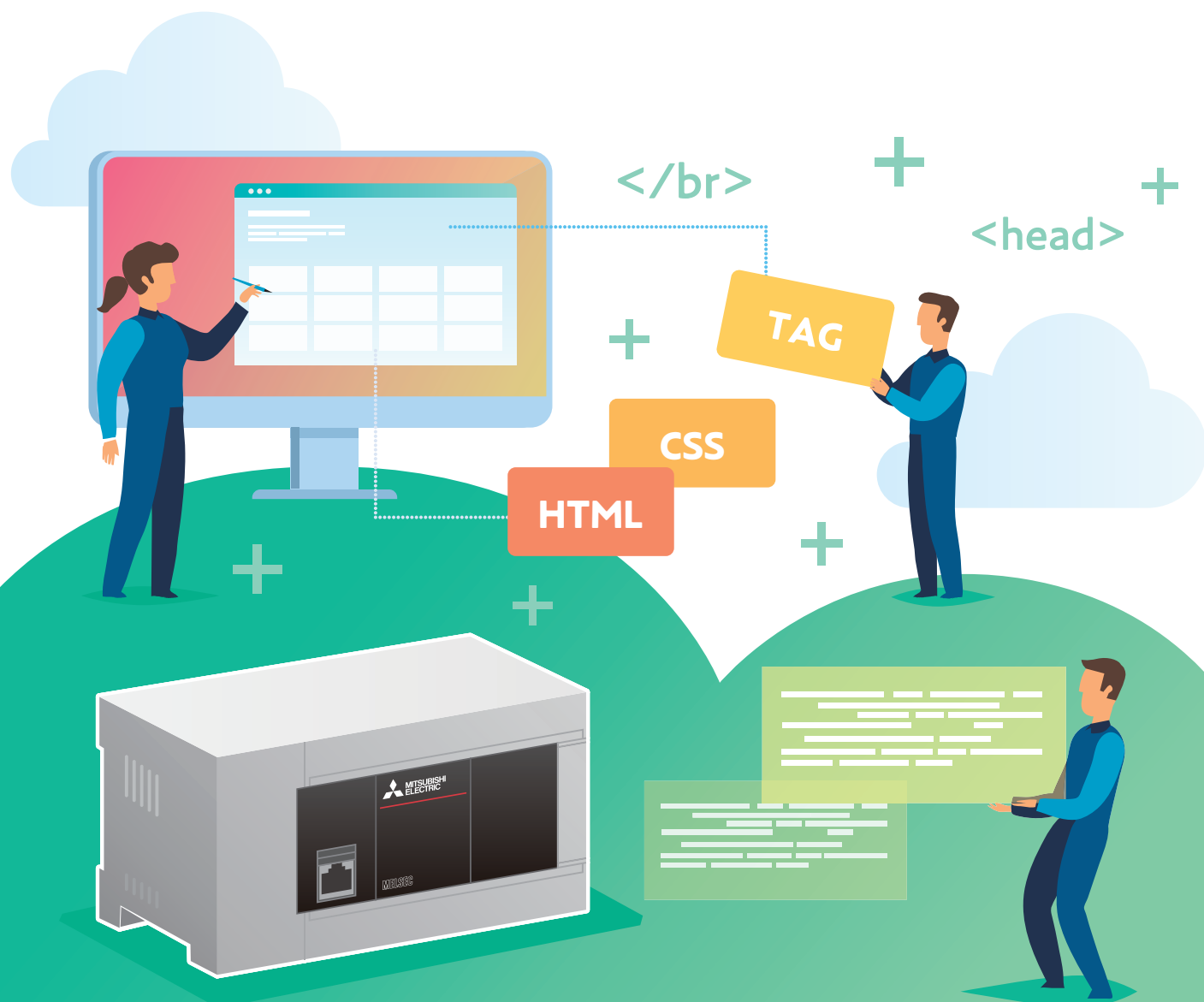


FACTORY AUTOMATION

三菱電機マイクロシーケンサ
MELSEC iQ-Fシリーズ

MELSEC iQ-F
series

Webサーバ機能 アプリケーションガイド Webページを活用してみよう ユーザWebページ HTML作成編



はじめに

このたびは、MELSEC iQ-Fシリーズをお買い上げいただき誠にありがとうございます。

本書は、FX5 CPUユニットのWebサーバ機能に関する設定について述べたものです。

ご使用前に、本書および関連製品のマニュアルをお読みいただき、その仕様を十分ご理解のうえ正しくご使用いただきますようお願いいたします。

ご使用に際してのお願い

- ・この製品は一般工業を対象とした汎用品として製作されたもので、人命にかかわるような状況下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。
- ・この製品を原子力用、電力用、航空宇宙用、医療用、乗用移動体用の機器あるいはシステムなどの特殊用途への適用をご検討の際には、当社の営業窓口まで照会してください。
- ・この製品は厳重な品質体制の下に製造しておりますが、この製品の故障により重大な故障または損失の発生が予測される設備への適用に際しては、バックアップやフェールセーフ機能を系統的に設置してください。

おことわり

- ・製品を設置する際にご不明な点がある場合、電気の知識（電気工事士あるいは同等以上の知識）を有する専門の電気技師に相談してください。この製品の操作や使い方についてご不明な点がある場合は、巻末記載の技術相談窓口へ相談してください。
- ・本書、技術資料、カタログなどに記載されている事例は参考用のため、動作を保証するものではありません。ご採用に際しては機器・装置の機能や安全性をお客様自身でご確認のうえ、ご使用ください。
- ・本書の内容に関しては、改良のため予告なしに仕様などを変更することがありますので、あらかじめご了承ください。
- ・本書の内容に関しては万全を期しておりますが、万一ご不明な点や誤りなどお気づきのことがありましたら、お手数ですが巻末記載の当社支社または支店までご連絡ください。その際、巻末記載のドキュメント番号もあわせてお知らせください。

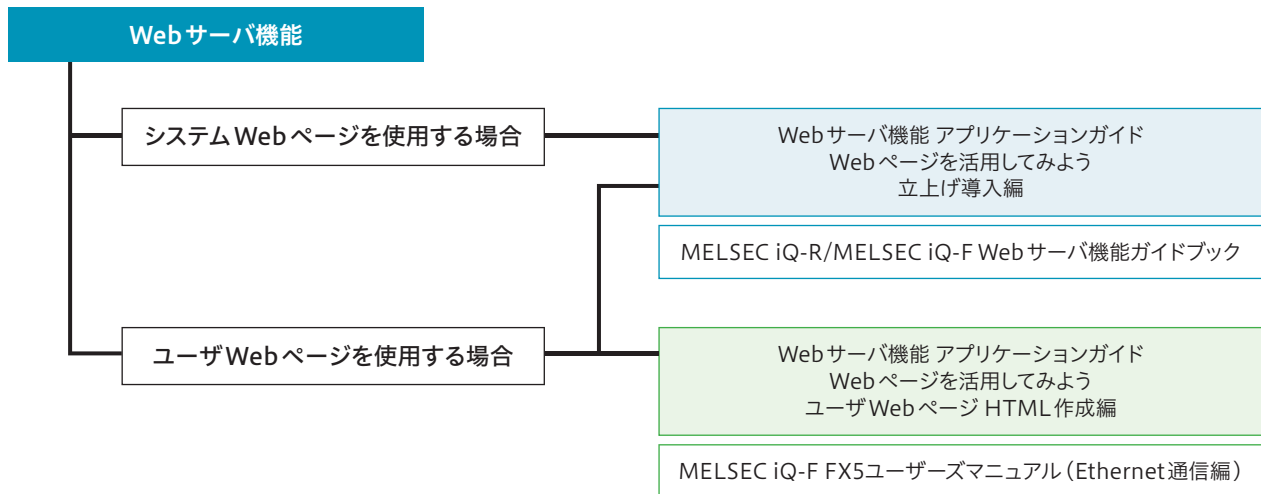
目次

はじめに	2
関連カタログ / マニュアル	4
注意事項	5
制限事項	5
1. HTML<タグ>作成の基礎	6
1.1 HTMLとは	6
1.2 HTML 構造	7
1.3 タグの基本ルール	9
1.4 タグの種類	10
1.5 ファイル名の付け方	12
2. パソコンで簡単なHTML文書を作成し表示してみよう	13
2.1 用意するもの	13
2.2 HTML ファイルの作成	13
2.3 Web ブラウザ表示	15
2.4 Web ブラウザでHTMLソースの表示方法	16
3. カスタマイズで簡単にユーザWeb ページを作成しよう	17
3.1 はじめる前に	17
3.2 用意するもの	19
3.3 流用方法	21
3.4 ユーザWeb ページ作成	26
3.5 ファイル保存先	59
3.6 トラブルシューティング	62
4. Style Sheet (CSS) の働きを知る	64
4.1 機能紹介	64
4.2 共通事項	65
4.3 全体構造 (UserWebStyle.css ファイル)	67
4.4 設定内容	67
4.5 Style Sheet の内容	68
5. JavaScript (JS) で表示できる部品の働きを知る	75
5.1 JavaScript (JS) とは	75
5.2 共通事項	77
5.3 表	81
5.4 グラフ	87
5.5 画像・図形	98
5.6 操作ボタン	107
6. CGIによるデバイスモニタ画面を作成しよう	113
6.1 CGIとは	113
6.2 CGIの概要と機能	113
6.3 共通仕様	116
6.4 デバイス読出しWeb画面の作成	118
6.5 デバイス書き込みWeb画面の作成	125
7. リファレンス	132
7.1 Style Sheet リファレンス	132
7.2 HTML タグ リファレンス	137
7.3 カラーネーム・カラーコード	143

関連カタログ/マニュアル

▶ カタログとマニュアル構成

使用する Web ページによって参照するカタログやマニュアルは、それぞれ下記のとおりとなります。



▶ 参考カタログ/マニュアル

製品の詳細仕様や注意事項、制限事項など、下記のカタログやマニュアルをご覧ください。

最新の e-Manual および PDF は、三菱電機 FA サイトからダウンロードできます。

www.MitsubishiElectric.co.jp/fa

名称 [カタログ/マニュアル番号]	説明	提供形態
Web サーバ機能 アプリケーションガイド Web ページを活用してみよう 立上げ導入編 [L (名)08642]	MELSEC iQ-F の Web サーバ機能を使ううえで必要な機材の用意から、Web ページを表示するまでの導入手順を説明しています。	PDF
Web サーバ機能 アプリケーションガイド Web ページを活用してみよう ユーザ Web ページ HTML 作成編 [L (名)08644] (本書)	ユーザ Web ページで独自の Web ページを作成する場合にご覧ください。 HTML 作成における Style Sheet によるデザインや JavaScript 部品、CGI 部品の詳細を説明しています。	PDF
MELSEC iQ-R/MELSEC iQ-F Web サーバ機能ガイドブック [SH-081981]	Web サーバ機能の仕様、運転までの手順、トラブルシューティングについて記載しています。	e-Manual, PDF
MELSEC iQ-F FX5 ユーザーズマニュアル (Ethernet 通信編) [JY997D55101]	Ethernet 通信の機能に関する内容を記載しています。 Web サーバ機能については、ユーザ Web ページで利用できる CGI 部品の詳細について記載しています。	e-Manual, PDF, 製本物

Point

e-Manual とは、専用のツールを使用して閲覧できる三菱電機 FA 電子書籍マニュアルです。
e-Manual には下記のような特長があります。

- ・ 探したい情報を複数のマニュアルから一度に検索 (マニュアル横断検索)
- ・ 頻繁に参照する情報をお気に入りに登録可能

注意事項

外部機器からの不正アクセスについて

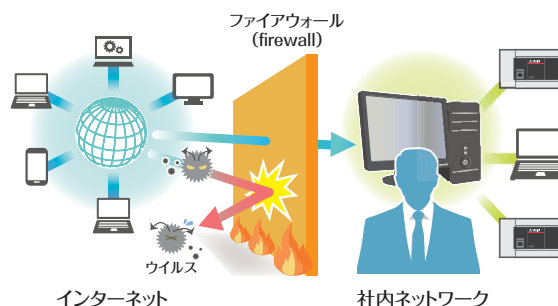
外部機器からの不正アクセスに対して、シーケンサシステムの安全を保つ必要があるときは、お客様による対策も盛り込んでください。不正アクセスにより発生するシステムトラブル上の諸問題に対して、当社は一切その責任を負うことができません。Webサーバ機能におけるユーザ認証は、外部機器からの不正アクセス（プログラムやデータの破壊など）を防止するための1つの手段であり、不正アクセスを完全に防止するものではありません。

不正アクセスの対策例を示します。

- ・ファイアウォールを設置する
- ・中継局としてパソコンを設置し、アプリケーションプログラムで送受信データの中継を制御する
- ・アクセス権を制御できる外部機器を中継局として設置する（アクセス権を制御できる外部機器については、ネットワーク接続業者、または機器販売業者にお問い合わせください）

ファイアウォール機能とは

- ・**フィルタリング機能**
不正なパケットを遮断して、許可されたパケットだけを通過させます。
- ・**アドレス変換機能**
外部のネットワークと内部のネットワークにおいて、相互にIPアドレスを割り当てる機能です。
- ・**遠隔操作、監視機能**
別のコンピュータからファイアウォールの設定を行ったり、ログの確認ができる機能です。



制限事項

Webページの応答性能について

Webページの応答性能は、下記の要因により変動します。

また、ネットワークの回線状態やFX5 CPUのEthernet機能の通信状況によっては、応答性能が低下します。

- ・Ethernet回線の負荷率（回線の混み具合）
- ・同時に使用するコネクション数（他のコネクションの更新処理）
- ・FX5 CPUのEthernet機能に設定したWebサーバ機能以外の通信機能も含めた通信負荷状態
- ・Webページ更新要求時にFX5 CPUのスキャンタイムが大きい状態

Webサーバへのアクセスについて

MELSEC iQ-Fシリーズに対して、同時にログインできるユーザ数は最大4ユーザまでです。

Webサーバ機能使用中に、ファイル読み込みが正常に完了しなかった場合や、ページのリロードを要求するメッセージが表示される場合は、再度読み込みしてください。

通信タイムアウト時間について

通信タイムアウト時間はブラウザの仕様に依存するため、必ずしもWebサーバのタイムアウト時間（5分）と同じではありません。また、エラー画面もブラウザによってデザインが異なります。

特記事項

注意事項や制限事項は、“FX5ユーザズマニュアル（Ethernet通信編）[JY997D55101]”も合わせてご覧ください。

1. HTML<タグ>作成の基礎

HTMLとは

HTML 構造

タグの基本ルール

タグの種類

ファイル名の付け方

1.1 HTMLとは

HTMLって何？

HTML (エイチティーエムエル) とは、Hyper Text Markup Language の略で、Web ページを作成するために開発された言語です。現在、インターネット上で公開されてる Web ページの多くは HTML で作成されており、Web ブラウザ (Internet Explorer® や Google Chrome® など) を使用して閲覧します。Web ブラウザは、読み込んだ HTML ファイルや画像ファイルなどを閲覧しやすい形に整え、Web ページとして表示します。

それでは実際に三菱電機 FA サイトの HTML ファイルのソースを Internet Explorer® で表示してみましょう。

ソース表示の方法 (Internet Explorer® による操作例です。)

Internet Explorer® 上で右クリック→[ソースの表示]を選択
ツールバーの[表示]→[ソース]を選択
[F12] キーを押す

左記の方法で、Web ブラウザの下部にソースを表示します。

三菱電機 FA サイト 画面は、Internet Explorer® による表示例です。



HTML ファイルの
ソース表示

HTML のルールは単純な構造と文字列で構成されており、容易に作成できます。

1. HTML<タグ>作成の基礎

HTMLとは

HTML 構造

タグの基本ルール

タグの種類

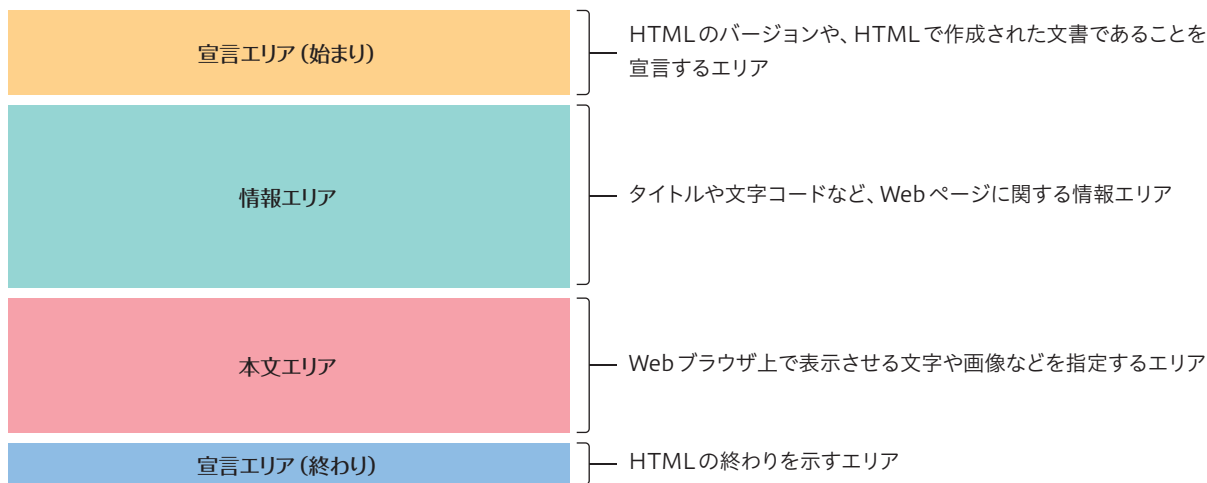
ファイル名の付け方

1.2 HTML構造

Web ページを作成する場合、Web ページを記述するための言語を使用しますが、HTML 以外にもXHM やSGML、XHTML など、さまざまな言語があります。

HTML 構造と各エリアの意味

HTML の基本構造と各エリアの意味は下記となります。



HTML の例

HTML ファイルに記述する最小限のタグの構成です。下記はHTMLの構成ができているため、Web ブラウザに表示できます。

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="UTF-8">
    <title>Web サーバページ</title>
  </head>
  <body>
    画像や文章の内容
  </body>
</html>
```

宣言エリア (始まり)

情報エリア

本文エリア

宣言エリア (終わり)

1

HTMLヘタグ作成の基礎

2

パソコンで簡単なHTML文書を作成し表示してみよう

3

カスタマイズで簡単にユーザーWebページを作成しよう

4

Style Sheet (CSS) の働きを知る

5

JavaScript (JS) で表示できる部品の働きを知る

6

CGIによるデバイスモニタ画面を作成しよう

7

リファレンス

1. HTML<タグ>作成の基礎

HTMLとは

HTML構造

タグの基本ルール

タグの種類

ファイル名の付け方

HTMLの要素

HTMLを構成する要素は多数ありますが、その中でも基本的な要素について紹介します。

<code><!DOCTYPE html></code>	DOCTYPE宣言 HTML文書の先頭に記述し、HTMLのバージョンを宣言するタグ。 <code><!DOCTYPE html></code> は、「このHTMLは、HTML5で記述する」という意味。
<code><html></code> :開始タグ <code></html></code> :終了タグ	html要素 この文書の言語がHTMLであることを示すタグ。 すべての要素は、 <code><html></code> と <code></html></code> で囲われた部分に記述する。
<code><head></code> :開始タグ <code></head></code> :終了タグ	head要素 メタ情報やタイトル情報など、文書全体の情報を <code><head></code> と <code></head></code> で囲われた部分に記述する。
<code><meta></code>	meta要素 メタ情報(メタデータ)とは、「情報についての情報」という意味で、このHTML文書に関する情報を示すタグ。 <code><meta charset="UTF-8"></code> と記載した場合、「このHTML文書は、UTF-8の文字コードで書いている」ことを表している。
<code><title></code> :開始タグ <code></title></code> :終了タグ	title要素 Webブラウザのタイトルバーに表示するタイトル名を指定するタグ。 【使用例】 <code><title>三菱電機FAサイト</title></code>
<code><body></code> :開始タグ <code></body></code> :終了タグ	body要素 Webページで表示させる文字や画像、表などを指定するタグ。 <code><body></code> ～ <code></body></code> で囲われた範囲内だけが、Webページ上に表示される。

タグのアルファベット入力について

HTMLのタグに使用する文字は、大文字・小文字を区別しないため、`<html>`と`<HTML>`は同じ意味になります。本書ではタグを小文字で記載します。

 `<body>` `<BODY>`

1. HTML<タグ>作成の基礎

HTMLとは

HTML 構造

タグの基本ルール

タグの種類

ファイル名の付け方

1.3 タグの基本ルール

HTML では、すべての情報を“タグ”で記載します。段落や改行、画像など、それぞれに指定するタグがあります。本節では基本的なタグの使用方法について説明します。

タグって何？

<html>や<head>など、HTMLを作成する際に使用する目印（マーク）のことです。タグの使用方法はHTMLの規格（ルール）によって決まっており、HTMLのルールで記述した文書を「HTML文書」と呼びます。また、HTMLには「HTML1.0」や「HTML4.01」などのバージョンがあり、バージョンによって記載方法や使用できるタグの仕様に違いがあります。

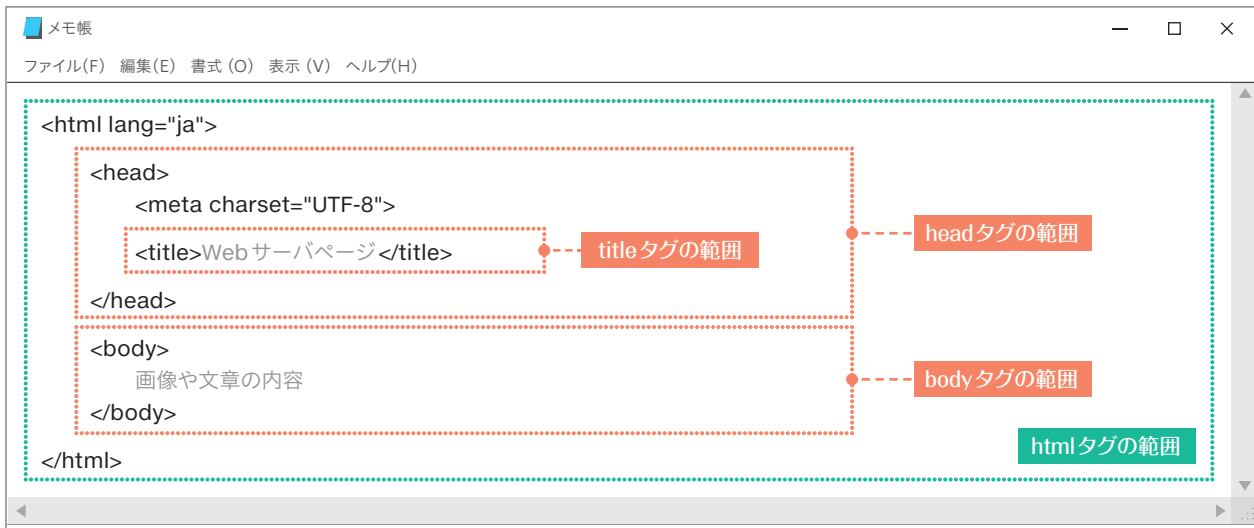
タグの構造

タグは、下記の2種類に大別します。

分類	フォーマット	内容
入れ子使用のタグ	<タグ名>～</タグ名>	内容をタグで入れ子にするようにして記載します。<タグ名>を開始タグ、</タグ名>を終了タグと呼びます。
単独使用のタグ	<タグ名>	タグは単独で使用するものもあります。（<meta>タグ、タグなど）

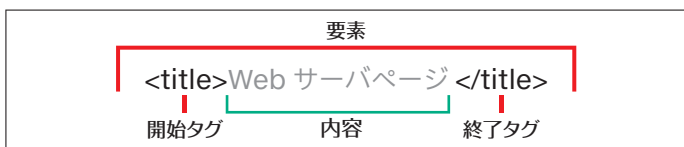
また、タグの内容の中にタグを記載して、階層構造にすることもできます。基本的にHTMLはタグの階層構造にして記述しますが、階層を深くすると読みづらくなるので注意が必要です。

■ タグの入れ子構造



タグと要素

開始タグから終了タグまでの全体を指して“要素”と呼びます。



Point

開始タグの後ろ、終了タグの前の半角スペースは無視します。また、内容の途中の2つ以上の半角スペースは、半角スペース1つとして扱われます。

1. HTML<タグ>作成の基礎

HTMLとは

HTML 構造

タグの基本ルール

タグの種類

ファイル名の付け方

1.4 タグの種類

HTMLを使用するうえで、よく使用するタグについて紹介します。

ページの構造を記述するタグ

<header> ヘッダー	Web ページの上部に表示する内容を記載します。
<footer> フッタ	Web ページの最下部に表示する内容を記載します。

Web ページ

header

←ページのタイトル など

footer

←著作権表示 など

文章を記述するタグ

<h1~h6> 見出し	文章の見出しは、<h1~6>タグで記載します。タグの数字を大きくするほど、見出しの文字サイズは小さくなります。 HTML 文書 <pre><h1>h1の見出し</h1> <h2>h2の見出し</h2> <h3>h3の見出し</h3> <h4>h4の見出し</h4> <h5>h5の見出し</h5> <h6>h6の見出し</h6></pre> Web ブラウザ表示 <pre>h1の見出し h2の見出し h3の見出し h4の見出し h5の見出し h6の見出し</pre>
<p> 段落	文章の段落は、<p>~</p>で囲んで記載します。 HTML 文書 <pre><p>段落 1</p> <p>段落 2</p></pre> Web ブラウザ表示 <pre>段落 1 段落 2</pre>

 改行	長い文章は表示領域の右端で自動的に改行しますが、意図的に改行したい場合に使用します。 HTML 文書 <pre>改行位置は
タグで任意に指定できます。</pre> Web ブラウザ表示 <pre>改行位置は タグで任意に指定できます。</pre>

Point

HTML では、HTML ファイル内での改行は無視します。分かりやすい表示のために、<p>タグや
タグを効果的に使用しましょう。

1. HTML<タグ>作成の基礎

HTMLとは

HTML 構造

タグの基本ルール

タグの種類

ファイル名の付け方

<table> 表

表は<table>タグを使用して作成します。下記の例のように、表の中の要素は<tr>タグと<td>タグを使用して記述します。タグの詳細は7.2 HTMLタグ リファレンス-テーブルを参照してください。

HTML 文書

```
<table>
<tr>
<td>りんご</td><td>Apple</td>
</tr>
<td>みかん</td><td>Orange</td>
</tr>
</table>
```

Web ブラウザ表示

<td>	</td>	<td>	</td>
<tr>	りんご	Apple	</tr>
<tr>	みかん	Orange	</tr>

その他のタグ

 画像を表示

画像はimgタグで画像ファイルを指定して表示します。画像ファイルの形式は、PNG、JPEG、GIFをよく使用します。(PNGの使用が推奨されています。)

書式:

HTML 文書

```

```



company-logo.png

Web ブラウザ表示

Company Name Logo

<a> リンクにジャンプする

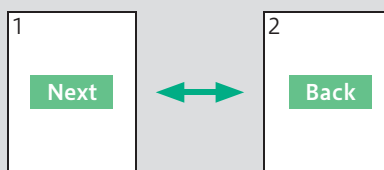
文章の一部や画像をクリックすることで、別のページにジャンプできます。

書式:ここをクリックでリンク先にジャンプする

HTML 文書

```
<a href="2.html">Next</a>
<a href="1.html">Back</a>
```

Web ブラウザ表示



関連ページ

7.2 HTMLタグ リファレンス

1

HTMLヘタ格作成の基礎

2

パソコンで簡単なHTML文書を作成し表示してみよう

3

カスタマイズで簡単にユーザーWebページを作成しよう

4

Style Sheet (CSS) の働きを知る

5

JavaScript (JS) で表示できる部品の働きを知る

6

CGIによるデバイスモニタ画面を作成しよう

7

リファレンス

1. HTML<タグ>作成の基礎

HTMLとは

HTML 構造

タグの基本ルール

タグの種類

ファイル名の付け方

1.5 ファイル名の付け方

ファイル名に使用可能な文字

下表の文字が使用できます。

種類	文字
半角アルファベット	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
半角英数	0 1 2 3 4 5 6 7 8 9
半角記号 (限定)	- (ハイフン), _ (アンダーバー)

ファイル名の付け方

ユーザ Web ページのトップページとして使用する HTML ファイルは、ファイル名を必ず“index.html”としてください。その他のページで使用する HTML ファイルは、任意のファイル名が使用できます。

例 **menu.html**
名前 拡張子

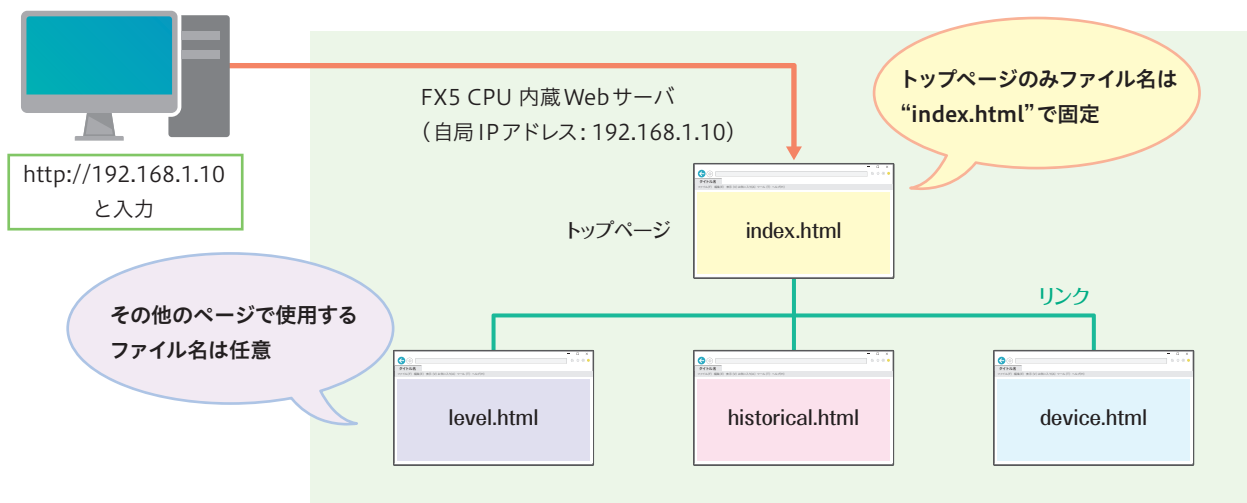
.htmlと.htmの2種類があるのはなぜ？

より詳しく知る

通常の HTML ファイルの拡張子は、".html"ですが、パソコンの古い OS (MS-DOS® や Windows® 3.1) では、拡張子を3文字までしか扱えなかったため、".html"を".htm"、".jpeg"を".jpg"と記述していました。このため.htmlと.htmが存在しています。.htmlと.htmは、どちらも使用できますが、Web サーバの設定仕様によっては片方しか使用できません。

▶ ファイルアクセスの仕組み

Web ブラウザのアドレスバーに、パラメータで設定した FX5 CPU ユニットの IP アドレスを入力し、Web サーバへアクセスすることで自動的に“index.html”を呼び出します。



Point

ファイル名には、表示内容に合ったわかりやすい名前を使用するようにしましょう。複雑な名前や通し番号は、URL を直接入力したり、リンクの設定をするときに間違いやすくなります。

2. パソコンで簡単なHTML文書を作成し表示してみよう

用意するもの

HTMLファイル作成

Webブラウザ表示

HTMLソース表示

実際にWindows® のメモ帳で作成したHTMLファイルをWebブラウザ (Internet Explorer®) で表示してみましょう。
また、Webブラウザを使用して、表示したWebページのHTML (ソース) を確認する方法を知っておきましょう。

2.1 用意するもの

次のいずれかのWindows® パソコンを用意してください。

Windows® パソコン	Microsoft® Windows® 10 Microsoft® Windows® 8 Microsoft® Windows® 7
使用するソフトウェア (Windows® の標準ソフト)	メモ帳 HTMLファイルの作成 Webブラウザ Internet Explorer® 11 作成したHTMLファイルの表示と確認

2.2 HTMLファイルの作成

Windows® に標準で搭載されているメモ帳を使用して実際にHTMLファイルを作成し、Webブラウザで表示するまでの手順をWindows® 10の例で説明します。

Windows® のメモ帳を開く

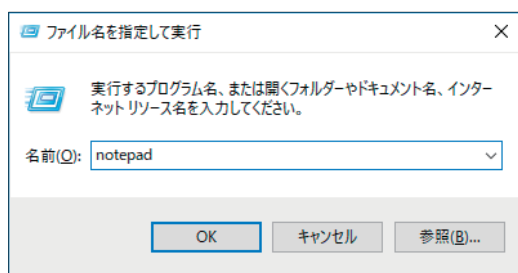
次のいずれかの方法で、メモ帳を開きます。

▶ ショートカットキーを利用して開く方法

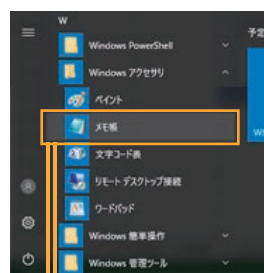
[Windows® のロゴキー ] + [R] キーを同時に押す ▶
「control」を「notepad」に変更 ▶ [OK] をクリック

▶ スタート画面から開く方法

Windows® アクセサリ内のメモ帳をクリック



*: 画面はWindows® 10の例



メモ帳の画面

2. パソコンで簡単なHTML文書を作成し表示してみよう

用意するもの

HTMLファイル作成

Webブラウザ表示

HTMLソース表示

2

HTML文書の入力

HTMLは、テキスト文だけで構成されているので、汎用的なテキストエディタでWebページを作成や編集できます。

メモ帳に右記のHTML文書を入力

