
14	<code>&lt;script&gt;</code>
⋮	⋮
⋮	<code>&lt;/head&gt;</code>

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

Before Starting | Preparation | Reuse | User Web Page Creation | **File Save Destination** | Troubleshooting

## Style Sheet files

### File name

File name	Function	Description	Default storage destination
UserWebStyle.css	Style settings	File with commonly defined styles	./css/UserWebStyle.css

### HTML document

This section explains how to read CSS files from HTML files.

It may be necessary to change the path depending on the file storage location.

#### When the “index.html” and “UserWebStyle.css” files are on the same level

HTML document	Folder structure on the SD memory card
<pre>&lt;script src="./UserWebStyle.css"&gt;&lt;/script&gt;</pre>	

#### When the “UserWebStyle.css” file is stored in the css folder (default storage destination)

HTML document	Folder structure on the SD memory card
<pre>&lt;script src="./css/UserWebStyle.css"&gt;&lt;/script&gt;</pre>	

### How to write statements in the HTML file

Write the statements between the `<head>` and `</head>` tags. However, write the statements above the JavaScript starting line (`<script>`).

Line No.	HTML
1	<code>&lt;!DOCTYPE html&gt;</code>
2	<code>&lt;html xmlns="http://www.w3.org/1999/xhtml"&gt;</code>
3	<code>&lt;head&gt;</code>
4	<code>&lt;!-- charset setting *Because the Web server setting is UTF-8, specify UTF-8. --&gt;</code>
5	<code>&lt;meta charset="UTF-8"&gt;</code>
6	<code>&lt;meta http-equiv="X-UA-Compatible" content="IE=edge"/&gt;</code>
7	<code>&lt;!-- Set the title. --&gt;</code>
8	<code>&lt;title&gt;Sample&lt;/title&gt;</code>
9	<code>&lt;link href="./css/UserWebStyle.css" rel="stylesheet" media="all" /&gt;</code>
10	<code>&lt;!-- Reading the library JavaScript (Change the path to match the storage location.) --&gt;</code>
11	<code>&lt;script src="./UserWebLib.js"&gt;&lt;/script&gt;</code>
12	<code>&lt;!-- Write the proprietary JavaScript from this point. --&gt;</code>
13	<code>&lt;!-- Write the user JavaScript here. --&gt;</code>
14	<code>&lt;script&gt;</code>
...	<code>...</code>
	<code>&lt;/head&gt;</code>

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

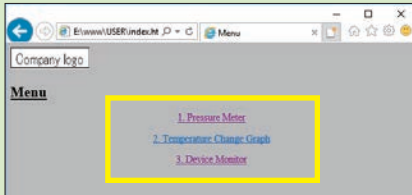
Before Starting | Preparation | Reuse | User Web Page Creation | File Save Destination | Troubleshooting

## 3.6 Troubleshooting

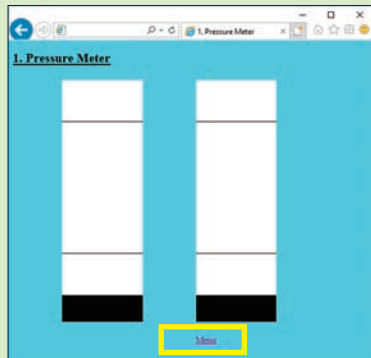
### Problem details

Web page switching buttons are displayed as underlined text, not as buttons.

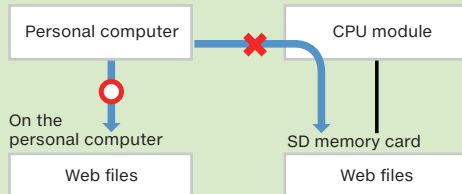
[Example 1]



[Example 2]



For the HTML file on the personal computer, the Web page switching button can be displayed with the correct style in a Web browser. However, it cannot be displayed correctly on the PLC-side Web page.



### Points to check

The Style Sheet file has not been read. Check the following details.

- The storage location of the Style Sheet is incorrect.
- The file name is incorrect.

[Example 1] When the Style Sheet file is stored in the css folder

**Bad** `<link href="/css/button-menu.css" rel="stylesheet" media="all"/>`  
**Good** `<link href="/css/button-menu.css" rel="stylesheet" media="all"/>`

[Example 2] When the Style Sheet file is stored in the css folder

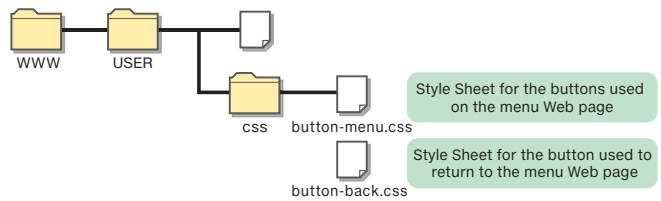
**Bad** `<link href="button-back.css" rel="stylesheet" media="all" />`  
**Good** `<link href="/css/button-back.css" rel="stylesheet" media="all" />`

- There is a mismatch involving the name in the Style Sheet class attributes: the name written on the HTML button does not match the name written in the Style Sheet file.

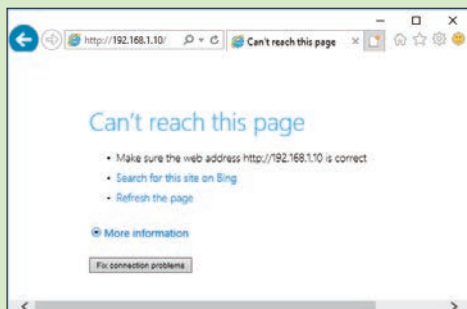
[Example 1]

**Bad** HTML `<a href="level.html" class="button1">1. Pressure Meter</a>`  
 Style Sheet `a.button{`  
**Good** HTML `<a href="level.html" class="button1">1. Pressure Meter</a>`  
 Style Sheet `a.button1{`

Check whether the Style Sheet files (button-menu.css and button-back.css) are stored on the SD memory card.



The Web page is not displayed in a Web browser. An error is displayed instead.



All the HTML files cannot be referenced from the HTML file that you attempted to display. Check the following details.

- The file name is incorrect.
- The storage locations of the link destination HTML files are incorrect.

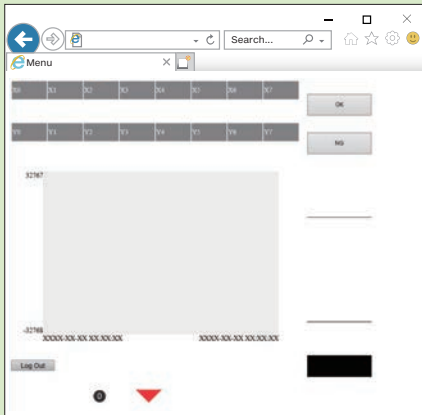
[Ex.] When all the HTML files are stored in the same location

`<a href="level.htm" class="button1">1. Pressure Meter</a>`  
 → File name extension mistake  
`<a href="/user/historical.html" class="button2">2. Temperature Change Graph</a>`  
 → Storage location specification mistake  
`<a href="device.html" class="button3">3. Device Monitor</a>`  
 → ["] missing after device.html

# 3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE

## Problem details

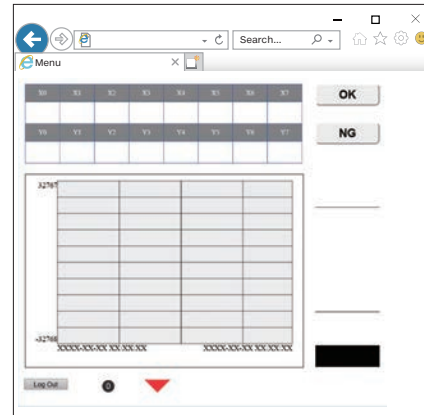
Table or graph lines are not displayed. Alternatively, OK, NG, and other buttons are displayed with the standard design of the Web browser.



## Points to check

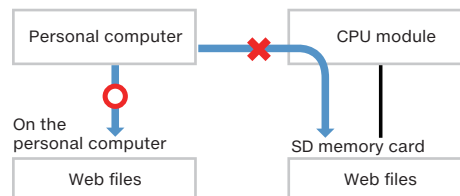
The Style Sheet file (UserWebStyle.css) cannot be referenced from the HTML file.

[Correct display status]



Possible causes are listed below.

- (1) The link specification `<link href="file name" rel="stylesheet" media="all" />` is not written in the HTML.
  - HTML
    - `<link href="./css/UserWebStyle.css" rel="stylesheet" media="all" />`
- (2) The Style Sheet file name does not match the HTML link specification `<link href="file name" rel="stylesheet" media="all" />`.
  - HTML
    - `<link href="./css/UserWebStyle.css" rel="stylesheet" media="all" />`
  - Style Sheet file name: [UserWebStyle.css]
- (3) The storage location of the Style Sheet file in the HTML link specification is incorrect.
  - HTML
    - `<link href="./css/UserWebStyle.css" rel="stylesheet" media="all" />`
  - Save destination: [www/css folder]
- (4) If HTML files on the personal computer can be displayed in a Web browser but the PLC-side Web page cannot be displayed correctly, check whether the Style Sheet file (UserWebStyle.css) is stored on the SD memory card.



# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

What Are Style Sheets?

Common Items

Overall Structure

Setting Details

Style Sheet Details

## 4.1 Function Introduction

A Style Sheet is a way to change the style of a Web page such as its design and layout. Whereas HTML defines the elements and structure of the Web page, the Style Sheet specifies how to decorate the elements and structure.

When specifying the style of a Web page created with a language such as HTML or XHTML, CSS (Cascading Style Sheets), one type of Style Sheet languages is most commonly used. Therefore, Style Sheets may generally be referred to as "CSS".

### ► Web page that does not use a Style Sheet

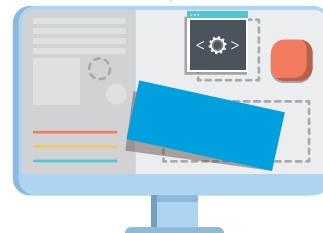
Web page created with HTML alone, without using a Style Sheet, results in a dull page in which the text and images are arranged from top to bottom. HTML is used to write the text documents that are displayed in browsers.



Create the foundation of the Web page with HTML.

### ► Web page that uses a Style Sheet

In relation to the foundational Web page created with HTML, "design" and "layout" are controlled with the Style Sheet. A Style Sheet can be used to arrange the appearance of the page by changing the text and background colors, drawing lines in various colors, and adjusting the margins.



The colors, lines, and layout can be changed with a Style Sheet.

#### Point

Use a CSS file to arrange the appearance of the text displayed in a browser with an HTML file. Separating the text and appearance into different files has a variety of advantages including making it possible to write these items by focusing on their separate details and making it easier to divide work among multiple people.

### ► An example of what can be done with Style Sheets

- When the page is created only with HTML

#### HTML code

```
<p>This is an example sentence. </p>
```

#### Browser display

This is an example sentence.

- Change the character color to orange.

#### HTML code

```
<p>This is an example sentence. </p>
```

#### CSS code

```
p {
  color: orange
}
```

#### Browser display

This is an example sentence.

- Change the font size to 20 px.

#### HTML code

```
<p>This is an example sentence. </p>
```

#### CSS code

```
p {
  color: orange
  font-size: 20px
}
```

#### Browser display

This is an example sentence.



# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

What Are Style Sheets?

Common Items

Overall Structure

Setting Details

Style Sheet Details

## 4.2 Common Items

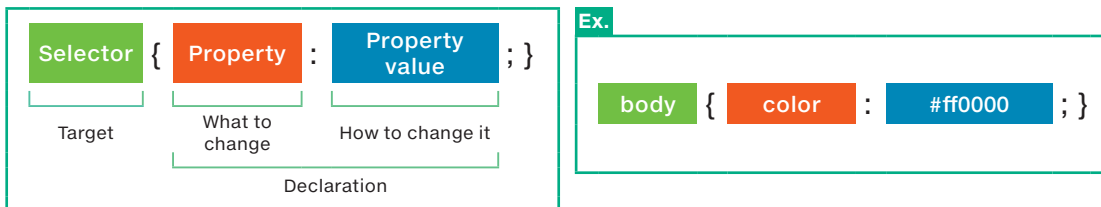
This section explains the common items that you need to know to use Style Sheets.

### Terminology

Item	Description
Selector	Target to which the Style Sheet will be applied.
Property	Name of the style to apply.
Property value	Details of the style to apply.

### Style Sheet writing rules

Each Style Sheet statement consists of a selector, property, and property value.



- The “property:property value” pair is called a **Declaration**.
- The declaration block for the selector is enclosed in curly brackets ( `{ }` ).
- Multiple declarations can be specified. Each declaration is delimited with a semicolon ( `;` ).

### Style Sheet setting types

There are three ways to set the Style Sheet.

#### ► Read the Style Sheet from a CSS file with a `<link>` tag.

Write the Style Sheet into a file with the “.css” extension, and then link to the CSS file from an HTML file to read the Style Sheet. To link to the CSS file from an HTML file, use the `<link>` tag within the `<head>` tag.

**Ex.** Linking to “UserWebStyle.css” in the user Web page library

```
<head>
  <link href="./css/UserWebStyle.css" rel="stylesheet" media="all" />
</head>
```

#### ► Write the Style Sheet with the `<style>` tag.

Use the `<style>` tag to write the Style Sheet inside the `<head>` tag in an HTML document.

**Ex.** Adding a Style Sheet that makes the body character color red

```
<head>
  <style>
    body { color: #ff0000; }
  </style>
</head>
```

#### ► Add the Style Sheet to the elements.

Add the Style attribute to an element to write the Style Sheet directly. When you write the Style Sheet directly in an element, the selector is not necessary.

**Ex.** Adding a Style Sheet to the `<h1>` Heading 1 `<h1>` element

```
<h1 style="color: #ff0000;">Heading 1</h1>
```

# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

[What Are Style Sheets?](#)
[Common Items](#)
[Overall Structure](#)
[Setting Details](#)
[Style Sheet Details](#)

## Style Sheet setting example

This section shows a setup example of using the <style> tag to write a Style Sheet.

To set the character color (property) of the body (selector) to red (property value), write the Style Sheet as shown below.

```
body { color:#ff0000; }
```

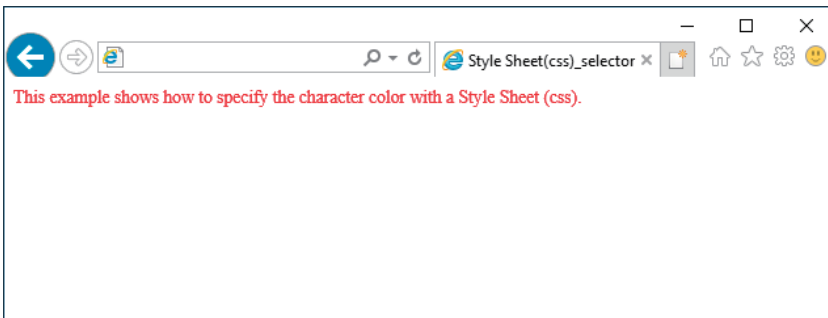
The above Style Sheet changes the color of the text between the <body> and </body> tags to “#ff0000 (red)”.

## ▶ HTML example

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charset setting *Because the Web server setting is UTF-8, specify UTF-8.) -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- Set the title. -->
    <title>Style Sheet(css)_selector</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta http-equiv="Content-Style-Type" content="text/css">