```
<!DOCTYPE html>
<html>
<head>
<title>〇〇製造装置</title>
<meta>
</head>
<body>
<p><u>〇〇製造装置 (3 号機) △△工程</u></p>
<p>生産状態</p>
</body>
</html>
```

3

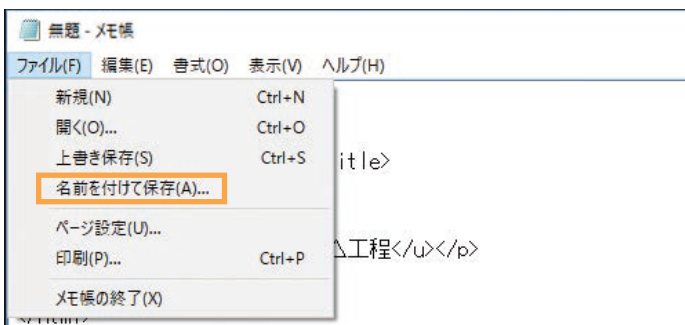
カスタマイズで簡単にユーザーWebページを作成しよう

4

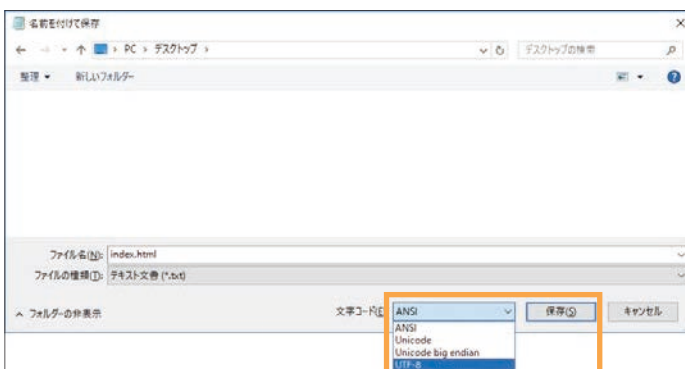
Style Sheet (CSS) の働きを知ろう

作成したHTML文書のファイル保存

[ツールバー:ファイル] ▶
[名前を付けて保存] をクリック



トップページのHTMLファイル名は
「index.html」▶
文字コード [UTF-8] を選択 ▶
[保存] をクリック



index . html

↑ ↑
ファイル名 拡張子

トップページのファイル名は「index.html」で保存してください。トップページ以外のファイル名は任意ですが、拡張子は「html」としてください。

7

リファレンス

2. パソコンで簡単なHTML文書を作成し表示してみよう

用意するもの

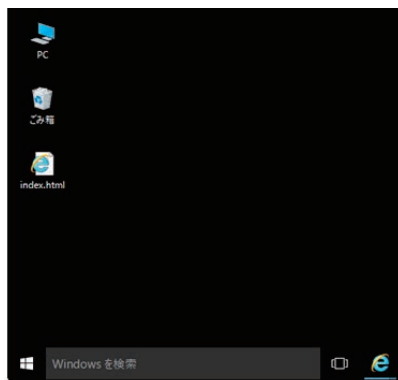
HTMLファイル作成

Webブラウザ表示

HTMLソース表示

2.3 Webブラウザ表示

保存した「index.html」ファイルを
ダブルクリック



▶ Webブラウザ表示画面

自動的にInternet Explorer® が立ち上がり、Web ページとして表示します。



2. パソコンで簡単なHTML文書を作成し表示してみよう

用意するもの

HTMLファイル作成

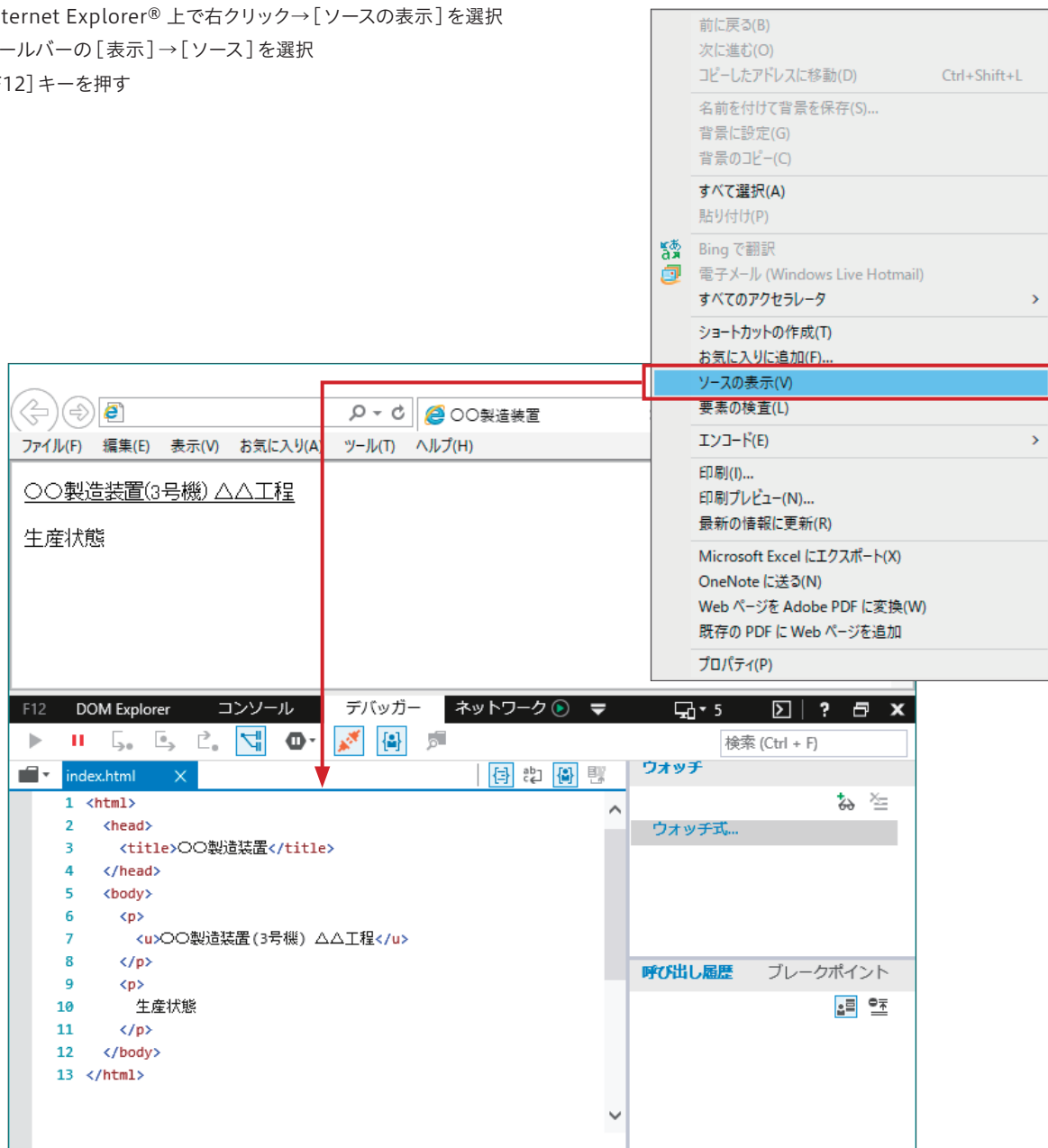
Webブラウザ表示

HTMLソース表示

2.4 WebブラウザでHTMLソースの表示方法

下記のいずれかの方法で、Webブラウザ下部に行番号とHTML文書を表示します。

- ・ Internet Explorer® 上で右クリック→[ソースの表示]を選択
- ・ ツールバーの[表示]→[ソース]を選択
- ・ [F12] キーを押す



3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

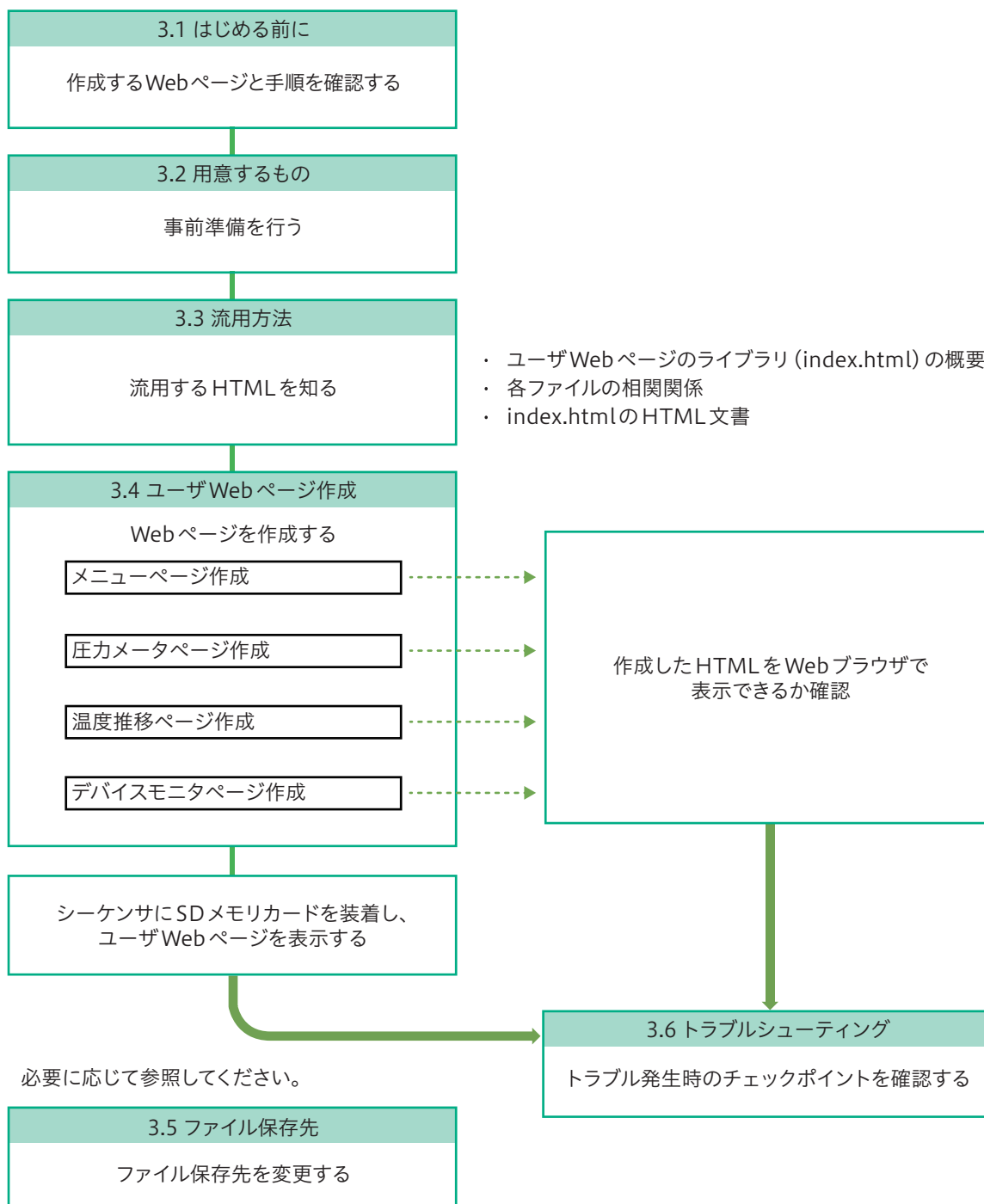
トラブルシューティング

三菱電機 FA サイトからダウンロード可能な HTML ファイル (Web サーバ機能用ライブラリ) を活用して、ユーザ Web ページの作成方法について説明します。

3.1 はじめる前に

作成の流れと関連ページ

Web ページを作成する手順を説明します。



3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

作成するユーザ Web ページ全体

本章では、ユーザ Web ページで使用可能な JavaScript 部品と CGI 部品を用い、下記のような HTML ページを作成します。JavaScript や CGI 部品の詳細は、それぞれの章を参考にして変更してください。

関連ページ

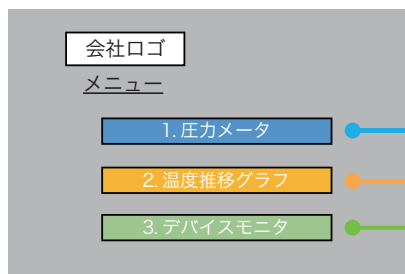
- ・ JavaScript の詳細 …… 5. JavaScript (JS) で表示できる部品の動きを知る
- ・ CGI の詳細 …………… 6. CGI によるデバイスモニタ画面を作成しよう

▶ メニュー 各ページへの切り替え

ユーザ Web ページを新規作成する。

3.4 ユーザ Web ページ作成

- ー メニューページ作成を参照してください。



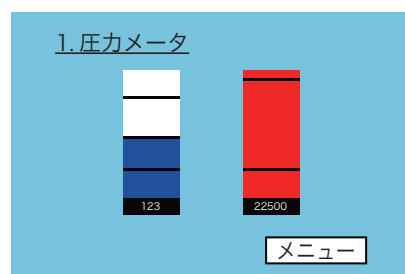
画面の切り替え

▶ レベル表示 圧力状態表示

ユーザ Web ページのライブラリを流用して作成する。

3.4 ユーザ Web ページ作成

- ー 圧力メータページ作成を参照してください。



▶ デバイスモニタ デバイスの読出し・書込み

6 章の CGI 例を流用して作成する。

3.4 ユーザ Web ページ作成

- ー デバイスモニタページ作成を参照してください。

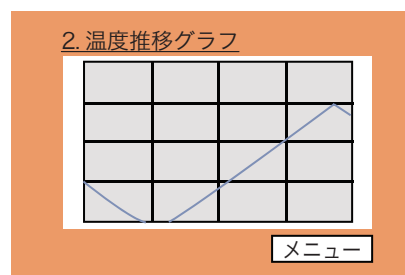


▶ ヒストリカルグラフ 温度の推移表示

ユーザ Web ページのライブラリを流用して作成する。

3.4 ユーザ Web ページ作成

- ー 温度推移ページ作成を参照してください。



3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめに

用意するもの

活用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

3.2 用意するもの

関連資料
関連ページ

- ・ Web サーバ機能 アプリケーションガイド Web ページを活用してみよう 立上げ導入編 [マニュアル番号: L(名)08642]
- ・ 6. ユーザ Web ページを使ってみよう

STEP1. Web サーバ HTML ファイルの準備

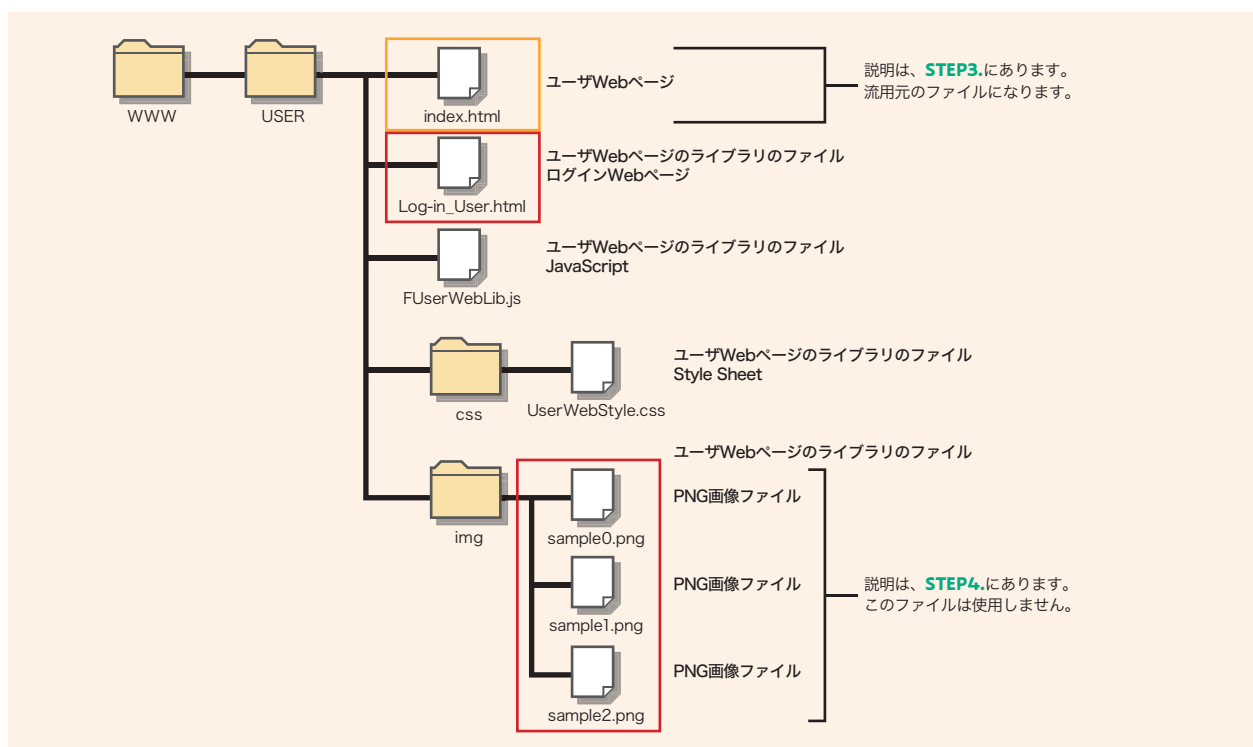
三菱電機 FA サイトから、ユーザ Web ページのライブラリをダウンロードします。
ダウンロードしたファイルを解凍します。

STEP2. ファイルとフォルダの確認

ダウンロードしたファイルやフォルダの構造が、下記のようにになっているか確認してください。

[ファイルとフォルダ構造]

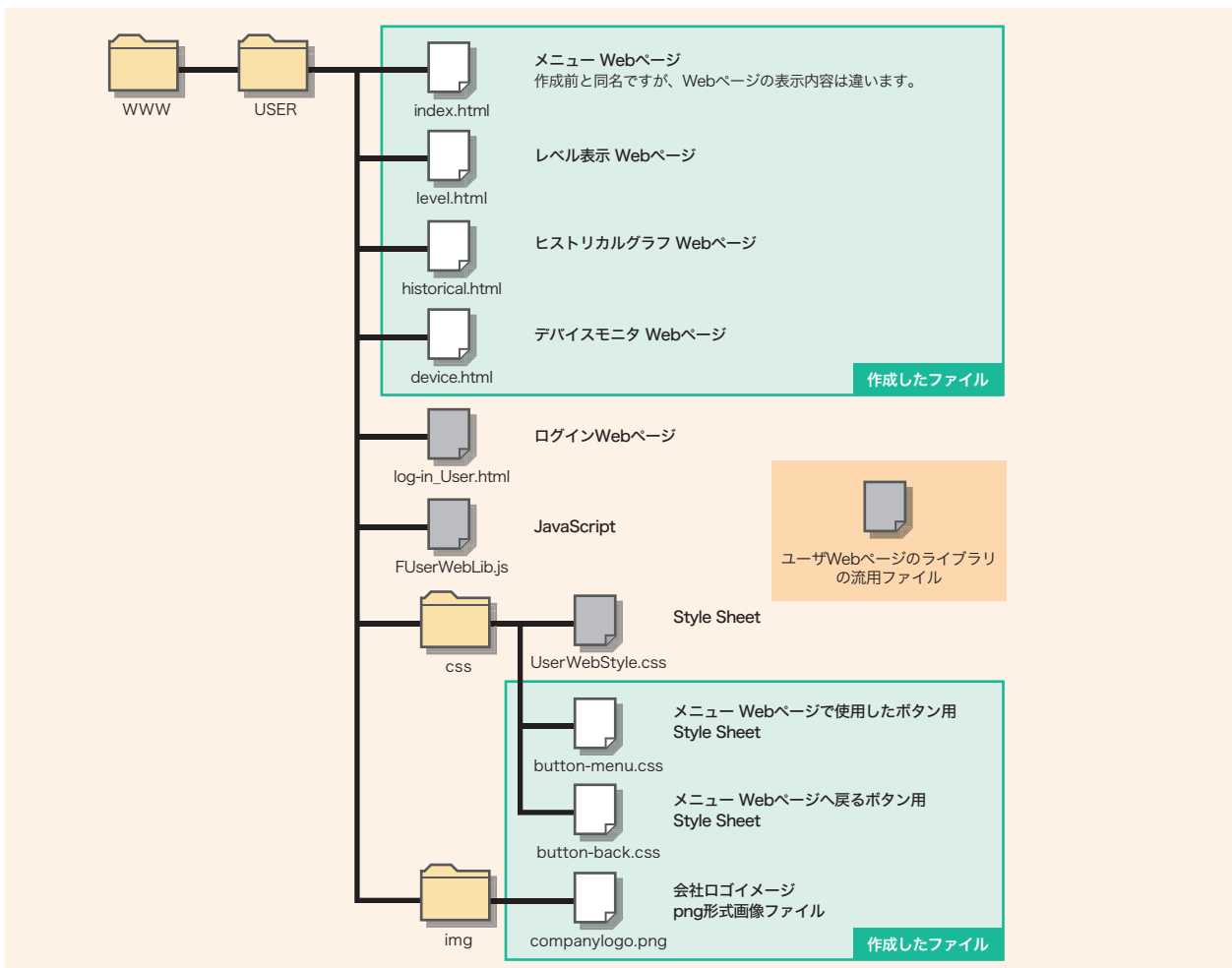
ユーザ Web ページのライブラリのファイル



3. カスタマイズで簡単にユーザ Web ページを作成しよう

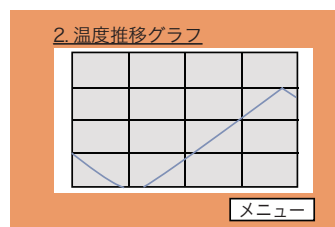
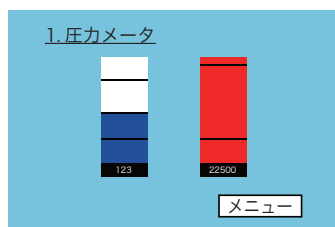
[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザ Web ページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)

▶ 作成後のファイル



STEP3. HTML ファイルの作成方法

レベル表示 Web ページ (level.html) とヒストリカルグラフ Web ページ (historical.html) は index.html (三菱電機 FA サイトからダウンロードした HTML ファイル) を流用して作成します。流用方法は 3.3 流用方法を参照してください。



STEP4. 不要なファイルの削除

本書では、img フォルダ内の下記画像ファイルを使用しないため削除します。

画像ファイル	sample0.png	sample1.png	sample2.png
--------	-------------	-------------	-------------

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

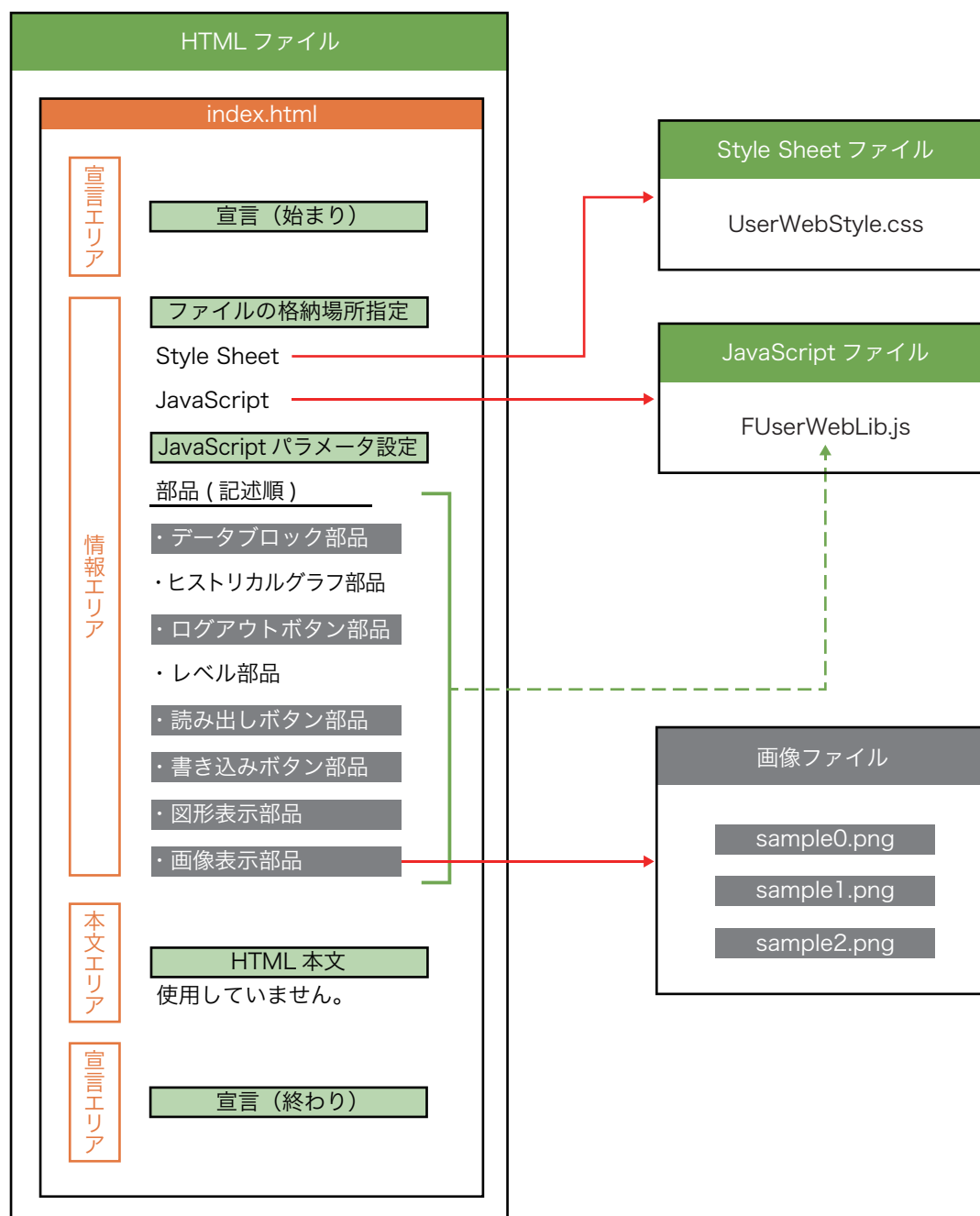
3.3 流用方法

“index.html”の概要

ユーザ Web ページのライブラリに含まれる、各ファイルの関連と各表示部品は下図のようになっています。

▶ 各ファイルの相関関係

白抜き JavaScript パラメータ設定の部品と画像ファイルは本書の作成例では使用しません。



1

HTML ヘタタ作成の基礎

2

パソコンで簡単なHTML文書を作成し表示してみよう

3

カスタマイズで簡単にユーザ Web ページを作成しよう

4

Style Sheet (CSS) の働きを知る

5

JavaScript (JS) で表示できる部品の働きを知る

6

CGIによるデバイスモニター画面を作成しよう

7

リファレンス

3. カスタマイズで簡単にユーザーWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)

index.htmlのHTML文書

レベル表示Webページ (level.html) とヒストリカルグラフ Web ページ (historical.html) の作成時に流用するユーザーWeb ページのライブラリ (index.html) のHTML文書を下記に示します。

行番号	HTML
1	<!DOCTYPE html>
2	<html xmlns="http://www.w3.org/1999/xhtml">
3	<head>
4	<!-- charsetの設定※Webサーバ設定がUTF-8のためUTF-8を設定する -->
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible" content="IE=edge"/>
7	<!-- タイトル名を設定する -->
8	<title>サンプル</title>
9	<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />
10	<!-- ライブラリ JavaScript読み込み (パスは格納位置に応じて変更する) -->
11	<script src="/FUserWebLib.js"></script>
12	<!-- 独自のJavaScriptはこれより後に記載する -->
13	<!-- ユーザーのJavaScriptをここに書く -->
14	<script>
15	var updateInterval = 5;
16	var dspLanguage = 'ja-JP';
17	
18	// データブロック部品
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	lnCol: 'blue',
52	xPos: 20,
53	yPos: 40
54	}
55	WSDatblk(dataBlockParam);
56	

3. カスタマイズで簡単にユーザ Web ページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザ Web ページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)

行番号	HTML
57	// ヒストリカルグラフ部品
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	// ログアウトボタン部品
92	logoutBtnParam = {
93	xPos: 20,
94	yPos: 730,
95	btnHeigh: 26,
96	btnWidth: 100,
97	btnTxt: 'ログアウト'
98	}
99	WSLogoutBtn(logoutBtnParam);
100	

3. カスタマイズで簡単にユーザーWebページを作成しよう

はじめの前に

用意するもの

流用方法

ユーザーWebページ作成

ファイル保存先

トラブルシューティング

行番号	HTML
101	// レベル部品
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	// 書き込みボタン部品
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	wrConfirm: 1,
140	language: 0,
141	xPos: 700,
142	yPos: 40,
143	});
144	
145	// 書き込みボタン部品
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	wrConfirm: 1,
157	language: 0,
158	xPos: 700,
159	yPos: 130,
160	});
161	

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめの前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

行番号	HTML
162	// 図形表示部品
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	// 画像表示部品
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. カスタマイズで簡単にユーザWebページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザWebページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

3.4 ユーザWebページ作成

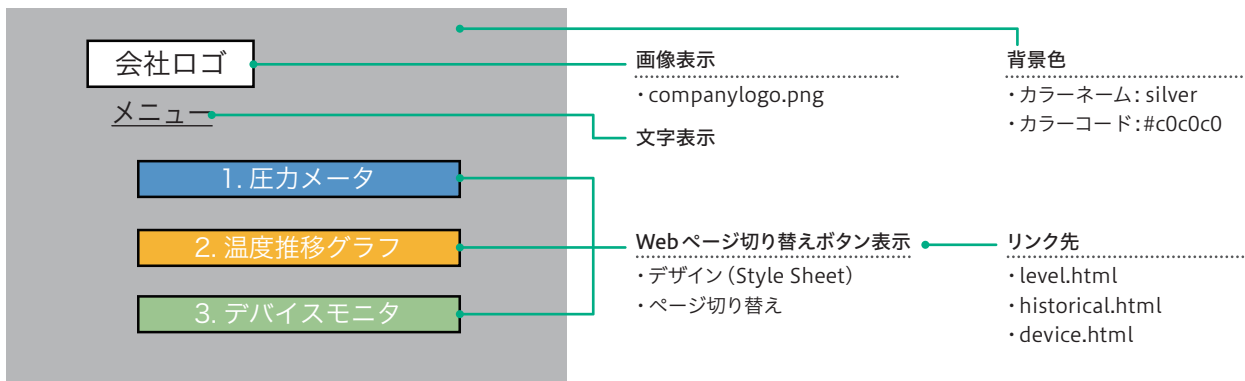
メニューページ作成

この作成例では、画像とWebページ切り替えボタンを表示します。Webページ切り替えボタンのデザインはStyle Sheetで行います。

本Webページは、index.htmlから流用しません。

仕様

[デザイン例]



使用する機能

機能		作成方法	用例や特記事項	参照節
背景色		Style Sheet	HTML ファイル内に Style Sheet を記述します。	7.1
画像表示		 タグ	HTML タグを使用した画像表示です。 	7.2
文字表示	文字サイズ	タグ	見出し3	
	アンダーライン	<u></u> タグ	<u>この範囲の文字にアンダーラインを引く</u>	
	段落	タグ	段落を示し、改行します。	7.1
Web ページ 切り替えボタン表示 (メニュー選択)	デザイン	Style Sheet	クリックすると右側のように変化します。 文字 クリック前 文字 クリック後	
	ページ切り替え	<a> タグ	表示テキスト	
	中央揃え	Style Sheet	Web ページの横幅を変更しても中央へ移動します。	

作成するファイル

ファイル種類	ファイル名	備考
HTML	index.html	ユーザWebサーバの最初に表示されるページのファイル名は "index.html" になります。変更しないでください。
画像	companylogo.png	ファイル名は任意です。変更した場合はHTMLファイルのリンク設定のタグ内の画像ファイル名も変更してください。
Style Sheet	button-menu.css	ファイル名は任意です。変更した場合はHTMLファイルのリンク設定のStyle Sheetファイル名も変更してください。

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

▶ 作成手順

STEP1. HTML ファイルの新規作成

関連ページ

操作の詳細は、下記も併せて参照してください。
2. パソコンで簡単な HTML 文書を作成し表示してみよう

- 1.Windows® のメモ帳を立ち上げます。
2. 下記 HTML の<!DOCTYPE html> から</html>の箇所を作成します。
3. メモ帳のツールバーの [ファイル] - [名前を付けて保存] を選択します。
4. ファイル名に “index.html” と入力し、[保存] ボタンをクリックします。分かりやすい場所にファイルを保存してください。

[HTML]

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charsetの設定※Webサーバ設定がUTF-8のためUTF-8を設定する -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- タイトル名を設定する -->
    <title>メニュー</title>
    <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
    <meta http-equiv="Content-Style-Type" content="text/css">
    <!-- Web ページ切り替えボタンのデザイン用 Style Sheet ファイルのリンク設定 -->
    <link href="/css/button-menu.css" rel="stylesheet" media="all"/>
    <!-- Web ページの背景色設定 (Style Sheet を HTML ファイル内に作成) -->
    <style>
      body {
        background-color: #c0c0c0;
      }
    </style>

    <!-- Web ページ切り替えボタンのセンター揃え用 (Style Sheet を HTML ファイル内に作成) -->
    <style type="text/css">
    <!--
    .web-center {
      text-align:center; /* Internet Explorer用の中央寄せ */
      margin-left:auto; /* Internet Explorer以外用の中央寄せ */
      margin-right:auto; /* Internet Explorer以外用の中央寄せ */
      width:300px; /* 幅 */
      height:180px; /* 高さ */
      color:#ffffff; /* 文字色 */
    }
    -->
    </style>
  </head>
  <body>
    <!-- 会社ロゴ画像の表示設定 -->
    
    <!-- Web ページのタイトル文字表示 -->
    <h2><u>メニュー</u></h2>
    <!-- Web ページ切り替えボタン (3 個) の表示 -->
    <div class="web-center">
      <p>
        <a href="index1.html" class="button1">1. 圧力メータ</a>
      </p>
      <p>
        <a href="index2.html" class="button2">2. 温度推移グラフ</a>
      </p>
      <p>
        <a href="index3.html" class="button3">3. デバイスモニタ</a>
      </p>
    </div>
  </body>
</html>
```

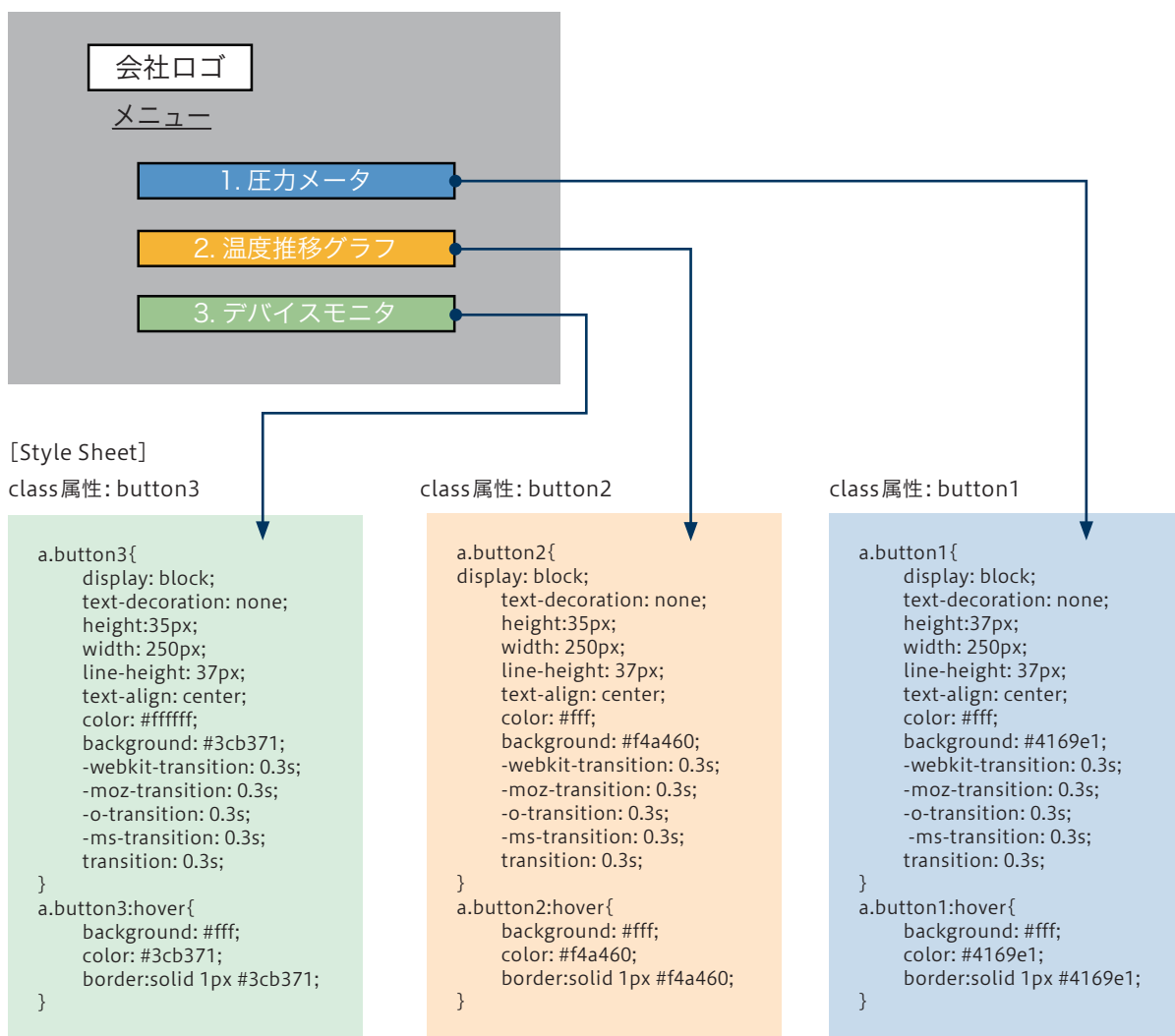

3. カスタマイズで簡単にユーザーWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニタページ作成](#)

STEP2. Style Sheet ファイルの作成

1. Windows® のメモ帳を立ち上げます。
 2. メモ帳に下記Style Sheetの内容をすべて入力します。
 3. メモ帳のツールバーの[ファイル]-[名前を付けて保存]を選択します。
 4. ファイル名に "button-menu.css" と入力し、[保存] ボタンをクリックします。
- Style Sheetの詳細は、7.1 Style Sheet リファレンスを参照してください。

[デザイン例] 色区分で各Webページ切り替えボタンのデザインを示します。



3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

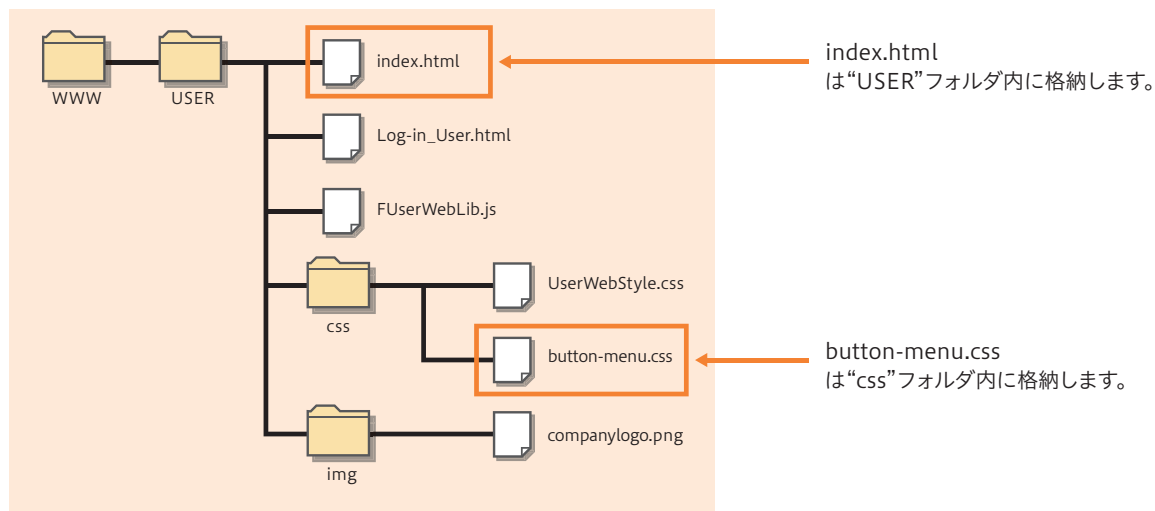
温度推移ページ作成

デバイスモニタページ作成

STEP3. 作成した各ファイルの構成

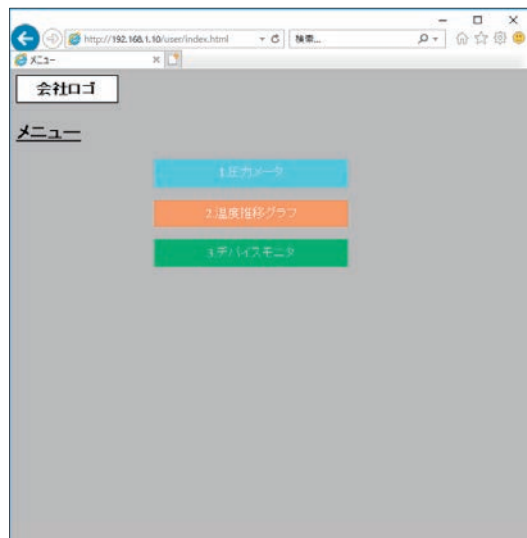
作成したファイルを下記ファイル構成になるように格納してください。

[構成後のファイル構成]



STEP4. Web ブラウザで正常に表示するか確認

“index.html” ファイルをダブルクリックします。



3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

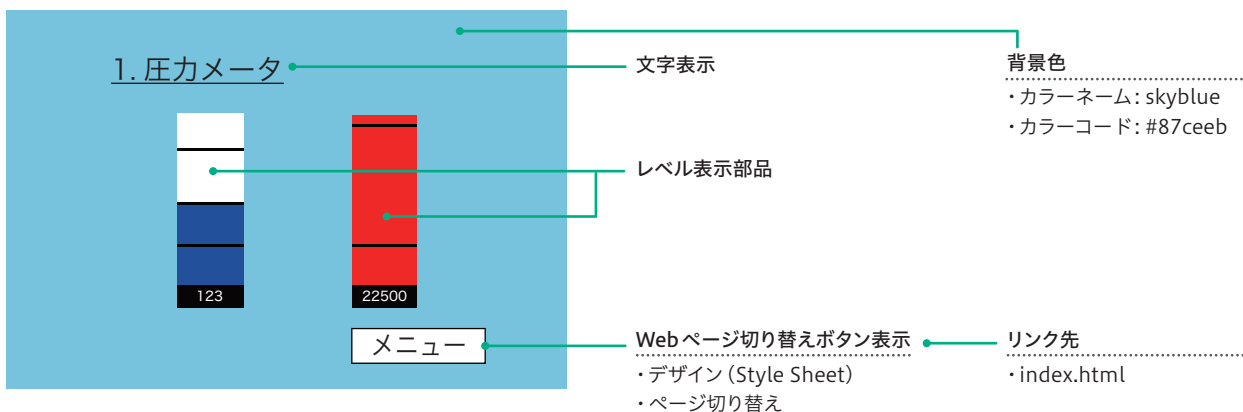
圧力メータページ作成

ここでは、圧力メータページの作成手順を説明します。

この作成例では、レベル表示部品と Web ページ切り替えボタンを表示します。レベル表示部品は“index.html”から流用して編集、Web ページ切り替えボタンのデザインは Style Sheet で行います。

仕様

[デザイン例]



使用する機能

機能	作成方法	用例や特記事項	参照節
背景色	Style Sheet	HTML ファイル内に Style Sheet を記述します。	7.1
画像表示	タグ	HTML タグを使用した画像表示です。 	7.2
文字表示	文字サイズ アンダーライン 段落	タグ <u></u>タグ タグ	7.2
レベル表示部品 (2 個)	ユーザ Web ページのライブラリの HTML ファイルにある“レベル表示部品”を流用します。		5.4
Web ページ切り替えボタン表示 (メニューへ戻る)	デザイン ページ切り替え 表示開始座標	JavaScript Style Sheet	7.1

作成するファイル

ファイル種類	ファイル名	備考
HTML	level.html	メニューページからリンクします。 ファイル名を変更する場合は、メニューページの修正も必要です。
Style Sheet	button-menu.css	ファイル名は任意です。変更した場合は HTML ファイルのリンク設定の Style Sheet ファイル名も変更してください。

必要なファイル

ファイル種類	ファイル名	備考
JavaScript	FUserWebLib.js	レベル表示部品を表示するために必要です。
Style Sheet	UserWebStyle.css	<link>タグで左記ファイルをリンク設定して使用します。

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

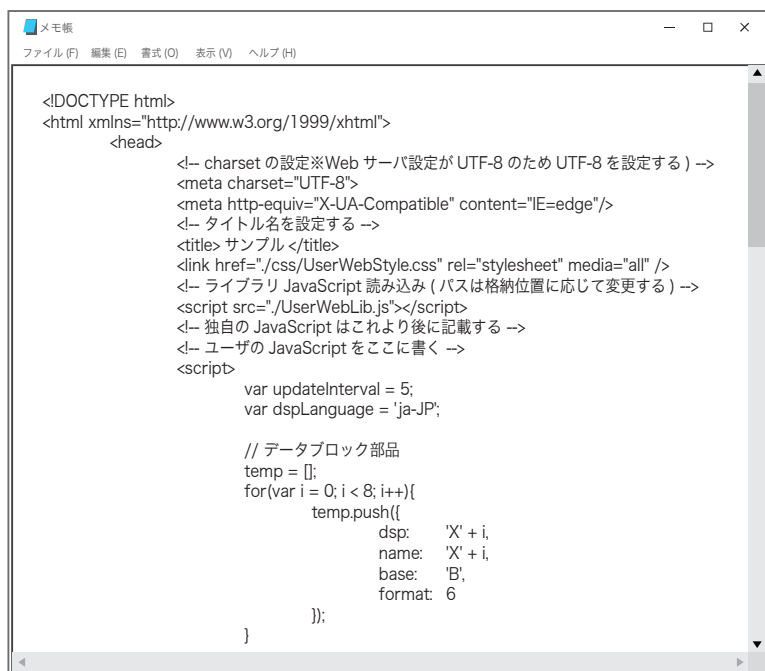
デバイスモニターページ作成

▶ 作成手順

三菱電機 FA サイトからダウンロードした、ユーザ Web ページのライブラリに含まれている、“index.html” ファイルを流用して作成します。

STEP1. 流用元ファイルをメモ帳で開く

ユーザ Web ページのライブラリの“index.html” ファイルをメモ帳で開きます。



STEP2. 別ファイル名“level.html”で保存

1. メモ帳のツールバーの [ファイル]-[名前を付けて保存] を選択します。
2. ファイル名に “level.html” と入力し、[保存] ボタンをクリックします。

流用作成した HTML を間違えて元ファイルに上書きすることを予防します。

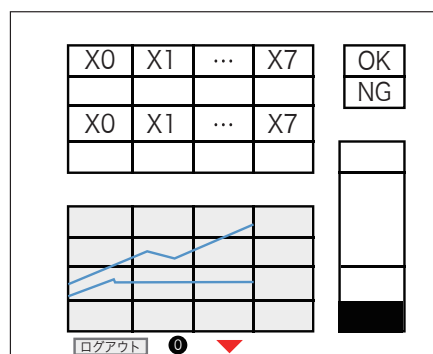
STEP3. 作成する Web ページに不要な HTML の行 (箇所) を削除

レベル表示部品の箇所以外を削除していきます。

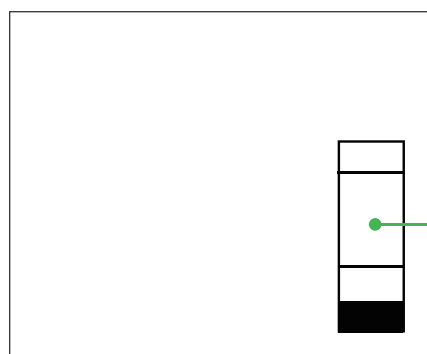
削除する行の詳細は、次ページを参照してください。

STEP3 以降は “level.html” を使用します。修正ファイルを間違えないよう注意してください。

変更前



変更後



3. カスタマイズで簡単にユーザーWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニタページ作成](#)

流用箇所

圧力メータ(レベル部品)を使用したWebページを作成するには、下表の流用箇所を示したユーザーWebページのライブラリ(index.html)の下記箇所を削除して流用します。

行番号	HTML	圧力メータ
1	<!DOCTYPE html>	削除しない
2	<html xmlns="http://www.w3.org/1999/xhtml">	
3	<head>	
4	<!-- charsetの設定※Webサーバ設定がUTF-8のためUTF-8を設定する -->	
5	<meta charset="UTF-8">	
6	<meta http-equiv="X-UA-Compatible" content="IE=edge"/>	
7	<!-- タイトル名を設定する -->	
8	<title>サンプル</title>	
9	<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />	
10	<!-- ライブラリJavaScript読み込み(パスは格納位置に応じて変更する) -->	
11	<script src="/FUserWebLib.js"></script>	
12	<!-- 独自のJavaScriptはこれより後に記載する -->	
13	<!-- ユーザーのJavaScriptをここに書く -->	
14	<script>	
15	var updateInterval = 5;	削除する
16	var dspLanguage = 'ja-JP';	
17		
18	// データブロック部品	
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	lnCol: 'blue',	
52	xPos: 20,	
53	yPos: 40	
54	}	
55	WSDatblk(dataBlockParam);	
56		

3. カスタマイズで簡単にユーザ Web ページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザ Web ページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニターページ作成](#)

行番号	HTML	圧力メータ
57	// ヒストリカルグラフ部品	削除する
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	// ログアウトボタン部品	
92	logoutBtnParam = {	
93	xPos: 20,	
94	yPos: 730,	
95	btnHeigh: 26,	
96	btnWidth: 100,	
97	btnTxt: 'ログアウト'	
98	}	
99	WSLogoutBtn(logoutBtnParam);	
100		

1 HTML ヘタケ V 作成の基礎

2

パソコンで簡単な HTML 文書を作成し表示してみよう

3

カスタマイズで簡単にユーザ Web ページを作成しよう

4

Style Sheet (CSS) の働きを知る

5

JavaScript (JS) で表示できる部品の働きを知る

6

CGI によるデバイスモニター画面を作成しよう

7

リファレンス

3. カスタマイズで簡単にユーザWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニターページ作成](#)

行番号	HTML	圧力メータ
101	// レベル部品	削除しない
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,	削除する
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	// 書き込みボタン部品	
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: 0,	
141	xPos: 700,	
142	yPos: 40,	
143	});	
144		
145	// 書き込みボタン部品	
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: 0,	
158	xPos: 700,	
159	yPos: 130,	
160	});	
161		

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニターページ作成

行番号	HTML	圧力メータ
162	// 図形表示部品	削除する
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,	削除しない
180	high: 10000,	
181	col: 'blue',	
182	},	
183	},	
184	xPos: 300,	
185	yPos: 730,	
186	});	
187		
188	// 画像表示部品	
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. カスタマイズで簡単にユーザ Web ページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザ Web ページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニタページ作成](#)

2

パソコンで簡単な HTML 文書を作成し表示してみよう

STEP4. </script> タグを追加

44 行目の空白行に `</script>` タグを追加してください。

行番号	HTML
18	...
19	// レベル部品
20	...
21	...
22	...
23	xPos: 700,
24	yPos: 250,
25	});
26	<code></script></code> ← 追加
27	<code></head></code>
28	<code><body></code>
29	<code></body></code>
30	<code></html></code>

3

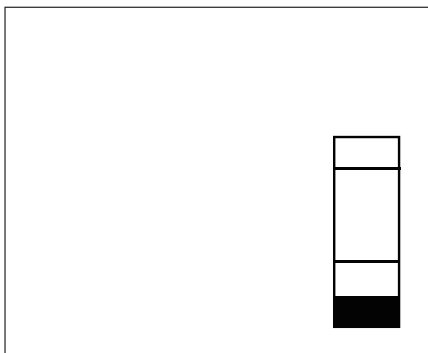
カスタマイズで簡単にユーザ Web ページを作成しよう

STEP5. レベル表示部品 (1 個目) のサイズ、表示座標などの修正

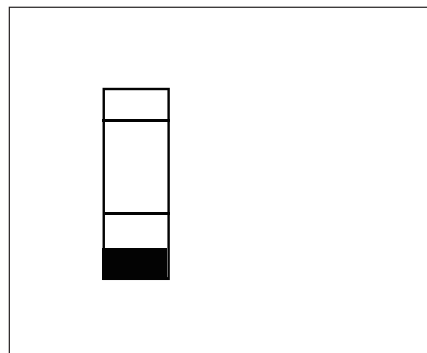
レベル表示部品 (1 個目) のサイズ、表示座標などを修正してください。

行番号	HTML
18	...
19	// レベル部品
20	WSLevel({
21	devName: 'D0', デバイス指定
22	...
23	levLength: 400, ← 200, レベルのサイズ
24	levWidth: 150, ← 100,
25	...
26	devValWidth: 150, ← 100, デバイス値表示サイズ
27	devValHeight: 50, ← 50,
28	xPos: 700, ← 100, 表示開始座標
29	yPos: 250, ← 50,
30	});

変更前



変更後



6

CGIによるデバイスモニタ画面を作成しよう

7

リファレンス

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

STEP6. レベル表示部品 (2 個目) を追加

行番号	HTML
18	// レベル部品
41	xPos: 100,
42	yPos: 50,
43	});
45	</script>
46	</head>
47	<body>
48	</body>
49	</html>

18~43 行をコピーし、43 行と 44 行の間に貼り付けます。

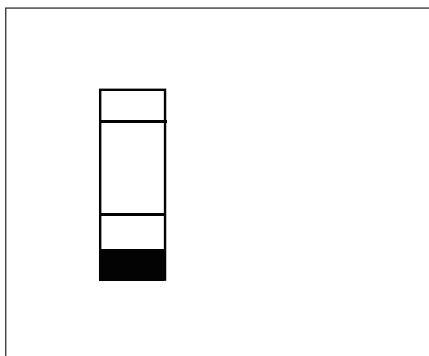
STEP7. レベル表示部品 (2 個目) の表示位置とサイズを修正

行番号	HTML
44	// レベル部品
45	WSLevel({
46	devName: 'D0',
66	devValHeight: 50,
67	xPos: 100,
68	yPos: 50,
69	});
70	</script>
71	</head>
72	<body>
73	</body>
74	</html>

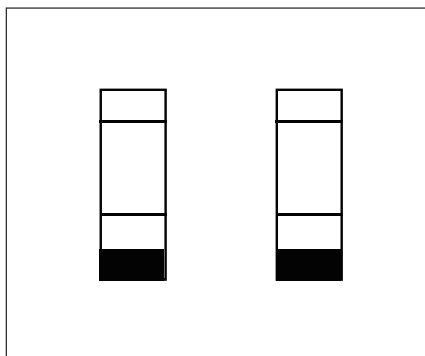
← 'D1', デバイス指定

← 300, 表示開始座標

変更前



変更後



3. カスタマイズで簡単にユーザーWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニタページ作成](#)

STEP8. 背景色を設定

背景色のHTMLを13行と14行の間に追加します。

背景色を白色にする場合、HTMLを追加する必要はありません。

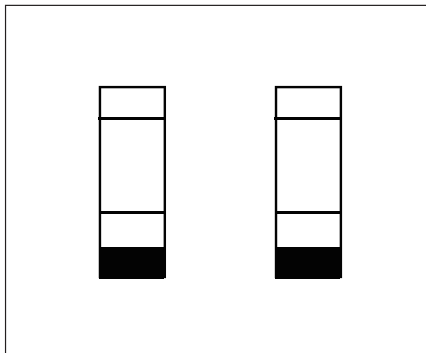
行番号	HTML
...	...
10	<!-- ライブラリJavaScript読み込み (パスは格納位置に応じて変更する) -->
11	<script src="/FUserWebLib.js"></script>
12	<!-- 独自のJavaScriptはこれより後に記載する -->
13	<!-- ユーザのJavaScriptをここに書く -->
	<style>
	body {
	background-color: #87ceeb;
	}
	</style>
14	<script>
15	var updateInterval = 5;
16	var dspLanguage = 'ja-JP';

STEP9. ページタイトルの変更とWebページ切り替えボタンの追加

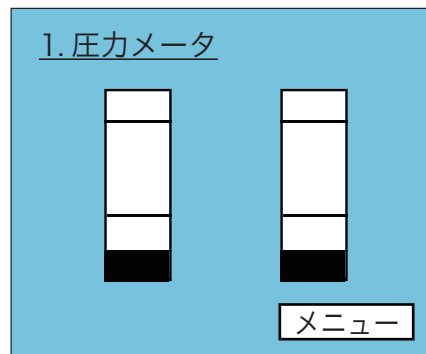
8行目のページタイトルを変更し、Webページ切り替えボタンのHTMLを、77行と78行の間に追加します。

行番号	HTML
...	...
8	<title>1. 圧力メータ</title> ← ページのタイトル文字表示
...	...
77	<body>
	<!-- Webページのタイトル文字表示 -->
	<u>1. 圧力メータ</u>
	<!-- Webページ切り替えボタンの表示 -->
	メニュー
78	</body>
79	</html>

変更前



変更後



3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

1 HTML ヘタケ V 作成の基礎

2

パソコンで簡単な HTML 文書を作成し表示してみよう

3

カスタマイズで簡単にユーザ Web ページを作成しよう

4

Style Sheet (CSS) の働きを知ろう

5

JavaScript (JS) で表示できる部品の働きを知る

6

CGI によるデバイスモニタ画面を作成しよう

7

リファレンス

STEP10. Web ページ切り替えボタン用 Style Sheet ファイルのリンク<link> 設定追加

Web ページ切り替えボタン用 Style Sheet ファイルのリンク<link> 設定の HTML を、9 行と 10 行の間に追加します。

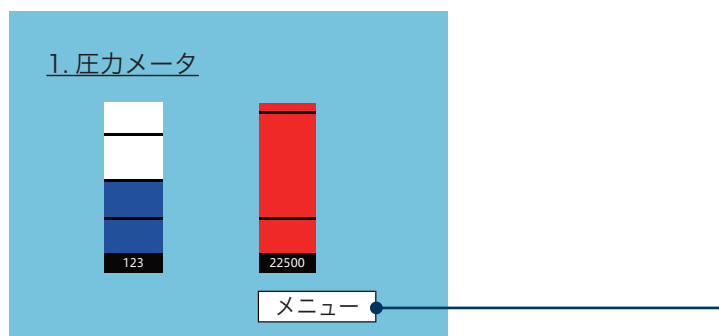
行番号	HTML
...	...
7	<!-- タイトル名を設定する -->
8	<title>1. 圧力メータ</title>
9	<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />
	<link href="/css/button-back.css" rel="stylesheet" media="all" />
10	<!-- ライブラリ JavaScript 読み込み (パスは格納位置に応じて変更する) -->

STEP11. Web ページ切り替えボタン用 Style Sheet ファイルの作成

1. Windows® のメモ帳を立ち上げます。
2. 下記の Style Sheet を作成します。
3. メモ帳のツールバーの [ファイル]-[名前を付けて保存] を選択します。
4. ファイル名に "button-back.css" と入力し、[保存] ボタンをクリックします。