    <!--Character color specification for the text between <body> and </body> in the Web page-->
    <style>
      body { color:#ff0000; }
    </style>
  </head>
  <body>
    This example shows how to specify the character color with a Style Sheet (css).
  </body>
</html>
```

## ▶ Web browser display



# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

What Are Style Sheets?

Common Items

Overall Structure

Setting Details

Style Sheet Details

## 4.3 Overall Structure (UserWebStyle.css File)

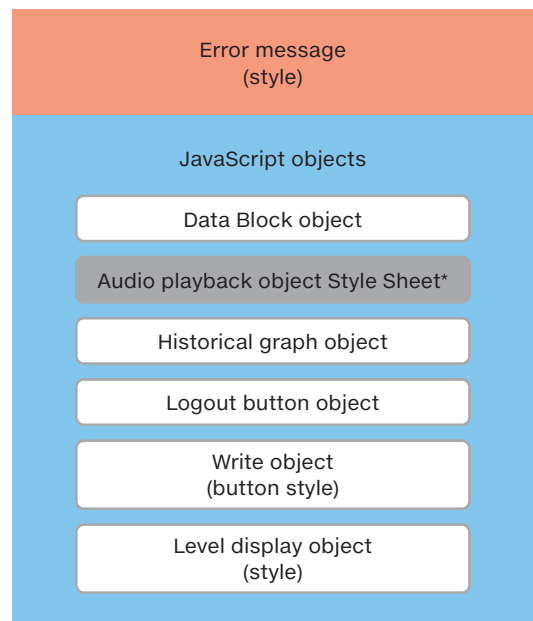
The HTML document <link> tag is used to read the Style Sheet file.



Defines the button styles and the character color for messages displayed when there are errors in the created HTML or JavaScript objects.

Each object has separate style settings.

### Internal structure of the Style Sheet file



\* This is not used by the user Web page library HTML file (index.html) from the Mitsubishi Electric representative.

## 4.4 Setting Details

The style of each object set with JavaScript takes priority, and even if the settings are changed with the Style Sheet (UserWebStyle.css), these changes are not applied. However, if specific settings are omitted in the JavaScript, the Style Sheet settings are applied.

Related Page

For details, refer to the following.  
5.1 What Is JavaScript (JS)? - Style Sheet (CSS)

Special Note

If you change the Style Sheet (UserWebStyle.css) in the user Web page library, do not change the Style Sheet class names. However, do not define Style Sheet class names with the same names.

# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

What Are Style Sheets?

Common Items

Overall Structure

Setting Details

Style Sheet Details

## 4.5 Style Sheet Details

This section explains the details of the Style Sheet file (UserWebStyle.css) in the user Web page library.

### Related Page

For details on Style Sheets, refer to the following.  
7.1 Style Sheet References

### Caution

When you change a Style Sheet file, setting parameters and values incorrectly will prevent the objects from being displayed in the Web browsers. Be sure to save a copy of the file before making changes.

### Error message style

#### ► Error message display example

The screenshot shows a web browser window with several error messages and a table. The error messages are:

- Setting of parameter "dev[0].base" is incorrect.
- Setting of parameter "dev[1].base" is incorrect.
- Setting of parameter "dev[2].base" is incorrect.
- Setting of parameter "dev[3].base" is incorrect.
- Setting of parameter "dev[4].base" is incorrect.
- Setting of parameter "dev[5].base" is incorrect.
- Setting of parameter "dev[6].base" is incorrect.
- Setting of parameter "dev[7].base" is incorrect.
- Invalid data format for device "D0"
- Invalid data format for device "D1"
- Invalid data format for device "D2"
- Invalid data format for device "D3"
- Invalid data format for device "D4"
- Invalid data format for device "D5"
- Invalid data format for device "D6"
- Invalid data format for device "D7"
- Setting of parameter "dev" is incorrect.
- Invalid data format for device "D0"
- Setting of parameter "direction" is incorrect.

The table has 4 columns and 10 rows. The first row contains the number -32768. The second and third rows contain the text XXXX-XX-XX XX:XX:XX. The fourth row contains the number 123. The fifth row contains the text NG. The sixth row contains the number 123. The seventh row contains the number 123. The eighth row contains the number 123. The ninth row contains the number 123. The tenth row contains the number 123.

Labels in the image point to:

- Error message display button**: A red square button with a white exclamation mark.
- Error message**: The text "Invalid data format for device "D0"".
- Error message display field**: A gray rectangular field containing the text "NG".

# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

What Are Style Sheets?

Common Items

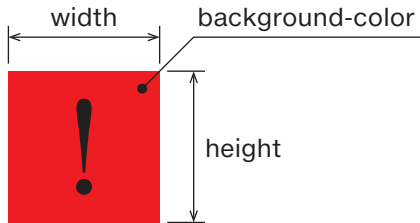
Overall Structure

Setting Details

Style Sheet Details

## ► List of parameters

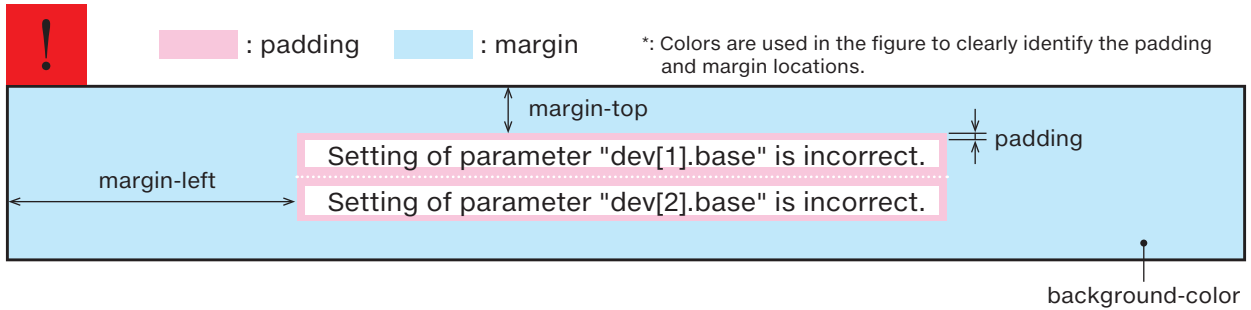
### • Error message display button



### • Cursor



### • Error message, error message display field



# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

What Are Style Sheets?

Common Items

Overall Structure

Setting Details

Style Sheet Details

## ► Style Sheet

Line No.	Style Sheet statement	Explanation
<b>Common settings</b>		<b>Settings for handling the display starting position</b>
1	.partsBase{	Selector
2	position: absolute;	Placement of the absolute position of the parent box
3	}	
<b>Global messages</b>		<b>Settings for the global message box</b>
4	.globalMessageArea{	Selector
5	position: absolute;	Placement of the absolute position of the parent box
6	top: 2px;	Distance from the reference position to the top edge of the box
7	left: 2px;	Distance from the reference position to the left edge of the box
8	z-index: 2147483647;	Specifies the box overlapping order.
9	background-color: white;	Background color
10	border: solid 1px black;	Parent box border specification Setting values [thickness: 1 pixel, line type: solid line, line color: black]
11	}	
<b>Error messages</b>		<b>Error message list line spacing</b>
12	.errorMessage{	Selector
13	margin: 0;	Outer margin specification (top, bottom, left, and right)
14	padding: 1px 0;	Inner margin specification (top and bottom: 1 pixel, left and right: 0 pixels)
15	}	
<b>Error message display field</b>		<b>Window that displays the error message list</b>
16	.errorField{	Selector
17	position: absolute;	Placement of the absolute position of the parent box
18	z-index: 2147483646;	Specifies the box overlapping order.
19	margin-top: 22px; /* errorButton height + margin-top */	Outer margin specification (top)
20	margin-left: 2px; /* Same value as errorButton margin-left */	Outer margin specification (left)
21	background-color: white;	Background color
22	border: solid 1px black;	Error message display field border specification Setting values [thickness: 1 pixel, line type: solid line, line color: black]
23	}	
<b>Display button</b>		<b>Button that displays the error message list</b>
24	.errorButton{	Selector
25	position: absolute;	Placement of the absolute position of the parent box
26	z-index: 2147483645;	Specifies the box overlapping order.
27	width: 20px;	Width
28	height: 20px;	Height
29	margin-top: 2px;	Outer margin specification (top)
30	margin-left: 2px;	Outer margin specification (left)
31	background-color: red;	Background color
32	border: solid 1px white;	Display button border specification Setting values [thickness: 1 pixel, line type: solid line, line color: white]
33	text-align: center;	Text centering
34	font-weight: bold;	Font decoration type specification Setting value [bold]
35	cursor: pointer;	Specifies the shape of the mouse cursor.
36	}	

# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

What Are Style Sheets?

Common Items

Overall Structure

Setting Details

Style Sheet Details

## Data Block object style

### List of parameters

Object 001	Object 002	Object 003	Object 004	Object 005
1000	900	800	700	600
Object 101	Object 102	Object 103	Object 104	Object 105
500	400	300	200	100

background-color:  
color:  
border:

### Style Sheet

Displayed with priority given to the JavaScript settings.

Line No.	Style Sheet statement	Explanation
37	<code>/* ----- */</code>	
38	<code>/* Data Block object */</code>	Parts enclosed in “/*” and “*/” are handled as comments.
39	<code>/* ----- */</code>	
40	<code>.datblk,</code>	
41	<code>.datblk_name,</code>	Selector
42	<code>.datblk_val{</code>	
43	<code>border: 1px solid black;</code>	Data cell border specification Setting values [thickness: 1 pixel, line type: solid line, line color: black]
44	<code>padding: 0;</code>	Inner margin specification (top, bottom, left, and right)
45	<code>text-align: center;</code>	Text alignment setting (center)
46	<code>color: black;</code>	Data cell character color specification
47	<code>background-color: white;</code>	Data cell background color specification
48	<code>}</code>	
49	<code>.datblk{</code>	Selector
50	<code>border-collapse: collapse;</code>	Specifies how to display the data cell line. Setting value [overlaps the lines of the adjacent cells]
51	<code>white-space: nowrap;</code>	Specifies how to display white space (consecutive single-byte spaces and tabs) and line breaks. Setting values [Handle line breaks as single-byte spaces and multiple consecutive spaces as a single space.]
52	<code>}</code>	
53	<code>.datblk_name,</code>	Selector
54	<code>.datblk_val{</code>	
55	<code>overflow: hidden;</code>	How to process text that protrudes outside of the cell Setting value [not displayed]
56	<code>}</code>	

## Audio playback object style

This function is not supported.

Line No.	Style Sheet statement	Explanation
57	<code>/* ----- */</code>	
58	<code>/* Audio playback object */</code>	Parts enclosed in “/*” and “*/” are handled as comments.
59	<code>/* ----- */</code>	
60	<code>.audio_btn{</code>	Selector
61	<code>padding: 0;</code>	
62	<code>text-align: center;</code>	
63	<code>color: black;</code>	
64	<code>overflow: hidden;</code>	
65	<code>}</code>	

# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

What Are Style Sheets?

Common Items

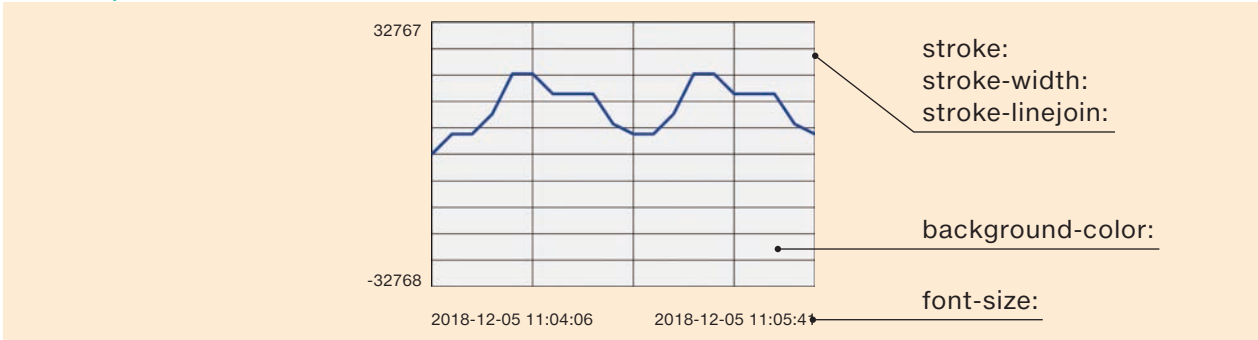
Overall Structure

Setting Details

Style Sheet Details

## Historical graph object style

### List of parameters



### Style Sheet

Line No.	Style Sheet statement	Explanation
66	<code>/* ----- */</code>	Parts enclosed in “/*” and “*/” are handled as comments.
67	<code>/* Historical graph object */</code>	
68	<code>/* ----- */</code>	
69	<code>.hstgrp{</code>	Selector
70	<code>border: 1px solid black;</code>	Not used
71	<code>background-color: white;</code>	
72	<code>}</code>	
73	<code>.hstgrp_frame{</code>	Selector
74	<code>stroke: black;</code>	Line color specification
75	<code>stroke-width: 1px;</code>	Line thickness specification
76	<code>stroke-linejoin: miter;</code>	Line joint and angle shape specifications
77	<code>}</code>	
78	<code>.hstgrp_text_x,</code>	Selector
79	<code>.hstgrp_text_y{</code>	
80	<code>font-size: 18px;</code>	Font size specification
81	<code>}</code>	
82	<code>.hstgrp_line{</code>	Selector
83	<code>stroke: blue;</code>	Not used
84	<code>stroke-width: 1px;</code>	
85	<code>stroke-linejoin: round;</code>	
86	<code>}</code>	



# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

What Are Style Sheets?

Common Items

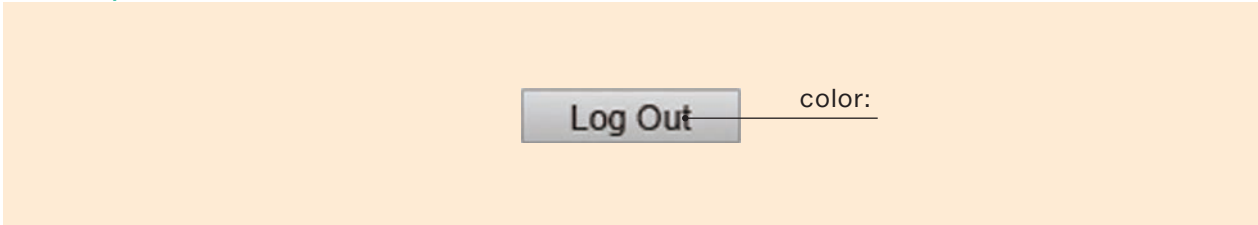
Overall Structure

Setting Details

Style Sheet Details

## Logout button object style

### ► List of parameters



### ► Style Sheet

Line No.	Style Sheet statement	Explanation
87	<code>/* ----- */</code>	Parts enclosed in “/*” and “*/” are handled as comments.
88	<code>/* Logout button object */</code>	
89	<code>/* ----- */</code>	
90	<code>.logout_btn{</code>	Selector
91	<code>padding: 0;</code>	Inner margin specification (top, bottom, left, and right)
92	<code>text-align: center;</code>	Alignment specification of the text within the button
93	<code>color: black;</code>	Character color specification
94	<code>overflow: hidden;</code>	Specification of how to process text that protrudes outside of the cell
95	<code>}</code>	Setting value [not displayed]

## Write object (button style)

### ► List of parameters



### ► Style Sheet

Line No.	Style Sheet statement	Explanation
96	<code>/* ----- */</code>	Parts enclosed in “/*” and “*/” are handled as comments.
97	<code>/* Write object (button style) */</code>	
98	<code>/* ----- */</code>	
99	<code>.write_btn{</code>	Selector
100	<code>height:50px;</code>	Not used
101	<code>width:100px;</code>	
102	<code>font-weight:bold;</code>	Font decoration type specification Setting value [bold]
103	<code>font-famiry:Meiryo;</code>	Character font type specification Setting value [Meiryo]
104	<code>font-size:20pt;</code>	Font size specification
105	<code>border-radius:10px</code>	
106	<code>}</code>	

# 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

What Are Style Sheets?

Common Items

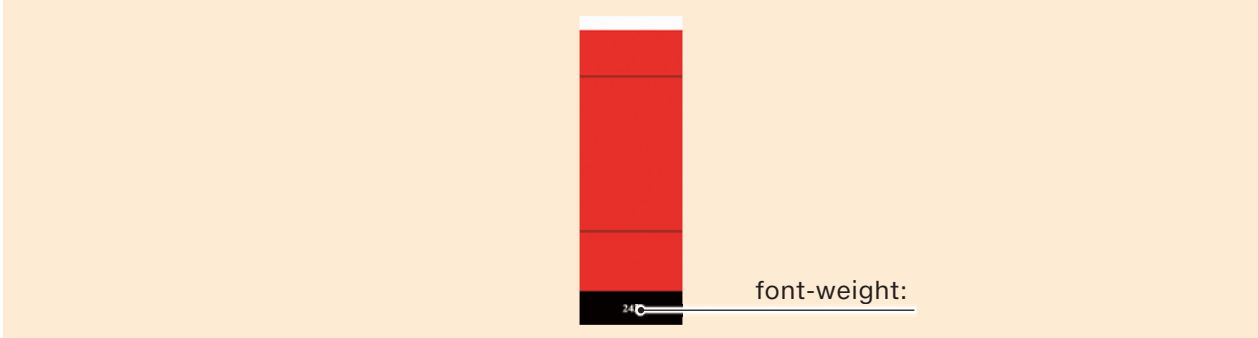
Overall Structure

Setting Details

Style Sheet Details

## Level display object (style)

### ► List of parameters



### ► Style Sheet

Line No.	Style Sheet statement	Explanation
107	<code>/* ----- */</code>	Parts enclosed in “/*” and “*/” are handled as comments.
108	<code>/* Level display object (style) */</code>	
109	<code>/* ----- */</code>	
110	<code>.partsLevelValue{</code>	Selector
111	<code>font-weight:bold;</code>	Font decoration type specification Setting value [bold]
112	<code>}</code>	

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

## 5.1 What Is JavaScript (JS)?

JavaScript is a language for adding functions to Web pages. This makes it possible to realize interactive operations such as displaying photograph slides and 2D/3D graphical animations.

By obtaining the user Web page library from the Mitsubishi Electric representative, you can use a JavaScript (JS) tool library.

This section explains the parameters of the JavaScript library objects. In this chapter, no details are listed regarding the general writing of JavaScript.

### List of objects

You can use the JavaScript objects in the user Web page library to read/write from/to devices from user Web pages. The objects in the following table are included JavaScript object library (FUserWebLib.js).

Table																					
Data Block object (WSDatblk)																					
<table border="1"> <thead> <tr> <th>Object 001</th> <th>Object 002</th> <th>Object 003</th> <th>Object 004</th> <th>Object 005</th> </tr> </thead> <tbody> <tr> <td>1000</td> <td>900</td> <td>800</td> <td>700</td> <td>600</td> </tr> <tr> <th>Object 101</th> <th>Object 102</th> <th>Object 103</th> <th>Object 104</th> <th>Object 105</th> </tr> <tr> <td>500</td> <td>400</td> <td>300</td> <td>200</td> <td>100</td> </tr> </tbody> </table>		Object 001	Object 002	Object 003	Object 004	Object 005	1000	900	800	700	600	Object 101	Object 102	Object 103	Object 104	Object 105	500	400	300	200	100
Object 001	Object 002	Object 003	Object 004	Object 005																	
1000	900	800	700	600																	
Object 101	Object 102	Object 103	Object 104	Object 105																	
500	400	300	200	100																	
Displays the specified device data as a table.																					
Graphs																					
Level display object (WSLevel)	Historical graph object (WSHstgrp)																				
Displays device value ratios as a bar graph.	Displays device data as a chronological order line graph.																				
Images																					
Image display object (WSPicture)	Figure display object (WSFigure)																				
Displays the specified image when the device value reaches the set value.	Displays the specified figure when the device value reaches the set value.																				
Operation buttons																					
Write button object (WSWrtBtn)	Logout button object (WSLogoutBtn)																				
Writes the specified value to the specified device. *: The text displayed on the button can be changed.	Executes the logout operation.																				

## 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

### Usable files

The following files can be used.

File	Extension	MIME type
HTML	.html	text/html
	.htm	text/htm
JavaScript	.js	text/javascript
CSS	.css	text/css
GIF image	.gif	image/gif
PNG image	.png	image/png
JPG/JPEG image	.jpgx/.jpeg	image/jpeg

### Terminology

#### Relationship between the extensions and MIME types of files on the Web

The concept of “extensions” and “MIME types” exists on the Web.

This is a mechanism for identifying the file type as text or HTML and for identifying the format of image files.

- The “extension” indicates what the file is. The extension is the last part of the file name from the dot (“.”) to the end.
- “MIME type” is a character string in the format of “type name/subtype name”. This “MIME type” is used to specify the data format between Web servers and Web browsers. When a Web browser requests the “xxx.png” file from a Web server, the Web server returns the details of “xxx.png” that it is an image/png type data. This enables the Web browser to process the received data correctly.
- The MIME type is specified by writing “Content-Type” in the HTML document.

**Ex.** `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">`  
`<meta http-equiv="Content-Style-Type" content="text/css">`

### Style Sheet (CSS)

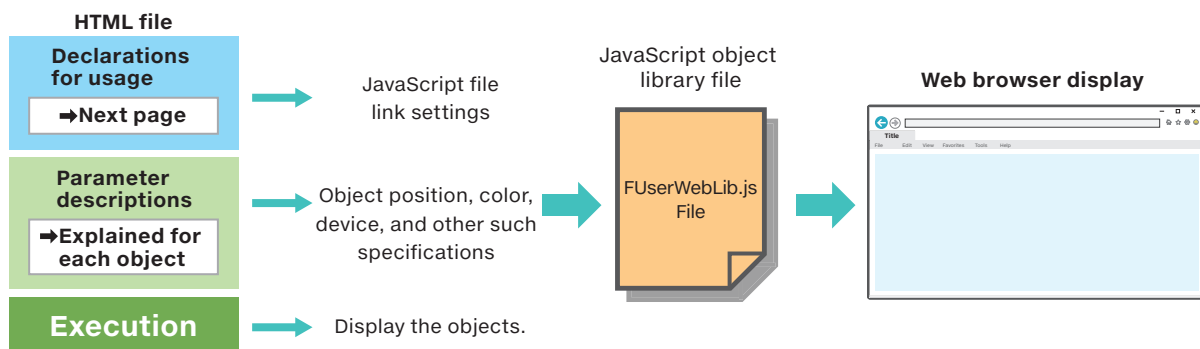
The writing of the styles of user Web page text, buttons, and graphs can be omitted. If the written details are omitted or are incorrect, the page is displayed according to the Style Sheet (UserWebStyle.css). The character font varies depending on the OS of the terminal that displays the user Web page.

Item	Default
Character color	Black
Background color	White
Line color	Black
Level display fill color	Blue
Graph line color	Blue
Font size	20 (in pixels)
Button character color	Follows the Web browser settings.
Button background color	Follows the Web browser settings.
Button line color	Follows the Web browser settings.

**Related Page** 4. LEARNING THE FUNCTIONS OF STYLE SHEETS (CSS)

### JavaScript description structure and processing flow

The roles of the different parts of a JavaScript description are classified as shown below. The object explanation pages are color-coded according to these roles.



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

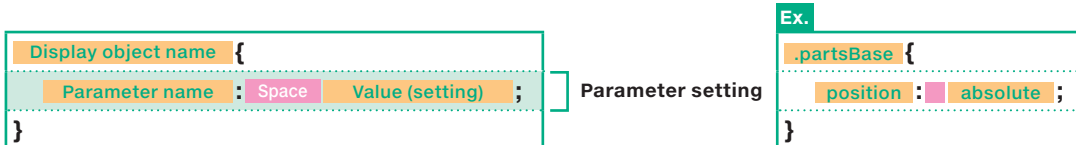
Operation Buttons

## 5.2 Common Items

This section explains the common items that you need to know to use JavaScript objects.

### JavaScript writing rules

The parameters that follow the display object name are organized as shown below.



- The parameter block is enclosed in curly brackets (`{ }`).
- Enclose character strings in single (`'`) or double (`"`) quotation marks. You can use either single (`'`) or double (`"`) quotation marks, but use the same type of symbol for the starting and ending marks.
- Parameter settings are composed of **Parameter name** + `:` + **Space** + **Value (setting)**.

### Parameter specification

This section explains the parameter specifications that follow the display object name.

	Description	Default
Specification omitted	When the parameter of an optional setting is omitted	Displayed with the default settings.
Incorrect specification	When the setting of a parameter is incorrect	
Data format	When set with a data format other than the specified one	For example, a parameter error occurs when a character string such as "XYZ" is specified for a parameter that is set with a numeric value. Set each parameter with its specified data format.
Coordinate specification	When the X and Y coordinates are omitted	The object is placed in the upper-left corner (coordinates 0, 0).
Color specification	Range check of the RGB value or color name	In the HTML specifications, a range check of items such as the RGB value and color name is not performed. If an incorrect setting is specified, the displayed details vary depending on the type of Web browser.
Device specification	When specifying a "U□\G□" device name	Use two "\" characters to specify a name as "U□\G□". (The first "\" is an escape sequence*.)

\* An escape sequence is a special character that is not output as-is and instead is used to control the output of characters such as to change the character color, move the cursor, and delete characters.

## 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)

### Devices that can be set

The settable formats and number bases of the devices that can be set in JavaScript objects are shown below.

✓: Settable, ×: Not settable

Device name	Classification	Bit	16-bit word				32-bit word				Single-precision real number (K)
			Signed		Unsigned		Signed		Unsigned		
			Decimal (K)	Hexadecimal (H)	Decimal (K)	Hexadecimal (H)	Decimal (K)	Hexadecimal (H)	Decimal (K)	Hexadecimal (H)	
<b>User devices</b>											
Input (X)	Bit	✓									
Output (Y)	Bit	✓									
Internal relay (M)	Bit	✓									
Latch relay (L)	Bit	✓									
Link relay (B)	Bit	✓									
Annunciator (F)	Bit	✓									
Link special relay (SB)	Bit	✓									
Step relay (S)	Bit	✓									
Timer (T)* <sup>1</sup>	Contact: TS	Bit	✓								
	Coil: TC	Bit	✓								
	Current value: TN	Word		✓	✓	×	×	×	×	×	×
Accumulation timer (ST)* <sup>1</sup>	Contact: STS	Bit	✓								
	Coil: STC	Bit	✓								
	Current value: STN	Word		✓	✓	×	×	×	×	×	×
Counter (C)* <sup>1</sup>	Contact: CS	Bit	✓								
	Coil: CC	Bit	✓								
	Current value: CN	Word		✓	✓	×	×	×	×	×	×
Long counter (LC)* <sup>1</sup>	Contact: LCS	Bit	✓								
	Coil: LCC	Bit	✓								
	Current value: LCN	Double word						✓	✓	×	×
Data register (D)	Word		✓	✓	✓	✓	✓	✓	✓	✓	✓
Link register (W)	Word		✓	✓	✓	✓	✓	✓	✓	✓	✓
Link special register (SW)	Word		✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>System device</b>											
Special relay (SM)	Bit	✓									
Special register (SD)	Word		✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>Module access device (U□\G□)*<sup>2</sup></b>											
Module access device (G)	Word		✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>Index register</b>											
Index register (Z)	Word		✓	✓	✓	✓	✓	✓	✓	✓	✓
Long index register (LZ)	Double word							✓	✓	✓	✓
<b>File register</b>											
File register (R)	Word		✓	✓	✓	✓	✓	✓	✓	✓	✓

\*1: When T, ST, C, or LC is specified, it is handled as the device of the current value (TN, STN, CN, or LCN).

\*2: When specifying a device name that includes the “\” character such as “U□\G□”, use two “\” characters to specify the name as “U□\\G□”. The first “\” (escape sequence) is a symbol that has the meaning of a special character, so the above expression is required.

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

## Device setting method

Refer to the correspondence table and set the devices.

JavaScript parameter name

<b>devName:</b>	Specified value	.....	① Device name (device name + device number)
<b>devBase:</b>	Specified value	.....	② Data number base
<b>devFormat:</b>	Specified value	.....	③ Data format

## Setting examples

• Specifying input X7

devName:	X7	....	Input X7
devBase:	B	....	Binary
devFormat:	6	....	Bit

• Specifying timer T25 (current value)

devName:	TN25	....	Timer T25
devBase:	K	....	Decimal
devFormat:	0	....	16-bit signed

## Correspondence table

Device name	Classification	JavaScript parameter name				
		① Device name	② Data number base	③ Data format		
		devName:	devBase:	devFormat:		
				② K	② H	
<b>User devices</b>						
Input (X)	Bit	X	B	6		
Output (Y)	Bit	Y	B	6		
Internal relay (M)	Bit	M	B	6		
Latch relay (L)	Bit	L	B	6		
Link relay (B)	Bit	B	B	6		
Annunciator (F)	Bit	F	B	6		
Link special relay (SB)	Bit	SB	B	6		
Step relay (S)	Bit	SB	B	6		
Timer (T)	Contact: TS	Bit	TS	B	6	
	Coil: TC	Bit	TC	B	6	
	Current value: TN	Word	TN	K, H	0	1
Accumulation timer (ST)	Contact: STS	Bit	STS	B	6	
	Coil: STC	Bit	STC	B	6	
	Current value: STN	Word	STN	K, H	0	1
Counter (C)	Contact: CS	Bit	CS	B	6	
	Coil: CC	Bit	CC	B	6	
	Current value: CN	Word	CN	K, H	0	1
Long counter (LC)	Contact: LCS	Bit	LCS	B	6	
	Coil: LCC	Bit	LCC	B	6	
	Current value: LCN	Double word	LCN	K, H	1	3
Data register (D)	Word	D	K, H	0, 2, 4	1, 3	
Link register (W)	Word	W	K, H	0, 2, 4	1, 3	
Link special register (SW)	Word	SW	K, H	0, 2, 4	1, 3	
<b>System device</b>						
Special relay (SM)	Bit	SM	B	6		
Special register (SD)	Word	SD	K, H	0, 2, 4	1, 3	
<b>Module access device (U□\G□)</b>						
Module access device (G)	Word	G	K, H	0, 2, 4	1, 3	
<b>Index register</b>						
Index register (Z)	Word	Z	K, H	0, 2, 4	1, 3	
Long index register (LZ)	Double word	LZ	K, H	2	3	
<b>File register</b>						
File register (R)	Word	R	K, H	0, 2, 4	1, 3	

## ② Data number base

Setting value	Description
K	Decimal
H	Hexadecimal
B	Binary

## ③ Data format

Setting value	Description
0	16-bit signed
1	16-bit unsigned
2	32-bit signed
3	32-bit unsigned
4	Single-precision real number
5	Not used
6	Bit

## 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

### Update interval

The update interval of the user Web page is set with the HTML variable `updateInterval`. This update interval is applied to all the user Web objects.

updateInterval	Setting range	HTML document
Not included	5 s interval (default)	Omitted
Included	5 to 120 s interval*	Example: Setting the update interval to 10 s <code>var updateInterval = 10;</code>

\*: If `updateInterval` is set to a value outside of the interval range of 5 to 120 s, operation will be performed with an interval of 5 s.

### Message display language

The message display language is set with the HTML variable `dspLanguage`. This section explains the variable setting and the message display language.

dspLanguage*	Setting range	HTML document
Not included	English	Omitted
ja-JP	Japanese	<code>var dspLanguage = 'ja-JP';</code>
en-US	English	<code>var dspLanguage = 'en-US';</code>
zh-CN	Chinese (Simplified)	<code>var dspLanguage = 'zh-CN';</code>

\*: If `dspLanguage` is set to a value outside of the range, the display language is set to English.

### ► How to write statements in the HTML file

Create the statements between the JavaScript starting and ending lines: `<script>` to `</script>`, which are placed between the `<head>` and `</head>` tags.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charset setting *Because the Web server setting is UTF-8, specify UTF-8.) -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- Set the title. -->
    <title>Sample</title>
    <link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />
    <!-- Reading the library JavaScript (Change the path to match the storage location.) -->
    <script src="/UserWebLib.js"></script>
    <!-- Write the proprietary JavaScript from this point. -->
    <!-- Write the user JavaScript here. -->
    <script>
      var updateInterval = 5;      Update interval
      var dspLanguage = 'en-US';  Message display language
      ...
    </script>
  </head>
  ...
</html>
```



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

**Table**

Graphs

Image/Figure

Operation Buttons

## 5.3 Table

### Data Block object (WSDatblk)

#### ► Function

Displays the specified device data as a table.

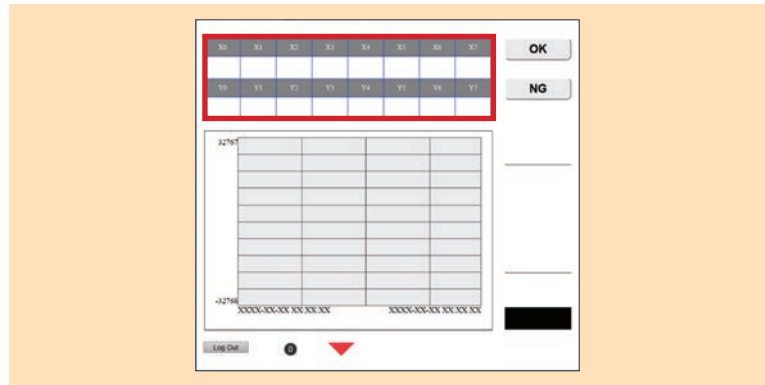
#### ► Creation example

Direction: Horizontal

Row count: Two

Data count: Eight

#### ► User Web page library window



### Object design

#### Caution

When the data format is set to single-precision real number, the data number base is changed to decimal.

#### ► HTML (JavaScript) structure

Create the HTML document for specifying the display devices consecutively as shown below with the number of rows or columns of the devices to display.

#### Usage declaration

Parameters (separate)

Display device specification (X0 to X7)

Parameters (common)

Data block (table construction) settings

Execution

X0	X1	X2	X3	X4	X5	X6	X7

#### Usage declaration

Parameters (separate)

Display device specification (X0 to X7)

Display device specification (Y0 to Y7)

Parameters (common)

Data block (table construction) settings

Execution

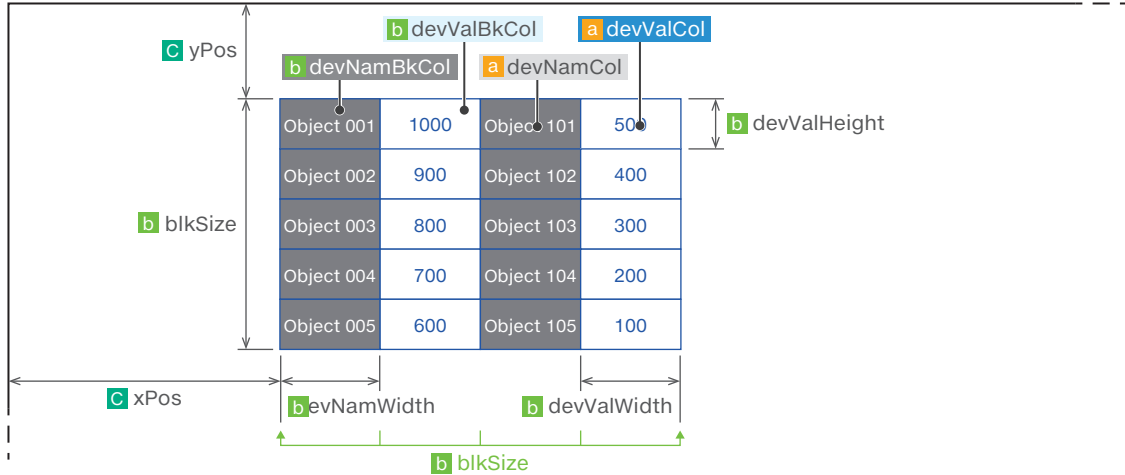
X0	X1	X2	X3	X4	X5	X6	X7
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

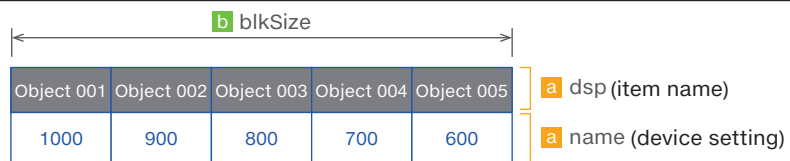
[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)

## ► List of parameters

### • Vertical **b** direction: 0



### • Horizontal **b** direction: 1



### **a** Item name and device settings

In the above list of parameters, dev sets the table item name and device. The settings in JavaScript are split into dsp:, name:, base:, and format:.

#### ► Item name setting ... dsp:

Any character string (example: sensor) can be displayed.

In the following example, a sequence number (1 to 4) starting with the variable "i" set with a for statement is added after 'sensor'.

```
Ex. for(var i = 1; i < 5; i++){
    temp.push({
        dsp: 'sensor'+i,
        name: 'X'+i,
        base: 'B',
        format: 6
    });
}
```

Display example

X1 = ON, X2 = OFF, X3 = OFF, X4 = ON

Sensor1	Sensor2	Sensor3	Sensor4
1	0	0	1

#### ► Device specification ... name:

The value of the specified device (example: the values of X1 to X4) can be displayed.

In the following example, 'X' is the device classification and the sequence number (1 to 4) starting with the variable "i" set with a for statement is the device number.

```
Ex. for(var i = 1; i < 5; i++){
    temp.push({
        dsp: 'sensor'+i,
        name: 'X'+i,
        base: 'B',
        format: 6
    });
}
```

Display example

X1 = ON, X2 = OFF, X3 = OFF, X4 = ON

Sensor1	Sensor2	Sensor3	Sensor4
1	0	0	1

**Related Page** [5.2 Common Items - Device Setting Method](#)

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

## ▶ Data number base ... base:

Value	Description
K	Decimal
H	Hexadecimal
B	Binary

**Related Page** 5.2 Common Items - Device Setting Method

## ▶ Data format ... format:

Value	Description	Value	Description
0	16-bit signed	4	Single-precision real number
1	16-bit unsigned	5	Not used
2	32-bit signed	6	Bit
3	32-bit unsigned		

**Related Page** 5.2 Common Items - Device Setting Method

## ▶ Device name (item name) display setting ... devNamDisp:

Script	Description
devNamDisp: 0	Do not display.
devNamDisp: 1	Display (when omitted).

Display example: Do not display device names.

1	0	0	1
---	---	---	---

Display example: Display device names.

X0	X1	X2	X3
1	0	0	1

## ▶ Character color specification ... devNamCol: devValCol:

Script	Description	Remarks
devNamCol:	Device name color specification	RGB value or color name
devValCol:	Device value color specification	Examples: #ff0000 (RGB value), red (color name)

### b Table construction settings

## ▶ Display direction specification ... direction:

Script	Description
direction: 0	Vertical (when omitted)
direction: 1	Horizontal

Display example: Vertical

X0	1	X2	0
X1	0	X3	1

Display example: Horizontal

X0	X1
1	0
X2	X3
0	1

## ▶ Block size (horizontal cell count) specification ... blkSize:

Script	Description	Remarks
blkSize:	Cell count specification	This parameter changes the number of rows when the display direction is vertical and the number of columns when the display direction is horizontal.

## 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)

### ▶ Cell size specification ... devNamWidth: devNamHeight: devValWidth: devValHeight:

Script	Description	Remarks
devNamWidth:	Device name cell	Width
devNamHeight:		Height
devValWidth:	Device value cell	Width
devValHeight:		Height

Positive real number (in pixels)

### ▶ Cell color specification ... devNamBkCol: devValBkCol:

Script	Description	Remarks
devNamBkCol:	Device name cell background color	RGB value or color name
devValBkCol:	Device value cell background color	Examples: #FF0000 (RGB value), red (color name)

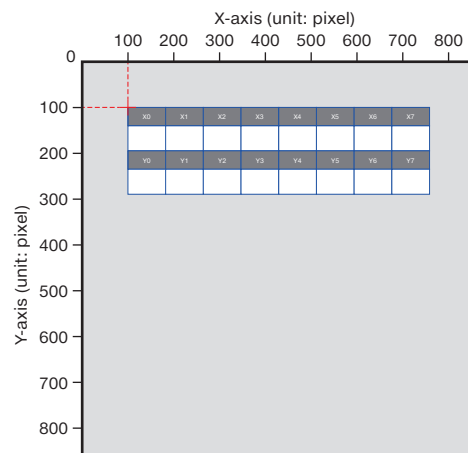
### C Display starting coordinates setting

### ▶ Web browser display starting position setting ... xPos: yPos:

Script	Description	Remarks
xPos:	X-axis coordinate	Setting range: Screen resolution of the personal computer (in pixels)
yPos:	Y-axis coordinate	

### ▶ How to view Web browser coordinates

**Ex.** (xPos, yPos) = (100, 100)



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

## ► HTML document explanation

Parameter handling

- ... Parameter setting required. An error occurs if this is not set.
- ... This parameter is not required when the device name display is set to "Do not display" (devNamDisp = 0).
- ... Details set with JavaScript.

### Caution

Write all the required parameters. If a required parameter is not written or if a setting value is outside of its range, an error occurs.

Line No.	JavaScript	Explanation				
1	// Data Block object (WSDatblk)	Write a comment.				
2	temp = [];	Declaration (label name)				
<b>Display device settings of lines 1 and 2</b>						
3	for(var i = 0; i < 8; i++){	<b>for statement start (variable name: i)</b> The lines within a for statement (from its start to its end) are executed repeatedly. In the statement on the left, "i" is used as the variable for counting the number of repetitions. The () part is composed of three statements: the default value of the variable for the number of repetitions, the end condition of the for statement, and the calculation for changing the number of repetitions. The calculation for changing the number of repetitions is "i++", so the variable is increased by one each time the contents of the for statement are executed until the for statement ends. Write the part in the () so that the range matches the input device number. The number of data items in the table is set with blkSize. Match i to that value + 1.				
4	temp.push({	<b>Device name setting start</b>				
5	dsp: 'X' + i,	<b>Name to display for the table item</b> You can write any name such as 'Input'. The example on the left displays X0 to X7.				
6	name: 'X' + i,	<b>Device specification</b> Specify the device number with device classification "+i". Set the PLC device name.				
7	base: 'B',	<b>Device data number base</b> <table border="1" style="float: right;"> <thead> <tr> <th>Setting value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>B</td> <td>Binary</td> </tr> </tbody> </table>	Setting value	Description	B	Binary
Setting value	Description					
B	Binary					
8	format: 6	<b>Device data format</b> <table border="1" style="float: right;"> <thead> <tr> <th>Setting value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>6</td> <td>Bit</td> </tr> </tbody> </table>	Setting value	Description	6	Bit
Setting value	Description					
6	Bit					
9	});	<b>Device name setting end</b>				
10	}	<b>for statement end</b>				
<b>Display device settings of lines 3 and 4</b>						
11	for(var i = 0; i < 8; i++){	<b>for statement start (variable name: i)</b>				
12	temp.push({	<b>Device name setting start (The details are the same as those for line number 3 above.)</b>				
13	dsp: 'Y' + i,	<b>Name to display for the table item</b> You can write any name such as 'Output'. The example on the left displays Y0 to Y7.				
14	name: 'Y' + i,	<b>Device specification</b> Specify the device number with device classification + i. Set the PLC device name.				
15	base: 'B',	<b>Device data number base</b> <table border="1" style="float: right;"> <thead> <tr> <th>Setting value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>B</td> <td>Binary</td> </tr> </tbody> </table>	Setting value	Description	B	Binary
Setting value	Description					
B	Binary					
16	format: 6	<b>Device data format</b> <table border="1" style="float: right;"> <thead> <tr> <th>Setting value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>6</td> <td>Bit</td> </tr> </tbody> </table>	Setting value	Description	6	Bit
Setting value	Description					
6	Bit					
17	});	<b>Device name setting end</b>				
18	}	<b>for statement end</b>				

## 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)

Line No.	JavaScript	Explanation
<b>Table overall settings</b>		
19	<code>dataBlockParam = {</code>	<b>JavaScript parameter setting start</b>
20	<code>  dev:                  temp,</code>	<b>Specify the multiple device names (values) set with the for statement on the previous page.</b>
21	<code>  direction:          1,</code>	<b>Display direction</b> 0: Vertical <b>1: Horizontal</b>
22	<code>  blkSize:             8,</code>	<b>Block size</b>
23	<code>  devNamDisp:         1,</code>	<b>Device name display</b> 0: Do not display. <b>1: Display (when omitted).</b>
24	<code>  devNamCol:           'white',</code>	<b>Device name cell color specification and size</b> Character color: Specified with a color name in this example Background color: Specified with a color code (RGB value) in this example Cell width (in pixels) Cell height (in pixels)
25	<code>  devNamBkCol:         '#808080',</code>	
26	<code>  devNamWidth:         100,</code>	
27	<code>  devNamHeight:        40,</code>	Cell height (in pixels)
28	<code>  devValCol:           'blue',</code>	<b>Device value cell color specification and size</b> Character color: Specified with a color name in this example Background color: Specified with a color name in this example Cell width (in pixels) Cell height (in pixels) Line color: Specified with a color name in this example
29	<code>  devValBkCol:         'white',</code>	
30	<code>  devValWidth:         80,</code>	
31	<code>  devValHeight:        50,</code>	
32	<code>  InCol:                'blue',</code>	Line color: Specified with a color name in this example
33	<code>  xPos:                 20,</code>	<b>Display starting position</b> X-coordinate (in pixels)
34	<code>  yPos:                 40</code>	Y-coordinate (in pixels)
35	<code>}</code>	<b>JavaScript parameter setting end</b>
<b>Execution</b>		
36	<code>  WSDatblk(dataBlockParam);</code>	Executes the display of the Data Block object.

Parameters (common)

Execution

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

**Graphs**

Image/Figure

Operation Buttons

Level display object

Historical graph object

## 5.4 Graphs

### Level display object (WSLevel)

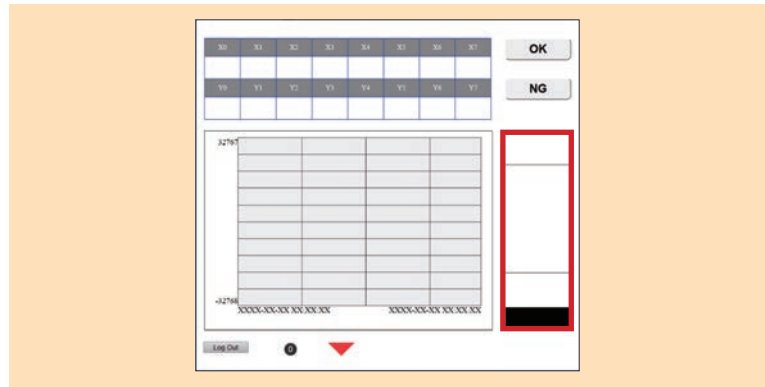
#### ► Function

Displays the current value of the specified device as a level within the range of the upper limit value and lower limit value.

#### ► Creation example

Direction: Vertical

#### ► User Web page library window

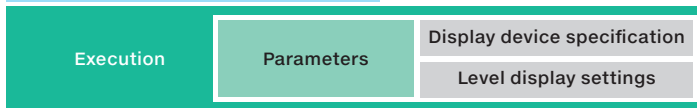


### Object design

#### ► HTML (JavaScript) structure

Create the object with the following structure.

#### Usage declaration

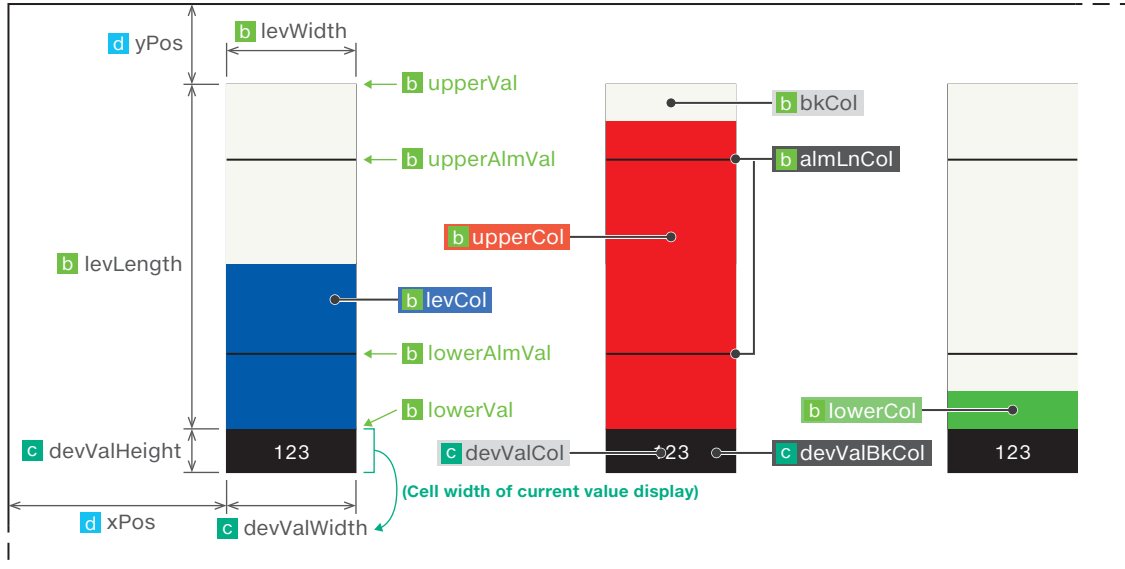


# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

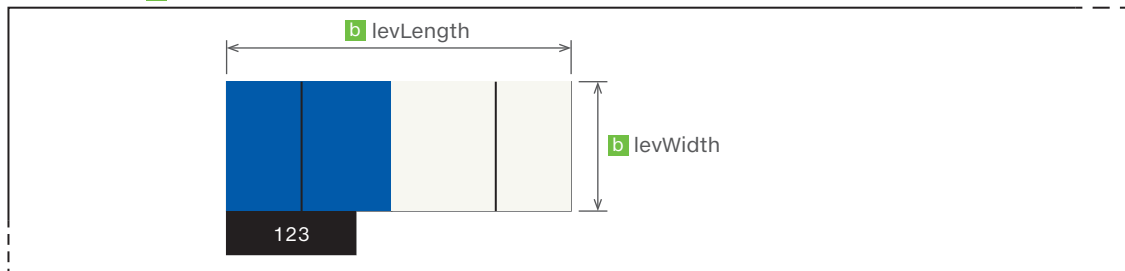
[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)
[Level display object](#)
[Historical graph object](#)

## ► List of parameters

### • Vertical **b** direction: 0



### • Horizontal **b** direction: 1



#### a Device settings

► Device name ... devName: Device classification + device number

Related Page [5.2 Common Items - Device Setting Method](#)

► Data format ... valFormat:

Value	Description
0	16-bit signed (when omitted)
1	16-bit unsigned
2	32-bit signed
3	32-bit unsigned
4	Single-precision real number

Related Page [5.2 Common Items - Device Setting Method](#)



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

**Graphs**

Image/Figure

Operation Buttons

Level display object

Historical graph object

## b Level display settings

### ▶ Level direction ... direction:

Script	Description
direction: 0	Vertical (when omitted)
direction: 1	Horizontal

### ▶ Upper/lower limit value ... upperVal: lowerVal:

Script	Description
upperVal:	Upper limit value
lowerVal:	Lower limit value

### ▶ Upper/lower limit of alarm value ... upperAlmVal: lowerAlmVal:

Script	Description
upperAlmVal:	Upper limit of alarm value
lowerAlmVal:	Lower limit of alarm value

### ▶ Alarm value line display ... dspAlmLn:

Script	Description
dspAlmLn: 0	Do not display.
dspAlmLn: 1	Display (when omitted).

Display example:  
Do not display.



Display example:  
Display.



### ▶ Current value display ... dspVal:

Script	Description
dspVal: 0	Do not display.
dspVal: 1	Display (when omitted).

Display example:  
Do not display.



Display example:  
Display.



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

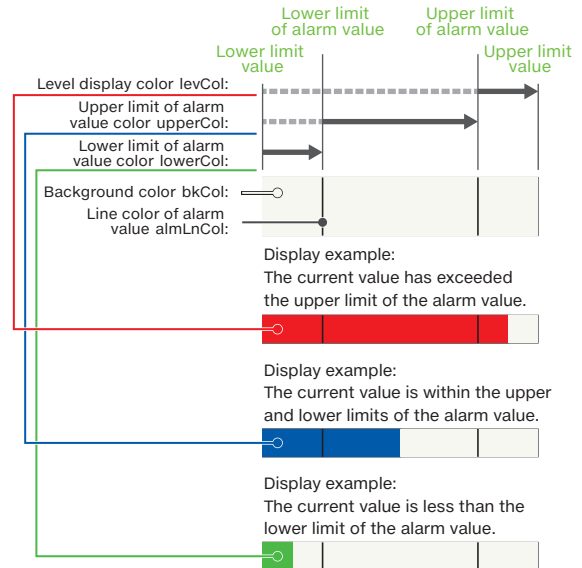
[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)
[Level display object](#)
[Historical graph object](#)

## ► Color specification ... levCol: upperCol: lowerCol: bkCol: almLnCol:

The level display color varies depending on the upper/lower limit of the alarm value.

Script	Description
levCol:	Level display color
upperCol:	Upper limit of alarm value color*
lowerCol:	Lower limit of alarm value color*
bkCol:	Background color
almLnCol:	Line color of alarm value

\*: When this parameter is omitted, the color is the same as the level display color.



## ► Size specification ... levLength: levWidth

Script	Description	Remarks
levLength:	Level length	Positive real number (in pixels)
levWidth:	Level width	

### C Current value display cell

## ► Size specification ... devValWidth: devValHeight:

Script	Description	Remarks
devValWidth:	Cell width of current value	Positive real number (in pixels)
devValHeight:	Cell height of current value	

## ► Cell color specification ... devValCol: devValBkCol:

Script	Description	Remarks
devValCol:	Character color of current value	RGB value or color name Examples: #FF0000 (RGB value), red (color name)
devValBkCol:	Background color of current value	

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

**Graphs**

Image/Figure

Operation Buttons

Level display object

Historical graph object

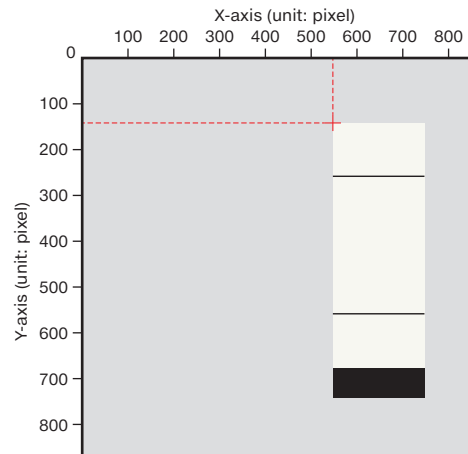
## d Display starting coordinates setting

### ▶ Web browser display starting position setting ... xPos: yPos:

Script	Description	Remarks
xPos:	X-axis coordinate	Personal computer setting range: Screen resolution (in pixels)
yPos:	Y-axis coordinate	

### ▶ How to view Web browser coordinates

**Ex.** (xPos, yPos) = (550, 150)



## 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)
[Level display object](#)
[Historical graph object](#)

### ► HTML document explanation

Parameter handling

... Parameter setting required. An error occurs if this is not set.

... Details set with JavaScript.

#### Caution

Write all the required parameters. If a required parameter is not written or if a setting value is outside of its range, an error occurs.

Line No.	JavaScript	Explanation
Execution	1 // Level display object	Write a comment.
	2 WSLLevel({	Data Block object parameter setting + display execution JavaScript parameter setting start
Parameters	3 devName: 'D0',	Device specification Specify the device number with device classification + i (set the PLC device name).
	4 direction: 0,	Display direction 0: Vertical (when omitted) 1: Horizontal
	5 levCol: 'mediumblue',	Level display color: Specified with a color name in this example
	6 upperCol: 'red',	Upper limit of alarm value color: Specified with a color name in this example
	7 lowerCol: '#00FF00',	Lower limit of alarm value color: Specified with a color code (RGB value) in this example
	8 bkCol: 'white',	Background color: Specified with a color name in this example
	9 upperVal: 32767,	Upper limit value: Range of values that can be handled with the specified device
	10 lowerVal: -32768,	Lower limit value: Range of values that can be handled with the specified device
	11 upperAlmVal: 20000,	Upper limit of alarm value: Range of values that can be handled with the specified device
	12 lowerAlmVal: -20000,	Lower limit of alarm value: Range of values that can be handled with the specified device
	13 dspAlmLn: 1,	Alarm value line display 0: Do not display. 1: Display (when omitted).
	14 almLnCol: 'black',	Line color of alarm value: Specified with a color name in this example
	15 levLength: 400,	Level length: Positive real number (in pixels)
	16 levWidth: 150,	Level width: Positive real number (in pixels)
	17 dspVal: 1,	Current value display 0: Do not display. 1: Display (when omitted).
	18 valFormat: 0,	Device data format 0: 16-bit signed (when omitted) 3: 32-bit unsigned 1: 16-bit unsigned 4: Single-precision real number 2: 32-bit signed
	19 devValCol: 'white',	Character color of current value
	20 devValBkCol: 'black',	Background color of current value
	21 devValWidth: 150,	Cell width of current value (in pixels)
	22 devValHeight: 50,	Cell height of current value (in pixels)
	23 xPos: 700,	Display starting position X-coordinate (in pixels)
	24 yPos: 250,	Y-coordinate (in pixels)
	25 });	JavaScript parameter setting end

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

**Graphs**

Image/Figure

Operation Buttons

Level display object

Historical graph object

## Historical graph object (WSHstgrp)

### Function

Displays a device value as a chronological order line graph.