“圧力メータページ”、“温度推移ページ”、“デバイスモニタページ”で使用します。

[デザイン例] Web ページ切り替えボタンのデザインを示します。



[Style Sheet]
class 属性: 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;
}
```

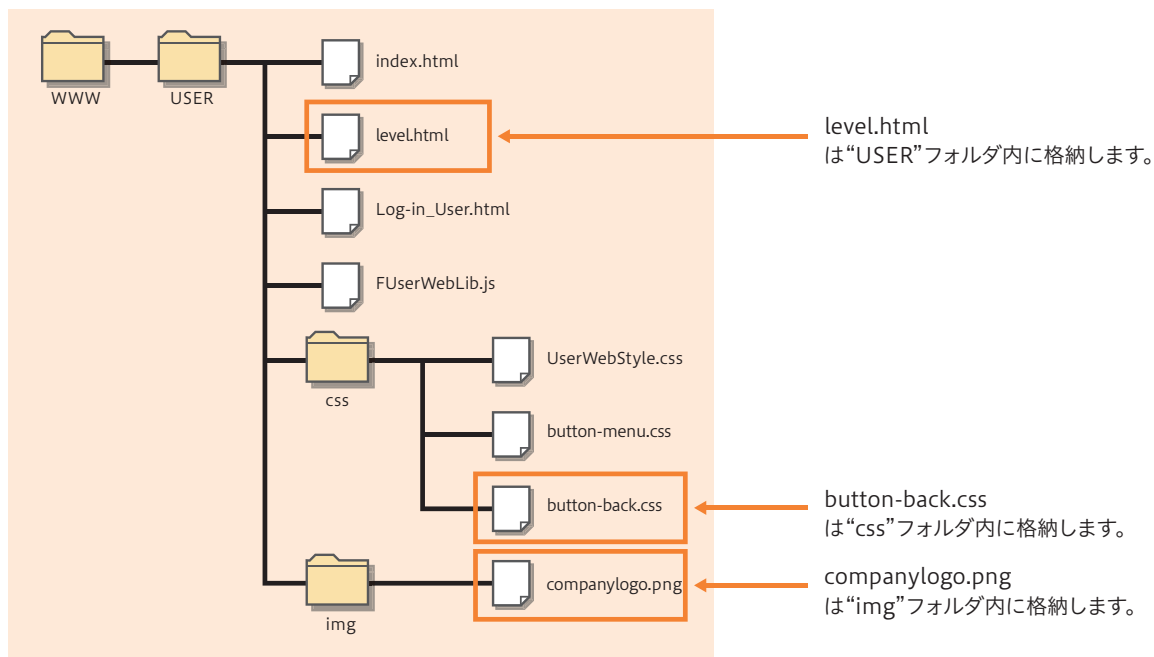
3. カスタマイズで簡単にユーザーWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニタページ作成](#)

STEP12. 作成した各ファイルの構成

作成したファイルを下記ファイル構成になるように格納してください。

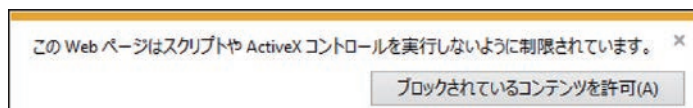
[構成後のファイル構成]



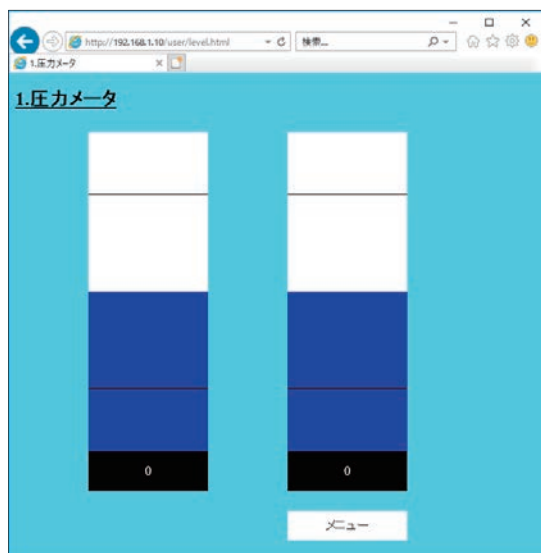
STEP13. Webブラウザで正常に表示できるかを確認

“level.html”ファイルをダブルクリックします。

このダイアログボックスが表示される場合は、“ブロックされているコンテンツを許可”ボタンをクリックしてください。



正常に表示されたWebページ



3. カスタマイズで簡単にユーザーWebページを作成しよう

はじめのうに

用意するもの

流用方法

ユーザーWebページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

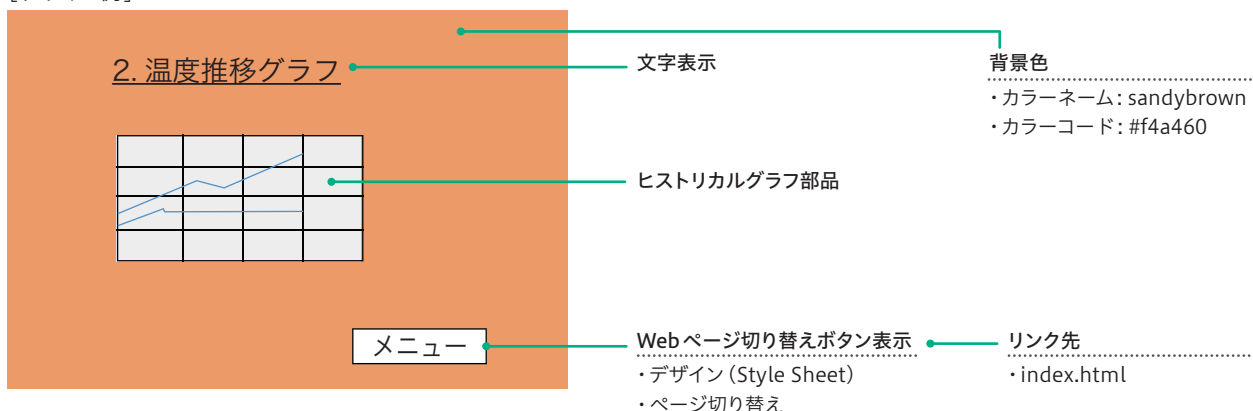
温度推移ページ作成

ここでは、温度推移ページの作成手順を例に説明します。

この作成例では、ヒストリカルグラフ部品とWebページ切り替えボタンを表示します。ヒストリカルグラフ部品は“index.html”から流用して編集、Webページ切り替えボタンのデザインはStyle Sheetで行います。

仕様

[デザイン例]



使用する機能

機能	作成方法	用例や特記事項	参照節
背景色	Style Sheet	HTML ファイル内に Style Sheet を記述します。	7.1
文字表示	文字サイズ	タグ	7.2
	アンダーライン	<u></u> タグ	
	段落	タグ	
ヒストリカルグラフ部品		ユーザーWebページのライブラリのHTMLファイルにある“ヒストリカルグラフ部品”を流用します。	5.4
	デバイス	D0とD1	
	サイズ	grHeight、grWidth パラメータを変更します。	
	表示開始座標	xPos:、yPos: のパラメータを変更します。	
Webページ切り替えボタン表示 (メニューへ戻る)	デザイン	クリックすると右側のように変化します。 文字 クリック前 文字 クリック後	7.1
	ページ切り替え	<a> タグ	
	表示開始座標	Style Sheet	

作成するファイル

ファイル種類	ファイル名	備考
HTML	historical.html	メニューページからリンクします。 ファイル名を変更する場合は、メニューページの修正も必要です。
Style Sheet	button-back.css	ファイル名は任意です。変更時はHTMLファイルのリンク設定のStyle Sheetファイル名も変更してください。

必要なファイル

ユーザーWebページのライブラリにあるファイルをそのまま使用します。

ファイル種類	ファイル名	備考
JavaScript	FUserWebLib.js	ヒストリカルグラフ部品を表示するために必要です。
Style Sheet	UserWebStyle.css	<link>タグで左記ファイルをリンク設定して使用します。

3. カスタマイズで簡単にユーザーWebページを作成しよう

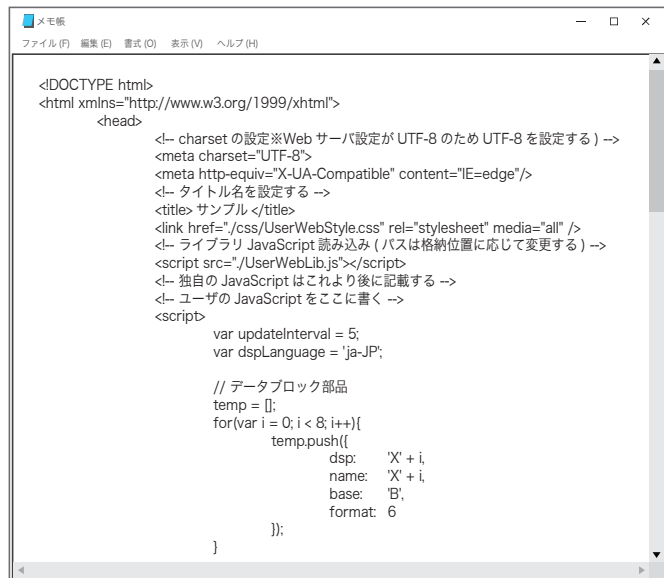
[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニタページ作成](#)

▶ 作成手順

三菱電機FAサイトからダウンロードした、ユーザーWebページのライブラリに含まれている、“index.html”ファイルを流用して作成します。

STEP1. 流用元ファイルをメモ帳で開く

ユーザーWebページのライブラリの中にある“index.html”ファイルをメモ帳で開きます。



STEP2. 別ファイル名“historical.html”で保存

1. メモ帳のツールバーの[ファイル]-[名前を付けて保存]を選択します。
2. ファイル名に“historical.html”と入力し、[保存]ボタンをクリックします。

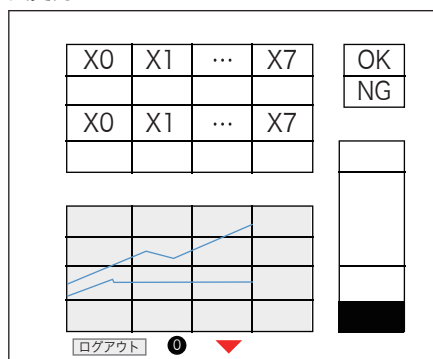
流用作成したHTMLを間違って元ファイルに上書きすることを予防します。

STEP3. 作成するWebページに不要なHTMLの行(箇所)を削除

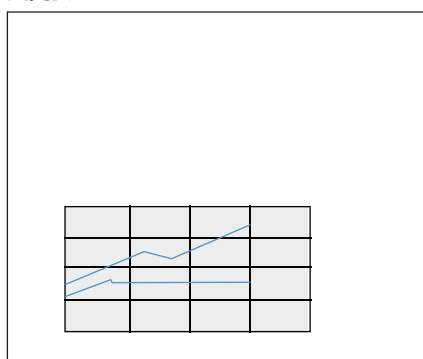
レベル表示部品の箇所以外を削除していきます。

STEP3以降は“historical.html”を使用します。修正ファイルを間違えないよう注意してください。

変更前



変更後



3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

流用箇所

温度推移グラフ (ヒストリカルグラフ部品) を使用した Web ページを作成するには、下表の流用箇所で示したユーザ Web ページのライブラリ (index.html) の下記箇所を削除して流用します。

行番号	HTML	温度推移グラフ
1	<!DOCTYPE html>	削除しない
2	<html xmlns="http://www.w3.org/1999/xhtml">	
3	<head>	
4	<!-- charsetの設定※Webサーバ設定がUTF-8のためUTF-8を設定する -->	
5	<meta charset="UTF-8">	
6	<meta http-equiv="X-UA-Compatible" content="IE=edge"/>	
7	<!-- タイトル名を設定する -->	
8	<title>サンプル</title>	
9	<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />	
10	<!-- ライブラリ JavaScript 読み込み (パスは格納位置に応じて変更する) -->	
11	<script src="/FUserWebLib.js"></script>	
12	<!-- 独自の JavaScript はこれより後に記載する -->	
13	<!-- ユーザの JavaScript をここに書く -->	
14	<script>	削除する
15	var updateInterval = 5;	
16	var dspLanguage = 'ja-JP';	
17		
18	// データブロック部品	
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	lnCol: 'blue',	
52	xPos: 20,	
53	yPos: 40	
54	}	
55	WSDatblk(dataBlockParam);	
56		

3. カスタマイズで簡単にユーザーWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニターページ作成](#)

行番号	HTML	温度推移グラフ
57	// ヒストリカルグラフ部品	削除しない
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	});	削除する
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	// ログアウトボタン部品	
92	logoutBtnParam = {	
93	xPos: 20,	
94	yPos: 730,	
95	btnHeigh: 26,	
96	btnWidth: 100,	
97	btnTxt: 'ログアウト'	
98	}	
99	WSLogoutBtn(logoutBtnParam);	
100		

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニターページ作成

行番号	HTML	温度推移グラフ
101	// レベル部品	
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	// 書き込みボタン部品	
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: 0,	
141	xPos: 700,	
142	yPos: 40,	
143	});	
144		
145	// 書き込みボタン部品	
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: 0,	
158	xPos: 700,	
159	yPos: 130,	
160	});	
161		

削除する

1 HTMLヘタケV作成の基礎

2

パソコンで簡単なHTML文書を作成し表示してみよう

3

カスタマイズで簡単にユーザ Web ページを作成しよう

4

Style Sheet (CSS) の働きを知る

5

JavaScript (JS) で表示できる部品の働きを知る

6

CGIによるデバイスモニター画面を作成しよう

7

リファレンス

3. カスタマイズで簡単にユーザWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニターページ作成](#)

行番号	HTML	温度推移グラフ
162	// 図形表示部品	削除する
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	// 画像表示部品	削除しない
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. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

STEP4. `</script>` タグを追加

51 行目の空白行に `</script>` タグを追加してください。

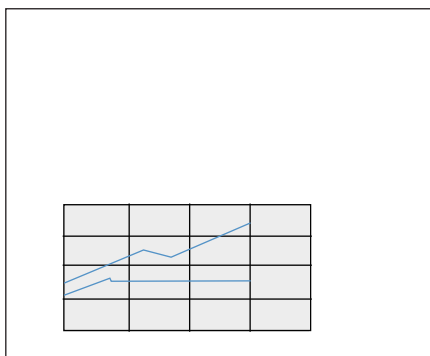
行番号	HTML
18	<code>// ヒストリカルグラフ部品</code>
19	<code>temp = [];</code>
20	<code>num = 2;</code>
21	<code>:</code>
22	<code>:</code>
47	<code>lowerMargin: 55,</code>
48	<code>rightMargin: 25,</code>
49	<code>}</code>
50	<code>WSHstgrp(hstGrpParam);</code>
51	<code></script></code> ← 追加
52	<code></head></code>
53	<code><body></code>
54	<code></body></code>
55	<code></html></code>

STEP5. ヒストリカルグラフ部品のサイズ、表示座標などの修正

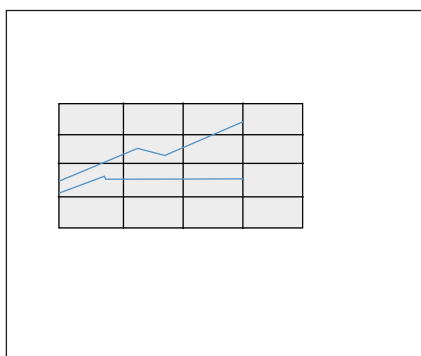
ヒストリカルグラフ部品のサイズ、表示座標などを修正してください。

行番号	HTML
18	<code>// ヒストリカルグラフ部品</code>
19	<code>temp = [];</code>
20	<code>num = 2;</code>
21	<code>:</code>
22	<code>:</code>
29	<code>hstGrpParam = {</code>
30	<code> xPos: 20,</code>
31	<code> yPos: 250,</code> ← 60,
32	<code> grElmNum: num,</code>
33	<code> :</code>
34	<code> :</code>
43	<code> grHeight: 380,</code> ← 250,
44	<code> grWidth: 550,</code> ← 450,
45	<code> upperMargin: 15,</code>
46	<code> leftMargin: 75,</code> ← 55,
47	<code> lowerMargin: 55,</code>
48	<code> rightMargin: 25,</code>
49	<code>}</code>
50	<code>WSHstgrp(hstGrpParam);</code>
51	<code></sctipt></code>
52	<code></head></code>
53	<code><body></code>
54	<code></body></code>
55	<code></html></code>

変更前



変更後



3. カスタマイズで簡単にユーザーWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニタページ作成](#)

STEP6. 背景色を設定

背景色のHTMLを13行と14行の間に追加します。

背景色を白色にする場合、HTMLを追加する必要はありません。

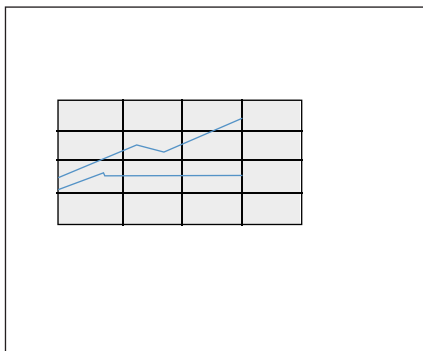
行番号	HTML
...	...
10	<!-- ライブラリJavaScript読み込み (パスは格納位置に応じて変更する) -->
11	<script src="/FUserWebLib.js"></script>
12	<!-- 独自のJavaScriptはこれより後に記載する -->
13	<!-- ユーザーのJavaScriptをここに書く -->
	<style>
	body {
	background-color: sandybrown;
	}
	</style>
14	<script>
15	var updateInterval = 5;
16	var dspLanguage = 'ja-JP';

STEP7. ページタイトルの変更とWebページ切り替えボタンの追加

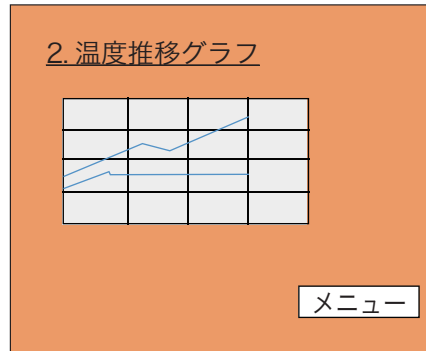
8行目のページタイトルを変更し、Webページ切り替えボタンのHTMLを、58行と59行の間に追加します。

行番号	HTML
...	...
8	<title>2. 温度推移グラフ</title>
...	...
58	<body>
	<!-- Web ページのタイトル文字表示 -->
	<u>2. 温度推移グラフ</u>
	<!-- Web ページ切り替えボタンの表示 -->
	メニュー
59	</body>
60	</html>

変更前



変更後



3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

活用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

STEP8. Web ページ切り替えボタン用 Style Sheet ファイルのリンク<link> 設定追加

Web ページ切り替えボタン用 Style Sheet ファイルのリンク<link> 設定の HTML を、9 行と 10 行の間に追加します。

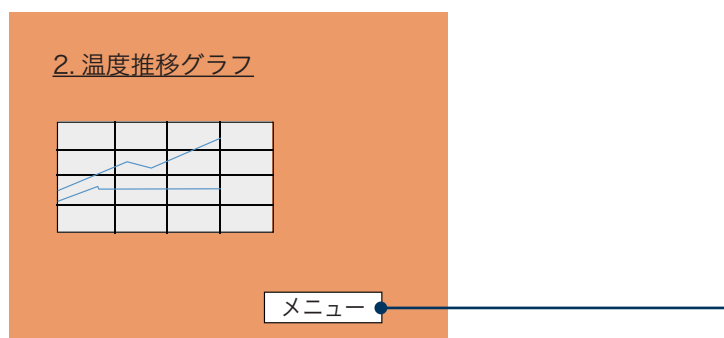
行番号	HTML
...	...
7	<!-- タイトル名を設定する -->
8	<title>2. 温度推移グラフ</title>
9	<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />
	<link href="/css/button-back.css" rel="stylesheet" media="all" />
10	<!-- ライブラリ JavaScript 読み込み (パスは格納位置に応じて変更する) -->

STEP9. Web ページ切り替えボタン用 Style Sheet ファイルの作成

1. Windows® のメモ帳を立ち上げます。
2. 下記の Style Sheet を作成します。
3. メモ帳のツールバーの [ファイル]-[名前を付けて保存] を選択します。
4. ファイル名に "button-back.css" と入力し、[保存] ボタンをクリックします。

“圧力メータページ”、“温度推移ページ”、“デバイスモニタページ”で使用します。

[デザイン例] Web ページ切り替えボタンのデザインを示します。



[Style Sheet]

class 属性: 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;
}
```

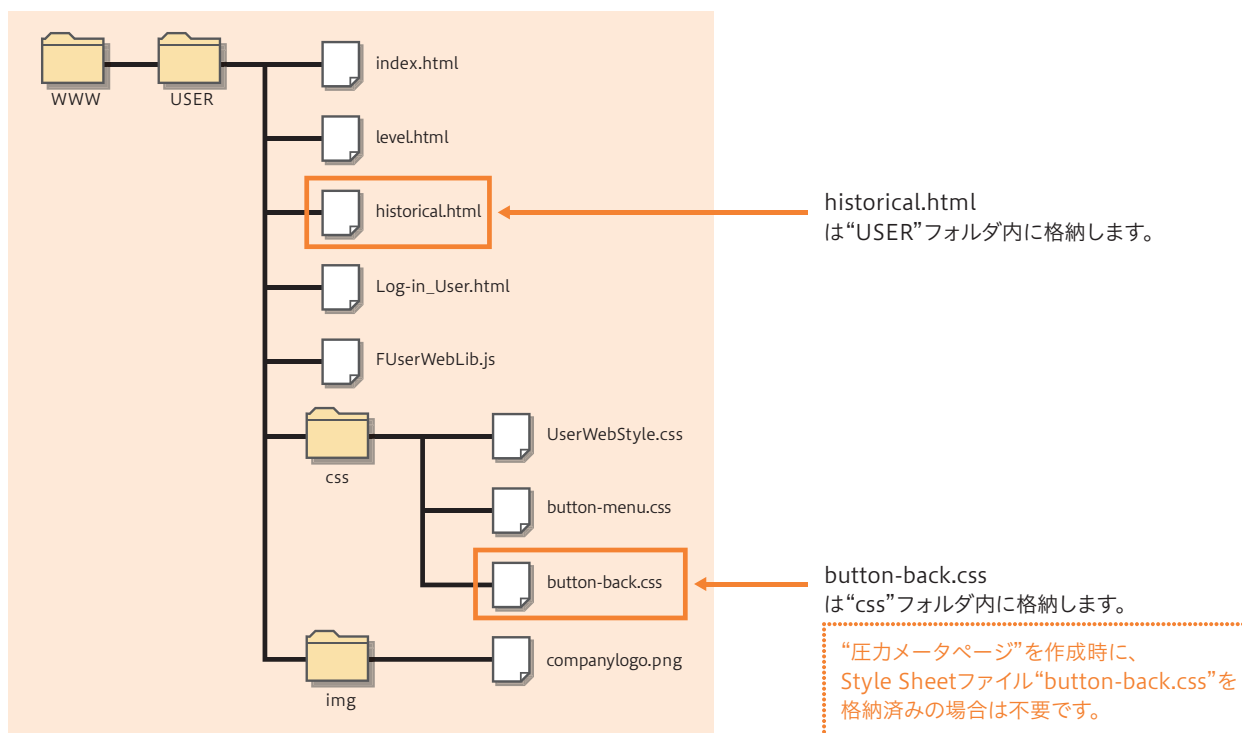
3. カスタマイズで簡単にユーザーWebページを作成しよう

[はじめの前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニタページ作成](#)

STEP10. 作成した各ファイルの構成

作成したファイルを下記ファイル構成になるように格納してください。

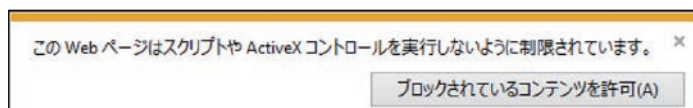
[構成後のファイル構成]



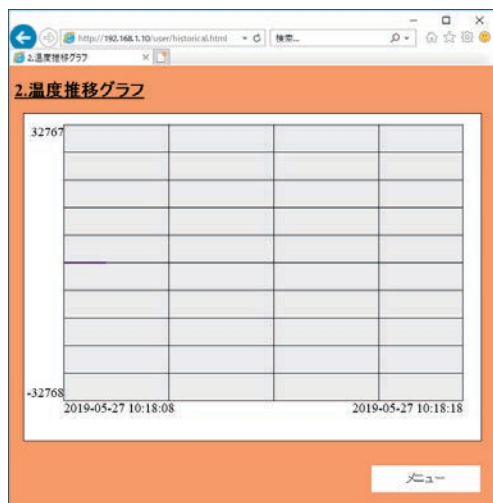
STEP11. Webブラウザで正常に表示できるかを確認

“historical.html”ファイルをダブルクリックします。

このダイアログボックスが表示される場合は、“ブロックされているコンテンツを許可”ボタンをクリックしてください。



正常に表示されたWebページ



3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめの例

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

デバイスモニタページ作成

ここでは、デバイスモニタページの作成手順を例に説明します。

この作成例では、デバイス読出し CGI、デバイス書込み CGI、Web ページ切り替えボタンを表示します。Web ページ切り替えボタンのデザインは Style Sheet で行います。

本 Web ページは、index.html から流用しません。下記の章を参照して HTML を作成してください。

→ 6. CGI によるデバイスモニタ画面を作成しよう

仕様

[デザイン例]

3. デバイスモニタ

デバイス名	データ型	値
D10	16bit 整数	1234
D11	32bit 整数	123456
M0	ビット	0

読出

デバイス名	データ型	値	
D10	16bit 整数	100	書込
D11	32bit 整数	600	書込
M0	ビット	1	書込

メニュー

文字表示

デバイス読出し CGI

デバイス書込み CGI

Web ページ切り替えボタン表示

リンク先

背景色

- ・カラーネーム: mediumseagreen
- ・カラーコード: #3cb371

リンク先

- ・index.html

使用する機能

機能		作成方法	用例や特記事項	参照節
背景色		Style Sheet	HTML ファイル内に Style Sheet を記述します。	7.1
文字表示	文字サイズ	タグ	見出し3	7.2
	アンダーライン	<u></u> タグ	<u>この範囲の文字にアンダーラインを引く</u>	
	段落	タグ	段落を示し、改行します。	
デバイス読出し CGI		本書の CGI 例 (HTML) を流用して作成します。		6.4
	デバイス	CGI	D10 と D11, M0	
	リンクファイル		RdDevRnd.cgi ファイル (シーケンサ内蔵)	
デバイス書込み CGI		本書の CGI 例 (HTML) を流用して作成します。		6.5
	デバイス	CGI	D10 と D11, M0	
	リンクファイル		WrDev.cgi ファイル (シーケンサ内蔵)	
Web ページ 切り替えボタン表示 (メニューへ戻る)	デザイン	Style Sheet	クリックすると右側のように変化します。 文字 クリック前 文字 クリック後	7.1
	ページ切り替え	<a> タグ	表示テキスト	
	表示開始座標	Style Sheet	<a> タグ内に Style Sheet を埋め込み、右下寄りに表示する。	

作成するファイル

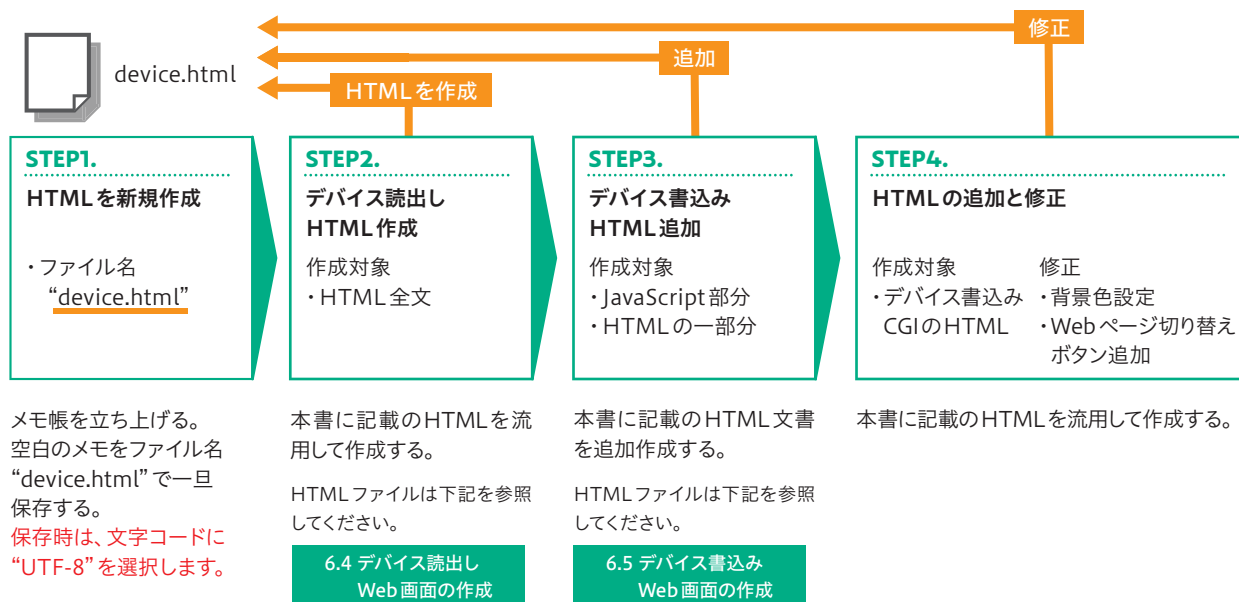
ファイル種類	ファイル名	備考
HTML	device.html	メニューページからリンクします。 ファイル名を変更する場合は、メニューページの修正も必要です。
Style Sheet	button-back.css	ファイル名は任意です。変更時は HTML ファイルのリンク設定の Style Sheet ファイル名も変更してください。

3. カスタマイズで簡単にユーザーWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニターページ作成](#)

▶ 手順

CGI 部品を使用して、デバイスを読み出し / 書き込みする HTML を作成する概要手順を説明します。



STEP1. デバイスモニターページのHTMLファイルを新規作成

- 1.Windows® のメモ帳を立ち上げます。
- 2.メモ帳のツールバーの[ファイル]-[名前を付けて保存]を選択します。
- 3.ファイル名に“device.html”と入力し、文字コードに“UTF-8”を選択してから[保存]ボタンをクリックします。分かりやすい場所にファイルを保存してください。

STEP2以降は“device.html”を使用します。修正ファイルを間違えないよう注意してください。

STEP2. デバイス読み出し CGI の HTML 作成

下記ページに記載している全文<!DOCTYPE html>～</html>のHTMLを流用して作成します。

関連ページ 6.4 デバイス読み出しWeb画面の作成 – HTML作成例 – HTML

[作成内容]

デバイス読み出し CGI のサンプルHTML			新規作成するHTML	
	行番号	HTML	device.html	
宣言エリア (始まり)	1	<!DOCTYPE html>	<!DOCTYPE html>	
情報エリア	⋮	⋮	⋮	
本文エリア	⋮	⋮	⋮	
宣言エリア (終わり)	117	</html>	</html>	

本書に記載の
HTMLを
作成

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

STEP3. デバイス書込み CGI (サンプル HTML) を追加

本書記載のデバイス書込み CGI から2カ所を、次の箇所に追加します。

関連ページ 6.5 デバイス書込み Web 画面の作成 - HTML 作成例 - HTML

- 1 デバイス書込み CGI サンプル HTML10 行目 `<script>` ~ 79 行 `</script>` 間を device.html の `</script>` ~ `</head>` 間に挿入します。
- 2 デバイス書込み CGI サンプル HTML83 行目 `<form>` ~ 111 行 `</form>` 間を device.html の `</form>` ~ `</body>` 間に挿入します。

[作成内容]

デバイス書込み CGI のサンプル HTML			作成している HTML	
	行番号	HTML		device.html
宣言エリア (始まり)	1	<code><!DOCTYPE html></code>		<code><!DOCTYPE html></code>
情報エリア	10	<code><script></code>		<code><script></code>
	⋮	⋮		⋮
	79	<code></script></code>	1 挿入	<code></script></code>
本文エリア	80	<code></head></code>		<code></head></code>
	82	<code><body></code>		<code><body></code>
	83	<code><form></code>		<code><form></code>
	⋮	⋮	2 挿入	⋮
	111	<code></form></code>		<code></form></code>
宣言エリア (終わり)	112	<code></body></code>		<code></body></code>
	113	<code></html></code>		<code></html></code>

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

STEP4. デバイス書込み CGI 共存のための HTML 修正

1 ページの Web ページにデバイス読出し CGI とデバイス書込み CGI の両方を使用するための修正を行います。

修正内容はデバイス読出し CGI で使用している ID 名と重複しないよう、デバイス書込み CGI の ID 名を 10 番台に修正した例です。

本修正を行わなかった場合は、シーケンサとデバイスの読出しや書込みが正常にできません。

[修正内容]

(1)			(2)			(3)		
行番号	変更前	変更後	行番号	変更前	変更後	行番号	変更前	変更後
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"

[修正後の HTML]

行番号	HTML
201	<form>
202	<table id="devtbl2" class="devtbl" border="1">
203	<tr>
204	<th>デバイス名</th>
205	<th>データ型</th>
206	<th>値</th>
207	</tr>
208	<tbody>
209	<tr> 変更前変更後
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="16bit 整数"/></td>
212	<td><input type="text" id="DATA10" name="DATA1" class="input" value="3"/></td>
213	<td><input type="button" value=" 書込 " 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="32bit 整数"/></td>
218	<td><input type="text" id="DATA11" name="DATA1" class="input" value="10"/></td>
219	<td><input type="button" value=" 書込 " 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="ビット"/></td>
224	<td><input type="text" id="DATA12" name="DATA1" class="input" value="1"/></td>
225	<td><input type="button" value=" 書込 " class="input" onclick="WriteDeviceBlockTbl('devtbl2',3)"/></td>
226	</tr>
227	</tbody>
228	</table>
229	</form>

(1)

(2)

(3)

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニタページ作成

STEP5. 背景色を設定

背景色の HTML を 9 行と 10 行の間に追加します。

背景色を白色にする場合、HTML を追加する必要はありません。

行番号	HTML
...	...
7	<!-- タイトル名を設定する -->
8	<title>デバイス読み出し CGI サンプル</title>
9	<!-- ユーザの JavaScript をここに書く -->
	<style>
	body {
	background-color: mediumseagreen;
	}
	</style>
10	<script>
11	// CGI リクエスト用の関数
12	function ReadDeviceRandomTbl(devtblid) {

STEP6. ページタイトルの変更と Web ページ切り替えボタンの追加

8 行目のページタイトルを変更し、Web ページ切り替えボタンの追加 HTML を、158 行と 159 行の間に追加します。

行番号	HTML
...	...
8	<title>3. デバイスモニタ</title>
...	...
156	</script>
157	</head>
158	<body>
	<!-- Web ページのタイトル文字表示 -->
	<u>3. デバイスモニタ</u>
	<!-- Web ページ切り替えボタンの表示 -->
	メニュー
159	<form>
160	<table id="devtbl" class="devtbl" border="1">

変更前

デバイス名	データ型	値
D10	16bit 整数	1234
D11	32bit 整数	123456
M0	ビット	0

読出

デバイス名	データ型	値	
D10	16bit 整数	100	書込
D11	32bit 整数	600	書込
M0	ビット	1	書込

変更後

3. デバイスモニタ		
デバイス名	データ型	値
D10	16bit 整数	1234
D11	32bit 整数	123456
M0	ビット	0

読出

デバイス名	データ型	値	
D10	16bit 整数	100	書込
D11	32bit 整数	600	書込
M0	ビット	1	書込

メニュー

3. カスタマイズで簡単にユーザーWebページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザーWebページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニタページ作成](#)

STEP7. Webページ切り替えボタン用Style Sheetファイルのリンク<link>設定追加

Webページ切り替えボタン用Style Sheet ファイルのリンク<link>設定の追加HTMLを、8行と9行の間に追加します。

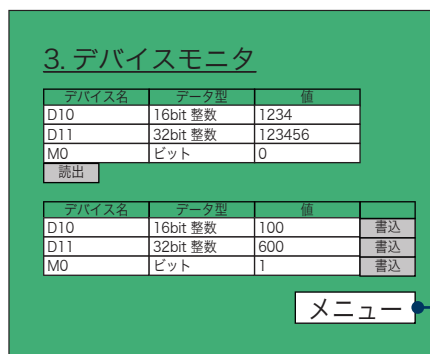
行番号	HTML
⋮	⋮
7	<!-- タイトル名を設定する -->
8	<title>3. デバイスモニタ</title>
	<link href="/css/button-back.css" rel="stylesheet" media="all" />
9	<!-- ユーザーのJavaScript をここに書く -->

STEP8. Webページ切り替えボタン用Style Sheet ファイルの作成

1. Windows® のメモ帳を立ち上げます。
2. 下記のStyle Sheetを作成します。
3. メモ帳のツールバーの[ファイル]-[名前を付けて保存]を選択します。
4. ファイル名に“button-back.css”と入力し、[保存]ボタンをクリックします。

“圧力メータページ”、“温度推移ページ”、“デバイスモニタページ”で使用します。

[デザイン例] Webページ切り替えボタンのデザインを示します。



[Style Sheet]

class属性: 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. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめの前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

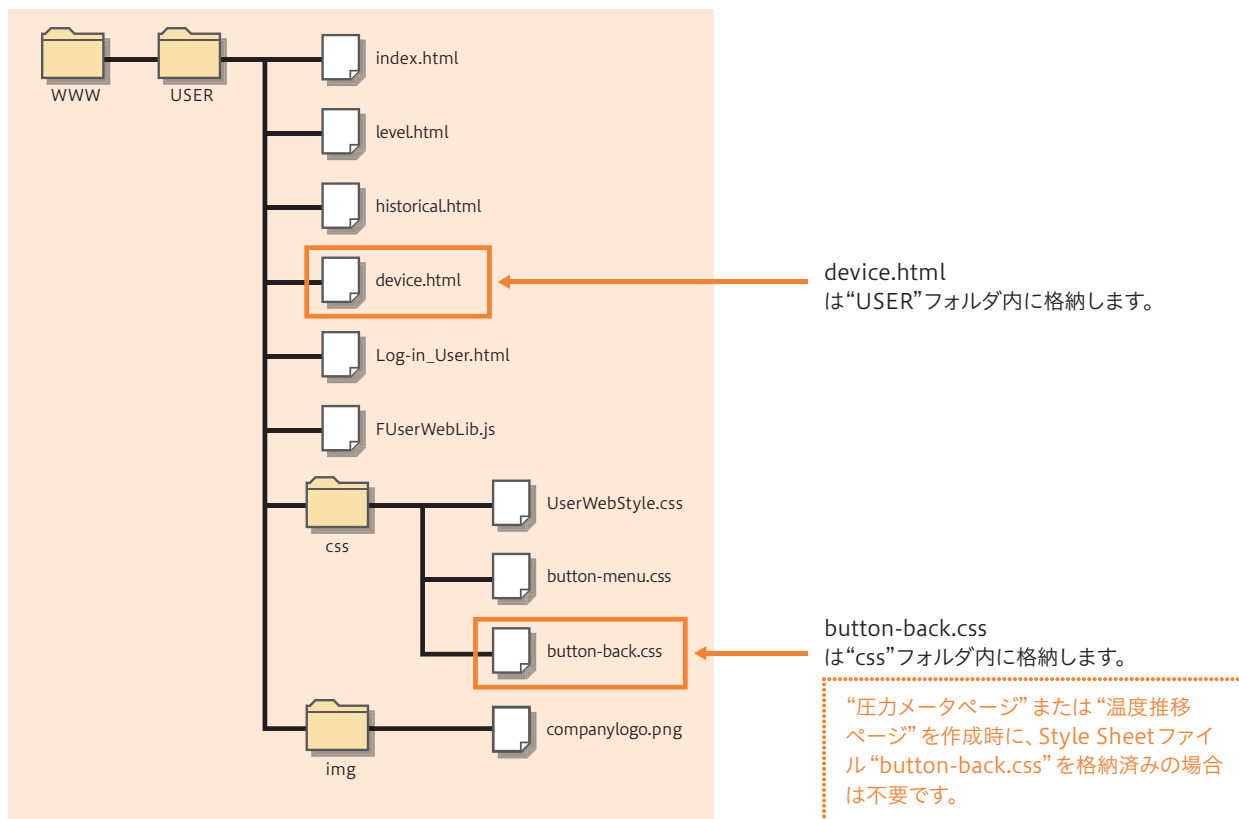
メニューページ作成

圧力メータページ作成

温度推移ページ作成

デバイスモニターページ作成

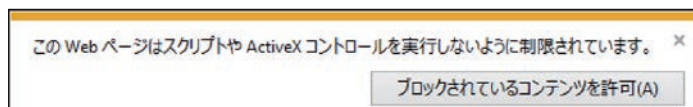
STEP9. 作成した各ファイルの構成



STEP10. Web ブラウザで正常に表示するか確認

“device.html” ファイルをダブルクリックします。

このダイアログボックスが表示される場合は、“ブロックされているコンテンツを許可” ボタンをクリックしてください。



3. カスタマイズで簡単にユーザ Web ページを作成しよう

[はじめる前に](#)[用意するもの](#)[流用方法](#)[ユーザ Web ページ作成](#)[ファイル保存先](#)[トラブルシューティング](#)[メニューページ作成](#)[圧力メータページ作成](#)[温度推移ページ作成](#)[デバイスモニタページ作成](#)