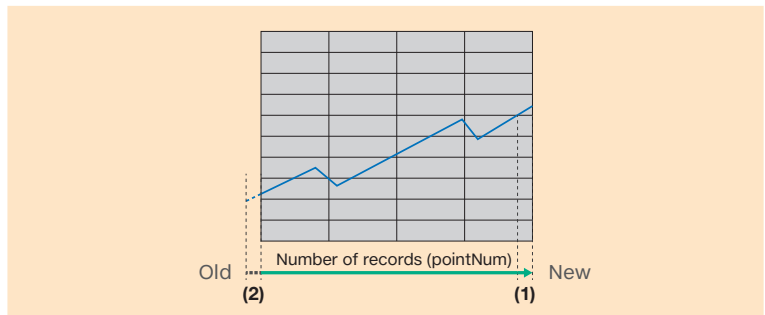
### User Web page library window



### Operation

The device value is read per update interval. When the upper limit on the number of records is reached, the oldest record is deleted and the display is shifted to the left.

- (1) Update interval portion
- (2) An update interval portion is outside the display range.



## Object design

### Restriction

- The numeric values that can be handled with the historical graph object are fixed to decimal values.
- If a state in which the communication load is high continues, device values may be lost.

### HTML (JavaScript) structure

Create the HTML document for specifying the display devices consecutively as shown below with the number of device lines to display.

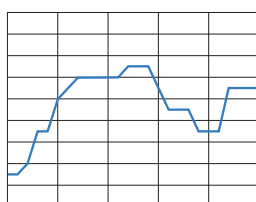
When there is one line

Usage declaration

Parameters (separate) Display device specification (first line)

Parameters (common) Historical graph settings

Execution



First line

When there are three lines

Usage declaration

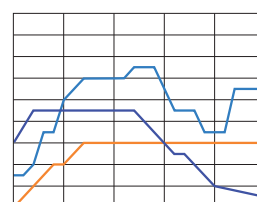
Parameters (separate) Display device specification (first line)

Display device specification (second line)

Display device specification (third line)

Parameters (common) Historical graph settings

Execution



First line

Second line

Third line

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

**Graphs**

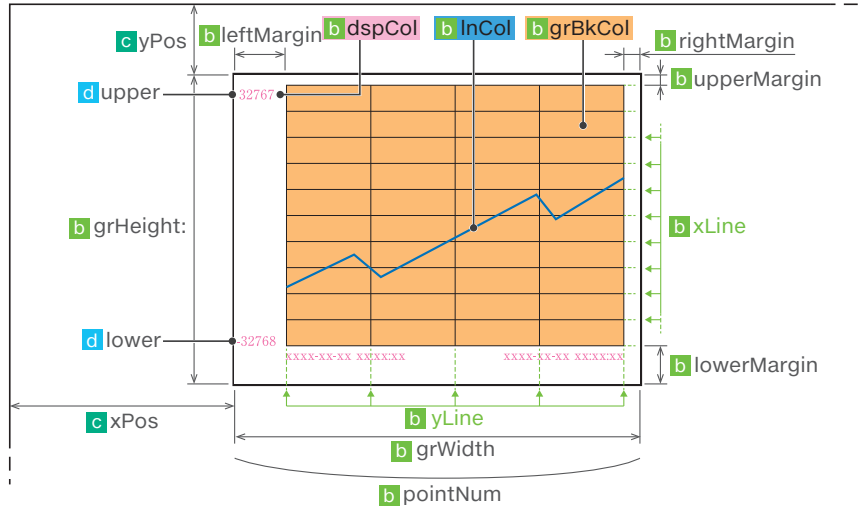
Image/Figure

Operation Buttons

Level display object

Historical graph object

## ► List of parameters



### a Device settings

► Device name ... devName: Device classification + device number

Related Page 5.2 Common Items - Device Setting Method

► Data format ... devFormat:

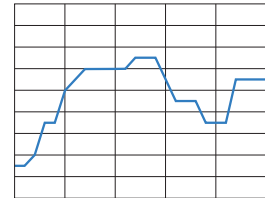
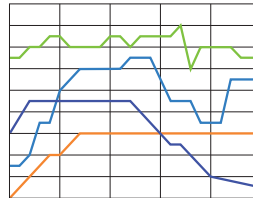
Value	Description	Value	Description	Value	Description	Value	Description
0	16-bit signed	2	32-bit signed	4	Single-precision real number	6	Bit
1	16-bit unsigned	3	32-bit unsigned	5	Not used		

Related Page 5.2 Common Items - Device Setting Method

### b Historical graph display settings

► Number of graph elements ... grElmNum:

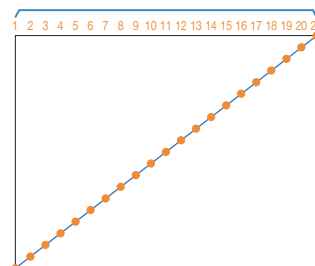
Script	Setting range	Description
grElmNum:	1 to 32	Line count setting



► Number of records ... pointNum:

Script	Setting range	Description
pointNum:	5 to 100	Number of data points along the horizontal axis of the line graph

Ex. pointNum: 21



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What is JavaScript?

Common Items

Table

**Graphs**

Image/Figure

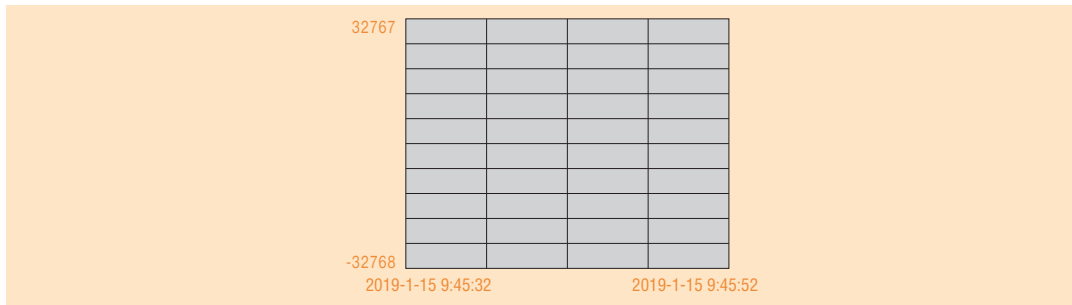
Operation Buttons

Level display object

Historical graph object

## ▶ Upper/lower limit value ... upper: lower:

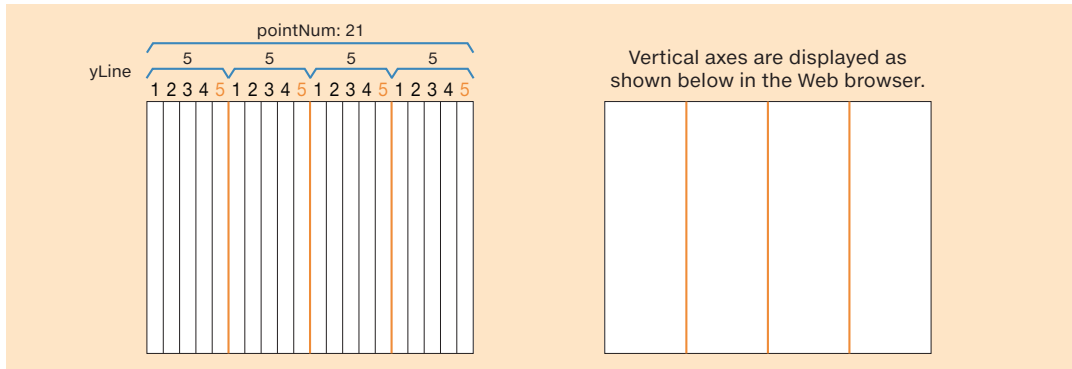
Script	Description	Remarks
upper:	Upper limit value	Device value (Y-axis) upper limit value and lower limit value settings An error occurs if the value is outside of this range. For the bit data format, specify 1 for the upper limit value and 0 for the lower limit value.
lower:	Lower limit value	



## ▶ Vertical-axis interval (record) ... yLine:

Script	Setting range	Description
yLine:	0 to 99	Interval setting for displaying the vertical-axis graduation lines The specified value indicates the number of records in each interval.

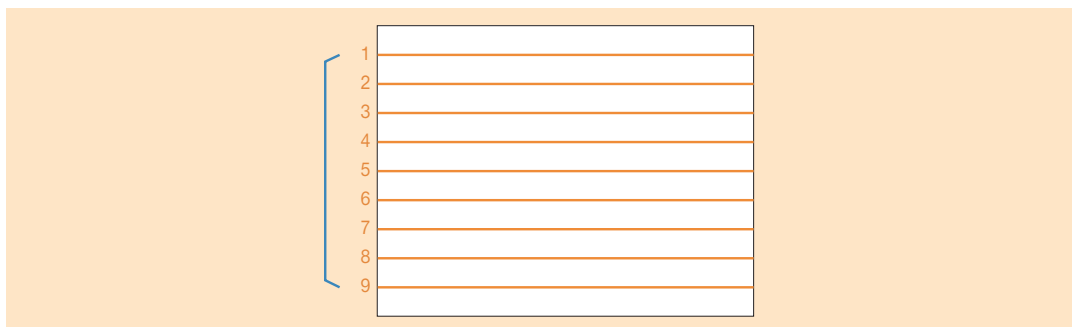
**Ex.]** yLine: 5 (vertical-axis interval)



## ▶ Number of horizontal-axis lines ... xLine:

Script	Setting range	Description
xLine:	0 to 99	Setting for the number of lines to display for horizontal-axis graduation lines

**Ex.]** xLine: 9

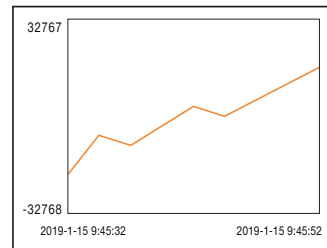
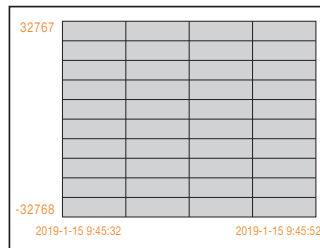
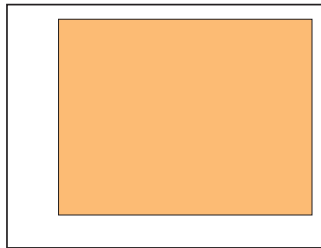


# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)
[Level display object](#)
[Historical graph object](#)

## ► Color specification ... InCol: dspCol: grBkCol:

Script	Description
InCol:	Graph line color
dspCol:	Character color
grBkCol:	Graph background color



## ► Size specification ... grHeight: grWidth:

Script	Description	Remarks
grHeight:	Graph height	Positive real number (in pixels)
grWidth:	Graph width	

## ► Margin specification ... rightMargin: leftMargin: upperMargin: lowerMargin:

Script	Description	Remarks
rightMargin:	Right margin	Positive real number (in pixels) Set to 0 when the parameter is omitted.
leftMargin:	Left margin	
upperMargin:	Upper margin	
lowerMargin:	Lower margin	

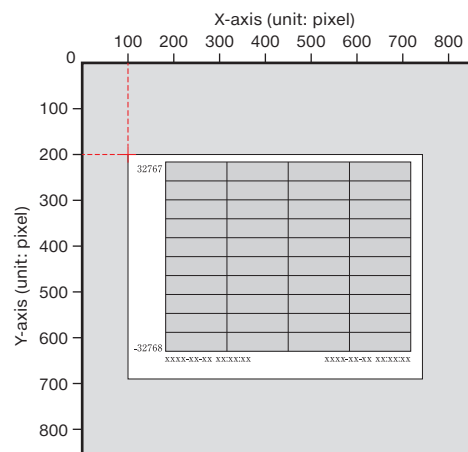
### C Display starting coordinates setting

## ► Web browser display starting position setting ... xPos: yPos:

Script	Description	Remarks
xPos:	X-axis coordinate	Setting range: Screen resolution (in pixels)
yPos:	Y-axis coordinate	

## ► How to view Web browser coordinates

Ex.] (xPos, yPos) = (100, 200)





# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

**Graphs**

Image/Figure

Operation Buttons

Level display object

Historical graph object

## ► HTML document explanation

Parameter handling

... Parameter setting required. An error occurs if this is not set.

### Caution

Write all the required parameters. If a required parameter is not written or if a setting value is outside of its range, an error occurs.

Line No.	JavaScript	Explanation				
1	// Historical graph object	Write a comment.				
2	temp = [];					
3	num = 2;	Number of graph elements, setting range: 1 to 32				
4	temp.push({	Specification of the first line				
5	devName: 'D0',	Device name: Device classification + device number				
6	lnCol: 'red',	Graph line color: Specified with a color name in this example				
7	});					
8	temp.push({	Specification of the second line				
9	devName: 'D1',	Device name: Device classification + device number				
10	lnCol: 'blue',	Graph line color: Specified with a color name in this example				
11	});					
12	hstGrpParam = {					
13	xPos: 20,	Display starting position X-coordinate (in pixels)				
14	yPos: 250,	Y-coordinate (in pixels)				
15	grElmNum: num,	Argument num sets the number of graph elements with num = 2 on line number 3.				
16	devFormat: 0,	Device data format				
		<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>16-bit signed</td> </tr> </tbody> </table>	Value	Description	0	16-bit signed
Value	Description					
0	16-bit signed					
17	grElm: temp,	The argument temp sets the device name and graph line color to a value that does not exceed the value of num on line number 3.				
18	grBkCol: '#F0F0F0',	Graph background color: Specified with a color code in this example				
19	dspCol: 'black',	Character color: Specified with a color name in this example				
20	pointNum: 20,	Number of records				
21	pointInt: 2,					
22	upper: 32767,	Upper limit value				
23	lower: -32768,	Lower limit value				
24	xLine: 9,	Number of horizontal-axis lines: 0 to 99				
25	yLine: 5,	Vertical-axis interval (number of records) at which to display lines: 0 to 99				
26	grHeight: 380,	Graph height (in pixels)				
27	grWidth: 550,	Graph width (in pixels)				
28	upperMargin: 15,	Upper margin: Positive real number (in pixels; set to 0 when the parameter is omitted)				
29	leftMargin: 75,	Left margin: Positive real number (in pixels; set to 0 when the parameter is omitted)				
30	lowerMargin: 55,	Lower margin: Positive real number (in pixels; set to 0 when the parameter is omitted)				
31	rightMargin: 25	Right margin: Positive real number (in pixels; set to 0 when the parameter is omitted)				
32	}	JavaScript parameter setting end				
33	WSHstgrp(hstGrpParam);	Executes the display of the Data Block object.				

Display device specification (lines 1 and 2)

Execution

## 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

Image display object

Figure display object

### 5.5 Image/Figure

#### Image display object (WSPicture)

##### ► Function

Displays the specified image file when the device value reaches a value in the set range.

##### ► Creation example

Monitor device: D0

Number of images: Three (including the default display image)

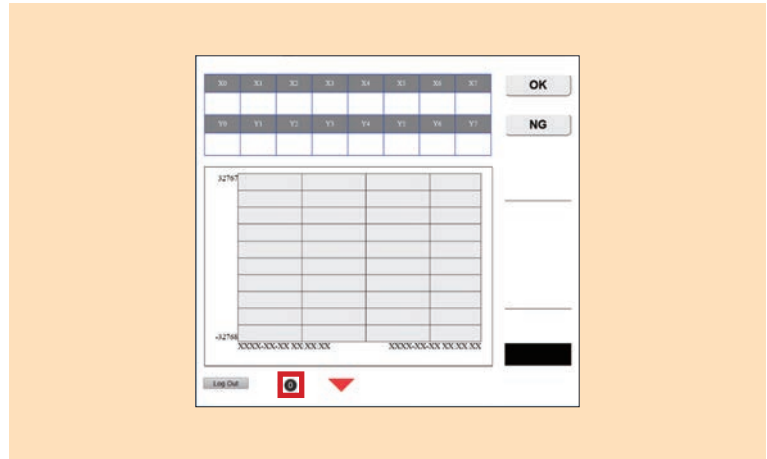
Set image file names:

sample0.png

sample1.png

sample2.png

##### ► User Web page library window



#### Object design

##### Restriction

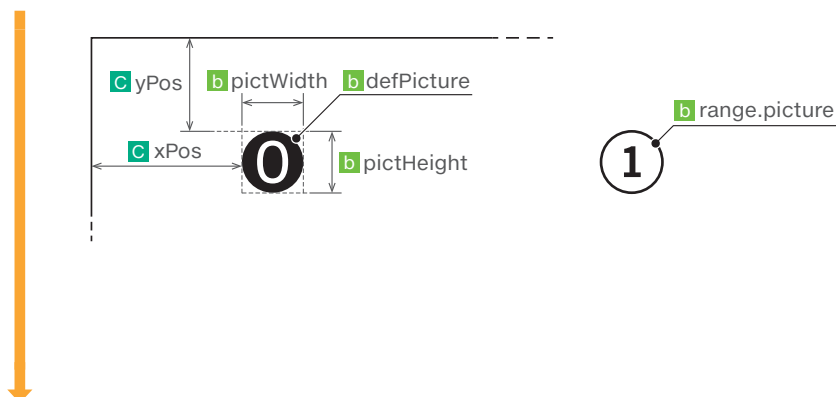
- As a guideline, ensure that the total size of all the image files in a single window is 100 kbyte or less.
- If setting ranges overlap, the images of the setting range with the lower number are displayed. For example, if the device value is within setting range 1 and setting range 2, the display will be that for setting range 1.

##### ► HTML (JavaScript) structure

Create the HTML document for the parameters consecutively as shown below to match the number of images that switch the display.

Execution	
Parameters (common)	Image display object settings
	Default display image
	Image (first)
Parameters (separate)	Image (second)
	⋮
	Image (fifth)

##### ► List of parameters



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

**Image/Figure**

Operation Buttons

Image display object

Figure display object

## a Device settings

▶ **Device name ... devName:** Device classification + device number

**Related Page** 5.2 Common Items - Device Setting Method

▶ **Data format ... valFormat:**

Value	Description	Value	Description
0	16-bit signed	4	Single-precision real number
1	16-bit unsigned	5	Not used
2	32-bit signed	6	Bit
3	32-bit unsigned		

**Related Page** 5.2 Common Items - Device Setting Method

## b Image settings

▶ **Image count setting ... rangeNum:**

Specifies the number of images to display. (The default display image is not included.)

Script	Description	Remarks
rangeNum:	Switch-to image count setting	Setting range: 1 to 5

▶ **Image file setting ... defPicture: range.picture:**

Specifies an image file. The image is displayed with the size specified by the image size setting.

Script	Description	Remarks
defPicture:	Default image file specification	Displayed when the device value is outside the setting range. Extension: .jpg, .jpeg, .gif, .png
range.picture:	Switch-to image file specification	Displayed when the device value is inside the setting range. Extension: .jpg, .jpeg, .gif, .png

**Related Page** 5.1 What Is JavaScript (JS)? - Usable files

▶ **Image size setting ... pictHeight: pictWidth:**

Script	Description	Remarks
pictHeight:	Image height	Value: Positive real number (in pixels)
pictWidth:	Image width	If a negative number is specified, it is handled as its absolute value.

## 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)
[Image display object](#)
[Figure display object](#)

### ▶ Device value range setting ... range.low: range.high:

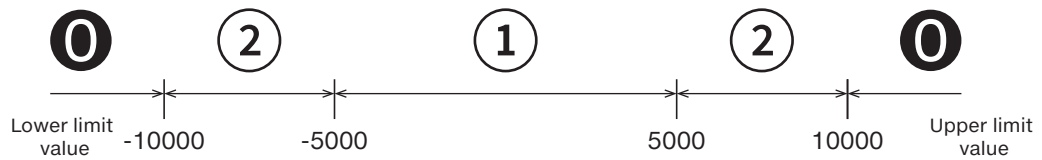
Script	Description	Remarks	
range.low:	Lower limit value	Lower limit value of the device values that display the image	Value: Positive real number (in pixels)
range.high:	Upper limit value	Upper limit value of the device values that display the image	

**Ex.** rangeNum: 2

① Default display

① Image (first) range.low: -5000, range.high: 5000

② Image (second) range.low: -10000, range.high: 10000



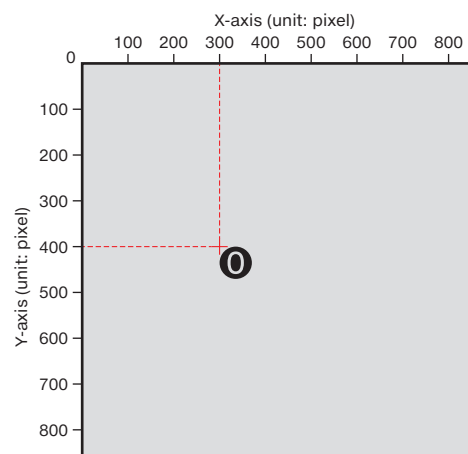
### c Display starting coordinates setting

### ▶ Web browser display starting position setting ... xPos: yPos:

Script	Description	Remarks
xPos:	X-axis coordinate	Setting range: Screen resolution of the personal computer (units of pixels)
yPos:	Y-axis coordinate	

### ▶ How to view Web browser coordinates

**Ex.** (xPos, yPos) = (300, 400)



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

**Image/Figure**

Operation Buttons

Image display object

Figure display object

## ► HTML document explanation

Parameter handling

... Parameter setting required. An error occurs if this is not set.

### Caution

Write all the required parameters. If a required parameter is not written or if a setting value is outside of its range, an error occurs.

Line No.	JavaScript	Explanation	
Execution	1 // Image display object	Write a comment.	
	2 WSPicture{	Image display object parameter setting + display execution	
Parameters	3 devName: 'D0',	Device specification Device classification + device number	
	4 devFormat: 0,	Device data format	
	5 pictHeight: 30,	Display range height of image file	
	6 pictWidth: 30,	Display range width of image file	
	7 defPicture: './img/sample0.png',	Name of image file to be displayed as default	
	8 rangeNum 2,	Setting range count: Setting range: 1 to 5	
	9 range:{		
	10 {		
	11 low: -5000,	Settings for changing the image (first)	Lower limit value of the device values that display the image
	12 high: 5000,		Upper limit value of the device values that display the image
	13 picture: './img/sample1.png',		Specify an image file.
	14 },		
	15 {	Settings for changing the image (second)	Lower limit value of the device values that display the image
16 low: -10000,	Upper limit value of the device values that display the image		
17 high: 10000,	Specify an image file.		
18 picture: './img/sample2.png',			
19 },			
20 },			
21 xPos: 150,	Display starting position	X-coordinate (in pixels)	
22 yPos: 50,		Y-coordinate (in pixels)	
23 });			

## 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

Image display object

Figure display object

### Figure display object (WSFigure)

#### ► Function

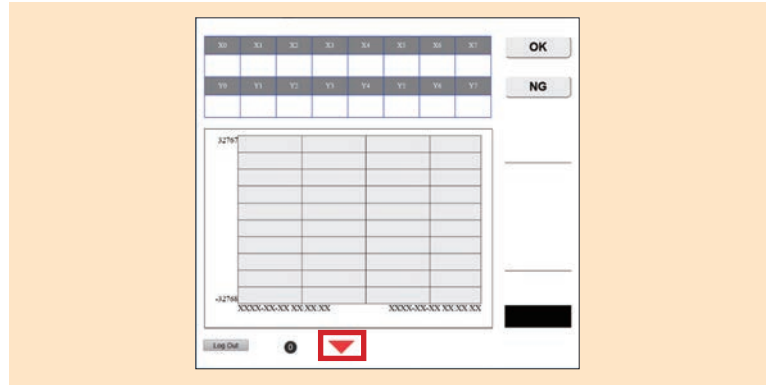
Displays a figure with the specified color when the device value reaches a value in the set range.

#### ► Creation example

Shape: Inverted triangle

Figure colors: Three (including the default display figure color)

#### ► User Web page library window



### Object design

#### Restriction

- If setting ranges overlap, the images of the setting range with the lower number are displayed. For example, if the device value is within setting range 1 and setting range 2, the display will be that for setting range 1.
- The device value is monitored at a fixed interval.

#### ► HTML (JavaScript) structure

Set only the range portion for the colors that you want to display.

Execution	
Parameters (common)	Figure display object settings
	Default display figure color
Parameters (separate)	Figure color setting (first)
Parameters (separate)	Figure color setting (second)

Example of set image file names:

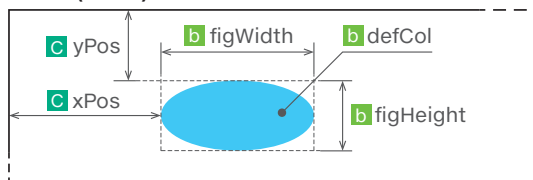
sample0.png

sample1.png

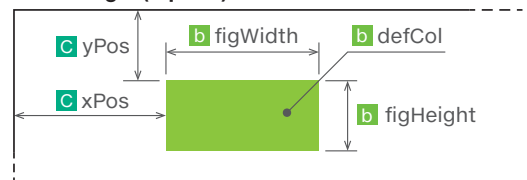
sample2.png

#### ► List of parameters

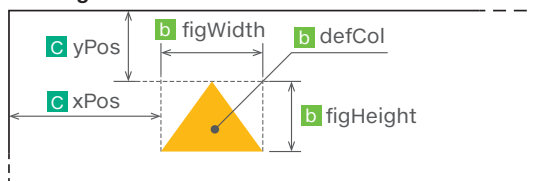
##### • Oval (circle)



##### • Rectangle (square)



##### • Triangle



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

Image display object

Figure display object

## a Device settings

▶ Device name ... devName: Device classification + device number

Related Page 5.2 Common Items - Device Setting Method

▶ Data number base ... base:

Value	Description
K	Decimal
H	Hexadecimal
B	Binary

Related Page 5.2 Common Items - Device Setting Method

▶ Data format ... devFormat:

Value	Description	Value	Description
0	16-bit signed	4	Single-precision real number
1	16-bit unsigned	5	Not used
2	32-bit signed	6	Bit
3	32-bit unsigned		

Related Page 5.2 Common Items - Device Setting Method

▶ Device value range setting ... range.low: range.high:

Script	Description	Remarks
range.low:	Lower limit value	Lower limit value of the device values that display the figure color Value: Positive real number (in pixels)
range.high:	Upper limit value	Upper limit value of the device values that display the figure color

## b Figure settings

▶ Setting range count ... rangeNum:

Specifies the number of colors (1 to 5) to change. (The default display color is not included.)

▶ Shape... figType:

Value	Description	Remarks
'Oval',	Oval	When specifying a circle, set the figure height to the figure width.
'Rect',	Rectangle	When specifying a square, set the figure height to the figure width.
'Tri',	Triangle	When specifying an inverted triangle, set the height of the figure size to a negative value.

▶ Figure size setting ... figWidth: figHeight:

Script	Description	Remarks
figWidth:	Figure width	Real number (in pixels)
figHeight:	Figure height	If a negative number is specified, it is handled as its absolute value.

## 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)
[Image display object](#)
[Figure display object](#)

### ▶ Figure color specification ... defCol: rangeCol:

Script	Description	Remarks
defCol:	Default display color	The color of the figure displayed when the device value is outside the setting range. (Default display figure color)
range.col:	Change color	The color of the figure displayed when each device value is inside the setting range.

RGB value or color name  
Examples: #FF9900 (RGB value), red (color name)

**Related Page** [7.3 Color Name/Color Code](#)

### Operation example

Centered on 0, an upper limit value and a lower limit value are set on, respectively, the positive side and negative side.



### JavaScript example (extract)

When three change colors are set, the JavaScript is as shown below.

JavaScript	Explanation
figType: 'Oval',	Oval figure
defCol: 'red',	Default display color setting
rangeNum: 3,	Setting for the number of colors to change
range: [	
{	
low: -500,	
high: 500,	Change setting (first)
col: 'green',	
},	
{	
low: -1000,	
high: 1000,	Change setting (second)
col: 'blue',	
},	
{	
low: -1500,	
high: 1500,	Change setting (third)
col: '#FF9900',	
},	
],	



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

Image display object

Figure display object

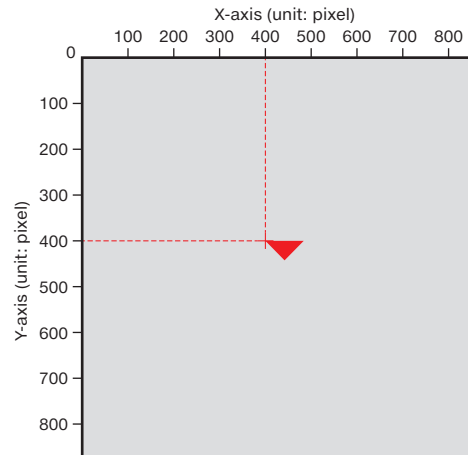
## C Display starting coordinates setting

### ► Web browser display starting position setting ... xPos: yPos:

Script	Description	Remarks
xPos:	X-axis coordinate	Setting range: Screen resolution of the personal computer (in pixels)
yPos:	Y-axis coordinate	

### ► How to view Web browser coordinates

Ex. (xPos, yPos) = (400, 400)



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)
[Image display object](#)
[Figure display object](#)

## ► HTML document explanation

Parameter handling

... Parameter setting required. An error occurs if this is not set.

### Caution

Write all the required parameters. If a required parameter is not written or if a setting value is outside of its range, an error occurs.

Line No.	JavaScript	Explanation
1	// Figure display object	Write a comment.
2	WSFigure({	Figure display object parameter setting + display execution
3	devName: 'D0',	Device specification: Device classification + device number
4	devFormat: 0,	Device data format
5	figType: 'tri',	Figure type, example: Triangle
6	figHeight: -30,	Figure size (If a negative number is specified, it is handled as its absolute value.)
7	figWidth: 60,	Figure size (If a negative number is specified, it is handled as its absolute value.)
8	defCol: 'red',	Default display color: Specified with a color name in this example
9	rangeNum: 2,	Setting range count
10	range:[	Setting range n: 1 to 5 Specifies the number of colors to change.
11	{	Specification start of the setting range to change
12	low: -5000,	Lower limit value of the device values that display the figure color
13	high: 5000,	Upper limit value of the device values that display the figure color
14	col: 'green',	Display color when the value is within the upper and lower limit values
15	},	
16	{	
17	low: -10000,	Lower limit value of the device values that display the figure color
18	high: 10000,	Upper limit value of the device values that display the figure color
19	col: 'blue',	Display color when the value is within the upper and lower limit values
20	},	
21	],	Specification end of the setting range to change
22	xPos: 300,	Display starting position
23	yPos: 730,	Display starting position
24	});	JavaScript parameter setting end

Execution

Parameters

Value	Description
0	16-bit signed

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

**Operation Buttons**

Write button object

Logout button object

## 5.6 Operation Buttons

### Write button object (WSWrBtn)

#### Function

Writes the specified value to the specified device.

#### Creation example



The character string within the button can be changed.

#### User Web page library window



### Object design

#### HTML (JavaScript) structure

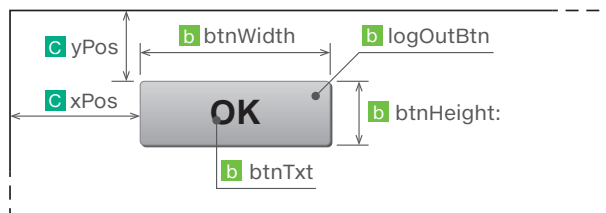
Create the HTML document for the parameters as shown below.

#### Usage declaration

Parameters Writing device specification

#### Execution

#### List of parameters



#### a Device settings

Device name ... devName: Device classification + device number

Related Page 5.2 Common Items - Device Setting Method

Data number base ... devBase:

Value	Description
K	Decimal
H	Hexadecimal
B	Binary

Related Page 5.2 Common Items - Device Setting Method

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)
[Write button object](#)
[Logout button object](#)

## ▶ Data format ... devFormat:

Value	Description	Value	Description
0	16-bit signed	4	Single-precision real number
1	16-bit unsigned	5	Not used
2	32-bit signed	6	Bit
3	32-bit unsigned		

**Related Page** [5.2 Common Items - Device Setting Method](#)

## ▶ Write value ... wrVal:

Script	Description	Remarks
wrVal:	Write value	The setting range varies depending on the number base or data format of the device.

**Special Note** Specify the input value with a character string. (Enclose it in single quotation marks ['].)  
Example: wrVal: '1'

## b Button design

### ▶ Style Sheet class element name ... wrBtn:

Script	Description	Remarks
wrBtn:	class element name	Used with Style Sheet selectors

### ▶ Button display text ... btnTxt:

Script	Description
btnTxt:	Text displayed on the button (A blank is displayed when this parameter is omitted.)

### ▶ Button size specification ... btnWidth: btnHeight:

Script	Description	Remarks
btnWidth:	Button width	Positive real number (in pixels)
btnHeight:	Button height	

### ▶ Write confirmation message presence ... wrConfirm:

Specifies whether to display a write confirmation message in the terminal OS when the button is operated (when the device value is written).

→ For details, refer to Message language ... language:.

Script	Description	Remarks
wrConfirm: 0	Do not display.	0 (do not display the write confirmation message) when this parameter is omitted.
wrConfirm: 1	Display.	

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

Write button object

Logout button object

## ▶ Message language ... language:

Script	Description	Remarks
language:0	Japanese	Set to 1 (English) when the parameter is omitted.
language:1	English	
language:2	Chinese (Simplified)	

**Ex.** When the OS is Windows®

language:0

language:1

language:2

The title bar and button text in the dialog box shown above are displayed in the language set with the language setting in Windows®. The windows shown above are from the Japanese version of Windows® 10.

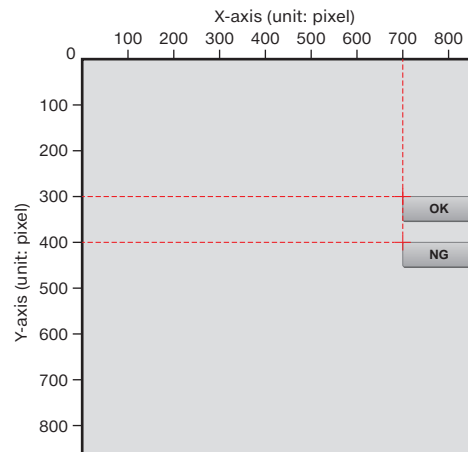
## C Display starting coordinates setting

### ▶ Web browser display starting position setting ... xPos: yPos:

Script	Description	Remarks
xPos:	X-axis coordinate	Setting range: Screen resolution (in pixels)
yPos:	Y-axis coordinate	

### ▶ How to view Web browser coordinates

**Ex.** OK button: (xPos, yPos) = (700, 300)  
 NG button: (xPos, yPos) = (700, 400)



# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)
[Write button object](#)
[Logout button object](#)

## ► HTML document explanation

Parameter handling

... Parameter setting required. An error occurs if this is not set.

### Caution

Write all the required parameters. If a required parameter is not written or if a setting value is outside of its range, an error occurs.

Line No.	JavaScript	Explanation	
<b>OK button</b>			
Execution	1 // Write button object	Write a comment.	
	2 WSWrtBtn{	Write button object parameter setting + display execution	
Display device specification (lines 1 and 2)	3 devName :'X0',	Device specification      Device classification + device number	
	4 devBase :'B',	Device data number base	
	5 devFormat :6,	Device data format	
	6 wrVal :'1',	Sets the write value to '1' (turns ON X0). Sets the input value with a character string. (Enclose it in single quotation marks ['].)	
	7 wrBtn :'write_btn',	Style Sheet class element name	
	8 btnTxt :'OK',	Sets the text displayed on the button to 'OK'.	
	9 btnWidth :150,	Button size	Width (in pixels)
	10 btnHeight :50,		Height (in pixels)
	11 wrConfirm :1,	Write confirmation message presence	
	12 language :0,	Message language	
	13 xPos :700,	Display starting position	X-coordinate (in pixels)
	14 yPos :40,		Y-coordinate (in pixels)
	15 }	JavaScript parameter setting end	

Line No.	JavaScript	Explanation	
<b>NG button</b>			
Execution	16 // Write button object	Write a comment.	
	17 WSWrtBtn{	Write button object display execution	
Display device specification (lines 1 and 2)	18 devName :'X0',	Device specification      Device classification + device number	
	19 devBase :'B',	Device data number base	
	20 devFormat :6,	Device data format	
	21 wrVal :'0',	Sets the write value to '0' (turns OFF X0). Sets the input value with a character string. (Enclose it in single quotation marks ['].)	
	22 wrBtn :'write_btn',	Style Sheet class element name	
	23 btnTxt :'NG',	Sets the text displayed on the button to 'NG'.	
	24 btnWidth :150,	Button size	Width (in pixels)
	25 btnHeight :50,		Height (in pixels)
	26 wrConfirm :1,	Write confirmation message presence	
	27 language :0,	Message language	
	28 xPos :700,	Display starting position	X-coordinate (in pixels)
	29 yPos :130,		Y-coordinate (in pixels)
	30 }	JavaScript parameter setting end	

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

What Is JavaScript?

Common Items

Table

Graphs

Image/Figure

Operation Buttons

Write button object

Logout button object

## Logout button object (WSLogoutBtn)

### Function

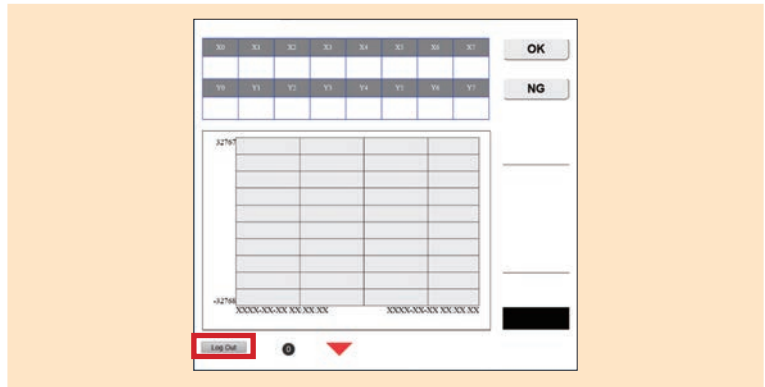
Place the button for logging out.

### Creation example



The character string within the button can be changed.

### User Web page library window



## Object design

### Restriction

- Only one logout button object can be used per Web page.
- Be sure to store the user login page (Log-in\_User.html) on the SD memory card.
- If a parameter is not specified, the default style specified with the Style Sheet (UserWebStyle.css) is used.

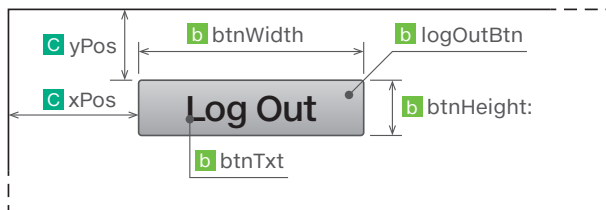
### HTML (JavaScript) structure

Parameters

Display settings

Execution

### List of parameters



#### a Button display settings

##### Size specification ... btnWidth: btnHeight:

Script	Description
btnWidth:	Button size Width (in pixels)
btnHeight:	Height (in pixels)

##### Button display text ... btnTxt:

Script	Description
btnTxt:	Set text displayed on the button (A blank is displayed when this parameter is omitted.)

Ex.	Script	Description
	btnTxt: 'ログアウト'	Japanese
	btnTxt: 'Log Out'	English
	btnTxt: '退出'	Chinese

# 5. LEARNING THE FUNCTIONS OF OBJECTS THAT CAN BE DISPLAYED WITH JAVASCRIPT (JS)

[What Is JavaScript?](#)
[Common Items](#)
[Table](#)
[Graphs](#)
[Image/Figure](#)
[Operation Buttons](#)
[Write button object](#)
[Logout button object](#)

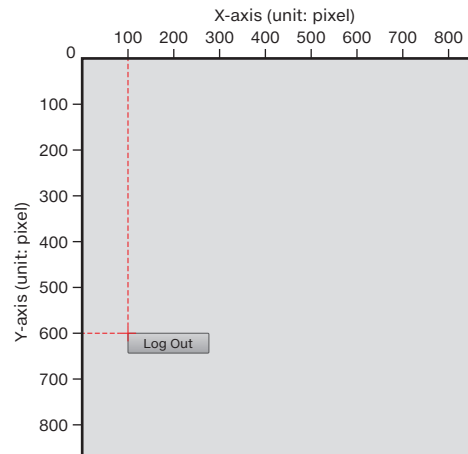
## b Display starting coordinates setting

### ► Web browser display starting position setting ... xPos: yPos:

Script	Description	Remarks
xPos:	X-axis coordinate	Setting range: Screen resolution (in pixels)
yPos:	Y-axis coordinate	

### ► How to view Web browser coordinates

**Ex.** (xPos, yPos) = (100, 600)



## ► HTML document explanation

Parameter handling

... Parameter setting required. An error occurs if this is not set.

### Caution

Write all the required parameters. If a required parameter is not written or if a setting value is outside of its range, an error occurs.

	Line No.	JavaScript	Explanation
Parameters	1	// Logout button object	Write a comment.
	2	logoutBtnParam = {	JavaScript parameter setting start
	3	xPos        20,	Display starting position
	4	yPos        730,	
	5	btnWidth   26,	Button size
	6	btnHeight  100,	
	7	btnTxt     'Log Out'	Text displayed on the button
	8	}	JavaScript parameter setting end
Execution	9	WSLogoutBtn(logoutBtnParam);	Execution



# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What Is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

## 6.1 What Is CGI?

CGI stands for “Common Gateway Interface”.

Its mechanism is as follows: in response to a request from a Web browser or a similar source, a program is executed on a Web server. The execution result is then returned to the client and is displayed on the Web browser. You can use CGI objects to create a simple user Web page with a small file size.

The device reading and writing CGI objects shown below can be used in user Web pages.

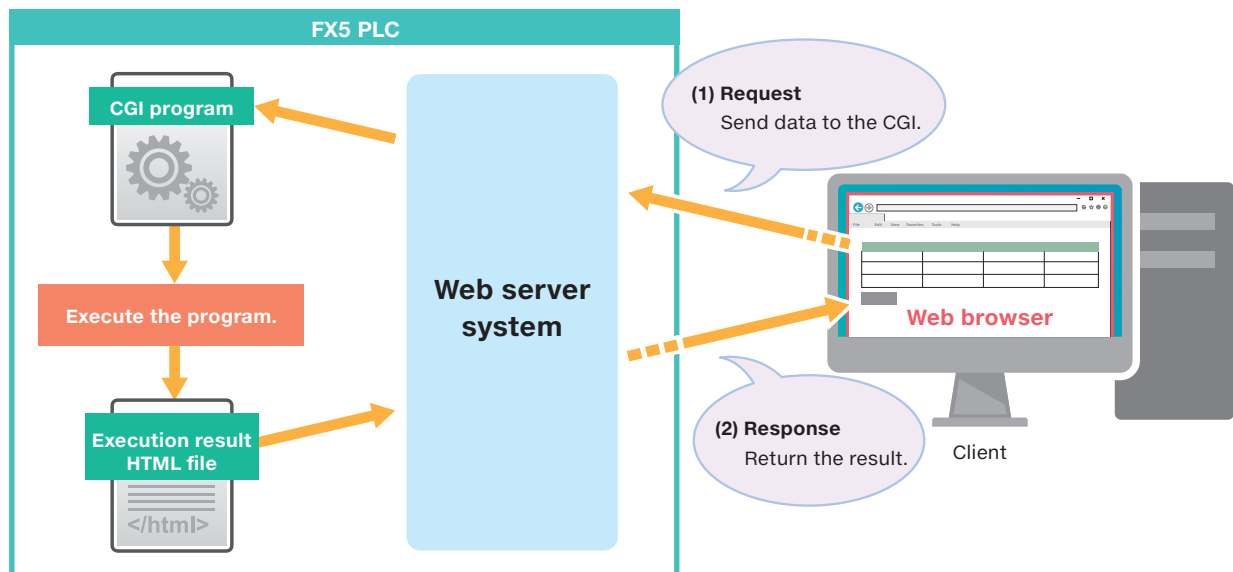
Item	CGI name	Function	Reference section
Device reading CGI	RdDevRnd.cgi	Reads the current value of the specified device.	6.4
Device writing CGI	WrDev.cgi	Writes the setting value to the specified device.	6.5

## 6.2 Overview and Functions of CGI

### Overview of CGI

A Web server system operates on the FX5 PLC when the Web server function is enabled with a GX Works3 parameter.

The mechanism of CGI is as follows: in response to access from a Web browser, a CGI program is executed within the Web server in the PLC, and then the result is returned to the Web browser.



# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

## Functions of CGI

The device reading/writing CGI objects shown below can be used in user Web pages. There are two types of CGI objects: **[those for reading from devices]** and **[those for writing to devices]**. You can use CGI to read/write values from/to devices by clicking the read/write buttons.

### Function of the CGI used to read from devices

Click [Read] to update and display the monitor values.

The default values are the device names and data formats created with the HTML. The values are blank until the [Read] button is clicked. Also, the displayed device names and data formats can be changed.

CGI name	RdDevRnd.cgi
Function	Reads the current value of the specified device.

### Window example

Device name	Data type	Value
D10	16-bit integer	1234
D11	32-bit integer	123456
M0	Bit	0

Read

Devices set with the HTML

Constants set with the HTML

### Operation procedure

As an example, this section explains the procedure for changing D11 on line 2 to D20 and monitoring the devices.

#### 1 Click Read.

Device name	Data type	Value
D10	16-bit integer	
D11	32-bit integer	
M0	Bit	

Read

#### 2 Display the monitor values.

Device name	Data type	Value
D10	16-bit integer	1234
D11	32-bit integer	123456
M0	Bit	0

Read

#### 3 Click (for example) the D11 cell.

Device name	Data type	Value
D10	16-bit integer	1234
D11	32-bit integer	123456
M0	Bit	0

Read

Select the device name to display the Close button (x).

#### 4 Clear (for example) D11.

Device name	Data type	Value
D10	16-bit integer	1234
D11	32-bit integer	123456
M0	Bit	0

Read

Click the Close button (x) to delete the device name.

#### 5 Enter D20.

Device name	Data type	Value
D10	16-bit integer	1234
D20	32-bit integer	123456
M0	Bit	0

Read

#### 6 Click Read to update the D20 monitor value.

Device name	Data type	Value
D10	16-bit integer	1234
D20	32-bit integer	0
M0	Bit	0

Read

Update and display the values of D20 (the device that was changed), D10, and M0.

### Related Page

For Web page creation method of the device monitor using CGI parts, refer to the following.

- Specifications of the devices which can be used  
6.3 Common Specifications
- Procedure of device reading Web page creation  
6.4 Device Reading Web Page Creation

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

## Function of the CGI used to write to devices

Click [Write] to write the entered values.

The default values are the device names, data formats, and values created with the HTML. Also, the device names, data formats, and values can be changed and written.

CGI name	WrDev.cgi
Function	Writes the setting value to the specified device.

## Window example

Device name	Data type	Value	
D10	16-bit integer	3	Write
D11	32-bit integer	10	Write
M0	Bit	1	Write

Devices set with the HTML

Constants set with the HTML

## Operation procedure

As an example, this section explains the procedure for changing D11 on line 2 to D20 and writing 55 to D20.