2

パソコンで簡単な HTML 文書を作成し表示してみよう

3

カスタマイズで簡単にユーザ Web ページを作成しよう

4

Style Sheet (CSS) の働きを知ろう

5

JavaScript (JS) で表示デバイスの動きを知る

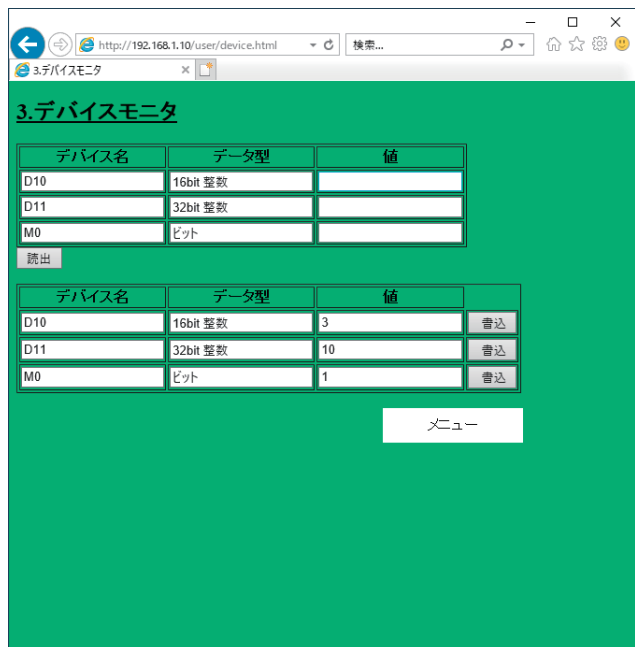
6

CG によるデバイスモニタ画面を作成しよう

7

リファレンス

正常に表示された Web ページ



以上で、すべてのユーザ Web ページの作成は終了です。

Web ページが正常に表示されない場合、3.6 トラブルシューティングを参照してください。

動作に関しては、下記をご覧ください。

関連資料

[Web サーバ機能 アプリケーションガイド Web ページを活用してみよう 立上げ導入編 \[マニュアル番号: L \(名\)08642\]](#)

3. カスタマイズで簡単にユーザWebページを作成しよう

はじめに

用意するもの

活用方法

ユーザWebページ作成

ファイル保存先

トラブルシューティング

3.5 ファイル保存先

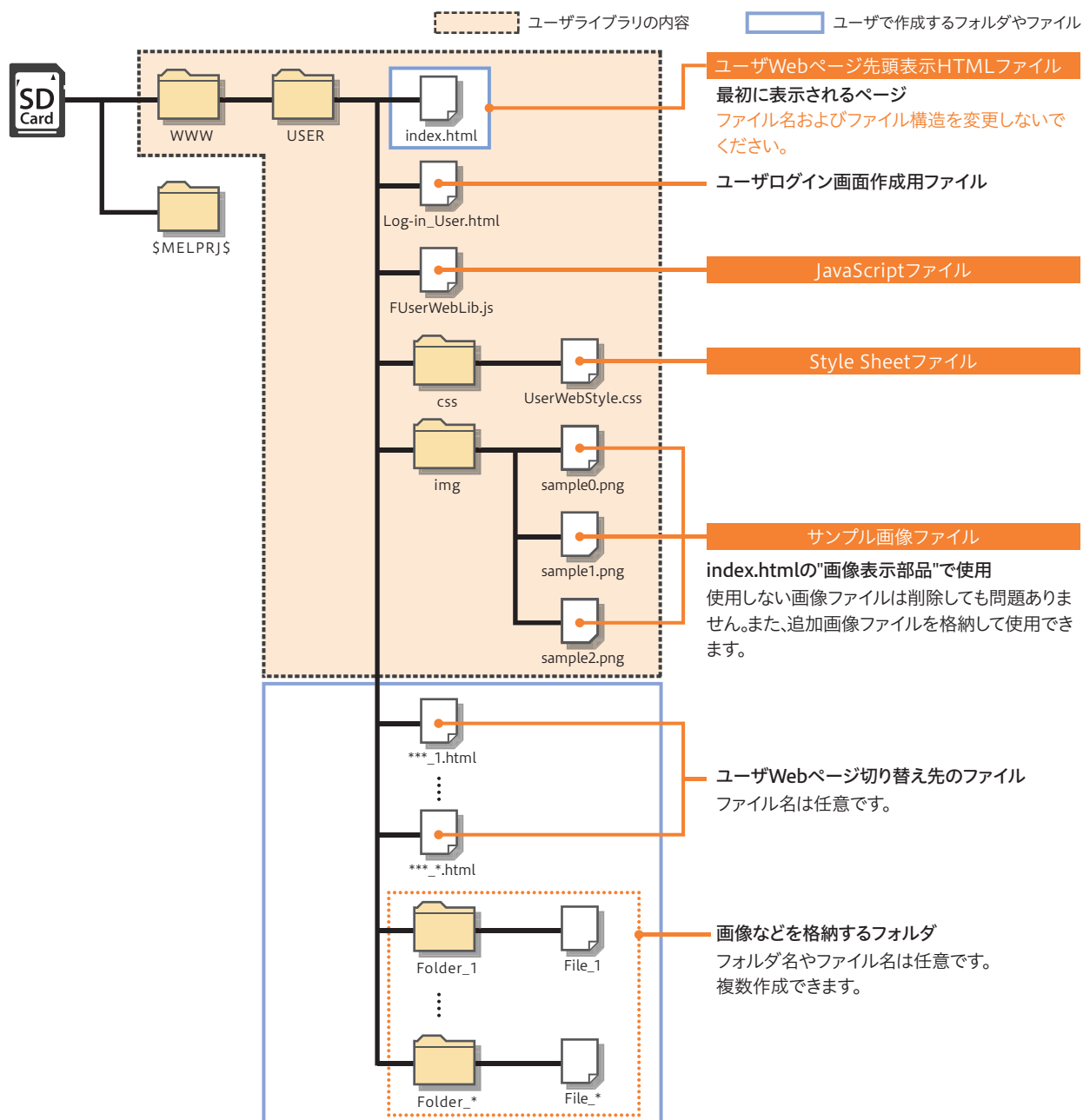
作成したユーザWebサーバのファイルはSDメモ리카ードへ格納します。本節はユーザWebサーバのファイル保存先について説明します。ファイル構造を変更する場合は、次の説明を参考に検討してください。

ユーザWebページのライブラリのファイル構造

ユーザWebページのライブラリのファイル構造を説明します。

下図はユーザWebページのライブラリで提供しているファイル群と、お客様が作成したファイル群をSDメモ리카ードに格納した場合のファイル構造です。ユーザライブラリのフォルダ構造やファイル名を変更すると正常に動作しない場合があります。

▶ ユーザWebサーバ用ファイル群



3. カスタマイズで簡単にユーザWebページを作成しよう

はじめの割に

用意するもの

流用方法

ユーザWebページ作成

ファイル保存先

トラブルシューティング

通常変更する必要はありませんが、フォルダによる分類や別のJavaScriptファイルもフォルダ内にまとめて格納する場合は参考にしてください。JavaScriptファイルとStyle Sheetファイルの場合について説明します。

JavaScriptファイルの場合

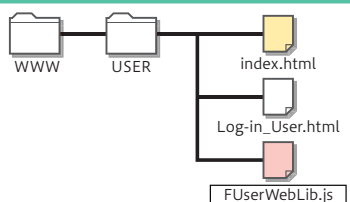
▶ ファイル名

ファイル名	機能	内容	初期格納先
FUserWebLib.js	JavaScript部品のライブラリ	ユーザWebページのライブラリに含まれているファイル	./FUserWebLib.js

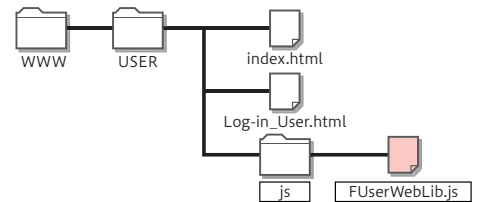
▶ HTML文書

HTMLファイル内からJavaScriptファイルの読み込み方法とファイルの格納場所により、パスの変更方法について説明します。

▶ “index.html”ファイルと“FUserWebLib.js”ファイルが同じ階層にある場合（初期格納先）

HTML文書	フォルダとファイル構造
<code><script src="./FUserWebLib.js"></script></code>	

▶ フォルダ（例: js）を作成し“FUserWebLib.js”ファイルを格納した場合

HTML文書	フォルダとファイル構造
<code><script src="./js/FUserWebLib.js"></script></code>	

▶ HTMLファイルへの記載方法

`<head>～</head>` 間に作成しますが、JavaScript開始行 `<script>` より上に作成します。

行番号	HTML
1	<code><!DOCTYPE html></code>
2	<code><html xmlns="http://www.w3.org/1999/xhtml"></code>
3	<code><head></code>
4	<code><!-- charsetの設定※Webサーバ設定がUTF-8のためUTF-8を設定する --></code>
5	<code><meta charset="UTF-8"></code>
6	<code><meta http-equiv="X-UA-Compatible" content="IE=edge"/></code>
7	<code><!-- タイトル名を設定する --></code>
8	<code><title>サンプル</title></code>
9	<code><link href="./css/UserWebStyle.css" rel="stylesheet" media="all" /></code>
10	<code><!-- ライブラリJavaScript読み込み（パスは格納位置に応じて変更する）--></code>
11	<code><script src="./FUserWebLib.js"></script></code>
12	<code><!-- 独自のJavaScriptはこれより後に記載する --></code>
13	<code><!-- ユーザのJavaScriptをここに書く --></code>
14	<code><script></code>
⋮	<code>⋮</code>
	<code></head></code>

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

Style Sheet ファイルの場合

▶ ファイル名

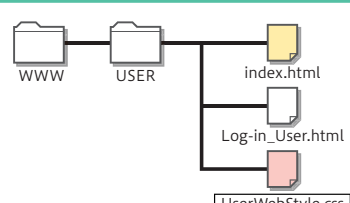
ファイル名	機能	内容	初期格納先
UserWebStyle.css	スタイルの設定	スタイルを共通定義したファイル	./css/UserWebStyle.css

▶ HTML 文書

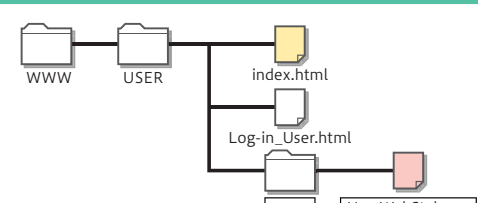
HTML ファイル内から CSS ファイルの読み込み方法を説明します。

ファイルの格納場所により、パスを変更する必要があります。

▶ “index.html” ファイルと “UserWebStyle.css” ファイルが同じ階層にある場合

HTML 文書	SD メモリカード内のフォルダ構造
<pre><script src="./UserWebStyle.css"></script></pre>	

▶ css フォルダに “UserWebStyle.css” ファイルが格納されている場合 (初期格納先)

HTML 文書	SD メモリカード内のフォルダ構造
<pre><script src="./css/UserWebStyle.css"></script></pre>	

▶ HTML ファイルへの記載方法

`<head>～</head>` 間に作成しますが、JavaScript 開始行 `<script>` より上に作成します。

行番号	HTML
1	<code><!DOCTYPE html></code>
2	<code><html xmlns="http://www.w3.org/1999/xhtml"></code>
3	<code><head></code>
4	<code><!-- charsetの設定※Webサーバ設定がUTF-8のためUTF-8を設定する --></code>
5	<code><meta charset="UTF-8"></code>
6	<code><meta http-equiv="X-UA-Compatible" content="IE=edge"/></code>
7	<code><!-- タイトル名を設定する --></code>
8	<code><title>サンプル</title></code>
9	<code><link href="./css/UserWebStyle.css" rel="stylesheet" media="all" /></code>
10	<code><!-- ライブラリJavaScript読み込み (パスは格納位置に応じて変更する) --></code>
11	<code><script src="./FUserWebLib.js"></script></code>
12	<code><!-- 独自のJavaScriptはこれより後に記載する --></code>
13	<code><!-- ユーザのJavaScriptをここに書く --></code>
14	<code><script></code>
⋮	⋮
	<code></head></code>

3. カスタマイズで簡単にユーザーWebページを作成しよう

はじめる前に

用意するもの

流用方法

ユーザーWebページ作成

ファイル保存先

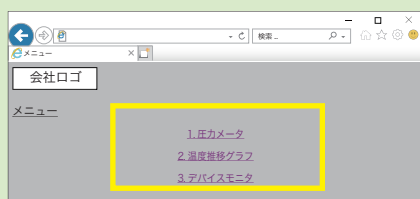
トラブルシューティング

3.6 トラブルシューティング

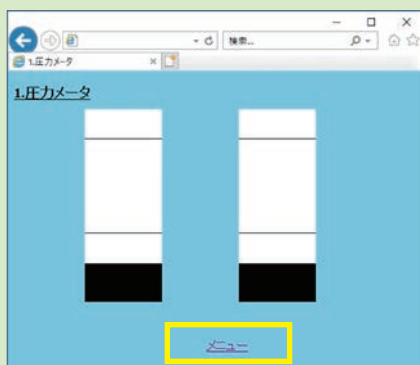
トラブル内容

Web ページ切り替えボタンのスタイルが文字と下線で表示され、ボタン枠がない。

[例 1]



[例 2]



チェックポイント

Style Sheet ファイルが読み込まれていません。
次の内容について確認してください。

- Style Sheet ファイルの格納場所が間違っている
- ファイル名が間違っている

[例 1] css フォルダ内に Style Sheet ファイルを格納している場合

誤 `<link href="/cssss/button-menu.css" rel="stylesheet" media="all"/>`
正 `<link href="/css/button-menu.css" rel="stylesheet" media="all"/>`

[例 2] css フォルダ内に Style Sheet ファイルを格納している場合

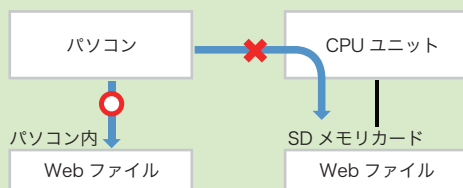
誤 `<link href="button-back.css" rel="stylesheet" media="all" />`
正 `<link href="/css/button-back.css" rel="stylesheet" media="all" />`

- Style Sheet の class 属性の名称が、HTML のボタン記述と Style Sheet ファイルの記述で一致していない。

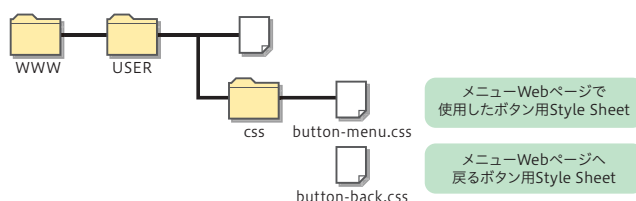
[例 1]

誤 HTML `<a href="level.html" class="button1">1.圧力メータ</a>`
Style Sheet `a.button {`
正 HTML `<a href="level.html" class="button1">1.圧力メータ</a>`
Style Sheet `a.button1 {`

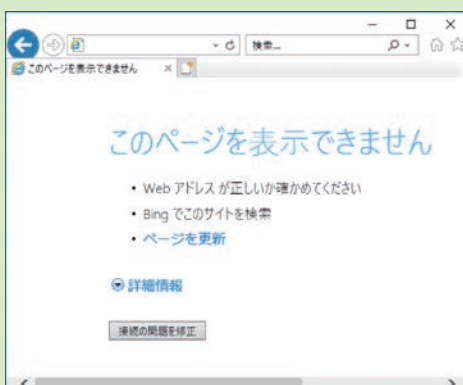
パソコン内の HTML ファイルは、Web ページ切り替えボタンのスタイルが正常に Web ブラウザで表示できる。しかし、シーケンサ側の Web ページは正常に表示できない。



Style Sheet ファイル (button-menu.css、button-back.css) が SD メモリカードに格納されているか確認してください。



Web ブラウザに Web ページが表示されず、エラー表示になる。



HTML ファイルから各 HTML ファイルが参照できていません。
次の内容について確認してください。

- ファイル名が間違っている
- リンク先 HTML ファイルの格納場所が間違っている

例 各 HTML ファイルは同じ場所に格納されている場合

`<a href="level.htm" class="button1">1.圧力メータ</a>`
→ファイル拡張子の間違い
`<a href="/.user/historical.html" class="button2">2.温度推移グラフ</a>`
→格納場所の指定間違い
`<a href="device.html" class="button3">3.デバイスモニタ</a>`
→device.htmlの後ろに「.」が無い

3. カスタマイズで簡単にユーザ Web ページを作成しよう

はじめる前に

用意するもの

流用方法

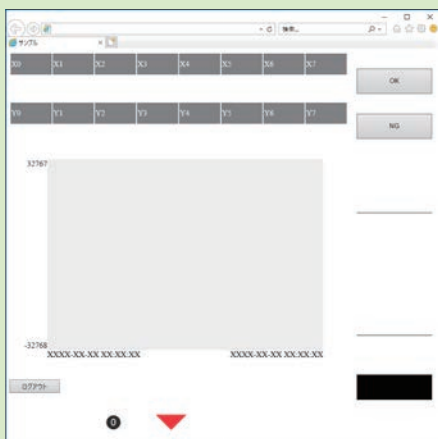
ユーザ Web ページ作成

ファイル保存先

トラブルシューティング

トラブル内容

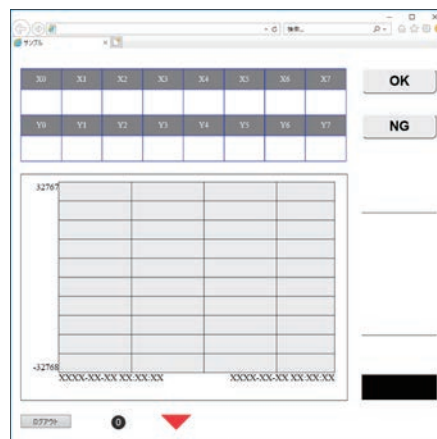
表やグラフの線が表示されない。また OK、NG ボタンなどのデザインが Web ブラウザの標準イメージで表示されている。



チェックポイント

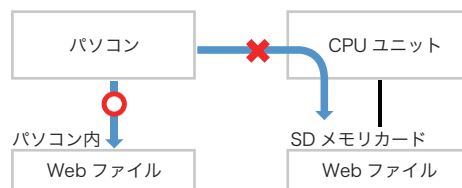
HTML ファイルから Style Sheet ファイル (UserWebStyle.css) が参照できていません。

[正しい状態]



下記の原因が考えられます。

- (1) HTML にリンク指定 `<link href="ファイル名" rel="stylesheet" media="all" />` の記載がない。
 - ・ HTML
`<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />`
- (2) HTML のリンク指定 `<link href="ファイル名" rel="stylesheet" media="all" />` の Style Sheet ファイル名が一致していない。
 - ・ HTML
`<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />`
 - ・ Style Sheet ファイル名 [UserWebStyle.css]
- (3) HTML のリンク指定で Style Sheet ファイルの格納場所を間違っている。
 - ・ HTML
`<link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />`
 - ・ 保存先 [www/css フォルダ内]
- (4) パソコン内の HTML ファイルを Web ブラウザで表示できるが、シーケンサ側の Web ページを正常に表示できない場合は、Style Sheet ファイル (UserWebStyle.css) が SD メモリカードに格納されているか確認してください。



4. Style Sheet (CSS) の働きを知る

Style Sheetとは

共通事項

全体構造

設定内容

Style Sheetの内容

4.1 機能紹介

Style Sheetとは、Webページのデザインやレイアウトなどのスタイルを変更するための仕組みです。HTMLがWebページ内の要素や構造を定義するのに対し、Style Sheetではそれらをどのように装飾するかを指定します。

HTMLやXHTMLなどで作成されるWebページにスタイルを指定する場合、Style Sheet言語の1つであるCSS (Cascading Style Sheets) が一般的に利用されているため、単にStyle Sheetと言えば「CSS」と指すことがあります。

▶ Style Sheetを使用していないWebページ

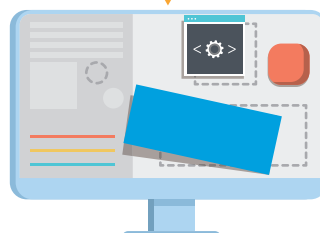
Style Sheetを使用せずにHTMLのみでWebページを作成すると、文字と画像が上から下に並んだ単調なページになります。HTMLはブラウザに表示させるテキスト文章を記載します。



HTMLでWebページの土台を作成する

▶ Style Sheetを使用しているWebページ

HTMLで作成されたWebページの土台に対して、Style Sheetで「デザイン」や「レイアウト」を制御します。Style Sheetにより文字や背景色の変更、多彩な線を引く、余白の調整など見栄えを整えることができます。



Style Sheetで、色・線・レイアウトを変更できる

Point

HTMLファイルでブラウザに表示するテキストの見栄えをCSSファイルで整えます。テキストと見栄えのファイルを分けることで、それぞれの内容に集中して記載でき、複数人での作業分担がしやすくなるなどのメリットがあります。

▶ Style Sheetでできることの一例

- ・ HTMLのみで作成した場合

HTMLコード

```
<p>これは例文です。</p>
```

ブラウザ表示

これは例文です。

- ・ 文字の色をオレンジに変更します。

HTMLコード

```
<p>これは例文です。</p>
```

CSSコード

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

ブラウザ表示

これは例文です。

- ・ 文字のサイズを20pxに変更します。

HTMLコード

```
<p>これは例文です。</p>
```

CSSコード

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

ブラウザ表示

これは例文です。

4. Style Sheet (CSS) の働きを知る

Style Sheetとは

共通事項

全体構造

設定内容

Style Sheetの内容

4.2 共通事項

Style Sheetを使用するうえで、知っておく必要がある共通事項について説明します。

用語説明

項目内容	内容
セレクト	Style Sheetを反映する対象
プロパティ	反映するスタイルの名称
プロパティの値	反映するスタイルの内容

Style Sheet記述ルール

Style Sheetの記述はセレクト、プロパティ、プロパティの値で構成します。

セレクト { プロパティ : プロパティの値 ; } 何に 何を どうする 宣言	例 body { color : #ff0000 ; }
--	---

- ・「プロパティ: プロパティの値」は **宣言** と呼びます。
- ・セレクトに対する宣言は中括弧 **{ }** で囲みます。
- ・宣言は **;** で区切り、複数指定できます。

Style Sheet 設定の種類

Style Sheetの設定方法は3種類あります。

▶ <link>タグでCSSファイルからStyle Sheetを読み出す

Style Sheetを拡張子が「.css」のファイルに記載し、HTMLファイルからCSSファイルのリンクをすることでStyle Sheetを読み出します。HTMLファイルからCSSファイルをリンクするには<head>タグ内で<link>タグを使用します。

例 ユーザWebページのライブラリ“UserWebStyle.css”をリンクする場合

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

▶ <style>タグでStyle Sheetを記述する

HTML文書の<head>タグ内で<style>タグを使用してStyle Sheetを記述します。

例 bodyの文字色を赤にするStyle Sheetを追加する場合

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

▶ 要素にStyle Sheetを追加する

要素にStyle属性を追加してStyle Sheetを直接記述します。要素に直接記述する場合、セレクトは不要です。

例 <h1> 見出し 1</h1>にStyle Sheetを追加する場合

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


4. Style Sheet (CSS) の働きを知る

Style Sheetとは

共通事項

全体構造

設定内容

Style Sheetの内容

Style Sheet設定例

<style>タグでStyle Sheetを記述する設定例を示します。

body (セクタ) の文字色 (プロパティ) を赤 (プロパティの値) にする場合、Style Sheetの記述は下記になります。

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

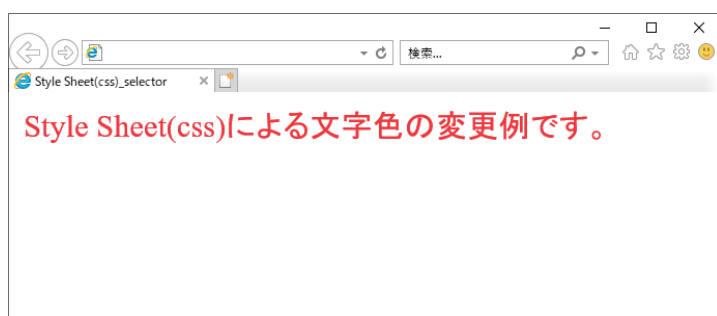
上記のStyle Sheetは<body>～</body>の間にあるテキストの色を「#ff0000 (赤)」に変更します。

▶ HTML例

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charsetの設定※Webサーバ設定がUTF-8のためUTF-8を設定する -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- タイトル名を設定する -->
    <title>Style Sheet (css)_selector</title>
    <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
    <meta http-equiv="Content-Style-Type" content="text/css">