1 Click (for example) the D11 cell.

Device name	Data type	Value	
D10	16-bit integer	3	Write
D11	32-bit integer	10	Write
M0	Bit	1	Write

Select the device name to display the Close button (x).

2 Clear (for example) D11.

Device name	Data type	Value	
D10	16-bit integer	3	Write
D11	32-bit integer	10	Write
M0	Bit	1	Write

Click the Close button (x) to delete the device name.

3 Enter D20.

Device name	Data type	Value	
D10	16-bit integer	3	Write
D20	32-bit integer	10	Write
M0	Bit	1	Write

4 Click (for example) the 10 cell.

Device name	Data type	Value	
D10	16-bit integer	3	Write
D20	32-bit integer	10	Write
M0	Bit	1	Write

Select the value and click the Close button (x).

5 Clear (for example) value 10.

Device name	Data type	Value	
D10	16-bit integer	3	Write
D11	32-bit integer	10	Write
M0	Bit	1	Write

Click the Close button (x) to delete the value.

6 Enter 55.

Device name	Data type	Value	
D10	16-bit integer	3	Write
D11	32-bit integer	55	Write
M0	Bit	1	Write

7 Click Write.

Device name	Data type	Value	
D10	16-bit integer	3	Write
D20	32-bit integer	55	Write
M0	Bit	1	Write

The value 55 is written to D20 on the PLC.

## Related Page

For Web page creation method of the device monitor using CGI parts, refer to the following.

- Specifications of the devices which can be used  
6.3 Common Specifications
- Procedure of device reading Web page creation  
6.5 Device Writing Web Page Creation

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

## 6.3 Common Specifications

### Data specified with CGI objects

#### ▶ Device name

Displays the device that can be accessed with the device reading CGI/device writing CGI.

Classification	Device
Bit device*1	X, Y, M, L, B, F, SB, S, TS, TC, STS, STC, CS, CC, LCS, LCC, SM
Word device*2	T (current value), ST (current value), C (current value), D, W, SW, SD, U□\G□, Z, R
Double word device	LC (current value), LZ

\*1: Use hexadecimal to specify octal device names (X\*\*\*, Y\*\*\*). (Example: To specify X20, specify X10 in CGI.)

\*2: When specifying the "U□\G□" device name, use two "\" characters to specify the name as "U□\\G□". The first "\" (escape sequence) is a symbol that has the meaning of a special character, so the above expression is required.

#### ▶ Device size

The usable device size varies depending on the device. Specify device number notations separately using octal, decimal, or hexadecimal depending on the device.

✓: Can be used with both the device reading CGI and the device writing CGI, x: Cannot be used

Device	Notation	Device size		
		Bit	Word	Double word
User devices				
Input (X)	Octal*1	✓	x	x
Output (Y)	Octal*1	✓	x	x
Internal relay (M)	Decimal	✓	x	x
Latch relay (L)	Decimal	✓	x	x
Link relay (B)	Hexadecimal	✓	x	x
Annunciator (F)	Decimal	✓	x	x
Link special relay (SB)	Hexadecimal	✓	x	x
Step relay (S)	Decimal	✓	x	x
Timer (T)*2	TS (contact)	Decimal	✓	x
	TC (coil)	Decimal	✓	x
	TN (current value)	Decimal	x	✓
Accumulation timer (ST)*2	STS (contact)	Decimal	✓	x
	STC (coil)	Decimal	✓	x
	STN (current value)	Decimal	x	✓
Counter (C)*2	CS (contact)	Decimal	✓	x
	CC (coil)	Decimal	✓	x
	CN (current value)	Decimal	x	✓
Long counter (LC)*2	LCS (contact)	Decimal	✓	x
	LCC (coil)	Decimal	✓	x
	LCN (current value)	Decimal	x	✓
Data register (D)	Decimal	x	✓	✓
Link register (W)	Hexadecimal	x	✓	✓
Link special register (SW)	Hexadecimal	x	✓	✓
System devices				
Special relay (SM)	Decimal	✓	x	x
Special register (SD)	Decimal	x	✓	✓
Module access device (U□\G□)				
G (U□\G□)	Decimal	x	✓	✓
Index registers				
Index register (Z)	Decimal	x	✓	✓
Long index register (LZ)	Decimal	x	x	✓
File register				
File register (R)	Decimal	x	✓	✓

\*1: This is handled as a hexadecimal value in CGI.

\*2: When T, ST, C, or LC is specified, it is handled as the device of the current value (TN, STN, CN, or LCN).

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

**Common Specifications**

Device Reading Web Page Creation

Device Writing Web Page Creation

## ► Device value

Use the notations shown below with the device values handled by CGI objects.

Value handling	Handle device values with hexadecimal notation in the String format (as character strings).
Conversions made by JavaScript	When using octal, decimal, or real number values in a Web page, use JavaScript (FUserWebLib.js) to convert such values into hexadecimal.
How to specify hexadecimal values	It is not necessary to add "0x" at the start of the value. Zero padding is also unnecessary. For example, to write the value "0x012F", write "12F". In the same manner, the read value is displayed as "12F" without "0x" added at its start and without any zero padding.
	Alphabet characters in hexadecimal values are not case sensitive. For example, to write the value "0x012F", write "12F" or "12f". The alphabet characters in read values are displayed in uppercase.

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

[What is CGI?](#)
[Overview and Functions of CGI](#)
[Common Specifications](#)
[Device Reading Web Page Creation](#)
[Device Writing Web Page Creation](#)
[Device reading CGI specifications](#)
[Layout method](#)
[HTML Creation Example](#)

## 6.4 Device Reading Web Page Creation

### Device reading CGI specifications

Name	Function
Device reading CGI	Requests the reading of the current value of the specified device.

### ▶ Access method and access information

The main methods of communicating with CGI are GET and POST. The Web server function supports POST.

Item	Description
Access method	method="POST"
Access destination information (URL)	/cgi/RdDevRnd.cgi

### ▶ Request specifications

The following table lists the parameters used by requests. Specify parameters with the query string format.

Specify DEV(n) and TYP(n) with sequence numbers. If these parameters are not specified with sequence numbers, an error will occur.

Parameter name	Data type	Description	Setting range	Reference section
NUM	string	Hexadecimal character string indicating the number of reading devices (n: 1 to 20)	Set this parameter so that the total number of devices specified for reading and writing per Web page is 32 or less.	Next page
DEV1	string	Device name 1	A character string containing 16 or less alphanumeric characters. (Characters are not case sensitive. Indirect specification, bit specification, digit specification, and index modification are not possible.)	6.3
TYP1	string	Device size 1	B: Bit W: Word D: Double word	6.3
:				
DEV(n)	string	Device name n	The same as parameter name [DEV1] given above	6.3
TYP(n)	string	Device size n	The same as parameter name [TYP1] given above	6.3

**Ex.** D0, M100 ... Reading 10 SD0 devices

NUM=A&DEV1=D0&TYP1=D&DEV2=M100& ... &DEV10=SD0&TYP10=W

#### Terminology

The **query string format** is a format used to pass data (parameters) to a Web server.

Assign values to parameters by adding the string [&parameter name=value] to the end of a URL. To pass multiple parameters, connect the query strings with [&].

Example: [http://www.melsec/iq-f&param1=0&param2="Sample"](http://www.melsec/iq-f&param1=0&param2=)

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

Device reading CGI specifications

Layout method

HTML Creation Example

- The maximum number of devices that can be used on a single Web page is 32

**Ex.** Web browser display window ..... Total: 10

Read

Device name	Data type	Value
D10	16-bit integer	
D11	16-bit integer	
D12	16-bit integer	
D13	16-bit integer	
D14	16-bit integer	

5

Write

Device name	Data type	Value	
D10	16-bit integer	25	Write
D11	16-bit integer	3	Write
D12	16-bit integer	30	Write
D13	16-bit integer	55	Write
D14	16-bit integer	8	Write

5

## ▶ Response specifications

The following table lists the parameters used by responses. Response data is in JSON format.

Parameter name	Data type	Description	
RET	string	Value	Execution result (hexadecimal character string) details
		0000	Normal
		0001	Not logged in
		0005	Illegal referer
		4005	Too many devices
		4030	Illegal device classification
		4031	Out of device range
		4041	Error: Specified buffer memory number + specified number of items to transfer is outside of the buffer memory area.
		4043	Non-existent module specification error
4080	CGI parameter error		
DATA	string	Hexadecimal character string of the read values (an array)	

**Ex.** Device reading CGI response data

```

{
  "RET": "0",
  "DATA": [
    "100",
    :
    "FABC"
  ]
}
```

In the message, the response on the left is transferred in the following format.

**Normal** {"RET": "0", "DATA": ["100", ..., "FABC"]}

**Error** {"RET": "4031"}

### Terminology

**JSON** is an abbreviation of "JavaScript Object Notation".

This is a language for writing the data structure and is used as a simple database. It was created by aiming for a format that can be easily handled by computers and easily viewed by users.

## 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

[What is CGI?](#)
[Overview and Functions of CGI](#)
[Common Specifications](#)
[Device Reading Web Page Creation](#)
[Device Writing Web Page Creation](#)
[Device reading CGI specifications](#)
[Layout method](#)
[HTML Creation Example](#)

### ▶ Displaying device values in real-number format on a Web page



Device name	Value
D0	B36FH
D1	35A5H

Values are stored as single-precision real numbers in D0 and D1. In this figure, the device values are monitored as hexadecimal values.



Device name	Value
DATA[0]	"35a5b36f"

Make a request with D: double word specified for the device size.



Device name	Data type	Value
D0	Single-precision real number	1.234568E-06

Convert the read data into real-number format with JavaScript.



# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

Device reading CGI specifications

Layout method

HTML Creation Example

## Web browser display layout method

In the device reading CGI, the table is created with the <table> tag.

The <input> tag is used to set cell contents such as item names and the names of the devices to monitor.

### Web browser display

Device name	Data type	Value
D10	32-bit integer	
SD0	16-bit integer	
M0	Bit	

The HTML document for the above table is shown below.

Device name	Data type	Value
D10	32-bit integer	
SD0	16-bit integer	
M0	Bit	

Different colors are used to indicate what objects are created by the HTML.

### HTML document

```
<form id="devform" name="readdev" method="post">
```

```
  <table id="devtbl" class="devtbl" border="1">
    <tr>
      <th>Device name</th><th>Data type</th><th>Value</th>
    </tr>
```

```
    <tbody>
```

```
      <tr>
```

```
        <td><input type="text" id="DEV1" name=" DEV1" class="input" value='D10' /></td>
```

```
        <td><input type="text" id="TYP1" name=" TYP1" class="input" value='32-bit integer' /></td>
```

```
        <td><input type="text" id="VAL1" name=" VAL1" class="read-input"/></td>
```

```
      </tr>
```

```
      <tr>
```

```
        <td><input type="text" id="DEV2" name="DEV2" class="input" value='SD0' /></td>
```

```
        <td><input type="text" id="TYP2" name="TYP2" class="input" value='16-bit integer' /></td>
```

```
        <td><input type="text" id="VAL2" name="VAL2" class="read-input"/></td>
```

```
      </tr>
```

```
      <tr>
```

```
        <td><input type="text" id="DEV3" name="DEV3" class="input" value='M0' /></td>
```

```
        <td><input type="text" id="TYP3" name="TYP3" class="input" value='bit' /></td>
```

```
        <td><input type="text" id="VAL3" name="VAL3" class="read-input"/></td>
```

```
      </tr>
```

```
    </tbody>
```

```
  </table>
```

```
  <input type="button" value="Read" class="input" onclick=" ReadDeviceRandomTbl(devtbl)"/>
```

```
</form>
```

Form

Table construction

D10 row

SD0 row

M0 row

### Related Page

For details on the form and table construction, refer to the following.  
7.2 HTML Tag References

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

Device reading CGI specifications

Layout method

HTML Creation Example

## HTML Creation Example

Use the following procedure to create the device reading Web page.

### STEP 1. Use Notepad to create the HTML on the next page.

```
Untitled - Notepad
File Edit Format View Help

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<!-- charset setting "Because the Web server setting is UTF-8, specify UTF-8." -->
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge"/>
<!-- Set the title -->
<title>Device Reading CGI Sample</title>
<!-- Write the user JavaScript here. -->
<script>
// CGI request function
function ReadDeviceRandom1bit(devtblid) {
var devtblitem = document.getElementById(devtblid);
var i, devitem, typitem;
var tblrows = devtblitem.rows.length;
var param;

// Setting the number of devices
param = "NUM=" + (tblrows - 1) + "&";
for (i=1; i <= tblrows; i++) {
// Device name parameter setting
devitem = document.getElementById(devtblitem.rows[i].cells[0].childNodes[0].id);
param += devitem.name + "=" + devitem.value + "&";

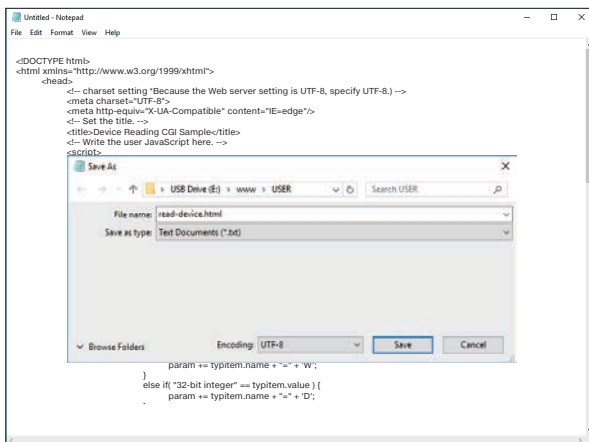
// Device size parameter setting
typitem = document.getElementById(devtblitem.rows[i].cells[1].childNodes[0].id);
if ("Bit" == typitem.value) {
param += typitem.name + "=" + 'B';
}
else if ("16-bit integer" == typitem.value) {
param += typitem.name + "=" + 'W';
}
else if ("32-bit integer" == typitem.value) {
param += typitem.name + "=" + 'D';
}
}
}
}
}

</script>
</head>
<body>
<table border="1">
| Device Name | Data Type | Value |
| --- | --- | --- |
| D10 | 16-bit Integer |  |
| D11 | 32-bit Integer |  |
| M0 | Bit |  |


<div>
<input type="button" value="Read"/>
</div>
</body>
</html>
```

Details on the HTML are also included in the MELSEC iQ-F FX5 User's Manual (Ethernet Communication) [JY997D56201].

### STEP 2. Save the file with an HTML file format.



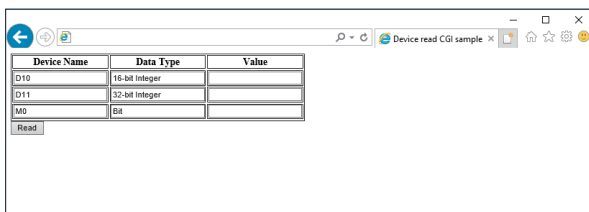
**Operation** >> Select [File] ► [Save As].

- ▶ Enter the File name (example: read-device.html).
- ▶ For the Encoding, select UTF-8, and then click [Save].

#### Point

If you specify a value other than UTF-8 for the Encoding, the text may be garbled when the HTML is displayed in a Web browser.

### STEP 3. Double-click the created file to display it in a Web browser.



### STEP 4. Store the created HTML file on an SD memory card, and then use the Web server function of the PLC to display this file.

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

Device reading CGI specifications

Layout method

HTML Creation Example

## ▶ Sample HTML

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charset setting *Because the Web server setting is UTF-8, specify UTF-8.) -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- Set the title. -->
    <title>Device Reading CGI Sample</title>
    <!-- Write the user JavaScript here. -->
    <script>
      // CGI request function
      function ReadDeviceRandomTbl(devtblid) {
        var devtblitem = document.getElementById(devtblid);
        var i, devitem, typitem;
        var tblrows = devtblitem.rows.length;
        var param;

        // Setting the number of devices
        param = "NUM=" + (tblrows - 1) + '&';
        for( i=1; i < tblrows; i++) {
          // Device name parameter setting
          devitem = document.getElementById(devtblitem.rows[i].cells[0].childNodes[0].id);
          param += devitem.name + "=" + devitem.value + '&';

          // Device size parameter setting
          typitem = document.getElementById(devtblitem.rows[i].cells[1].childNodes[0].id);
          if( "Bit" == typitem.value ) {
            param += typitem.name + "=" + 'B';
          }
          else if( "16-bit integer" == typitem.value ) {
            param += typitem.name + "=" + 'W';
          }
          else if( "32-bit integer" == typitem.value ) {
            param += typitem.name + "=" + 'D';
          }
          else {
            param += typitem.name + "=" + 'Q';
          }
          if( i < (tblrows - 1) ) param += '&';
        }
        // CGI request
        xhr = new XMLHttpRequest();
        xhr.open("POST", "/cgi/RdDevRnd.cgi", true);
        xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
        var FUNC = function() { ReadDeviceRandomTbl_Response(xhr, devtblid); }; // Response analysis function setting
        xhr.onreadystatechange = FUNC;
        xhr.send(param);
      }
    </script>
  </head>
  <body>
  </body>
</html>
```



Continued on the next page

## 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

Device reading CGI specifications

Layout method

HTML Creation Example



Continued from the previous page

```
// Response analysis function
function ReadDeviceRandomTbl_Response(xhr, devtblid) {
    // Check of the XMLHttpRequest client status
    // 0:UNSENT 1:OPENED 2:HEADERS_RECEIVED 3:LOADING 4:DONE
    if( 4 != xhr.readyState ) {
        // If the status is a value other than 4, DONE (operation complete), end the processing.
        return;
    }

    // HTTP response code check
    if ( 200 != xhr.status ) {
        // If the value is not "200 OK", display an error dialog box.
        alert("HTTP STATUS ERROR=" + xhr.status);
        return;
    }

    var i, dataitem;
    var devtblitem = document.getElementById(devtblid);
    var tblrows = devtblitem.rows.length; // Obtain the number of table rows (including the header).
    var res = JSON.parse( xhr.response ); // JSON character string analysis and processing

    // Judgment of the result from the CGI
    if( res.RET != "0000" ) {
        // If an error has occurred, display an error dialog box.
        alert("ERROR=" + res.RET);
    }
    else {
        // If the result is normal, reflect the obtained values in the table.
        for ( i = 1, m = 0; i < tblrows; i++, m++) {
            dataitem = document.getElementById(devtblitem.rows[i].cells[2].childNodes[0].id);
            // Set the table values to the reading result here (convert the hexadecimal character string to values).
            dataitem.value = parseInt(res.DATA[m], 16);
        }
        alert("Reading complete");
    }
}
</script>
</head>
<body>
<form>
<table id="devtbl" class="devtbl" border="1">
<tr>
<th>Device name</th>
<th>Data type</th>
<th>Value</th>
</tr>
<tbody>
<tr>
<td><input type="text" id="DEV1" name="DEV1" class="input" value="D10"/></td>
<td><input type="text" id="TYP1" name="TYP1" class="input" value="16-bit integer"/></td>
<td><input type="text" id="DATA1" name="DATA1" class="read-input" /></td>
</tr>
<tr>
<td><input type="text" id="DEV2" name="DEV2" class="input" value="D11"/></td>
<td><input type="text" id="TYP2" name="TYP2" class="input" value="32-bit integer"/></td>
<td><input type="text" id="DATA2" name="DATA2" class="read-input" /></td>
</tr>
<tr>
<td><input type="text" id="DEV3" name="DEV3" class="input" value="M0"/></td>
<td><input type="text" id="TYP3" name="TYP3" class="input" value="Bit"/></td>
<td><input type="text" id="DATA3" name="DATA3" class="read-input" /></td>
</tr>
</tbody>
</table>
<input type="button" value="Read" class="input" onclick="ReadDeviceRandomTbl('devtbl')"/>
</form>
</body>
</html>
```

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

Device writing CGI specifications

Layout method

HTML Creation Example

## 6.5 Device Writing Web Page Creation

### Device writing CGI specifications

Name	Function
Device writing CGI	Orders the writing of the specified value to the specified device.

### ► Access method and access information

The main methods of communicating with CGI are GET and POST. The Web server function supports POST.

Item	Description
Access method	method="POST"
Access destination information (URL)	/cgi/WrDev.cgi

### ► Request specifications

The following table lists the parameters used by requests. Specify parameters with the query string format.

Parameter name	Data type	Description	Setting range	Reference section
NUM	string	Number of devices to write to (1).	Set this to 1. (Set this parameter so that the total number of devices specified for reading and writing per Web page is 32 or less.)	Next page
DEV1	string	Device name	A character string containing 10 or less alphanumeric characters. (Characters are not case sensitive. Indirect specification, bit specification, digit specification, and index modification are not possible.)	6.3
TYP1	string	Device size	B: Bit W: Word D: Double word	6.3
DATA1	string	Write value	Hexadecimal character string	—

**Ex.** Writing FFFF to D0

NUM=1&DEV1=D0&TYP1=W&DATA1=FFFF

### Terminology

The **query string format** is a format used to pass data (parameters) to a Web server.

Assign values to parameters by adding the string [&parameter name=value] to the end of a URL. To pass multiple parameters, connect the query strings with [&].

Example: `http://www.melsec/iq-f&param1=0&param2="Sample"`

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

Device writing CGI specifications

Layout method

HTML Creation Example

- The maximum number of devices that can be used on a single Web page is 32

**Ex.** Web browser display window ..... Total: 10

Read

Device name	Data type	Value
D10	16-bit integer	
D11	16-bit integer	
D12	16-bit integer	
D13	16-bit integer	
D14	16-bit integer	

5

Write

Device name	Data type	Value	
D10	16-bit integer	25	<input type="button" value="Write"/>
D11	16-bit integer	3	<input type="button" value="Write"/>
D12	16-bit integer	30	<input type="button" value="Write"/>
D13	16-bit integer	55	<input type="button" value="Write"/>
D14	16-bit integer	8	<input type="button" value="Write"/>

5

## ▶ Response specifications

The following table lists the parameters used by responses. Response data is in JSON format.

Parameter name	Data type	Description	
RET	string	Value	Execution result (hexadecimal character string) details
		0000	Normal
		0002	No user rights (CGI was executed by a user without the permission or rights to write to devices.)
		0001	Not logged in
		0005	Illegal referer
		4005	Too many devices
		4030	Illegal device classification
		4031	Out of device range
		4041	Error: Specified buffer memory number + specified number of items to transfer is outside of the buffer memory area.
4043	Non-existent module specification error		
4080	CGI parameter error		
DATA	string	Hexadecimal character string of the read values (an array) from the devices written to	

**Ex.** Device writing CGI response data

```

{
  "RET": "0",
  "DATA": [
    "100"
  ]
}
```

In the message, the response on the left is transferred in the following format.

- Normal {"RET":"0","DATA":["100"]}
- Error {"RET":"4031"}

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

Device writing CGI specifications

Layout method

HTML Creation Example

## ▶ Displaying device values in real-number format on a Web page



Device name	Data type	Value
D0	Single-precision real number	1.234568E-06

Convert the entered data into hexadecimal with JavaScript.



Device name	Value
DATA1	"35a5b36f"

Make a request with D: double word specified for the device size.



Device name	Value
D0	B36FH
D1	35A5H

Values are stored as single-precision real numbers in D0 and D1. In this figure, the device values are monitored as hexadecimal values.

## 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

[What is CGI?](#)
[Overview and Functions of CGI](#)
[Common Specifications](#)
[Device Reading Web Page Creation](#)
[Device Writing Web Page Creation](#)
[Device writing CGI specifications](#)
[Layout method](#)
[HTML Creation Example](#)

### Web browser display layout method

In the device writing CGI, the table is created with the <table> tag.

The <input> tag is used to set cell contents such as item names and the names of the devices to monitor.

#### Web browser display

Device name	Data type	Value	
D10	32-bit integer	3	Write

The HTML document for the above table is shown below.

Device name	Data type	Value	
D10	32-bit integer	3	Write

Different colors are used to indicate what objects are created by the HTML.

#### HTML document

```
<form id="devform" name="readdev" method="post">
```

```
<table id="devtbl" class="devtbl" border="1">
```

```
<tr>
```

```
<th>Device name</th><th>Data type</th><th>Value</th>
```

```
</tr>
```

```
<tbody>
```

```
<tr>
```

```
<td><input type="text" id="DEV1" name=" DEV1" class=" input"value='D10'/></td>
```

```
<td><input type="text" id="TYP1" name=" TYP1" class=" input"value='32-bit integer'/></td>
```

```
<td><input type="text" id="DATA1" name=" DATA1" class=" input"value='3'/></td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

```
<input type="button" value="Write" class="input" onclick=" WriteDeviceBlockTbl(devbl,1,1)"/>
```

```
</form>
```

**Form**
**Table construction**



# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

Device writing CGI specifications

Layout method

HTML Creation Example

## HTML Creation Example

Use the following procedure to create the device writing Web page.