    <!-- Web ページの<body>～</body>間にあるテキストの文字色指定 -->
    <style>
      body { color:#ff0000; }
    </style>
  </head>
  <body>
    Style Sheet (css) による文字色の指定例です。
  </body>
</html>
```

▶ Webブラウザ表示



4. Style Sheet (CSS) の働きを知る

Style Sheet とは

共通事項

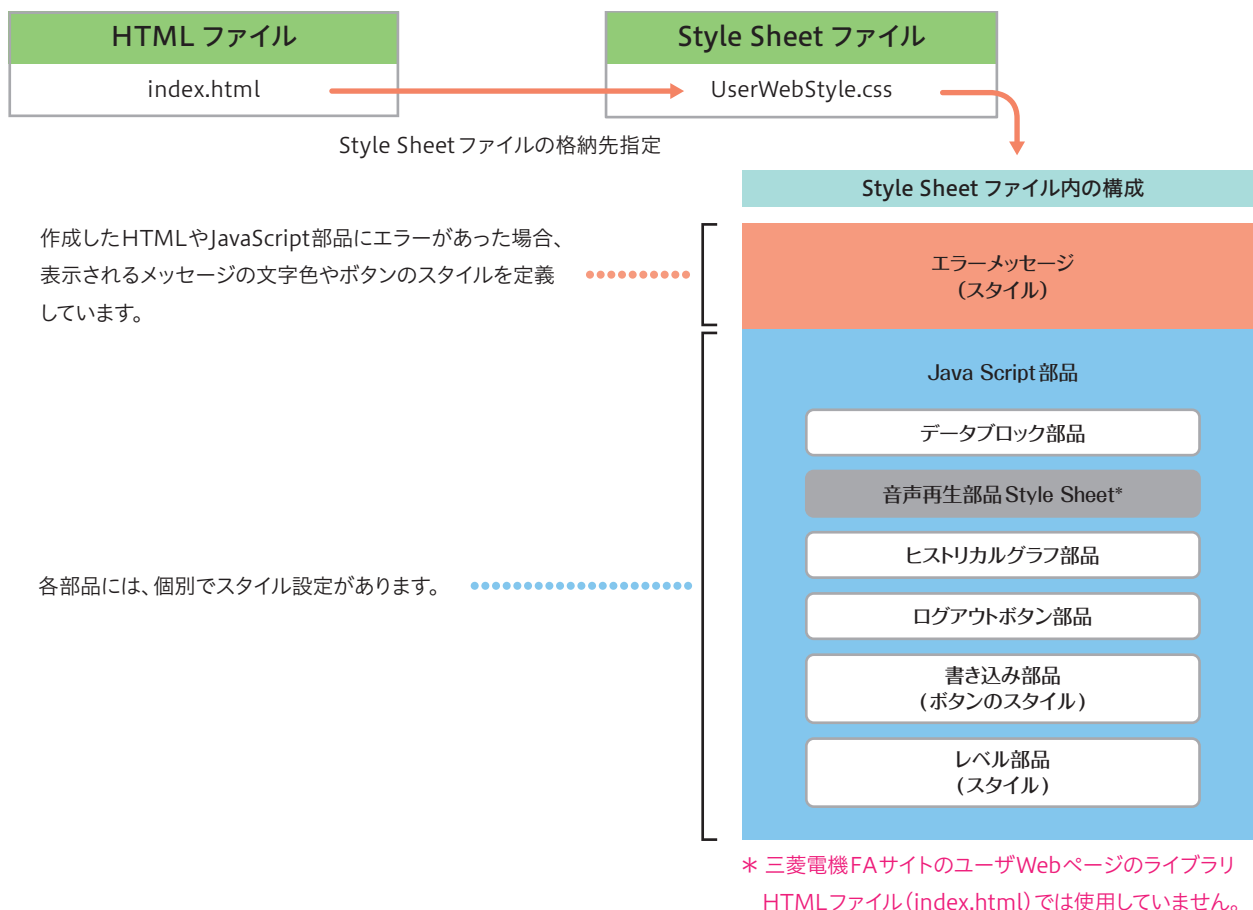
全体構造

設定内容

Style Sheet の内容

4.3 全体構造 (UserWebStyle.css ファイル)

HTML 文書の <link> タグで、Style Sheet ファイルを読み出しています。



4.4 設定内容

各部品のスタイルは、JavaScript で設定した各部品のスタイルが優先され、Style Sheet (UserWebStyle.css) で設定を変更しても反映されません。ただし、JavaScript で特定の設定を省略した場合、Style Sheet の設定が反映されます。

関連ページ

詳細は下記を参照してください。
5.1 JavaScript (JS) とは - Style Sheet (CSS)

特記事項

ユーザ Web ページのライブラリ内にある Style Sheet (UserWebStyle.css) を変更する場合、Style Sheet のクラス名称は変更しないでください。また、同じ名称の Style Sheet のクラス名称は定義しないでください。

4. Style Sheet (CSS) の働きを知る

Style Sheetとは | 共通事項 | 全体構造 | 設定内容 | Style Sheetの内容

4.5 Style Sheetの内容

ユーザWebページのライブラリ内のStyle Sheetファイル (UserWebStyle.css) の内容について説明します。

関連ページ

Style Sheetの詳細は、下記を参照してください。
7.1 Style Sheetリファレンス

注意事項

Style Sheetファイルを変更する場合、パラメータや設定値に間違った設定をすると、表示部品はWebブラウザで表示できなくなります。必ず変更前のファイルを保存してから、変更を行ってください。

エラーメッセージ用スタイル

▶ エラーメッセージ表示例

エラーメッセージ表示ボタン

エラーメッセージ

エラーメッセージ表示フィールド

NG

パラメータ "dev[0].base" の設定内容が間違っています。
パラメータ "dev[1].base" の設定内容が間違っています。
パラメータ "dev[2].base" の設定内容が間違っています。
パラメータ "dev[3].base" の設定内容が間違っています。
パラメータ "dev[4].base" の設定内容が間違っています。
パラメータ "dev[5].base" の設定内容が間違っています。
パラメータ "dev[6].base" の設定内容が間違っています。
パラメータ "dev[7].base" の設定内容が間違っています。
デバイス "D0" のデータ形式が不正です。
デバイス "D1" のデータ形式が不正です。
デバイス "D2" のデータ形式が不正です。
デバイス "D3" のデータ形式が不正です。
デバイス "D4" のデータ形式が不正です。
デバイス "D5" のデータ形式が不正です。
デバイス "D6" のデータ形式が不正です。
デバイス "D7" のデータ形式が不正です。
パラメータ "dev" の設定内容が間違っています。

デバイス "D0" のデータ形式が不正です。

パラメータ "direction" の設定内容が間違っています。

-32768

XXXX-XX-XX XX:XX:XX

XXXX-XX-XX XX:XX:XX

123

ログアウト

0

4. Style Sheet (CSS) の働きを知る

Style Sheetとは

共通事項

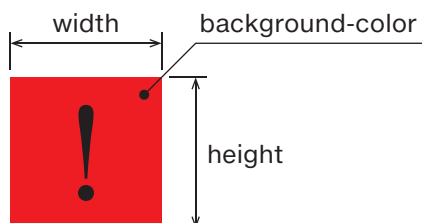
全体構造

設定内容

Style Sheetの内容

▶ パラメーター一覧

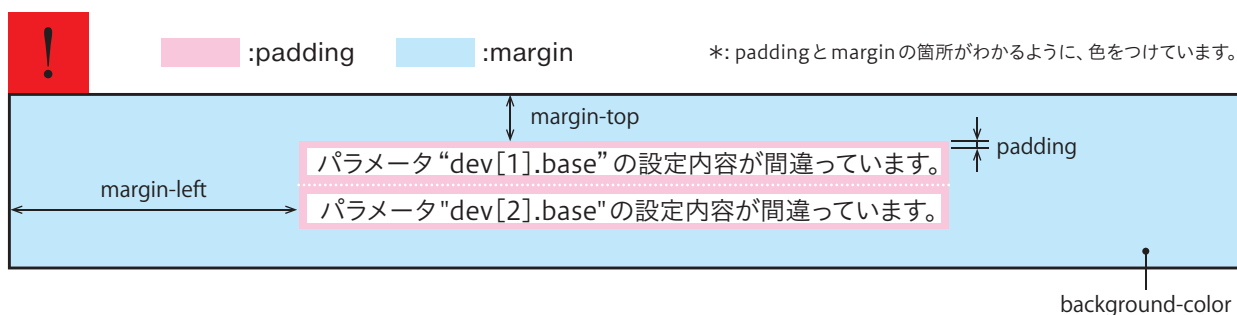
・エラーメッセージ表示ボタン



・カーソル



・エラーメッセージ, エラーメッセージ表示フィールド



4. Style Sheet (CSS) の働きを知る

Style Sheetとは

共通事項

全体構造

設定内容

Style Sheetの内容

► Style Sheet

行番号	Style Sheet 文	解説
共通設定		表示開始位置の扱い指定
1	.partsBase{	セレクト
2	position: absolute;	親ボックスの絶対位置の配置
3	}	
グローバルメッセージ		グローバルメッセージのボックスの設定
4	.globalMessageArea{	セレクト
5	position: absolute;	親ボックスの絶対位置の配置
6	top: 2px;	基準位置からボックス上端までの距離
7	left: 2px;	基準位置からボックス左端までの距離
8	z-index: 2147483647;	ボックスの重なるの順序を指定
9	background-color: white;	背景色
10	border: solid 1px black;	親ボックスの枠線指定 設定値 [太さ:1pixel、線種:実線、線色:黒]
11	}	
エラーメッセージ		エラーメッセージ一覧の行間
12	.errorMessage{	セレクト
13	margin: 0;	外側余白指定 (上下左右)
14	padding: 1px 0;	内側余白指定 (上下:1pixel、左右:0pixel)
15	}	
エラーメッセージ表示フィールド		エラーメッセージ一覧を表示するウインドウ
16	.errorField{	セレクト
17	position: absolute;	親ボックスの絶対位置の配置
18	z-index: 2147483646;	ボックスの重なるの順序を指定
19	margin-top: 22px; /* errorButtonのheight+margin-top */	外側余白指定 (上)
20	margin-left: 2px; /* errorButtonのmargin-leftと同値 */	外側余白指定 (左)
21	background-color: white;	背景色
22	border: solid 1px black;	エラーメッセージ表示フィールドの枠線指定 設定値 [太さ:1pixel、線種:実線、線色:黒]
23	}	
表示ボタン		エラーメッセージ一覧を表示するボタン
24	.errorButton{	セレクト
25	position: absolute;	親ボックスの絶対位置の配置
26	z-index: 2147483645;	ボックスの重なるの順序を指定
27	width: 20px;	横幅
28	height: 20px;	高さ
29	margin-top: 2px;	外側余白指定 (上)
30	margin-left: 2px;	外側余白指定 (左)
31	background-color: red;	背景色
32	border: solid 1px white;	表示ボタンの枠線指定 設定値 [太さ:1pixel、線種:実線、線色:白]
33	text-align: center;	テキストの中央揃え
34	font-weight: bold;	文字修飾の種類指定 設定値 [強調]
35	cursor: pointer;	マウスカーソルの形状を指定
36	}	

4. Style Sheet (CSS) の働きを知る

Style Sheetとは

共通事項

全体構造

設定内容

Style Sheetの内容

データブロック部品用スタイル

▶ パラメーター一覧

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

background-color:

color:

border:

▶ Style Sheet

JavaScriptの設定を優先表示

行番号	Style Sheet文	解説
37	/* ----- */	
38	/* データブロック部品 */	「/*」「*/」で囲まれた部分はコメントとして扱われます。
39	/* ----- */	
40	.datblk,	
41	.datblk_name,	セレクト
42	.datblk_val{	
43	border: 1px solid black;	データセルの枠線指定 設定値[太さ:1pixel、線種:実線、線色:黒]
44	padding: 0;	内側余白指定(上下左右)
45	text-align: center;	文字揃え設定(中央)
46	color: black;	データセルの文字色指定
47	background-color: white;	データセルの背景色指定
48	}	
49	.datblk{	セレクト
50	border-collapse: collapse;	データセルの線表示方法を指定 設定値[隣接するセルの線を重ねて表示]
51	white-space: nowrap;	ホワイトスペース(連続する半角スペース・タブ)や改行の表示方法を指定 設定値[改行は半角スペース、連続する複数のスペースは一つのスペースとして扱う]
52	}	
53	.datblk_name,	セレクト
54	.datblk_val{	
55	overflow: hidden;	セルからはみ出した文字の処理方法 設定値[表示しない]
56	}	

音声再生部品用スタイル

本機能には対応していません。

行番号	Style Sheet文	解説
57	/* ----- */	
58	/* 音声再生部品 */	「/*」「*/」で囲まれた部分はコメントとして扱われます。
59	/* ----- */	
60	.audio_btn{	セレクト
61	padding: 0;	
62	text-align: center;	
63	color: black;	
64	overflow: hidden;	
65	}	

4. Style Sheet (CSS) の働きを知る

Style Sheet とは

共通事項

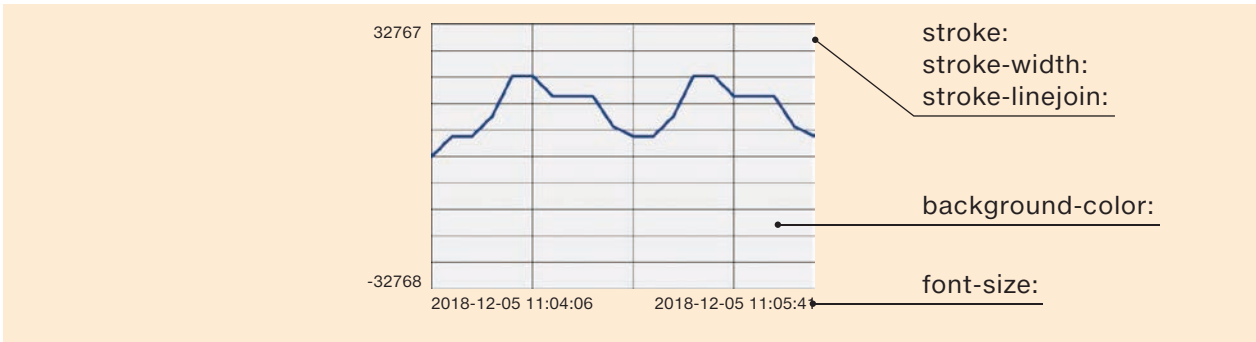
全体構造

設定内容

Style Sheet の内容

ヒストリカルグラフ部品用スタイル

パラメーター一覧



Style Sheet

行番号	Style Sheet 文	解説
66	<code>/* ----- */</code>	
67	<code>/* ヒストリカルグラフ部品 */</code>	「/」「*/」で囲まれた部分はコメントとして扱われます。
68	<code>/* ----- */</code>	
69	<code>.hstgrp{</code>	セレクト
70	<code>border: 1px solid black;</code>	未使用
71	<code>background-color: white;</code>	
72	<code>}</code>	
73	<code>.hstgrp_frame{</code>	セレクト
74	<code>stroke: black;</code>	線色指定
75	<code>stroke-width: 1px;</code>	線の太さ指定
76	<code>stroke-linejoin: miter;</code>	線の繋ぎや角の形状指定
77	<code>}</code>	
78	<code>.hstgrp_text_x,</code>	セレクト
79	<code>.hstgrp_text_y{</code>	
80	<code>font-size: 18px;</code>	文字サイズの指定
81	<code>}</code>	
82	<code>.hstgrp_line{</code>	セレクト
83	<code>stroke: blue;</code>	
84	<code>stroke-width: 1px;</code>	未使用
85	<code>stroke-linejoin: round;</code>	
86	<code>}</code>	

4. Style Sheet (CSS) の働きを知る

Style Sheet とは

共通事項

全体構造

設定内容

Style Sheet の内容

ログアウトボタン部品用スタイル

▶ パラメーター一覧

ログアウト color:

▶ Style Sheet

行番号	Style Sheet 文	解説
87	/* ----- */	
88	/* ログアウトボタン部品 */	「/*」「*/」で囲まれた部分はコメントとして扱われます。
89	/* ----- */	
90	.logout_btn{	セレクト
91	padding: 0;	内側余白指定 (上下左右)
92	text-align: center;	ボタン内の文字の揃え指定
93	color: black;	文字色指定
94	overflow: hidden;	セルからはみ出した文字の処理方法の指定 設定値 [表示しない]
95	}	

書き込み部品 (ボタンのスタイル)

▶ パラメーター一覧

設定

▶ Style Sheet

行番号	Style Sheet 文	解説
96	/* ----- */	
97	/* 書き込み部品 (ボタンのスタイル) */	「/*」「*/」で囲まれた部分はコメントとして扱われます。
98	/* ----- */	
99	.write_btn{	セレクト
100	height:50px;	未使用
101	width:100px;	
102	font-weight:bold;	文字修飾の種類指定 設定値 [強調]
103	font-family:Meiryo;	文字フォントの種類指定 設定値 [メイリオ]
104	font-size:20pt;	文字のサイズ指定
105	border-radius:10px	
106	}	

4. Style Sheet (CSS) の働きを知る

Style Sheetとは

共通事項

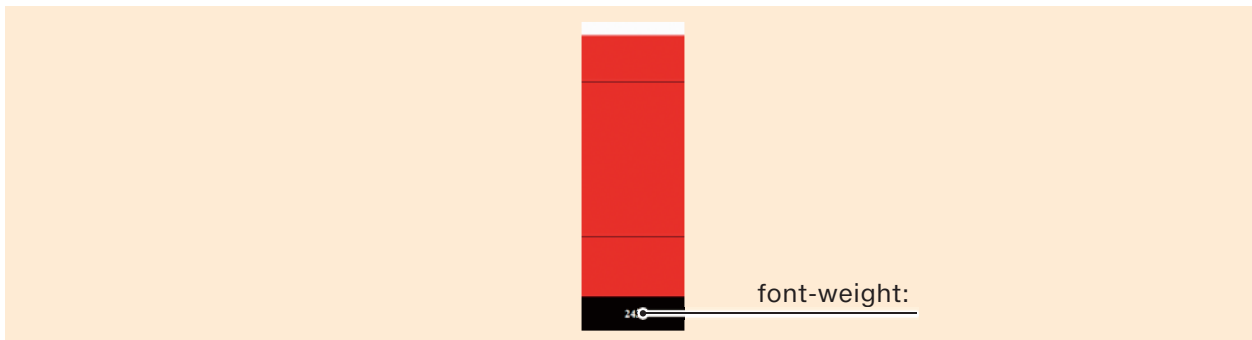
全体構造

設定内容

Style Sheetの内容

レベル部品 (スタイル)

▶ パラメーター一覧



▶ Style Sheet

行番号	Style Sheet 文	解説
107	<code>/* ----- */</code>	
108	<code>/* レベル部品 (スタイル) */</code>	「/*」「*/」で囲まれた部分はコメントとして扱われます。
109	<code>/* ----- */</code>	
110	<code>.partsLevelValue{</code>	セレクタ
111	<code>font-weight:bold;</code>	文字修飾の種類指定 設定値 [強調]
112	<code>}</code>	

5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

5.1 JavaScript(JS) とは

JavaScriptとは、Webページに動きをつけるための言語です。写真のスライド表示や、2D/3Dグラフィックのアニメーションなど、インタラクティブな動作が実現できます。

三菱電機FAサイトで公開しているユーザWebページのライブラリをダウンロードすることで、JavaScript(JS) ツールを利用できます。JavaScriptでの部品ライブラリのパラメータについて説明します。本章では、一般的なJavaScriptの記述については記載しません。

部品一覧

ライブラリにあるJavaScript部品ライブラリを使用すると、ユーザWebページでデバイスの読み出し/書き込みができます。JavaScript部品ライブラリ (FUserWebLib.js) には、下表の部品が含まれています。

表

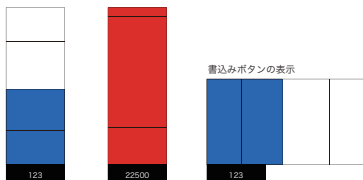
データブロック部品 (WSDatblk)

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

指定したデバイスデータを表形式で表示

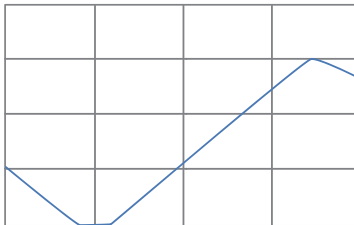
グラフ

レベル表示部品 (WSLevel)



デバイス値に対する割合を棒グラフで表示

ヒストリカルグラフ部品 (WSHstgrp)



デバイスデータを時系列の折れ線グラフで表示

画像

画像表示部品 (WSPicture)



デバイス値が設定した値になると、指定した画像を表示

図形表示部品 (WSFigure)



デバイス値が設定した値になると、指定した図形を表示

操作ボタン

書き込みボタン部品 (WSWrtBtn)

OK *

→X0に1を書き込む

NG *

→X0に0を書き込む

指定したデバイスに指定した値を書き込み

*: ボタン上に表示するテキストは変更可能

ログアウトボタン部品 (WSLogoutBtn)

ログアウト

ログアウト操作

5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

使用可能ファイル

下記のファイルが使用できます。

ファイル	拡張子	MIMEタイプ
HTML	.html	text/html
	.htm	text/htm
JavaScript	.js	text/javascript
CSS	.css	text/css
GIF画像	.gif	image/gif
PNG画像	.png	image/png
JPG/JPEG画像	.jpgx/.jpeg	image/jpeg

用語説明

Webにおけるファイルの拡張子とMIMEタイプの関係

Web上には「拡張子」と「MIME (マيلم) タイプ」という概念があります。

いずれもファイルタイプがテキストかHTMLか、画像が何形式のファイルかを知らせるための仕組みです。

・「拡張子」はそのファイルが何であるかを示します。ファイル名の最後、ドット“.”から後ろの部分です。

・「MIMEタイプ」は“タイプ名/サブタイプ名”の形式の文字列で、WebサーバとWebブラウザの間はこの「MIMEタイプ」を用いてデータ形式を指定しています。Webブラウザが“xxx.png”ファイルをWebサーバに要求する際、Webサーバは“xxx.png”の内容(データ)を「これは image/png タイプのデータです」と返答します。これにより、Webブラウザは受け取ったデータを正常に処理できます。

・MIMEタイプは、HTML文書に「Content-Type」を記述し指定します。

例 <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<meta http-equiv="Content-Style-Type" content="text/css">

Style Sheet (CSS)

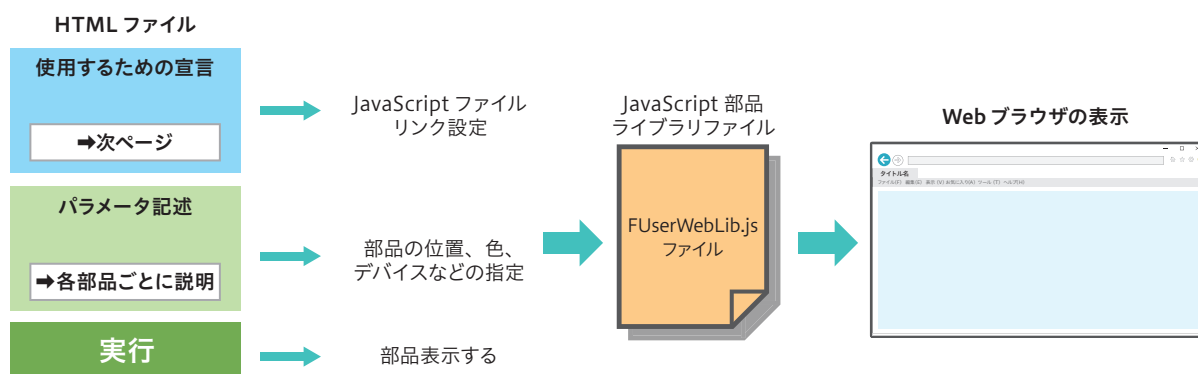
ユーザWebページの文字やボタン、グラフのスタイルは、記述が省略できます。記述を省略した場合や記述が誤っている場合、Style Sheet (UserWebStyle.css) に従った表示となります。文字フォントは、ユーザWebページを表示する端末のOSにより異なります。

項目	デフォルト
文字色	Black
背景色	White
線色	Black
レベル表示塗りつぶし色	Blue
グラフ線色	Blue
文字サイズ	20 (pixel単位)
ボタン文字色	Webブラウザの設定に従います。
ボタン背景色	Webブラウザの設定に従います。
ボタン線色	Webブラウザの設定に従います。

関連ページ 4.Style Sheet (CSS) の働きを知る

JavaScriptの記述構造と処理の流れ

JavaScriptの記述で役割別に分類すると次のとおりとなります。各部品の説明ページで役割によって色分けしています。



5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

5.2 共通事項

JavaScriptの各部品を使用するうえで、知っておく必要がある共通事項について説明します。

JavaScript記述ルール

表示部品名に続く、パラメータは下記のように構成します。

```
表示部品名 {  
  パラメータ名 : スペース 値(設定) ;  
}
```

パラメータ設定

例

```
.partsBase {  
  position : absolute ;  
}
```

- ・パラメータは中括弧【】で囲みます。
- ・文字列の場合は「'」または「"」で囲みます。「'」と「"」のどちらも使用可能ですが、前後とも同じ記号にしてください。
- ・パラメータ設定は、パラメータ名 + : + スペース + 値(設定) で構成します。

パラメータ指定

表示部品名に続く、パラメータ指定を説明します。

内容		デフォルト
指定省略	設定の要否が任意のパラメータを省略した場合	デフォルトの設定で表示します。
誤指定	任意のパラメータの設定が異常な場合	
データ形式	指定されたデータ形式以外で設定した場合	例えば数値で設定するパラメータに“XYZ”などの文字列を設定した場合は、パラメータ異常となります。各パラメータは指定されたデータ形式で設定してください。
座標指定	X座標、Y座標を省略した場合	部品は左上端(座標0,0)に配置します。
カラー指定	RGB値やカラー名の範囲チェック	HTML仕様ではRGB値、またはカラー名などの範囲チェックが行われません。正しくない設定をした場合は、Webブラウザの種類によって表示される内容が異なります。
デバイス指定	U□¥G□のデバイス名を指定する時	U□¥¥G□と「¥」を重ねて指定してください。(1つ目の¥はエスケープシーケンス*です。)

*エスケープシーケンスとは: 文字そのものを出力するのではなく、文字色の変更やカーソルの移動、文字の消去など、文字出力の制御を行う特殊な文字列のことです。

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptと3 | 共通事項 | 表 | グラフ | 画像・図形 | 操作ボタン

設定できるデバイス

JavaScript部品に設定できるデバイスの設定可能な形式と進数を下記に示します。

○: 設定可 ×: 設定不可

デバイス名		種別	ビット	ワード16bit				ワード32bit				単精度実数(K)
				符号付き		符号なし		符号付き		符号なし		
				10進数(K)	16進数(H)	10進数(K)	16進数(H)	10進数(K)	16進数(H)	10進数(K)	16進数(H)	
ユーザデバイス												
入力(X)		ビット	○									
出力(Y)		ビット	○									
内部リレー(M)		ビット	○									
ラッチリレー(L)		ビット	○									
リンクリレー(B)		ビット	○									
アナンシェータ(F)		ビット	○									
リンク特殊リレー(SB)		ビット	○									
ステップリレー(S)		ビット	○									
タイマ(T)*1	接点: TS	ビット	○									
	コイル: TC	ビット	○									
	現在値: TN	ワード		○	○	×	×	×	×	×	×	×
積算タイマ(ST)*1	接点: STS	ビット	○									
	コイル: STC	ビット	○									
	現在値: STN	ワード		○	○	×	×	×	×	×	×	×
カウンタ(C)*1	接点: CS	ビット	○									
	コイル: CC	ビット	○									
	現在値: CN	ワード		○	○	×	×	×	×	×	×	×
ロングカウンタ(LC)*1	接点: LCS	ビット	○									
	コイル: LCC	ビット	○									
	現在値: LCN	ダブルワード						○	○	×	×	×
データレジスタ(D)		ワード		○	○	○	○	○	○	○	○	○
リンクレジスタ(W)		ワード		○	○	○	○	○	○	○	○	○
リンク特殊レジスタ(SW)		ワード		○	○	○	○	○	○	○	○	○
システムデバイス												
特殊リレー(SM)		ビット	○									
特殊レジスタ(SD)		ワード		○	○	○	○	○	○	○	○	○
ユニットアクセスデバイス(U□¥G□)*2												
ユニットアクセスデバイス(G)		ワード		○	○	○	○	○	○	○	○	○
インデックスレジスタ												
インデックスレジスタ(Z)		ワード		○	○	○	○	○	○	○	○	○
ロングインデックスレジスタ(LZ)		ダブルワード						○	○	○	○	○
ファイルレジスタ												
ファイルレジスタ(R)		ワード		○	○	○	○	○	○	○	○	○

*1: T、ST、C、LCを指定した場合、現在値のデバイス(TN、STN、CN、LCN)として扱われます。
*2: U□¥G□のような¥付きのデバイス名を指定するときは、U□¥¥G□と「¥」を重ねて指定してください。
1つ目の¥(エスケープシーケンス) が特殊文字を意味する記号のため、このような表現が必要になります。

5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

デバイス設定方法

対応表を参照して、デバイス設定をします。

JavaScriptパラメータ名

devName: 任意の値 ① デバイス名 (デバイス名 + デバイス番号)
 devBase: 任意の値 ② データ進数
 devFormat: 任意の値 ③ データ形式

▶ 設定例

・ 入力X7を指定する場合

devName: X7 入力 X7
 devBase: B 2進数
 devFormat: 6 ビット

・ タイマT25 (現在値) を指定する場合

devName: TN25 タイマ T25
 devBase: K 10進数
 devFormat: 0 16bit符号付き

▶ 対応表

デバイス名	種別	JavaScriptパラメータ名			
		① デバイス名	② データ進数	③ データ形式	
		devName:	devBase:	devFormat:	
				② K	② H
ユーザデバイス					
入力 (X)	ビット	X	B	6	
出力 (Y)	ビット	Y	B	6	
内部リレー (M)	ビット	M	B	6	
ラッチリレー (L)	ビット	L	B	6	
リンクリレー (B)	ビット	B	B	6	
アナンシェータ (F)	ビット	F	B	6	
リンク特殊リレー (SB)	ビット	SB	B	6	
ステップリレー (S)	ビット	SB	B	6	
タイマ (T)	接点: TS	TS	B	6	
	コイル: TC	TC	B	6	
	現在値: TN	TN	K, H	0	1
積算タイマ (ST)	接点: STS	STS	B	6	
	コイル: STC	STC	B	6	
	現在値: STN	STN	K, H	0	1
カウンタ (C)	接点: CS	CS	B	6	
	コイル: CC	CC	B	6	
	現在値: CN	CN	K, H	0	1
ロングカウンタ (LC)	接点: LCS	LCS	B	6	
	コイル: LCC	LCC	B	6	
	現在値: LCN	LCN	K, H	1	3
データレジスタ (D)	ワード	D	K, H	0,2,4	1,3
リンクレジスタ (W)	ワード	W	K, H	0,2,4	1,3
リンク特殊レジスタ (SW)	ワード	SW	K, H	0,2,4	1,3
システムデバイス					
特殊リレー (SM)	ビット	SM	B	6	
特殊レジスタ (SD)	ワード	SD	K, H	0,2,4	1,3
ユニットアクセスデバイス (U□¥G□)					
ユニットアクセスデバイス (G)	ワード	G	K, H	0,2,4	1,3
インデックスレジスタ					
インデックスレジスタ (Z)	ワード	Z	K, H	0,2,4	1,3
ロングインデックスレジスタ (LZ)	ダブルワード	LZ	K, H	2	3
ファイルレジスタ					
ファイルレジスタ (R)	ワード	R	K, H	0,2,4	1,3

② データ進数

設定値	内容
K	10進数
H	16進数
B	2進数

③ データ形式

設定値	内容
0	16bit符号付き
1	16bit符号なし
2	32bit符号付き
3	32bit符号なし
4	単精度実数
5	未使用
6	ビット

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

更新周期

ユーザWebページの更新周期は、HTML上の変数updateIntervalに設定します。

この更新周期は、すべてのユーザWeb部品に適用されます。

updateInterval	設定範囲	HTML文書
記載なし	5秒周期(デフォルト)	省略
記載あり	5～120秒周期*	例: 10秒周期で更新設定 var updateInterval = 10;

*: updateIntervalが5～120秒周期の範囲外の場合、5秒周期で動作します。

メッセージ表示言語

メッセージ表示言語は、HTML上の変数dspLanguageに設定します。

変数設定とメッセージ表示言語について説明します。

dspLanguage*	設定範囲	HTML文書
記載なし	英語	省略
ja-JP	日本語	var dspLanguage = 'ja-JP';
en-US	英語	var dspLanguage = 'en-US';
zh-CN	中国語(簡体字)	var dspLanguage = 'zh-CN';

*: dspLanguageが範囲外の場合、表示言語は英語になります。

▶ HTMLファイルへの記載方法

<head>～</head>間にある、JavaScript開始行の<script>～</script>間に作成します。

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charsetの設定※Webサーバ設定がUTF-8のためUTF-8を設定する -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- タイトル名を設定する -->
    <title>サンプル</title>
    <link href="/css/UserWebStyle.css" rel="stylesheet" media="all" />
    <!-- ライブラリJavaScript読み込み(パスは格納位置に応じて変更する) -->
    <script src="/FUserWebLib.js"></script>
    <!-- 独自のJavaScriptはこれより後に記載する -->
    <!-- ユーザのJavaScriptをここに書く -->
    <script>
      var updateInterval = 5;      更新周期
      var dspLanguage = 'ja-JP';  メッセージ表示言語
      ...
    </script>
  </head>
  ...
</html>
```

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

5.3 表

データブロック部品 (WSDatblk)

▶ 機能

指定したデバイスデータを表形式で表示します。

▶ 作成例

方向: 横向き

行数: 2行

データ数: 8個

▶ ユーザWebページのライブラリ画面



部品の設計

注意事項

データ形式を単精度実数とする場合、データ進数は10進数になります。

▶ HTML (JavaScript) 構造

表示するデバイスの行、または列の個数は、下記のように表示デバイス指定のHTML文書を連続して作成します。

使用宣言

パラメータ (個別)

表示デバイス指定 (X0~X7)

パラメータ (共通)

データブロック (表組み) 設定

実行

X0	X1	X2	X3	X4	X5	X6	X7

使用宣言

パラメータ (個別)

表示デバイス指定 (X0~X7)

パラメータ (共通)

表示デバイス指定 (Y0~Y7)

データブロック (表組み) 設定

実行

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

5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

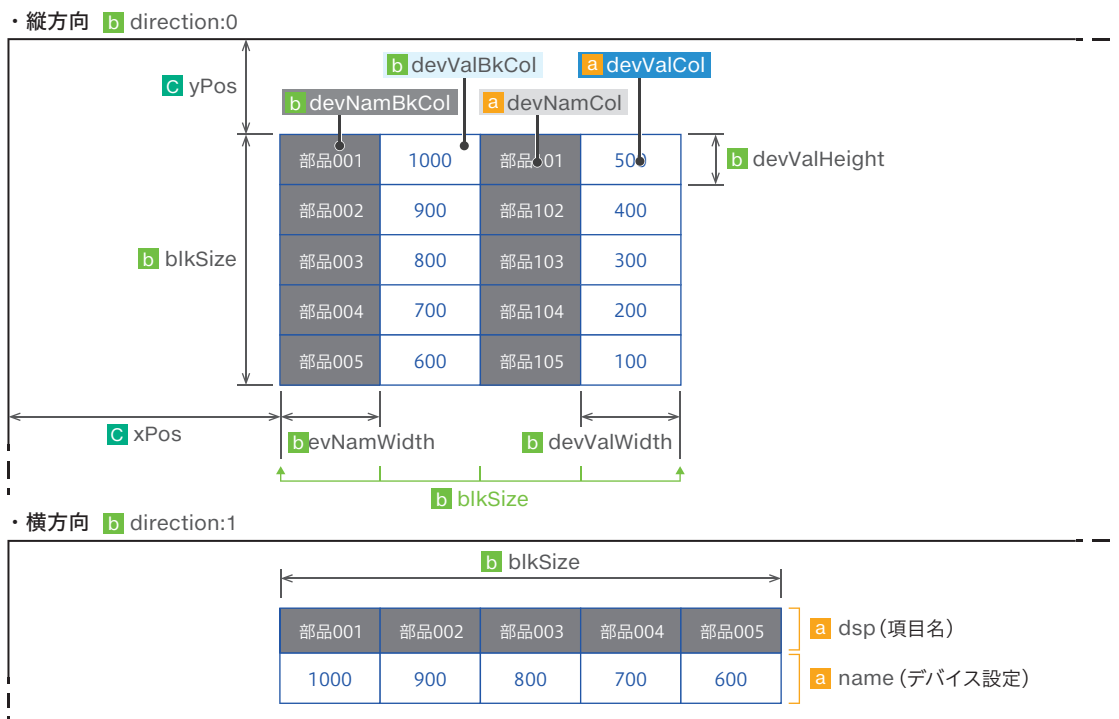
表

グラフ

画像・図形

操作ボタン

▶ パラメーター一覧



a 項目名とデバイス設定

上記パラメーター一覧のdevは、表の項目名とデバイス設定を行います。
JavaScriptの設定ではdsp:、name:、base:、format:に分けて設定します。

▶ 項目名称の設定… dsp:

任意の文字 (例: センサ) を表示できます。

下記例ではfor文で設定した変数“i”から始まる連番 (1~4) が'センサ'の後ろに付加します。

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

表示例
X1=ON、X2=OFF、X3=OFF、X4=ON の場合

センサ1	センサ2	センサ3	センサ4
1	0	0	1

▶ デバイス指定… name:

指定したデバイスの値 (例: X1~X4の値) を表示できます。

下記例では'X'がデバイス種別、for文で設定した変数“i”から始まる連番 (1~4) がデバイス番号となります。

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

表示例
X1=ON、X2=OFF、X3=OFF、X4=ON の場合

センサ1	センサ2	センサ3	センサ4
1	0	0	1

関連ページ 5.2 共通事項 - デバイス設定方法

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

▶ データ進数… base:

値	内容
K	10進数
H	16進数
B	2進数

関連ページ 5.2 共通事項－デバイス設定方法

▶ データ形式… format:

値	内容	値	内容
0	16bit符号付き	4	単精度実数
1	16bit符号なし	5	未使用
2	32bit符号付き	6	ビット
3	32bit符号なし		

関連ページ 5.2 共通事項－デバイス設定方法

▶ デバイス名(項目名称)の表示有無設定… devNamDisp:

Script	内容
devNamDisp: 0	表示しない
devNamDisp: 1	表示する(省略時)

表示例: デバイス名を表示しない

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

表示例: デバイス名を表示する

X0	X1	X2	X3
1	0	0	1

▶ 文字色指定… devNamCol: devValCol:

Script	内容	備考
devNamCol:	デバイス名の色指定	RGB値またはカラー名 例: #ff0000 (RGB値)、red (カラー名)
devValCol:	デバイス値の色指定	

b 表組み設定

▶ 表示方向指定… direction:

Script	内容
direction: 0	縦方向(省略時)
direction: 1	横方向

表示例: 縦方向

X0	1	X2	0
X1	0	X3	1

表示例: 横方向

X0	X1
1	0
X2	X3
0	1

▶ ブロックサイズ(横方向のセル数)指定… blkSize:

Script	内容	備考
blkSize:	セル数の指定	表示方向が縦方向の場合は行数、横方向の場合は列数が変動

5. JavaScript(JS) で表示できる部品働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

▶ セルのサイズ指定… devNamWidth: devNamHeight: devValWidth: devValHeight:

Script	内容	備考
devNamWidth:	デバイス名のセル	横幅
devNamHeight:		縦幅
devValWidth:	デバイス値のセル	横幅
devValHeight:		縦幅
		正の実数 (pixel単位)

▶ セルの色指定… devNamBkCol: devValBkCol:

Script	内容	備考
devNamBkCol:	デバイス名セルの場合のセルの背景色	RGB値またはカラー名
devValBkCol:	デバイス値セルの場合のセルの背景色	例: #FF0000 (RGB値)、red (カラー名)

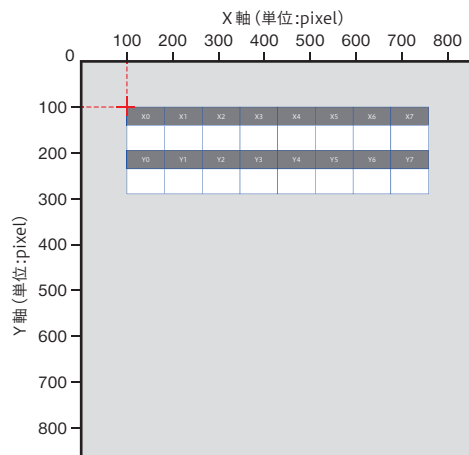
c 表示開始座標設定

▶ Webブラウザ表示開始位置設定… xPos: yPos:

Script	内容	備考
xPos:	X軸の座標	設定範囲: パソコンの画面解像度 (pixel単位)
yPos:	Y軸の座標	

▶ Webブラウザの座標の見方

例 (xPos, yPos) = (100, 100)



5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

▶ HTML文書の解説

パラメータの扱い

…パラメータ設定が必須。未設定の場合はエラーとなります。

…デバイス名表示を「表示しない」(devNamDisp = 0) を設定時は不要です。

…JavaScript で設定した内容です。

注意事項

必須パラメータはすべて記述してください。必須パラメータの記述がない場合、または設定値が範囲外の場合はエラーになります。

表示デバイス指定 (1,2行目)

行番号	JavaScript	解説				
1	// データブロック部品 (WSDatblk)	コメントを記述				
2	temp = [];	宣言 (ラベル名)				
1, 2 行目の表示デバイス設定						
3	for(var i = 0; i < 8; i++){	forの開始 (変数名 i) forの開始～forの終了までの行を繰り返し実行する。左記では繰り返し回数をカウントするための変数を i とする。() の中身は3つの文で構成されており、(繰り返し回数用変数の初期値; forの終了条件; 繰り返し回数用変数の演算方法)となる。繰り返し回数の演算方法は i++;forの終了まで実行したら1加算となる。 () 内は入力のデバイス番号に合わせた範囲となるように記述。表のデータ個数は blkSize で設定し、i をその数値の +1 に合わせる。				
4	temp.push({	デバイス名設定の開始				
5	dsp: 'X' + i,	表の項目で表示させる任意の名称 '入力' など任意の記述が可能。左記は X0～X7 を表示する例。				
6	name: 'X' + i,	デバイス指定 デバイス種別 + i でデバイス番号を指定。シーケンサのデバイス名を設定。				
7	base: 'B',	デバイスのデータ進数 <table><tr><th>設定値</th><th>内容</th></tr><tr><td>B</td><td>2進数</td></tr></table>	設定値	内容	B	2進数
設定値	内容					
B	2進数					
8	format: 6	デバイスのデータ形式 <table><tr><th>設定値</th><th>内容</th></tr><tr><td>6</td><td>ビット</td></tr></table>	設定値	内容	6	ビット
設定値	内容					
6	ビット					
9	});	デバイス名設定の終了				
10	}	forの終了				

表示デバイス指定 (3,4行目)

3, 4 行目の表示デバイス設定						
11	for(var i = 0; i < 8; i++){	forの開始 (変数名 i)				
12	temp.push({	デバイス名設定の開始 (詳細は上記行番号 3 と同様)				
13	dsp: 'Y' + i,	表の項目で表示させる任意の名称 '出力' など任意の記述が可能。左記は Y0～Y7 を表示する例。				
14	name: 'Y' + i,	デバイス指定 デバイス種別 + i でデバイス番号を指定。シーケンサのデバイス名を設定。				
15	base: 'B',	デバイスのデータ進数 <table><tr><th>設定値</th><th>内容</th></tr><tr><td>B</td><td>2進数</td></tr></table>	設定値	内容	B	2進数
設定値	内容					
B	2進数					
16	format: 6	デバイスのデータ形式 <table><tr><th>設定値</th><th>内容</th></tr><tr><td>6</td><td>ビット</td></tr></table>	設定値	内容	6	ビット
設定値	内容					
6	ビット					
17	});	デバイス名設定の終了				
18	}	forの終了				

表示デバイス指定 (1,2 行目)

表示デバイス指定 (3,4 行目)

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

行番号	JavaScript	解説
表全体の設定		
19	dataBlockParam = {	JavaScript / パラメータ設定の開始
20	dev: temp,	前ページのforで設定した複数のデバイス名(値)を指定
21	direction: 1,	表示方向 0: 縦方向 1: 横方向
22	blkSize: 8,	ブロックサイズ
23	devNamDisp: 1,	デバイス名表示 0: 表示しない 1: 表示する(省略時)
24	devNamCol: 'white',	文字色: カラー名で指定した例
25	devNamBkCol: '#808080',	背景色: 色コード(RGB値)で指定した例
26	devNamWidth: 100,	セル幅(pixel単位)
27	devNamHeight: 40,	セル高さ(pixel単位)
28	devValCol: 'blue',	文字色: カラー名で指定した例
29	devValBkCol: 'white',	背景色: カラー名で指定した例
30	devValWidth: 80,	セル幅(pixel単位)
31	devValHeight: 50,	セル高さ(pixel単位)
32	lnCol: 'blue',	線色: カラー名で指定した例
33	xPos: 20,	X座標(pixel単位)
34	yPos: 40	Y座標(pixel単位)
35	}	JavaScriptパラメータ設定の終了
実行		
36	WSDatblk(dataBlockParam);	データブロック部品の表示実行

パラメータ(共通)

実行

5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

レベル表示部品

ヒストリカルグラフ部品

5.4 グラフ

レベル表示部品 (WSLevel)

▶ 機能

指定したデバイスの現在値を、上限値と下限値の範囲に対するレベルで表示します。

▶ 作成例

方向: 縦方向

▶ ユーザWebページのライブラリ画面



部品の設計

▶ HTML (JavaScript) 構造

次の構造で作成します。

使用宣言

実行

パラメータ

表示デバイス指定

レベル表示設定

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

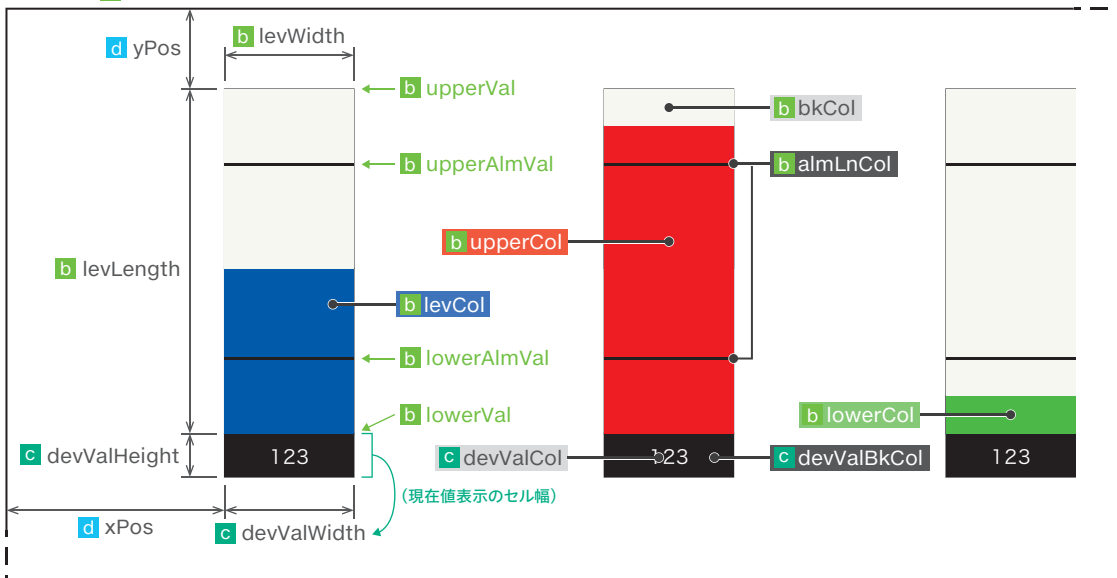
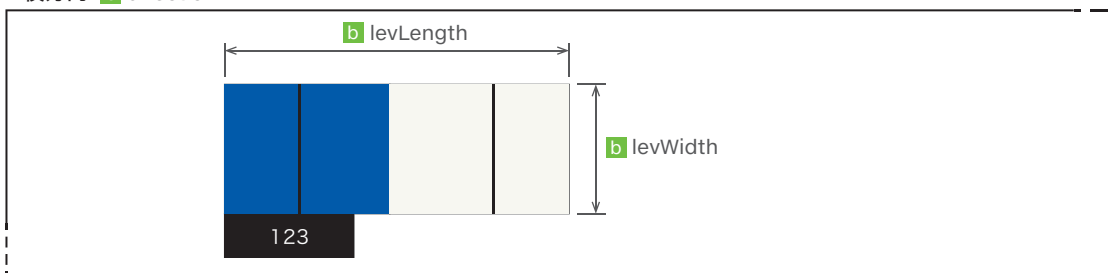
画像・図形

操作ボタン

レベル表示部品

ヒストリカルグラフ部品

▶ パラメーター一覧

・縦方向 **b** direction:0・横方向 **b** direction:1

a デバイス設定

▶ デバイス名… devName: デバイス種別 + デバイス番号

関連ページ 5.2 共通事項 - デバイス設定方法

▶ データ形式… valFormat:

値	内容
0	16bit符号付き (省略時)
1	16bit符号なし
2	32bit符号付き
3	32bit符号なし
4	単精度実数

関連ページ 5.2 共通事項 - デバイス設定方法

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

レベル表示部品

ヒストリカルグラフ部品

b レベル表示設定

▶ レベル方向… direction:

Script	内容
direction: 0	縦方向 (省略時)
direction: 1	横方向

▶ 上下限值… upperVal: lowerVal:

Script	内容
upperVal:	上限値
lowerVal:	下限値

▶ 警報値上下限值… upperAlmVal: lowerAlmVal:

Script	内容
upperAlmVal:	警報値上限値
lowerAlmVal:	警報値下限値

▶ 警報値線表示の有無… dspAlmLn:

Script	内容
dspAlmLn: 0	表示しない
dspAlmLn: 1	表示する (省略時)

表示例 : 表示しない



表示例 : 表示する



▶ 現在値表示の有無… dspVal:

Script	内容
dspVal: 0	表示しない
dspVal: 1	表示する (省略時)

表示例 : 表示しない



表示例 : 表示する



1 HTMLヘタケV作成の基礎

2 パソコンで簡単なHTML文書を作成し表示してみよう

3 カスタマイズで簡単にユーザーWebページを作成しよう

4 Style Sheet (CSS) の働きを知ろう

5 JavaScript (JS) で表示できる部品の働きを知る

6 CGによるデバイスモニタ画面を作成しよう

7 リファレンス

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

レベル表示部品

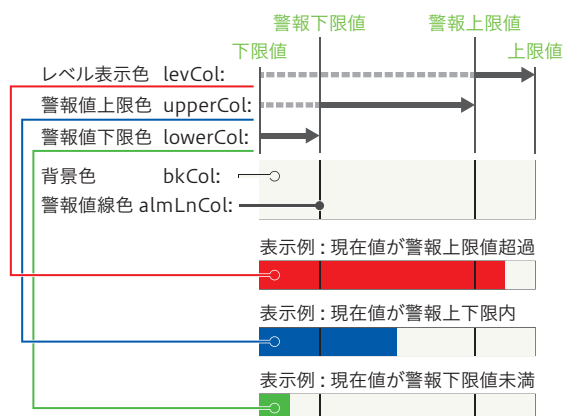
ヒストリカルグラフ部品

▶ 色指定… levCol: upperCol: lowerCol: bkCol: almLnCol:

レベル表示色は警報上下限值によって変化します。

Script	内容
levCol:	レベル表示色
upperCol:	警報値上限色*
lowerCol:	警報値下限色*
bkCol:	背景色
almLnCol:	警報値線色

*: パラメータ省略時はレベル表示色と同じ色になります。



▶ サイズ指定… levLength: levWidth

Script	内容	備考
levLength:	レベル長さ	正の実数 (pixel単位)
levWidth:	レベル幅	

□ 現在値表示セル

▶ サイズ指定… devValWidth: devValHeight:

Script	内容	備考
devValWidth:	現在値セル幅	正の実数 (pixel単位)
devValHeight:	現在値セル高さ	

▶ セル色指定… devValCol: devValBkCol:

Script	内容	備考
devValCol:	現在値文字色	RGB 値またはカラー名 例: #FF0000 (RGB 値)、red (カラー名)
devValBkCol:	現在値背景色	

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

レベル表示部品

ヒストリカルグラフ部品

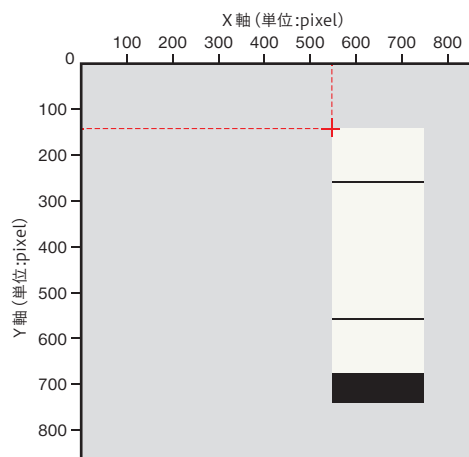
d 表示開始座標設定

▶ Webブラウザ表示開始位置設定… xPos: yPos:

Script	内容	備考
xPos:	X軸の座標	パソコンの設定範囲: 画面解像度(pixel単位)
yPos:	Y軸の座標	

▶ Webブラウザの座標の見方

例 (xPos, yPos) = (550, 150)



5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

レベル表示部品

ヒストリカルグラフ部品

▶ HTML文書の解説

パラメータの扱い

…パラメータ設定が必須。未設定の場合はエラーとなります。

…JavaScriptで設定した内容です。

注意事項

必須パラメータはすべて記述してください。必須パラメータの記述がない場合、または設定値が範囲外の場合はエラーになります。

行番号	JavaScript	解説
実行	1 // レベル部品	コメントを記載
	2 WSLevel({	データブロック部品のパラメータ設定+表示実行 JavaScript/パラメータ設定の開始
パラメータ	3 devName: 'D0',	デバイス指定 デバイス種別 + i でデバイス番号を指定 (シーケンサのデバイス名を設定)
	4 direction: 0,	表示方向 0: 縦方向 (省略時) 1: 横方向
	5 levCol: 'mediumblue',	レベル表示色: カラー名で指定した例
	6 upperCol: 'red',	警報値上限色: カラー名で指定した例
	7 lowerCol: '#00FF00',	警報値下限色: 色コード (RGB 値) で指定した例
	8 bkCol: 'white',	背景色: カラー名で指定した例
	9 upperVal: 32767,	上限値: 指定デバイスで扱える値の範囲
	10 lowerVal: -32768,	下限値: 指定デバイスで扱える値の範囲
	11 upperAlmVal: 20000,	警報値上限値: 指定デバイスで扱える値の範囲
	12 lowerAlmVal: -20000,	警報値下限値: 指定デバイスで扱える値の範囲
	13 dspAlmLn: 1,	警報値線表示 0: 表示しない 1: 表示する (省略時)
	14 almLnCol: 'black',	警報値線色: カラー名で指定した例
	15 levLength: 400,	レベル長さ: 正の実数 (pixel 単位)
	16 levWidth: 150,	レベル幅: 正の実数 (pixel 単位)
	17 dspVal: 1,	現在値表示の有無 0: 表示しない 1: 表示する (省略時)
	18 valFormat: 0,	デバイスのデータ形式 0: 16bit 符号付き (省略時) 3: 32bit 符号なし 1: 16bit 符号なし 4: 単精度実数 2: 32bit 符号付き
	19 devValCol: 'white',	現在値文字色
	20 devValBkCol: 'black',	現在値背景色
	21 devValWidth: 150,	現在値セル幅 (pixel 単位)
	22 devValHeight: 50,	現在値セル高さ (pixel 単位)
	23 xPos: 700,	表示開始位置 X座標 (pixel 単位)
	24 yPos: 250,	Y座標 (pixel 単位)
	25 });	JavaScript/パラメータ設定の終了

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

レベル表示部品

ヒストリカルグラフ部品

ヒストリカルグラフ部品 (WSHstgrp)

機能

デバイス値を時系列の折れ線グラフで表示します。

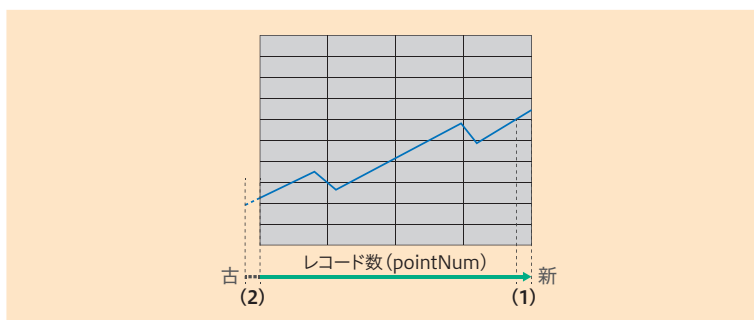
ユーザWebページのライブラリ画面



動作

デバイス値を更新周期ごとに読み出し、レコード数到達後は古いレコードを消去しながら表示を左へシフトします。

- (1) 更新周期分
- (2) 更新周期分が表示範囲外となります。



部品の設計

制約事項

- ・ヒストリカルグラフ部品で扱う数値は10進数固定です。
- ・通信負荷の高い状態が続くと、デバイスの値を取りこぼす場合があります。

HTML (JavaScript) 構造

表示するデバイスの折れ線の本数は、下記のように表示デバイス指定のHTML文書を連続して作成します。

折れ線が1本の場合

使用宣言

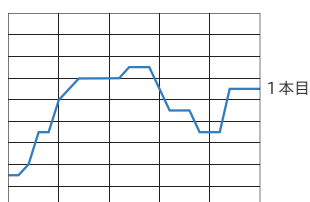
パラメータ (個別)

表示デバイス指定 (1本目)

パラメータ (共通)

ヒストリカルグラフ設定

実行



折れ線が3本の場合

使用宣言

パラメータ (個別)

表示デバイス指定 (1本目)

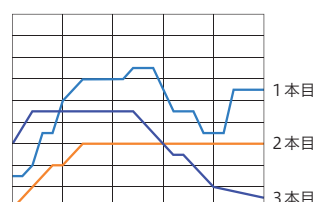
表示デバイス指定 (2本目)

表示デバイス指定 (3本目)

パラメータ (共通)

ヒストリカルグラフ設定

実行



5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

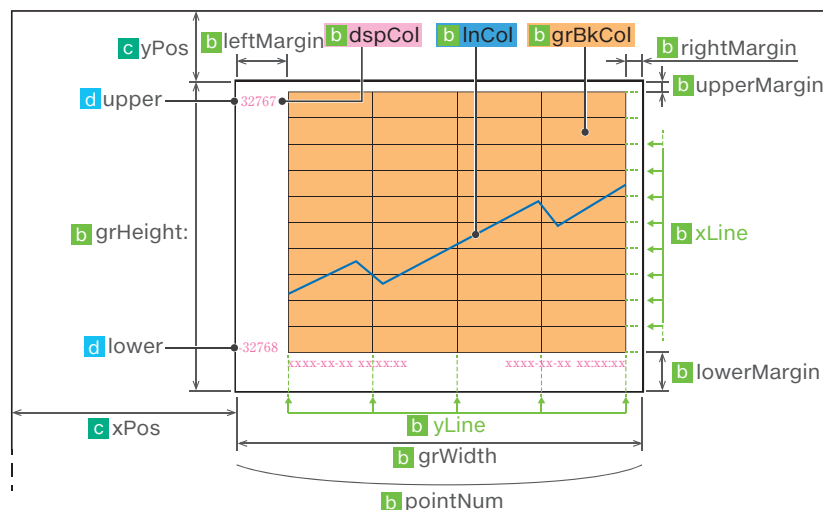
画像・図形

操作ボタン

レベル表示部品

ヒストリカルグラフ部品

▶ パラメーター一覧



a デバイス設定

▶ デバイス名… devName: デバイス種別 + デバイス番号

関連ページ 5.2 共通事項－デバイス設定方法

▶ データ形式… devFormat:

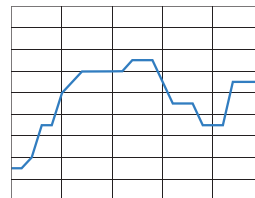
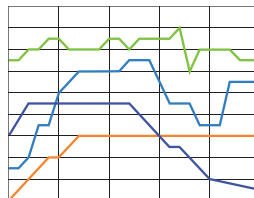
値	内容	値	内容	値	内容	値	内容
0	16bit 符号付き	2	32bit 符号付き	4	単精度実数	6	ビット
1	16bit 符号なし	3	32bit 符号なし	5	未使用		

関連ページ 5.2 共通事項－デバイス設定方法

b ヒストリカル グラフ表示設定

▶ グラフ要素数… grElmNum:

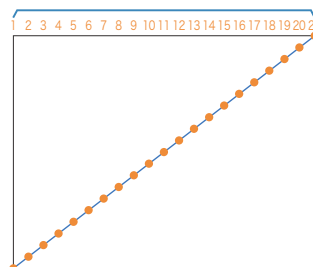
Script	設定範囲	内容
grElmNum:	1~32	折れ線の本数の設定



▶ レコード数… pointNum:

Script	設定範囲	内容
pointNum:	5~100	折れ線グラフの横軸にデータのポイント数

例 pointNum : 21



5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

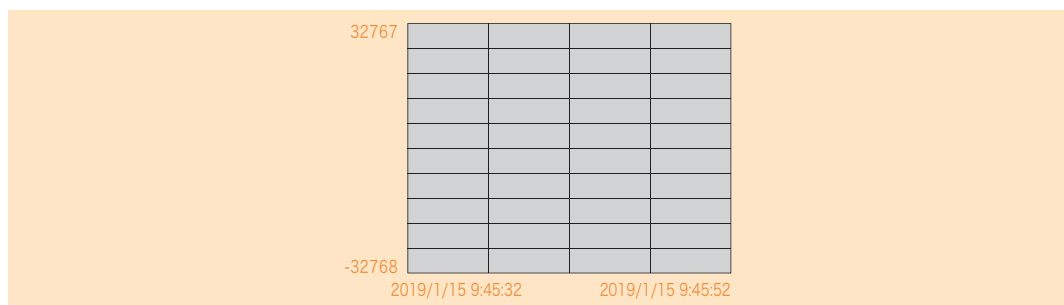
操作ボタン

レベル表示部品

ヒストリカルグラフ部品

▶ 上下限值… upper: lower:

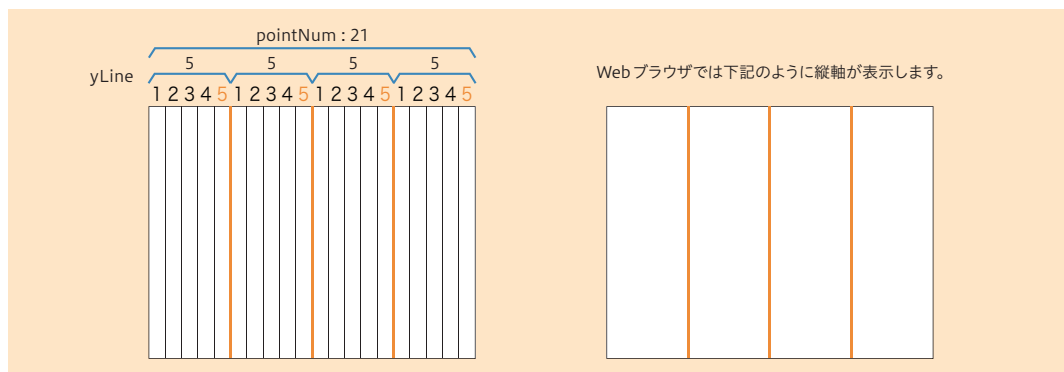
Script	内容	備考
upper:	上限値	デバイス値(Y軸)の上限値と下限値の設定 範囲以外の場合はエラーになります。
lower:	下限値	データ形式がビットの場合、上限値を1、下限値を0に指定してください。



▶ 縦軸間隔(レコード)… yLine:

Script	設定範囲	内容
yLine:	0~99	縦軸の目盛線を表示する間隔設定 レコードに対して指定した値の間隔となります。

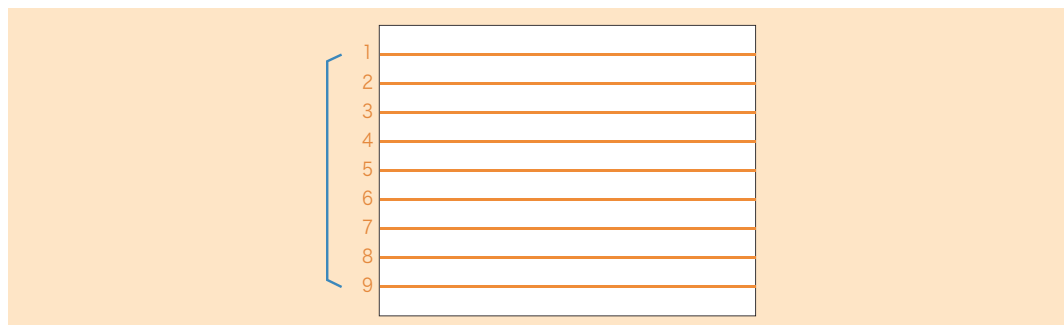
例 yLine : 5 (縦軸間隔)



▶ 横軸の本数… xLine:

Script	設定範囲	内容
xLine:	0~99	横軸の目盛線を表示する本数設定

例 xLine : 9



5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

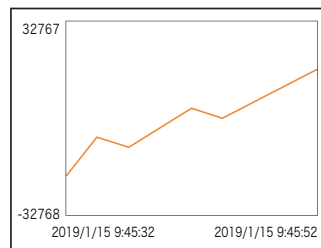
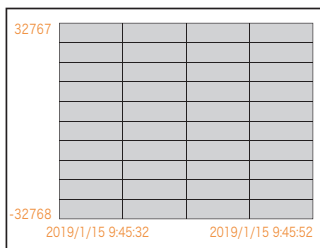
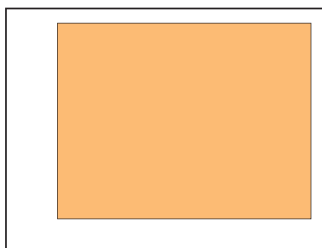
操作ボタン

レベル表示部品

ヒストリカルグラフ部品

▶ 色指定… InCol: dspCol: grBkCol:

Script	内容
InCol:	グラフ線色
dspCol:	文字色
grBkCol:	グラフ背景色



▶ サイズ指定… grHeight: grWidth:

Script	内容	備考
grHeight:	グラフ高さ	正の実数(pixel単位)
grWidth:	グラフ幅	

▶ 余白指定… rightMargin: leftMargin: upperMargin: lowerMargin:

Script	内容	備考
rightMargin:	右余白	正の実数(pixel単位) パラメータ省略時は0
leftMargin:	左余白	
upperMargin:	上余白	
lowerMargin:	下余白	

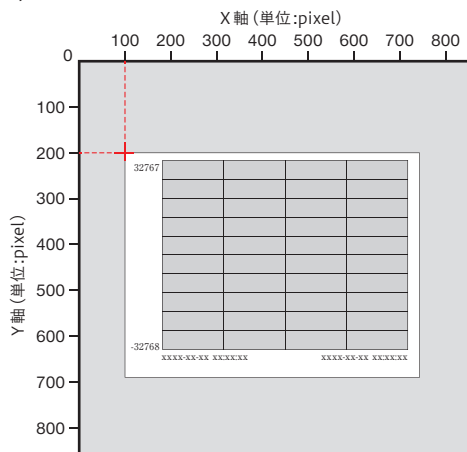
c 表示開始座標設定

▶ Webブラウザ表示開始位置設定… xPos: yPos:

Script	内容	備考
xPos:	X軸の座標	設定範囲: 画面解像度(pixel単位)
yPos:	Y軸の座標	

▶ Webブラウザの座標の見方

例 (xPos, yPos) = (100, 200)



5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

レベル表示部品

ヒストリカルグラフ部品

▶ HTML文書の解説

パラメータの扱い

…パラメータ設定が必須。未設定の場合はエラーとなります。

注意事項

必須パラメータはすべて記述してください。必須パラメータの記述がない場合、または設定値が範囲外の場合はエラーになります。

表示デバイス指定（1,2行目）

行番号	JavaScript	解説
1	// ヒストリカルグラフ部品	コメントを記載
2	temp = [];	
3	num = 2;	グラフ要素数 設定範囲: 1〜32まで
4	temp.push({	1本目の指定
5	devName: 'D0',	デバイス名: デバイス種別 + デバイス番号
6	lnCol: 'red',	グラフ線色: カラー名で指定した例
7	});	
8	temp.push({	2本目の指定
9	devName: 'D1',	デバイス名: デバイス種別 + デバイス番号
10	lnCol: 'blue',	グラフ線色: カラー名で指定した例
11	});	
12	hstGrpParam = {	
13	xPos: 20,	表示開始位置 X座標 (pixel単位)
14	yPos: 250,	Y座標 (pixel単位)
15	grElmNum: num,	引数numは行番号3のnum=2でグラフ要素数設定
16	devFormat: 0,	デバイスのデータ形式 値 内容 016bit符号付き
17	grElm: temp,	tempは引数で行番号3のnum内でデバイス名とグラフ線色設定
18	grBkCol: '#F0F0F0',	グラフ背景色: カラーコードで指定した例
19	dspCol: 'black',	文字色: カラー名で指定した例
20	pointNum: 20,	レコード数
21	pointInt: 2,	
22	upper: 32767,	上限値
23	lower: -32768,	下限値
24	xLine: 9,	横軸の線本数 0〜99
25	yLine: 5,	線を表示する縦軸間隔 (レコード) 0〜99
26	grHeight: 380,	グラフ高さ (pixel単位)
27	grWidth: 550,	グラフ幅 (pixel単位)
28	upperMargin: 15,	上余白: 正の実数 (pixel単位) (省略時は0)
29	leftMargin: 75,	左余白: 正の実数 (pixel単位) (省略時は0)
30	lowerMargin: 55,	下余白: 正の実数 (pixel単位) (省略時は0)
31	rightMargin: 25,	右余白: 正の実数 (pixel単位) (省略時は0)
32	}	JavaScriptパラメータ設定の終了
33	WSHstgrp(hstGrpParam);	データブロック部品の表示実行

実行

表示デバイス指定(1,2行目)

実行

5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

画像表示部品

図形表示部品

5.5 画像・図形

画像表示部品 (WSPicture)

機能

デバイス値が設定した範囲の値になったときに、指定した画像ファイルを表示します。

作成例

モニタデバイス: D0

画像数: 3 (デフォルト表示画像含む)

設定した画像ファイル名:

sample0.png

sample1.png

sample2.png

ユーザWebページのライブラリ画面



部品の設計

制約事項

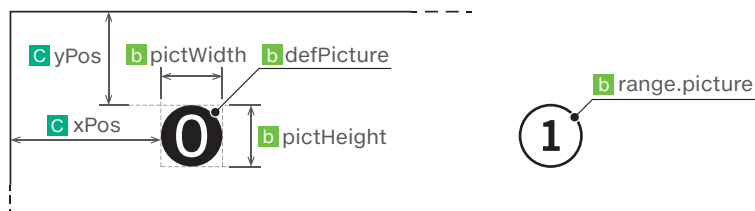
- ・ 1 画面内の画像ファイルの合計サイズは、100kbyte 以下を目安としてください。
- ・ 設定範囲が重複している場合、設定範囲の若い番号の画像を表示します。例えばデバイス値が設定範囲1と設定範囲2の両方にある場合、設定範囲1の表示となります。

HTML (JavaScript) 構造

表示を切り替える画像数に応じて、下記のようにパラメータのHTML文書を連続して作成します。

実行	
パラメータ (共通)	画像表示部品設定
	デフォルト表示画像
	画像 (1 個目)
パラメータ (個別)	画像 (2 個目)
	⋮
	画像 (5 個目)

パラメーター一覧



5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

画像表示部品

図形表示部品

a デバイス設定

▶ デバイス名… devName: デバイス種別 + デバイス番号

関連ページ 5.2 共通事項－デバイス設定方法

▶ データ形式… valFormat:

値	内容	値	内容
0	16bit符号付き	4	単精度実数
1	16bit符号なし	5	未使用
2	32bit符号付き	6	ビット
3	32bit符号なし		

関連ページ 5.2 共通事項－デバイス設定方法

b 画像設定

▶ 画像数設定… rangeNum:

表示する画像の数を指定します。(デフォルト表示画像は含めない)