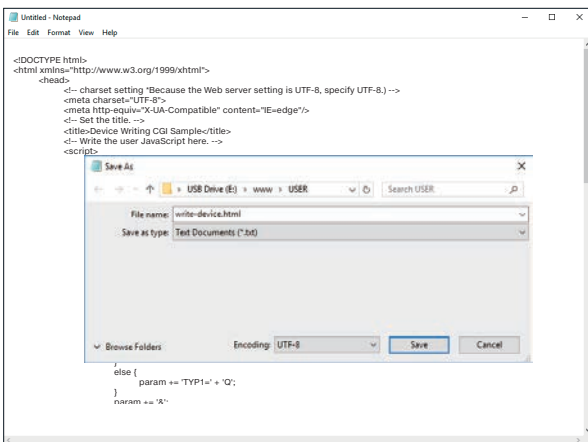
### STEP 1. Use Notepad to create the HTML on the next page.

```
Untitled - Notepad
File Edit Format View Help

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<!-- charset setting (Because the Web server setting is UTF-8, specify UTF-8.) -->
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge"/>
<!-- Set the title -->
<title>Device Writing CGI Sample</title>
<!-- Write the user JavaScript here. -->
<script>
function WriteDeviceBlockTbl(devtblid, row) {
var datalen;
var str;
var devtblitem = document.getElementById(devtblid);
// Setting the number of devices (fixed to 1)
var param = 'NUM=' + '1';
// Device name parameter setting
var devitem = document.getElementById(devtblitem.rows[row].cells[0].childNodes[0].id);
param += 'DEV1=' + devitem.value + '&';
// Device size parameter setting
var typitem = document.getElementById(devtblitem.rows[row].cells[1].childNodes[0].id);
if ('bit' == typitem.value) {
param += 'TYP1=' + 'B';
}
else if ('16-bit integer' == typitem.value) {
param += 'TYP1=' + 'W';
}
else if ('32-bit integer' == typitem.value) {
param += 'TYP1=' + 'D';
}
else {
param += 'TYP1=' + 'Q';
}
param += '&';
}
```

Details on the HTML are also included in the MELSEC iQ-F FX5 User's Manual (Ethernet Communication) [JY997D56201].

### STEP 2. Save the file with an HTML file format.



**Operation** >> Select [File] ▶ [Save As].

- ▶ Enter the File name (example: write-device.html).
- ▶ For the Encoding, select UTF-8, and then click
- ▶ [Save].

#### Point

If you specify a value other than UTF-8 for the Encoding, the text may be garbled when the HTML is displayed in a Web browser.

### STEP 3. Double-click the created file to display it in a Web browser.



### STEP 4. Store the created HTML file on an SD memory card, and then use the Web server function of the PLC to display this file.

## 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

[What is CGI?](#)
[Overview and Functions of CGI](#)
[Common Specifications](#)
[Device Reading Web Page Creation](#)
[Device Writing Web Page Creation](#)
[Device writing CGI specifications](#)
[Layout method](#)
[HTML Creation Example](#)

### ▶ Sample HTML

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charset setting *Because the Web server setting is UTF-8, specify UTF-8. -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- Set the title. -->
    <title>Device Writing CGI Sample</title>
    <!-- Write the user JavaScript here. -->
    <script>
      function WriteDeviceBlockTbl(devtblid, row) {
        var dataitem;
        var xhr;
        var devtblitem = document.getElementById(devtblid);
        // Setting the number of devices (fixed to 1)
        var param = 'NUM=1&';
        // Device name parameter setting
        var devitem = document.getElementById(devtblitem.rows[row].cells[0].childNodes[0].id);
        param += 'DEV1=' + devitem.value + '&';

        // Device size parameter setting
        var typitem = document.getElementById(devtblitem.rows[row].cells[1].childNodes[0].id);
        if( 'Bit' == typitem.value ) {
          param += 'TYP1=' + 'B';
        }
        else if( '16-bit integer' == typitem.value ) {
          param += 'TYP1=' + 'W';
        }
        else if( '32-bit integer' == typitem.value ) {
          param += 'TYP1=' + 'D';
        }
        else {
          param += 'TYP1=' + 'Q';
        }
        param += '&';

        // Data parameter setting
        var dataitem = document.getElementById(devtblitem.rows[row].cells[2].childNodes[0].id);
        param += 'DATA1=' + parseInt(dataitem.value).toString(16)

        // CGI request
        xhr = new XMLHttpRequest();
        xhr.open('POST', "/cgi/WrDev.cgi", true);
        xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
        var FUNC = function() { WriteDeviceBlockTbl_Response(xhr, typitem, dataitem); }; // Response analysis function setting
        xhr.onreadystatechange = FUNC;
        xhr.send(param);
      }
    </script>
  </head>
  <body>
  </body>
</html>
```



Continued on the next page

# 6. CREATING A DEVICE MONITOR WINDOW WITH CGI

What is CGI?

Overview and Functions of CGI

Common Specifications

Device Reading Web Page Creation

Device Writing Web Page Creation

Device writing CGI specifications

Layout method

HTML Creation Example



Continued from the previous page

```
// Response analysis function
function WriteDeviceBlockTbl_Response(xhr, typitem, dataitem) {
  // Check of the XMLHttpRequest client status
  // 0:UNSENT 1:OPENED 2:HEADERS_RECEIVED 3:LOADING 4:DONE
  if( 4 != xhr.readyState ) {
    // If the status is a value other than 4, DONE (operation complete), end the processing.
    return;
  }

  // HTTP response code check
  if ( 200 != xhr.status ) {
    // If the value is not "200 OK", display an error dialog box.
    alert("HTTP STATUS ERROR=" + xhr.status );
    return;
  }

  var value;
  var res = JSON.parse( xhr.response ); // JSON character string analysis and processing
  // Judgment of the result from the CGI
  if( res.RET != "0000" ) {
    // If an error has occurred, display an error dialog box.
    alert("ERROR=" + res.RET);
  }
  else {
    // If the result is normal, reflect the writing result values.
    dataitem.value = parseInt(res.DATA[0],16);
    alert("Writing complete");
  }
}
</script>
</head>
<body>
  <form>
    <table id="devtbl" class="devtbl" border="1">
      <tr>
        <th>Device name</th>
        <th>Data type</th>
        <th>Value</th>
      </tr>
      <tbody>
        <tr>
          <td><input type="text" id="DEV1" name="DEV1" class="input" value="D10"/></td>
          <td><input type="text" id="TYP1" name="TYP1" class="input" value="16-bit integer"/></td>
          <td><input type="text" id="DATA1" name="DATA1" class="input" value="3"/></td>
          <td><input type="button" value="Write" class="input" onclick="WriteDeviceBlockTbl('devtbl,1)"/></td>
        </tr>
        <tr>
          <td><input type="text" id="DEV2" name="DEV2" class="input" value="D11"/></td>
          <td><input type="text" id="TYP2" name="TYP2" class="input" value="32-bit integer"/></td>
          <td><input type="text" id="DATA2" name="DATA2" class="input" value="10"/></td>
          <td><input type="button" value="Write" class="input" onclick="WriteDeviceBlockTbl('devtbl,2)"/></td>
        </tr>
        <tr>
          <td><input type="text" id="DEV3" name="DEV3" class="input" value="M0"/></td>
          <td><input type="text" id="TYP3" name="TYP3" class="input" value="Bit"/></td>
          <td><input type="text" id="DATA3" name="DATA3" class="input" value="1"/></td>
          <td><input type="button" value="Write" class="input" onclick="WriteDeviceBlockTbl('devtbl,3)"/></td>
        </tr>
      </tbody>
    </table>
  </form>
</body>
</html>
```

# 7. REFERENCES

## Style Sheet

## HTML Tags

## Color Name/Color Code

### List of Properties

### Explanation

### Explanation for Usage Example

## 7.1 Style Sheet References

This section provides a simple explanation of the functions of the main properties of Style Sheets. For details on Style Sheets such as how to use them, refer to commercially available reference books.

### List of Properties

- This section explains the operations of the properties used by the user Web page library file (UserWebStyle.css).

No.	Property	Description
1	position:	Determines the position of the element.
2	z-index:	Specifies the overlapping order of the element.
3	width:	Element width
4	height:	Element height
5	margin-top:	Outer, upper margin of the element

No.	Property	Description
6	margin-left:	Outer, left margin of the element
7	background-color:	Element background color
8	border:	Element border
9	text-align:	Text position specification
10	font-weight:	Character thickness
11	cursor:	Cursor shape

### Explanation

- ▶ No. 1 [position:] ... Determines the position of the element.

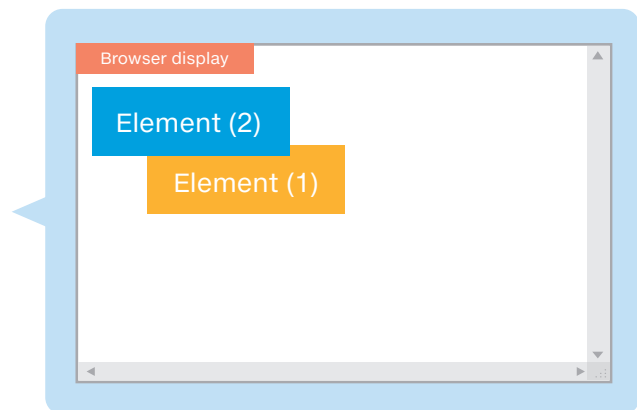
Value	Description
static	Do not specify a placement method (default value).
absolute	Placement in an absolute position (reference position: Upper-left corner of the Web browser (entire window))
relative	Placement in a relative position
fixed	Placement in an absolute position (reference position: Upper-left corner of the Web browser (current display))

### HTML

```
<div class="parent">
  <p>Element (1)</p>
  <p id="absolute">Element (2)</p>
</div>
```

### Style Sheet

```
.parent {
  padding: 20px;
  position: relative;
  /*Set the parent element to relative.*/}
#absolute {
  position: absolute;
  top: 0;
  left: 0;
}
/*Decorations are omitted.*/
```



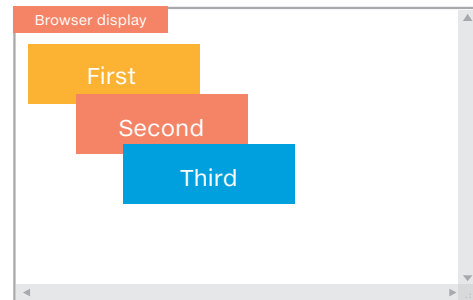
# 7. REFERENCES

Style Sheet	HTML Tags	Color Name/Color Code
List of Properties	Explanation	Explanation for Usage Example

## ► No. 2 [z-index:] ... Specifies the overlapping order of the element.

Value	Description
Integer value	Specifies the overlapping order with integers (zero is used as the reference, and the larger the value, the higher up the object is).
auto	The same layer as the parent object. (Default value)

If you do not specify the z-index, the element will be covered with the elements that are written later in the code.



```

HTML
<div id="example">
  <p class="one">First</p>
  <p class="two">Second</p>
  <p class="three">Third</p>
</div>

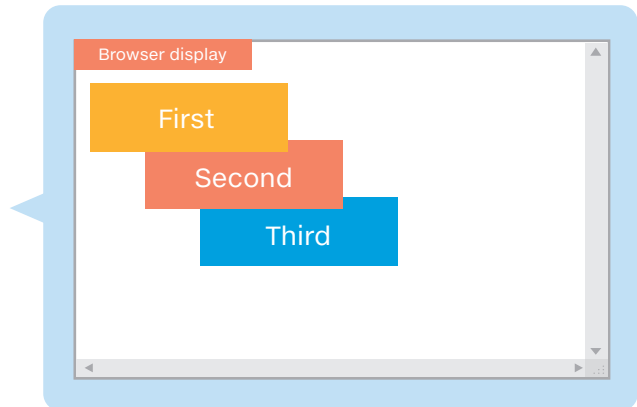
Style Sheet
/*Set the parent element to relative.*/
#example { position: relative; }

/*Set all three elements to absolute.*/
.one, .two, .three {
  position: absolute;
}

.one { z-index: 30; left: 0; top: 0; }
.two { z-index: 20; left: 20px; top: 20px; }
.three { z-index: 10; left: 40px; top: 40px; }

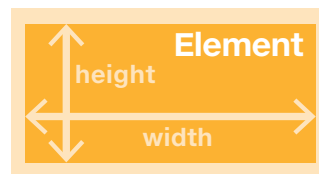
/*Coloring and other such settings are omitted.*/

```



## ► No. 3 [width:] ... Element width

## ► No. 4 [height:] ... Element height

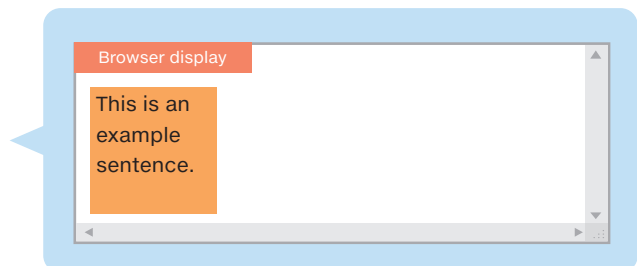


```

HTML
<p>This is an example sentence.</p>

Style Sheet
p { width: 100px;
  height: 100px;
  background: orange;
}

```



# 7. REFERENCES

Style Sheet

HTML Tags

Color Name/Color Code

List of Properties

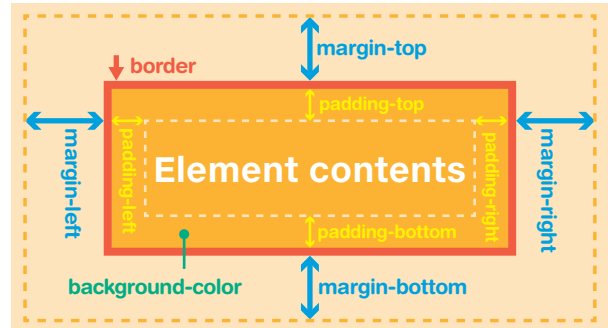
Explanation

Explanation for Usage Example

► No. 5 [margin-top:] ... Outer, upper margin of the element

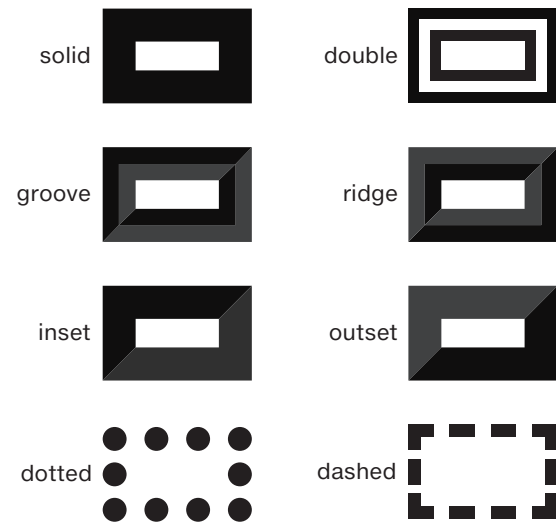
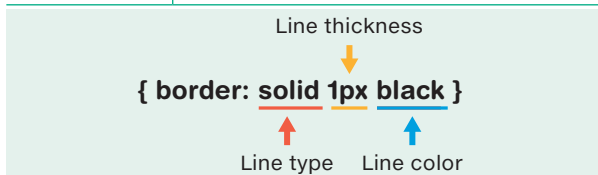
► No. 6 [margin-left:] ... Outer, left margin of the element

► No. 7 [background-color:] ... Element background color



► No. 8 [border:] ... Element border

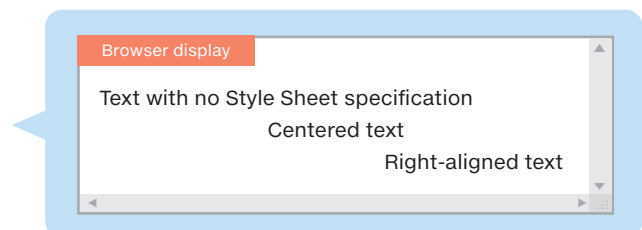
Value	Description
none	No border (default value)
hidden	No border
solid	Solid line (one line)
double	Double line (two lines)
groove	Concave line
ridge	Convex line
inset	Interior concave line (shadow on the upper-left part)
outset	Interior convex line (shadow on the lower-right part)
dotted	Dotted line
dashed	Dashed line



► No. 9 [text-align:] ... Text position specification

Value	Description
left	Left-aligned (default value)
center	Centered
right	Right-aligned

HTML
<code>&lt;p&gt;Text with no Style Sheet specification&lt;/p&gt;</code>
<code>&lt;p class="center"&gt;Centered text&lt;/p&gt;</code>
<code>&lt;p class="right"&gt;Right-aligned text&lt;/p&gt;</code>
Style Sheet
<code>.center { text-align: center; }</code>
<code>.right { text-align: right; }</code>









► No. 10 [font-weight:] ... Character thickness

Value	Description
normal	Standard thickness
bold	General thickness of bold text
lighter	Slightly thinner compared to bold text
bolder	Slightly thicker compared to bold text

# 7. REFERENCES

Style Sheet	HTML Tags	Color Name/Color Code
List of Properties	Explanation	Explanation for Usage Example

## ▶ No. 11 [cursor:] ... Cursor shape

Value	Description	Shape
auto	Default value (selected automatically depending on the situation)	-
default	Default	
pointer	Link	
crosshair	Crosshairs	
context-menu	Context menu	
cell	Cell selection	
help	Help	

XX {cursor: value;}  
For XX, write a selector that indicates on what the cursor needs to be placed in order for its shape to be changed.

The cursor shape varies depending on the OS of the terminal that displays the user Web page.

### Explanation for Usage Example

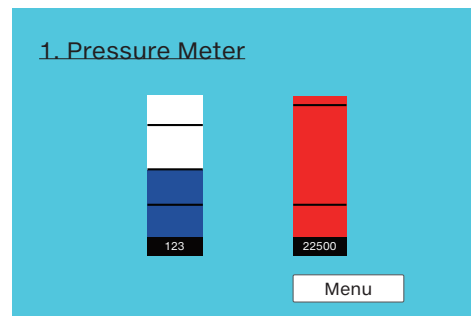
This section explains the operations of the properties of the Style Sheet of the "Menu" button used in "3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE".

No.	Property	Description
A	position:	Absolute/relative coordinate specification
B	left:	Coordinate position from the left side of the Web browser
C	top:	Coordinate position from the top of the Web browser

HTML

```
<a href="index.html" class="menu" style="position: absolute; left: 350px; top: 550px;" />Menu</a>
```

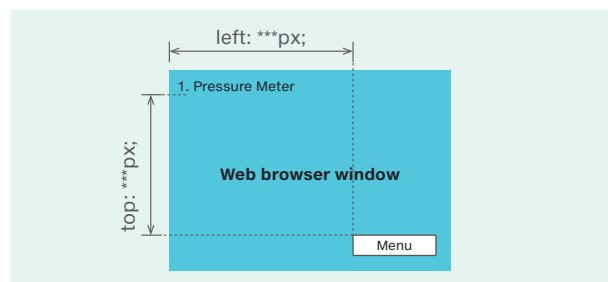
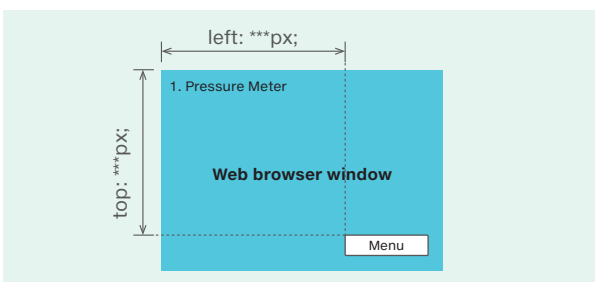
Style Sheet parameter part written in the HTML <a> tag



## ▶ No. A [position:] ... Display coordinate specification method selection

Value	Description	Remarks
static	Do not specify a placement method.	The element is placed at the default value position without applying top, bottom, left, and right.
absolute	Placement in an absolute position	The upper-left corner of the Web browser (entire window) is set as the reference position. The top, bottom, left, and right coordinate values are applied.
relative	Placement in a relative position	The position displayed when static is specified with the position property is set as the reference position. The top, bottom, left, and right coordinate values are applied.
fixed	Placement in an absolute position	The same as "absolute" position. However, the upper-left corner of the currently displayed Web browser is set as the reference position. (The element is displayed in an absolute position so that it stays in the same position even if the user scrolls through the page.)

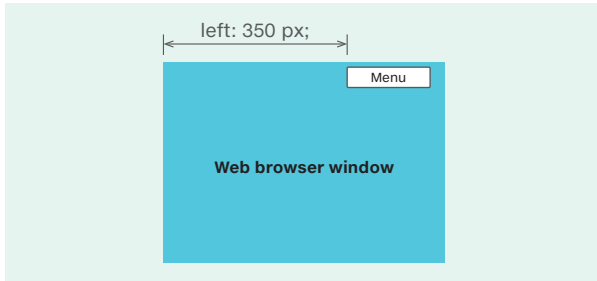
- When set to position: absolute (the setting used in this guide)
- When set to position: relative



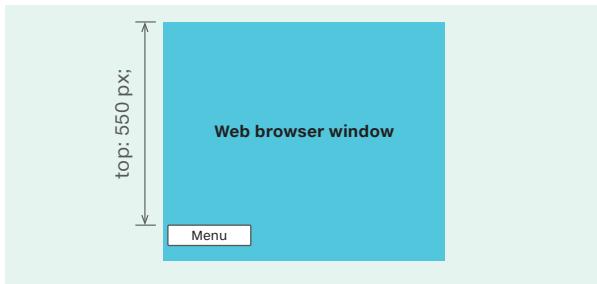
# 7. REFERENCES

Style Sheet	HTML Tags	Color Name/Color Code
List of Properties	Explanation	Explanation for Usage Example

## ▶ No. B [left:] ... Coordinate position from the left side of the Web browser



## ▶ No. C [top:] ... Coordinate position from the top of the Web browser





# 7. REFERENCES

Style Sheet	<b>HTML Tags</b>	Color Name/Color Code			
List of Tags	HTML Definition	Character string modification	Image	Link	Table

## 7.2 HTML Tag References

You can use tags (text enclosed in the less-than (<) and greater-than (>) signs) to specify a variety of items including text and image displays.

Some tags are used in a nested manner and some are used in a stand-alone manner.

### List of Tags

This section provides a simple explanation of the functions of the tags used in this guide. Tags that are not used in the user Web page library file (index.html) but are used in “3. USING CUSTOMIZATION TO EASILY CREATE A USER WEB PAGE” are included.

Value	Tag	Description
HTML definition	<meta>	Declaration
Character string modification	 	Line break
	<p>	Paragraph
	<i>	Italic
	<u>	Underline
Image	<img>	Image display of an image file
Link	<a> to </a>	Link to a different Web page
Table	<table> to </table>	Table construction

### HTML Definition

Function	Start	End	HTML document	Web browser display
Declaration	<meta>	None	<meta charset="UTF-8">	—

Element	charset	Kanji code
	ISO-2022-JP	JIS code
	Shift_JIS	Shift JIS code
	EUC-JP	EUC code

# 7. REFERENCES

Style Sheet

HTML Tags

Color Name/Color Code

List of Tags

HTML Definition

Character string modification

Image

Link

Table

## Character string modification


### ► Text

Tag			HTML document	Web browser display
Function	Start	End		
Line break	 	–	Display line one.   Display line two.	Display line one. Display line two.
Paragraph	<p>	</p>	<p>Display line one. </p> Display line two.	Display line one. Display line two.

### ► Font

Tag			HTML document	Web browser display
Function	Start	End		
Italic	<i>	</i>	<i>Display line one. </i>	<i>Display line one.</i>
Underline	<u>	</u>	<u>Display</u> line two.	<u>Display</u> line two.

## Image

Tag			HTML document	Web browser display
Function	Start	End		
Image display	<img>	–		

Element	Description	Image example
src="image file name"	Sets the storage location and file name of the image file.	
alt="alternative text"	Specifies the text that is displayed (read) in place of the image.	
border="0"	Specifies the border around the image in pixels (0: no border is displayed).	
width="300"	Width specification	
height="10"	Height specification	

sample0.png  
Size: 30 × 30

# 7. REFERENCES



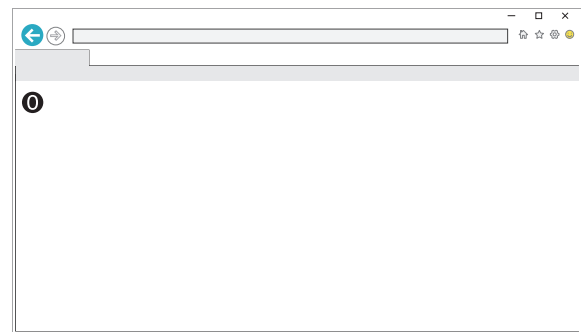
## Reference

This section provides an explanation using an image file (sample0.png) in the user Web page library as an example.

**STEP 1.** Create a new image and save that file (example: sample0.gif) to the personal computer.

**STEP 2.** Check the size of the created image.

Display the image by dragging it to a Web browser.



**STEP 3.** Right-click the displayed image and display its properties.

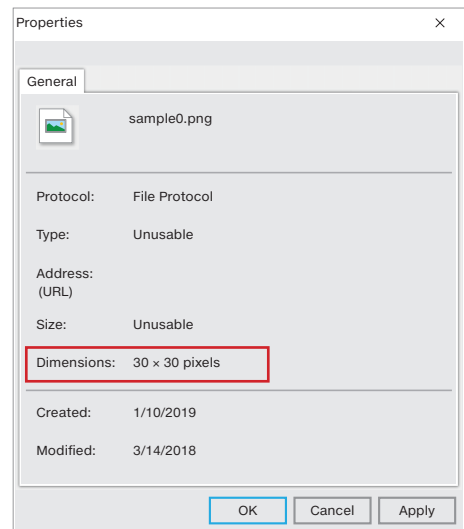
When displaying the image with a 1/1 ratio, enter the size displayed next to Dimensions: (in pixels).

► Setting example

```
width="30"
height="30"
```

For example, if you change 30 pixels to 15 pixels, the displayed image will be reduced to 1/4 of its original size.

You can also change the display size according to a percentage value with the size of the image file used as 100%.



**STEP 4.** Add the HTML as shown below between the <body> and </body> tags of the HTML document.

► Setting example

```
src="/img/sample0.png"
```

► HTML

```

```

# 7. REFERENCES

Style Sheet	<b>HTML Tags</b>	Color Name/Color Code
List of Tags	HTML Definition	Character string modification
		Image
		<b>Link</b>
		Table

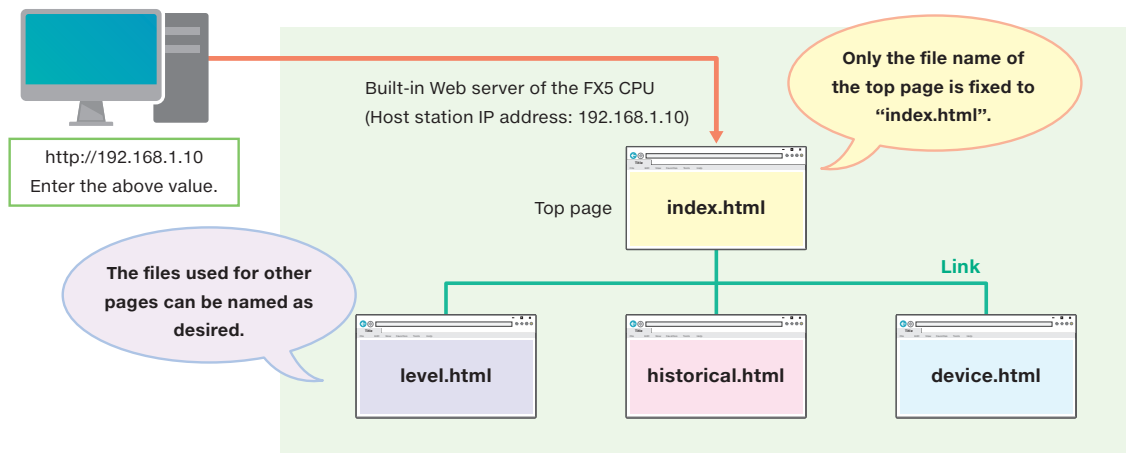
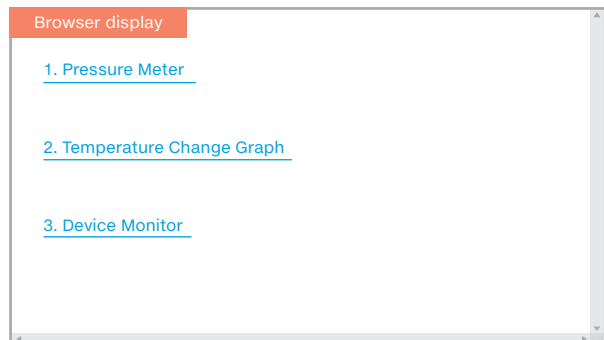
## Link

Function	Tag		HTML document	Web browser display
	Start	End		
Link	<code>&lt;a&gt;</code>	<code>&lt;/a&gt;</code>	<code>&lt;a href="historical.html"&gt;2. Temperature Change Graph&lt;/a&gt;</code>	The link is indicated with blue, underlined text.  <a href="#">2. Temperature Change Graph</a>

```

HTML
<p>
<a href="level.html">1. Pressure Meter</a>
</p>
<p>
<a href="historical.html">2. Temperature Change Graph</a>
</p>
<p>
<a href="device.html">3. Device Monitor</a>
</p>

```



# 7. REFERENCES

Style Sheet

HTML Tags

Color Name/Color Code

List of Tags

HTML Definition

Character string modification

Image

Link

Table

## Table

Function	Tag		HTML document	Web browser display		
	Start	End				
Table	<code>&lt;table&gt;</code>	<code>&lt;/table&gt;</code>	<code>&lt;table&gt;</code> Device name <code>&lt;/table&gt;</code>	Enclose the entire table in the <code>&lt;table&gt;</code> and <code>&lt;/table&gt;</code> tags.  Device name		
Table row	<code>&lt;tr&gt;</code>	<code>&lt;/tr&gt;</code>	<code>&lt;table border="1"&gt;</code> <code>&lt;tr&gt;&lt;td&gt;Device one&lt;/td&gt;&lt;td&gt;Device two&lt;/td&gt;&lt;/tr&gt;</code> <code>&lt;/table&gt;</code>	Create each row with the <code>&lt;tr&gt;</code> and <code>&lt;/tr&gt;</code> tags.  <table border="1"> <tr> <td>Device one</td> <td>Device two</td> </tr> </table>	Device one	Device two
Device one	Device two					
Table heading	<code>&lt;th&gt;</code>	<code>&lt;/th&gt;</code>	<code>&lt;table border="1"&gt;</code> <code>&lt;tr&gt;&lt;th&gt;Device name&lt;/th&gt;&lt;th&gt;Production count&lt;/th&gt;&lt;/tr&gt;</code> <code>&lt;/table&gt;</code>	For each row created with the <code>&lt;th&gt;</code> and <code>&lt;/th&gt;</code> tags, the text is emphasized.  <table border="1"> <tr> <td>Device name</td> <td>Production count</td> </tr> </table>	Device name	Production count
Device name	Production count					
Table contents	<code>&lt;td&gt;</code>	<code>&lt;/td&gt;</code>	<code>&lt;table border="1"&gt;</code> <code>&lt;tr&gt;&lt;td&gt;Device one&lt;/td&gt;&lt;/tr&gt;</code> <code>&lt;tr&gt;&lt;td&gt;Device two&lt;/td&gt;&lt;/tr&gt;</code> <code>&lt;/table&gt;</code>	Create each column with the <code>&lt;td&gt;</code> and <code>&lt;/td&gt;</code> tags.  <table border="1"> <tr> <td>Device one</td> </tr> <tr> <td>Device two</td> </tr> </table>	Device one	Device two
Device one						
Device two						

## List of attributes

Only some of the attributes are listed in this guide. Attributes that are not set are handled as zero.

### ► List of main attributes

Function	Attribute	HTML document	Web browser display							
Border	border	<code>&lt;table border="1"&gt;</code>	<table border="1"> <tr> <td>Device name</td> </tr> </table>	Device name						
Device name										
Cell color	bgcolor	<code>&lt;table bgcolor="green"&gt;</code>	<table> <tr> <td>Device name</td> </tr> </table>	Device name						
Device name										
Column-joining (horizontal direction) specification attribute	colspan	<code>&lt;td colspan="number of cells to join"&gt;</code>	<p>colspan="1"</p> <table> <tr> <td>A</td> <td>B</td> <td>C</td> </tr> <tr> <td>Device name</td> <td></td> <td></td> </tr> </table> <p>colspan="3"</p> <table> <tr> <td>Device name</td> </tr> </table>	A	B	C	Device name			Device name
A	B	C								
Device name										
Device name										
Row-joining (vertical direction) specification attribute	rowspan	<code>&lt;td rowspan="number of cells to join"&gt;</code>	<p>rowspan="1"</p> <table> <tr> <td>1</td> <td>Device name</td> </tr> <tr> <td>2</td> <td></td> </tr> <tr> <td>3</td> <td></td> </tr> </table> <p>rowspan="3"</p> <table> <tr> <td>Device name</td> </tr> </table>	1	Device name	2		3		Device name
1	Device name									
2										
3										
Device name										

# 7. REFERENCES

Style Sheet

HTML Tags

Color Name/Color Code

List of Tags

HTML Definition

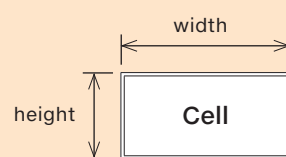
Character string modification

Image

Link

Table

## ► Cell size specification

Function	Attribute	HTML document	Web browser display
Cell width	width	<code>&lt;td width="200" height="100"&gt; Device name&lt;/td&gt;</code>	
Cell height	height		

## ► Text display position specification

	Function	Attribute	HTML document	Web browser display
Horizontal	Text alignment (left-aligned) (td tag default value)	left	<code>&lt;td align="left"&gt;</code>	Device name
	Text alignment (center) (th tag default value)	center	<code>&lt;td align="center"&gt;</code>	Device name
	Text alignment (right-aligned)	right	<code>&lt;td align="right"&gt;</code>	Device name
Vertical	Text alignment (top-aligned)	top	<code>&lt;td valign="top"&gt;</code>	Device name
	Text alignment (center) (Default value)	middle	<code>&lt;td valign="middle"&gt;</code>	Device name
	Text alignment (bottom-aligned)	bottom	<code>&lt;td valign="bottom"&gt;</code>	Device name

## ► Table creation example

	A	B	C	D	
Line one	A-1		C-1	D-1	Centered text
Line two	A-2	B-2	C-2	D-2	Left-aligned text (default value)
Line three	A-3	B-3	C-3		Left-aligned text (default value)

## ► HTML

	<code>&lt;table border="1"&gt;</code>
	<code>&lt;tr bgcolor="green"&gt;&lt;th width="100"&gt;A&lt;/th&gt;&lt;th width="100"&gt;B&lt;/th&gt;&lt;th width="100"&gt;C&lt;/th&gt;&lt;th width="100"&gt;D&lt;/th&gt;&lt;/tr&gt;</code>
Line one	<code>&lt;tr align="center" bgcolor="white"&gt;&lt;td colspan="2"&gt;A-1&lt;/td&gt;&lt;td&gt;C-1&lt;/td&gt;&lt;td&gt;D-1&lt;/td&gt;&lt;/tr&gt;</code>
Line two	<code>&lt;tr bgcolor="white"&gt;&lt;td&gt;A-2&lt;/td&gt;&lt;td&gt;B-2&lt;/td&gt;&lt;td&gt;C-2&lt;/td&gt;&lt;td rowspan="2"&gt;D-2&lt;/td&gt;&lt;/tr&gt;</code>
Line three	<code>&lt;tr bgcolor="white"&gt;&lt;td&gt;A-3&lt;/td&gt;&lt;td&gt;B-3&lt;/td&gt;&lt;td&gt;C-3&lt;/td&gt;&lt;/tr&gt;</code>
	<code>&lt;/table&gt;</code>

# 7. REFERENCES

Style Sheet

HTML Tags

Color Name/Color Code

16 colors (basic colors)

Web-safe colors, 216 colors

Web color table, 147 colors

## 7.3 Color Name/Color Code

### 16 colors (basic colors)

These are the (16) basic colors defined by HTML.

Color	HTML / CSS	Hex Code
	Color Name	#RRGGBB
	black	#000000
	silver	#c0c0c0
	maroon	#800000
	purple	#800080
	green	#008000
	olive	#808000
	navy	#000080
	teal	#008080
	gray	#808080
	white	#ffffff
	red	#ff0000
	fuchsia	#ff00ff
	lime	#00ff00
	yellow	#ffff00
	blue	#0000ff
	aqua	#00ffff

# 7. REFERENCES

Style Sheet

HTML Tags

Color Name/Color Code

16 colors (basic colors)

Web-safe colors, 216 colors

Web color table, 147 colors

## Web-safe colors, 216 colors

Web-safe colors is an assembly of 216 colors defined with red, green, and blue each split into 6 levels (6 × 6 × 6). The advantage of using a color scheme with web-safe colors is that while the range is limited to 216 colors, nearly the same colors can be reproduced without being greatly influenced by the performance of the computer.

Color	Hex Code #RRGGBB	Color	Hex Code #RRGGBB	Color	Hex Code #RRGGBB	Color	Hex Code #RRGGBB	Color	Hex Code #RRGGBB	Color	Hex Code #RRGGBB
	#000000		#003300		#006600		#009900		#00cc00		#00ff00
	#000033		#003333		#006633		#009933		#00cc33		#00ff33
	#000066		#003366		#006666		#009966		#00cc66		#00ff66
	#000099		#003399		#006699		#009999		#00cc99		#00ff99
	#0000cc		#0033cc		#0066cc		#0099cc		#00cccc		#00ffcc
	#0000ff		#0033ff		#0066ff		#0099ff		#00ccff		#00ffff
	#330000		#333300		#336600		#339900		#33cc00		#33ff00
	#330033		#333333		#336633		#339933		#33cc33		#33ff33
	#330066		#333366		#336666		#339966		#33cc66		#33ff66
	#330099		#333399		#336699		#339999		#33cc99		#33ff99
	#3300cc		#3333cc		#3366cc		#3399cc		#33cccc		#33ffcc
	#3300ff		#3333ff		#3366ff		#3399ff		#33ccff		#33ffff
	#660000		#663300		#666600		#669900		#66cc00		#66ff00
	#660033		#663333		#666633		#669933		#66cc33		#66ff33
	#660066		#663366		#666666		#669966		#66cc66		#66ff66
	#660099		#663399		#666699		#669999		#66cc99		#66ff99
	#6600cc		#6633cc		#6666cc		#6699cc		#66cccc		#66ffcc
	#6600ff		#6633ff		#6666ff		#6699ff		#66ccff		#66ffff
	#990000		#993300		#996600		#999900		#99cc00		#99ff00
	#990033		#993333		#996633		#999933		#99cc33		#99ff33
	#990066		#993366		#996666		#999966		#99cc66		#99ff66
	#990099		#993399		#996699		#999999		#99cc99		#99ff99
	#9900cc		#9933cc		#9966cc		#9999cc		#99cccc		#99ffcc
	#9900ff		#9933ff		#9966ff		#9999ff		#99ccff		#99ffff
	#cc0000		#cc3300		#cc6600		#cc9900		#cccc00		#ccff00
	#cc0033		#cc3333		#cc6633		#cc9933		#cccc33		#ccff33
	#cc0066		#cc3366		#cc6666		#cc9966		#cccc66		#ccff66
	#cc0099		#cc3399		#cc6699		#cc9999		#cccc99		#ccff99
	#cc00cc		#cc33cc		#cc66cc		#cc99cc		#ccccc		#ccffcc
	#cc00ff		#cc33ff		#cc66ff		#cc99ff		#ccccff		#ccffff
	#ff0000		#ff3300		#ff6600		#ff9900		#ffcc00		#ffff00
	#ff0033		#ff3333		#ff6633		#ff9933		#ffcc33		#ffff33
	#ff0066		#ff3366		#ff6666		#ff9966		#ffcc66		#ffff66
	#ff0099		#ff3399		#ff6699		#ff9999		#ffcc99		#ffff99
	#ff00cc		#ff33cc		#ff66cc		#ff99cc		#ffcccc		#ffffcc
	#ff00ff		#ff33ff		#ff66ff		#ff99ff		#ffccff		#ffffff



# 7. REFERENCES

Style Sheet

HTML Tags

Color Name/Color Code

16 colors (basic colors)

Web-safe colors, 216 colors

Web color table, 147 colors

## Web color table, 147 colors

There are 147 colors that can be specified by color names.

Color	HTML / CSS	Hex Code
	Color Name	#RRGGBB
lightsalmon		#FFA07A
salmon		#FA8072
darksalmon		#E9967A
lightcoral		#F08080
indianred		#CD5C5C
crimson		#DC143C
firebrick		#B22222
red		#FF0000
darkred		#8B0000
coral		#FF7F50
tomato		#FF6347
orangered		#FF4500
gold		#FFD700
orange		#FFA500
darkorange		#FF8C00
lightyellow		#FFFFE0
lemonchiffon		#FFFACD
lightgoldenrodyellow		#FAFAD2
papayawhip		#FFEFD5
moccasin		#FFE4B5
peachpuff		#FFDAB9
palegoldenrod		#EEE8AA
khaki		#F0E68C
darkkhaki		#BDB76B
yellow		#FFFF00
lawngreen		#7CFC00
chartreuse		#7FFF00
limegreen		#32CD32
lime		#00FF00
forestgreen		#228B22
green		#008000
darkgreen		#006400
greenyellow		#ADFF2F
yellowgreen		#9ACD32
springgreen		#00FF7F
mediumspringgreen		#00FA9A
lightgreen		#90EE90
palegreen		#98FB98
darkseagreen		#8FBC8F
mediumseagreen		#3CB371
seagreen		#2E8B57

Color	HTML / CSS	Hex Code
	Color Name	#RRGGBB
olive		#808000
darkolivegreen		#556B2F
olivedrab		#6B8E23
lightcyan		#E0FFFF
cyan		#00FFFF
aqua		#00FFFF
aquamarine		#7FFFD4
mediumaquamarine		#66CDAA
paleturquoise		#AFEEEE
turquoise		#40E0D0
mediumturquoise		#48D1CC
darkturquoise		#00CED1
lightseagreen		#20B2AA
cadetblue		#5F9EA0
darkcyan		#008B8B
teal		#008080
powderblue		#B0E0E6
lightblue		#ADD8E6
lightskyblue		#87CEFA
skyblue		#87CEEB
deepskyblue		#00BFFF
lightsteelblue		#B0C4DE
dodgerblue		#1E90FF
cornflowerblue		#6495ED
steelblue		#4682B4
royalblue		#4169E1
blue		#0000FF
mediumblue		#0000CD
darkblue		#00008B
navy		#000080
midnightblue		#191970
mediumslateblue		#7B68EE
slateblue		#6A5ACD
darkslateblue		#483D8B
lavender		#E6E6FA
thistle		#D8BFD8
plum		#DDA0DD
violet		#EE82EE
orchid		#DA70D6
fuchsia		#FF00FF
magenta		#FF00FF

# 7. REFERENCES

Style Sheet



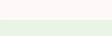

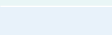



HTML Tags

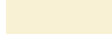















Color Name/Color Code

16 colors (basic colors)

Web-safe colors, 216 colors

Web color table, 147 colors

Color	HTML / CSS	Hex Code
	Color Name	#RRGGBB
	mediumorchid	#ba55d3
	mediumpurple	#9370db
	blueviolet	#8a2be2
	darkviolet	#9400d3
	darkorchid	#9932cc
	darkmagenta	#8b008b
	purple	#800080
	indigo	#4b0082
	pink	#ffc0cb
	lightpink	#ffb6c1
	hotpink	#ff69b4
	deeppink	#ff1493
	palevioletred	#db7093
	mediumvioletred	#c71585
	white	#ffffff
	snow	#ffaafa
	honeydew	#f0fff0
	mintcream	#f5fffa
	azure	#f0ffff
	aliceblue	#f0f8ff
	ghostwhite	#f8f8ff
	whitesmoke	#f5f5f5
	seashell	#fff5ee
	beige	#f5f5dc
	oldlace	#fdf5e6
	floralwhite	#fffaf0
	ivory	#ffff00
	antiquewhite	#faebd7
	linen	#faf0e6
	lavenderblush	#fff0f5
	mistyrose	#ffe4e1
	gainsboro	#dcdcdc
	lightgray	#d3d3d3
	silver	#c0c0c0
	darkgray	#a9a9a9
	gray	#808080
	dimgray	#696969
	lightslategray	#778899
	slategray	#708090
	darkslategray	#2f4f4f
	black	#000000

Color	HTML / CSS	Hex Code
	Color Name	#RRGGBB
	cornsilk	#fff8dc
	blanchedalmond	#ffebcd
	bisque	#ffe4c4
	navajowhite	#ffdead
	wheat	#f5deb3
	burlywood	#deb887
	tan	#d2b48c
	rosybrown	#bc8f8f
	sandybrown	#f4a460
	goldenrod	#daa520
	peru	#cd853f
	chocolate	#d2691e
	saddlebrown	#8b4513
	sienna	#a0522d
	brown	#a52a2a
	maroon	#800000

# WARRANTY

---

Please confirm the following product warranty details before using this product.

- “WARRANTY” in MELSEC iQ-F FX5U User’s Manual (Hardware) Manual number: JY997D55301
- “WARRANTY” in MELSEC iQ-F FX5UC User’s Manual (Hardware) Manual number: JY997D61401

# SAFETY GUIDELINES

---

- To ensure proper use of the product described in this guide, be sure to read the manuals of the product before use.
- This product has been manufactured as a general-purpose part for general industries, and has not been designed or manufactured to be incorporated in a device or system used for purposes related to human life.
- Before using the product for special purposes such as nuclear power, electric power, aerospace, medicine, or passenger movement vehicles, consult Mitsubishi Electric representative.
- This product has been manufactured under strict quality control. However, when installing the product where major accidents or losses could occur if the product fails, install appropriate backup or failsafe functions in the system
- For the precautions concerning design, wiring, and others, read SAFETY PRECAUTIONS provided in the relevant manuals.

# TRADEMARKS

---

Microsoft, Windows, and Internet Explorer are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Google Chrome is either a registered trademark or trademark of Google Inc.

The SD and SDHC logos are either registered trademarks or trademarks of SD-3C, LLC.

IOS (iOS) is either a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the United States and/or other countries, and iOS is used under license by Apple Inc.

Ethernet is a registered trademark of Fuji Xerox Co., Ltd. in Japan.

Oracle and JavaScript are registered trademarks of Oracle and/or its affiliates in the United States and/or other countries.

The company names, system names and product names mentioned in this guide are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as ‘™’ or ‘®’ are not specified in this guide.

# REVISIONS

---

\* The document number is given in the bottom-left corner of the back cover of this guide.

Revision date	Document number	Description
September 2019	L(NA)08645ENG-A	First edition

This guide confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights that may occur as a result of using the contents noted in this guide.

# Programmable Controller

## MELSEC iQ-F Series

**mitsubishi** **ELECTRIC CORPORATION**

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN  
<http://Global.MitsubishiElectric.com>

---