Script	内容	備考
rangeNum:	切替え画像数設定	設定範囲: 1～5

▶ 画像ファイル設定… defPicture: range.picture:

任意の画像ファイルをそれぞれ指定します。サイズは、画像サイズ設定で指定した値で表示します。

Script	内容	備考
defPicture:	デフォルト画像ファイル指定	デバイス値が設定範囲外の際に表示 拡張子: .jpg, .jpeg, .gif, .png
range.picture:	切替え画像ファイル指定	デバイス値が設定範囲内の際に表示 拡張子: .jpg, .jpeg, .gif, .png

関連ページ 5.1 JavaScript(JS)とは－使用可能ファイル

▶ 画像サイズ設定… pictHeight: pictWidth:

Script	内容	備考
pictHeight:	画像高さ	値: 正の実数 (pixel単位)
pictWidth:	画像横幅	負数を設定した場合は、絶対値として扱われます。

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

画像表示部品

図形表示部品

▶ デバイス値範囲設定 … range.low: range.high:

Script	内容	備考
range.low:	下限値	画像を表示するデバイス値の下限値
range.high:	上限値	画像を表示するデバイス値の上限値

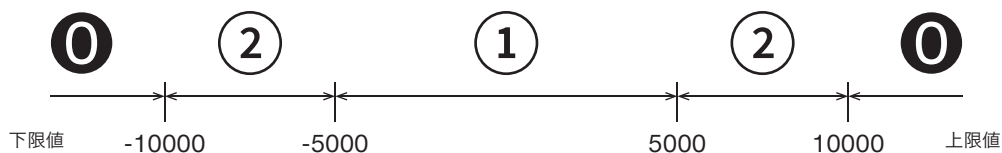
値: 正の実数 (pixel単位)

例 rangeNum: 2

① デフォルト表示

① 画像 (1 個目) range.low:-5000, range.high:5000

② 画像 (2 個目) range.low:-10000, range.high:10000



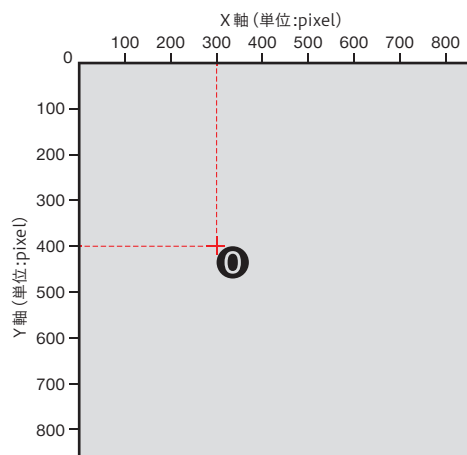
c 表示開始座標設定

▶ Webブラウザ表示開始位置設定 … xPos: yPos:

Script	内容	備考
xPos:	X軸の座標	設定範囲: パソコンの画面解像度 (pixel単位)
yPos:	Y軸の座標	

▶ Webブラウザの座標の見方

例 (xPos, yPos) = (300, 400)



5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

画像表示部品

図形表示部品

▶ HTML文書の解説

パラメータの扱い

…パラメータ設定が必須。未設定の場合はエラーとなります。

注意事項

必須パラメータはすべて記述してください。必須パラメータの記述がない場合、または設定値が範囲外の場合はエラーになります。

行番号		JavaScript	解説					
実行	1	// 画像表示部品	コメントを記載					
	2	WSPicture({	画像表示部品のパラメータ設定 + 表示実行					
パラメータ	3	devName: 'D0',	デバイス指定 デバイス種別 + デバイス番号					
	4	devFormat: 0,	デバイスのデータ形式	<table><tr><th>値</th><th>内容</th></tr><tr><td>0</td><td>16bit符号付き</td></tr></table>	値	内容	0	16bit符号付き
	値	内容						
	0	16bit符号付き						
	5	pictHeight: 30,	画像ファイル表示範囲の高さ					
	6	pictWidth: 30,	画像ファイル表示範囲の幅					
	7	defPicture: './img/sample0.png',	デフォルトの表示画像ファイル名					
	8	rangeNum 2,	設定範囲個数 設定範囲: 1〜5					
	9	range:[						
	10	{						
	11	low: -5000,	画像を変更する設定 (1個目)	画像を表示するデバイス値の下限值				
	12	high: 5000,		画像を表示するデバイス値の上限値				
	13	picture: './img/sample1.png',		任意の画像ファイルを指定				
	14	},						
	15	{	画像を変更する設定 (2個目)	画像を表示するデバイス値の下限值				
	16	low: -10000,		画像を表示するデバイス値の上限値				
	17	high: 10000,		任意の画像ファイルを指定				
	18	picture: './img/sample2.png',						
	19	},						
	20	],						
21	xPos: 150,	表示開始位置	X座標 (pixel単位)					
22	yPos: 50,		Y座標 (pixel単位)					
	23	});						

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

画像表示部品

図形表示部品

図形表示部品 (WSFigure)

機能

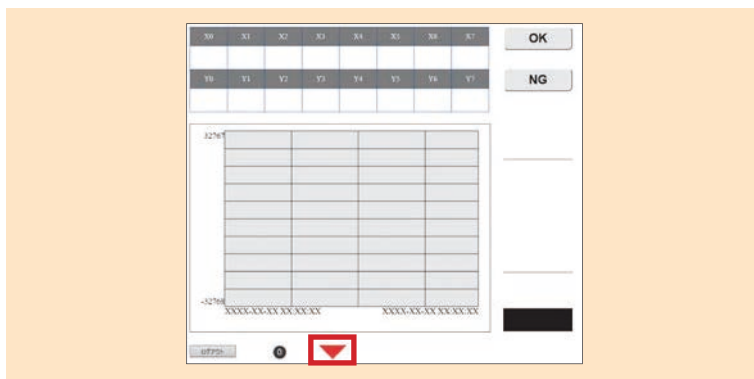
デバイス値が設定した範囲の値になったときに、指定した色で図形を表示します。

作成例

形状: 逆三角形

図形色: 3 (デフォルト表示図形色含む)

ユーザWebページのライブラリ画面



部品の設計

制約事項

- ・設定範囲が重複している場合、設定範囲の若い番号の画像を表示します。
例えば、デバイス値が設定範囲1と設定範囲2の両方にある場合、設定範囲1の表示となります。
- ・デバイス値の監視は定周期で実行されます。

HTML (JavaScript) 構造

表示させたい表示色の範囲分だけ設定します。

実行

パラメータ (共通)	図形表示部品設定 デフォルト表示図形色
パラメータ (個別)	図形色設定 (1 個目)
パラメータ (個別)	図形色設定 (2 個目)

設定した画像ファイル名の例

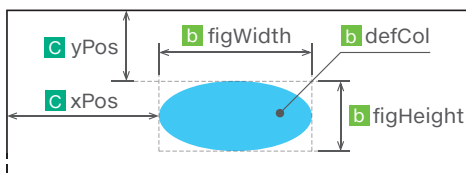
sample0.png

sample1.png

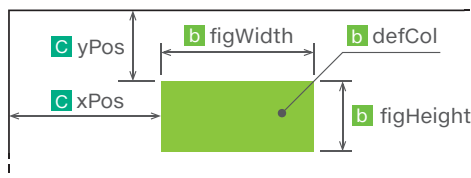
sample2.png

パラメーター一覧

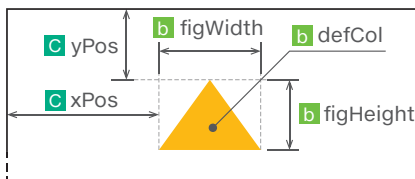
・だ円 (円)



・長方形 (正方形)



・三角形



5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

画像表示部品

図形表示部品

a デバイス設定

▶ デバイス名… devName: デバイス種別 + デバイス番号

関連ページ 5.2 共通事項 - デバイス設定方法

▶ データ進数… base:

値	内容
K	10進数
H	16進数
B	2進数

関連ページ 5.2 共通事項 - デバイス設定方法

▶ データ形式… devFormat:

値	内容	値	内容
0	16bit符号付き	4	単精度実数
1	16bit符号なし	5	未使用
2	32bit符号付き	6	ビット
3	32bit符号なし		

関連ページ 5.2 共通事項 - デバイス設定方法

▶ デバイス値範囲設定… range.low: range.high:

Script	内容	備考
range.low:	下限値	図形色を表示するデバイス値の下限値
range.high:	上限値	図形色を表示するデバイス値の上限値

値: 正の実数 (pixel単位)

b 図形設定

▶ 設定範囲個数… rangeNum:

変化させる色数 (1~5) を指定します。(初期表示色は含めない)

▶ 形状… figType:

値	内容	備考
'Oval',	だ円	円を指定する場合は、図形の高さ = 幅に設定
'Rect',	長方形	正方形を指定する場合は、図形の高さ = 幅に設定
'Tri',	三角形	逆三角形を指定する場合は、図形のサイズの高さに負数を設定

▶ 図形サイズ設定… figWidth: figHeight:

Script	内容	備考
figWidth:	図形の幅	実数 (pixel単位)
figHeight:	図形の高さ	負数を設定した場合は、絶対値として扱われます。

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

画像表示部品

図形表示部品

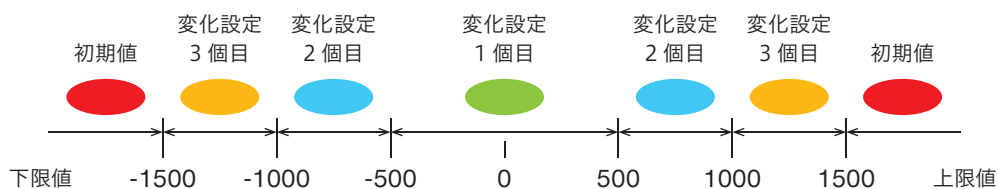
▶ 図形色指定… defCol: rangeCol:

Script	内容	備考
defCol:	初期表示色	デバイス値範囲設定範囲外の場合に表示される図形の色です。(デフォルト表示図形色) RGB値またはカラー名 例: #FF9900 (RGB値)、red (カラー名)
range.col:	変化色	

関連ページ 7.3 カラーネーム・カラーコード

動作例

0を中心として、プラス側とマイナス側にそれぞれ上限値と下限値を設定しています。



JavaScript例 (抜粋)

変化色を3点設定した場合は、下記ようになります。

JavaScript	解説
figType: 'Oval', defCol: 'red', rangeNum: 3, range:[{ low: -500, high: 500, col: 'green', }, { low: -1000, high: 1000, col: 'blue', }, { low: -1500, high: 1500, col: '#FF9900', },],	楕円図形 初期表示色設定 変化させる色数設定  変化設定 (1 個目)  変化設定 (2 個目)  変化設定 (3 個目) 

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

画像表示部品

図形表示部品

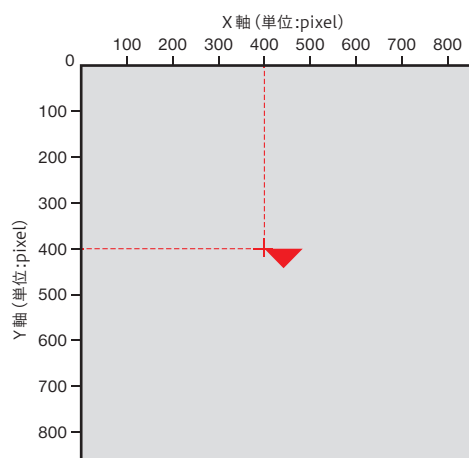
C 表示開始座標設定

▶ Webブラウザ表示開始位置設定 ... xPos: yPos:

Script	内容	備考
xPos:	X軸の座標	設定範囲: パソコンの画面解像度 (pixel単位)
yPos:	Y軸の座標	

▶ Webブラウザの座標の見方

例 (xPos, yPos) = (400, 400)



図形表示部品

行番号		JavaScript	解説					
実行	1	// 図形表示部品	コメントを記載					
	2	WSFigure({	図形表示部品のパラメータ設定+表示実行					
パラメータ	3	devName: 'D0',	デバイス指定 デバイス種別 + デバイス番号					
	4	devFormat: 0,	デバイスのデータ形式	<table><tr><th>値</th><th>内容</th></tr><tr><td>0</td><td>16bit 符号付き</td></tr></table>	値	内容	0	16bit 符号付き
	値	内容						
	0	16bit 符号付き						
	5	figType: 'tri',	図形の種類 例: 三角形					
	6	figHeight: -30,	図形サイズ (負数を設定した場合は、絶対値として扱う)	高さ 実数 (pixel単位)				
	7	figWidth: 60,		横幅 実数 (pixel単位)				
	8	defCol: 'red',	デフォルトの表示色: カラー名で指定した例					
	9	rangeNum: 2,	設定範囲個数	設定範囲 n: 1~5 変化させる色数を指定				
	10	range:[	変化する設定範囲の指定開始					
	11	{						
	12	low: -5000,	変化設定 (1 個目)	図形色を表示するデバイス値の下限値				
	13	high: 5000,		図形色を表示するデバイス値の上限値				
	14	col: 'green',		上下限值内の時の表示色				
	15	},						
	16	{						
	17	low: -10000,	変化設定 (2 個目)	図形色を表示するデバイス値の下限値				
	18	high: 10000,		図形色を表示するデバイス値の上限値				
	19	col: 'blue',		上下限值内の時の表示色				
	20	},						
	21	],	変化する設定範囲の指定終了					
	22	xPos: 300,	表示開始位置	X座標 (pixel単位)				
	23	yPos: 730,		Y座標 (pixel単位)				
	24	});	JavaScriptパラメータ設定の終了					

5. JavaScript(JS) で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

書き込みボタン部品

ログアウトボタン部品

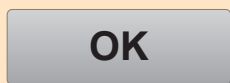
5.6 操作ボタン

書き込みボタン部品 (WSWrtBtn)

機能

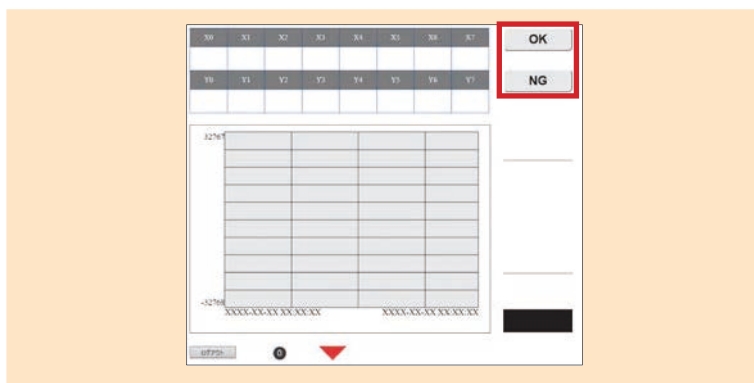
指定デバイスに、指定した値を書き込みます。

作成例



ボタン内の文字列は変更可能

ユーザWebページのライブラリ画面



部品の設計

HTML (JavaScript) 構造

下記のようにパラメータのHTML文書を作成します。

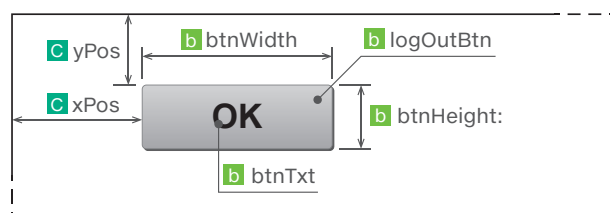
使用宣言

パラメータ

書き込みデバイス指定

実行

パラメーター一覧



a デバイス設定

▶ デバイス名… devName: デバイス種別 + デバイス番号

関連ページ 5.2 共通事項－デバイス設定方法

▶ データ進数… devBase:

値	内容
K	10進数
H	16進数
B	2進数

関連ページ 5.2. 共通事項－デバイス設定方法

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

書き込みボタン部品

ログアウトボタン部品

▶ データ形式… devFormat:

値	内容	値	内容
0	16bit符号付き	4	単精度実数
1	16bit符号なし	5	未使用
2	32bit符号付き	6	ビット
3	32bit符号なし		

関連ページ 5.2 共通事項－デバイス設定方法

▶ 書き込み値… wrVal:

Script	内容	備考
wrVal:	書き込み値	設定範囲は、デバイスの進数やデータ形式による

特記事項 入力値は文字列で指定します。(' 'で囲む) 例:wrVal: '1'

b ボタンデザイン

▶ Style Sheet用class要素名… wrBtn:

Script	内容	備考
wrBtn:	class要素名	Style Sheetのセレクトに使用

▶ ボタン表示テキスト… btnTxt:

Script	内容
btnTxt:	ボタン上に表示するテキスト(省略時は空白を表示)

▶ ボタンサイズ指定… btnWidth: btnHeigh:

Script	内容	備考
btnWidth:	ボタンの幅	正の実数(pixel単位)
btnHeight:	ボタンの高さ	

▶ 書き込み確認メッセージの有無… wrComfirm:

ボタン操作(デバイス値書換え)時に、端末OS上の書き込み確認メッセージ表示の有無を指定します。

→詳細はメッセージ言語(language:)を参照してください。

Script	内容	備考
wrComfirm:0	表示しない	パラメータ省略時は0(書き込み確認メッセージの表示をしない)
wrComfirm:1	表示する	

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

書き込みボタン部品

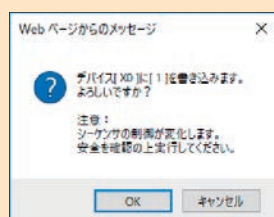
ログアウトボタン部品

▶ メッセージ言語… language:

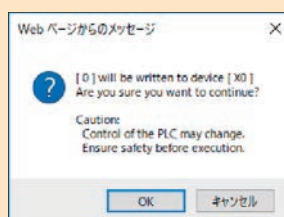
Script	内容	備考
language:0	日本語	省略時は1 (英語)
language:1	英語	
language:2	中国語 (簡体字)	

例 OSがWindows® の場合

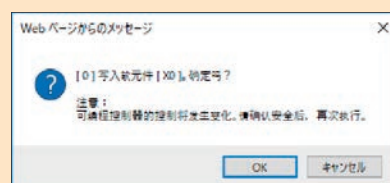
language:0



language:1



language:2



上記ダイアログボックスのタイトルバーやボタンの文字は、Windows® の言語設定で設定された言語で表示します。
上記画面は、Windows® 10 日本語版になります。

c 表示開始座標設定

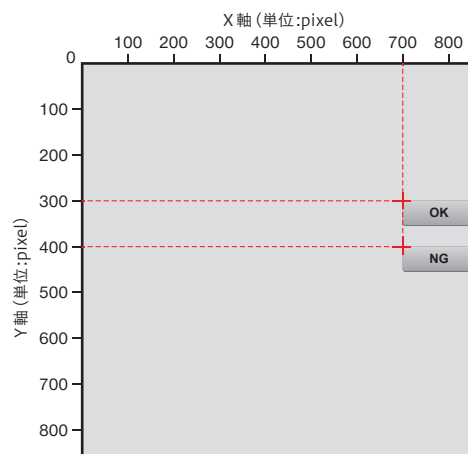
▶ Web ブラウザ表示開始位置設定… xPos: yPos:

Script	内容	備考
xPos:	X 軸の座標	設定範囲: 画面解像度 (pixel 単位)
yPos:	Y 軸の座標	

▶ Web ブラウザの座標の見方

例 OK ボタン: (xPos, yPos) = (700, 300)

NG ボタン: (xPos, yPos) = (700, 400)



5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは | 共通事項 | 表 | グラフ | 画像・図形 | 操作ボタン | 書き込みボタン部品 | ログアウトボタン部品

▶ HTML文書の解説

パラメータの扱い

…パラメータ設定が必須。未設定の場合はエラーとなります。

注意事項 必須パラメータはすべて記述してください。必須パラメータの記述がない場合、または設定値が範囲外の場合はエラーになります。

行番号		JavaScript	解説					
OKボタン								
実行	1	// 書き込みボタン部品	コメントを記載					
	2	WSWrtBtn({	書き込みボタン部品のパラメータ設定+表示実行					
表示デバイス指定(1,2行目)	3	devName : 'X0',	デバイス指定	デバイス種別 + デバイス番号				
	4	devBase : 'B',	デバイスのデータ進数	<table><tr><th>設定値</th><th>内容</th></tr><tr><td>B</td><td>2進数</td></tr></table>	設定値	内容	B	2進数
	設定値	内容						
	B	2進数						
	5	devFormat : 6,	デバイスのデータ形式	<table><tr><th>設定値</th><th>内容</th></tr><tr><td>6</td><td>ビット</td></tr></table>	設定値	内容	6	ビット
	設定値	内容						
	6	ビット						
	6	wrVal : '1',	書き込み値に'1' (X0をON) 設定。入力値は文字列 ('「」 で囲む) で設定。					
	7	wrBtn : 'write_btn',	Style Sheet用class要素名					
	8	btnTxt : 'OK',	ボタン上に表示するテキストに'OK'を設定					
	9	btnWidth : 150,	ボタンサイズ	幅 (pixel単位)				
	10	btnHeight : 50,		高さ (pixel単位)				
	11	wrComfirm : 1,	書き込み確認メッセージの有無					
12	language : 0,	メッセージ言語						
13	xPos : 700,	表示開始位置	X座標 (pixel単位)					
14	yPos : 40,		Y座標 (pixel単位)					
15	}	JavaScript/パラメータ設定の終了						

NGボタン

実行

16

// 書き込みボタン部品

コメントを記載

17

WSWrtBtn{

書き込みボタン部品の表示実行

表示デバイス指定（1,2行目）

18

devName

:

'X0',

デバイス指定

デバイス種別 + デバイス番号

19

devBase

:

'B',

デバイスのデータ進数

設定値

内容

B

2進数

20

devFormat

:

6,

デバイスのデータ形式

設定値

内容

6

ビット

21

wrVal

:

'0',

書き込み値に'0'(X0をOFF)設定。入力値は文字列（「」で囲む）で設定。

22

wrBtn

:

'write_btn',

Style Sheet用class要素名

23

btnTxt

:

'NG',

ボタン上に表示するテキストに'NG'を設定

24

btnWidth

:

150,

ボタンサイズ

幅 (pixel単位)

25

btnHeight

:

50,

高さ (pixel単位)

26

wrComfirm

:

1,

書き込み確認メッセージの有無

27

language

:

0,

メッセージ言語

28

xPos

:

700,

表示開始位置

X座標 (pixel単位)

29

yPos

:

130,

Y座標 (pixel単位)

30

}

JavaScript/パラメータ設定の終了

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは

共通事項

表

グラフ

画像・図形

操作ボタン

書き込みボタン部品

ログアウトボタン部品

ログアウトボタン部品 (WSLogoutBtn)

機能

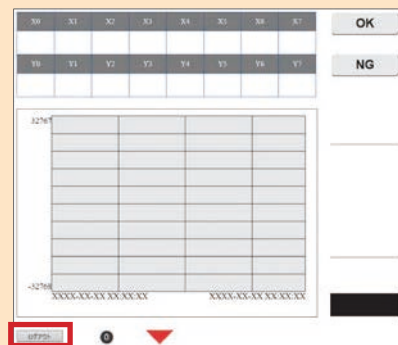
ログアウトするためのボタンを配置します。

作成例

ログアウト

ボタン内の文字列は変更可能

ユーザWebページのライブラリ画面



部品の設計

制約事項

- ・ログアウトボタン部品は、1つのWebページに1つのみ使用可能です。
- ・必ずユーザ用ログインページ (Log-in_User.html) をSDメモリカードに格納してください。
- ・パラメータに指定がない場合、Style Sheet (UserWebStyle.css) で指定しているデフォルトのスタイルになります。

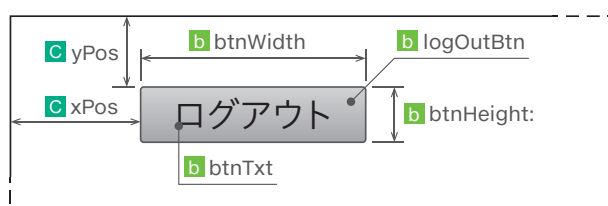
HTML (JavaScript) 構造

パラメータ

表示設定

実行

パラメーター一覧



a ボタン表示設定

サイズ指定… btnWidth: btnHeight:

Script	内容
btnWidth:	ボタンサイズ 横幅 (pixel単位)
btnHeight	高さ (pixel単位)

ボタン表示文字… btnTxt:

Script	内容
btnTxt:	ボタン上に表示するテキストを任意で設定 (省略時は空白を表示)

例 btnTxt: 'ログアウト' 日本語
btnTxt: 'Log Out' 英語
btnTxt: '退出' 中国語

5. JavaScript(JS)で表示できる部品の働きを知る

JavaScriptとは | 共通事項 | 表 | グラフ | 画像・図形 | 操作ボタン | 書き込みボタン部品 | ログアウトボタン部品

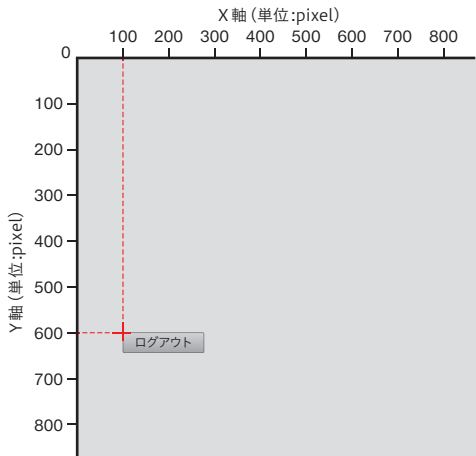
b 表示開始座標設定

▶ Webブラウザ表示開始位置設定 ... xPos: yPos:

Script	内容	備考
xPos:	X軸の座標	設定範囲: 画面解像度 (pixel単位)
yPos:	Y軸の座標	

▶ Webブラウザの座標の見方

例 (xPos, yPos) = (100, 600)



▶ HTML文書の解説

パラメータの扱い

…パラメータ設定が必須。未設定の場合はエラーとなります。

注意事項 必須パラメータはすべて記述してください。必須パラメータの記述がない場合、または設定値が範囲外の場合はエラーになります。

	行番号	JavaScript	解説	
パラメータ	1	// ログアウトボタン部品	コメントを記載	
	2	logoutBtnParam = {	JavaScript/パラメータ設定の開始	
	3	xPos 20,	表示開始位置	X座標 (pixel単位)
	4	yPos 730,		Y座標 (pixel単位)
	5	btnWidth 26,	ボタンサイズ	幅 (pixel単位)
	6	btnHeight 100,		高さ (pixel単位)
	7	btnTxt 'ログアウト'	ボタン表示文字	
	8	}	JavaScript/パラメータ設定の終了	
実行	9	WSLogoutBtn (logoutBtnParam);	実行	

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読みWeb画面の作成

デバイス書き込みWeb画面の作成

6.1 CGIとは

CGIとは、“コモン・ゲートウェイ・インターフェース (Common Gateway Interface)”です。

Web ブラウザなどからの要求によってwebサーバでプログラムを実行し、実行結果をクライアントに返し、Web ブラウザに表示させる仕組みです。CGI 部品を使用することで、シンプルでファイルサイズの小さいユーザ Web ページを作成できます。

ユーザ Web ページでは、下記のデバイスの読み出しと書き込み CGI 部品が使用できます。

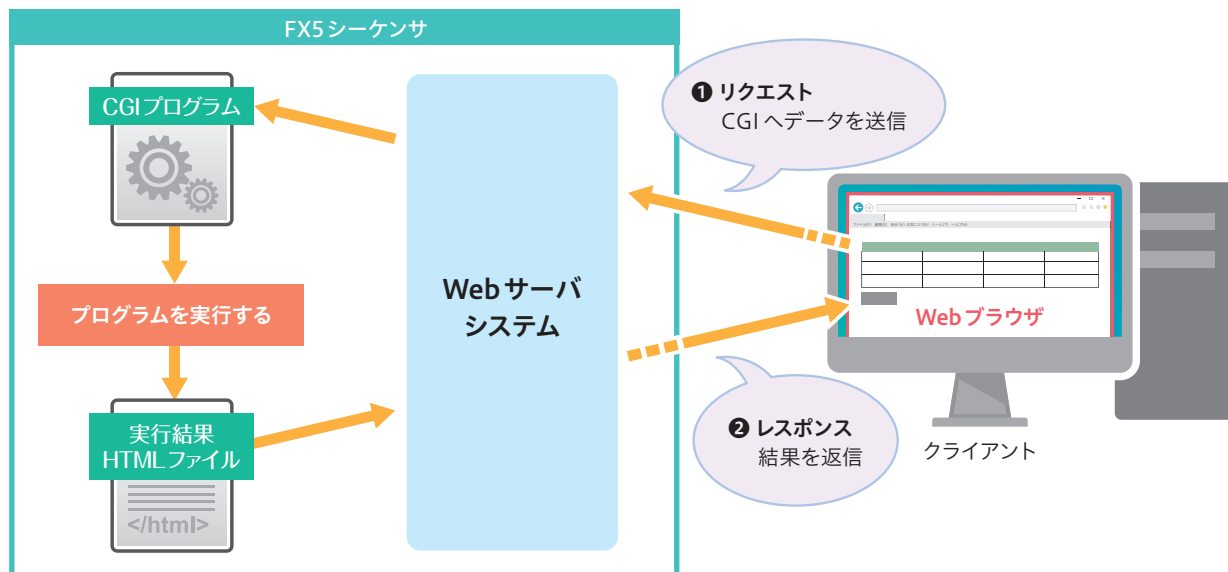
項目	CGI 名	機能	参照節
デバイス読み出し CGI	RdDevRnd.cgi	指定したデバイスの現在値を読み出す	6.4
デバイス書き込み CGI	WrDev.cgi	指定したデバイスに設定値を書き込む	6.5

6.2 CGIの概要と機能

CGIの概要

FX5シーケンサは、GX Works3のパラメータでWebサーバ機能を有効にすると、Webサーバシステムが動作します。

CGIは、Web ブラウザからのアクセスによってシーケンサ内にあるWebサーバ内でCGI プログラムが実行され、その結果をWeb ブラウザへ返すという仕組みです。



6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読出しWeb画面の作成

デバイス書込みWeb画面の作成

CGIの機能

ユーザWebページでは、下記のデバイスの読出し/書込みを行うCGI部品が使用できます。CGI部品は、[デバイス読出し用]と[デバイス書込み用]の2種類があります。CGIを使用することで、読出し/書込みボタンをクリックして任意のデバイスに対して値の読出し/書込み操作ができます。

▶ デバイス読出しで使用するCGIの機能

[読出] ボタンをクリックすることで、モニタ値を更新表示します。

初期値はHTMLで作成したデバイス名とデータ型となり、値は[読出] ボタンをクリックするまでブランクとなります。また、デバイス名やデータ型を変更し表示可能です。

CGI名	RdDevRnd.cgi
機能	指定したデバイスの現在値を読み出す

画面例

デバイス名	データ型	値
D10	16bit 整数	1234
D11	32bit 整数	123456
M0	ビット	0

読出

HTMLで設定されたデバイス

HTMLで設定された定数

▶ 操作手順

例として、2行目のD11をD20に変更してデバイスモニタする手順を説明します。

1 読出しボタンをクリック

デバイス名	データ型	値
D10	16bit 整数	
D11	32bit 整数	
M0	ビット	

読出

2 モニタ値を表示

デバイス名	データ型	値
D10	16bit 整数	1234
D11	32bit 整数	123456
M0	ビット	0

読出

3 D11(例)のセルをクリック

デバイス名	データ型	値
D10	16bit 整数	1234
D11	32bit 整数	123456
M0	ビット	0

読出

デバイス名を選択し、×ボタンを表示します。

4 D11(例)をクリア

デバイス名	データ型	値
D10	16bit 整数	1234
D11	32bit 整数	
M0	ビット	0

読出

×ボタンをクリックし、デバイス名を削除します。

5 D20を入力

デバイス名	データ型	値
D10	16bit 整数	1234
D20	32bit 整数	123456
M0	ビット	0

読出

6 読出しボタンをクリック→D20のモニタ値を更新

デバイス名	データ型	値
D10	16bit 整数	1234
D20	32bit 整数	0
M0	ビット	0

読出

変更したデバイスのD20やD10、M0の値を更新表示します。

関連ページ

CGI部品を使ったデバイスモニタのWebページ作成方法は、下記を参照してください。

- ・ 扱えるデバイスの仕様
6.3 共通仕様
- ・ デバイス読出しWeb画面の作成手順
6.4 デバイス読出しWeb画面の作成

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読出しWeb画面の作成

デバイス書込みWeb画面の作成

▶ デバイス書込みで使用するCGIの機能

[書込] ボタンをクリックすることで、入力した値を書き込みます。

初期値はHTMLで作成したデバイス名やデータ型、値となります。また、デバイス名やデータ型、値を変更し、書き込むことが可能です。

CGI名	WrDev.cgi
機能	指定したデバイスに設定値を書き込む

画面例

デバイス名	データ型	値	
D10	16bit 整数	3	書込
D11	32bit 整数	10	書込
M0	ビット	1	書込

HTMLで設定されたデバイス

HTMLで設定された定数

▶ 操作手順

例として、2行目のD11をD20に変更し、D20に55を書き込む手順を説明します。

1 D11 (例) のセルをクリック

デバイス名	データ型	値	
D10	16bit 整数	3	書込
D11	32bit 整数	10	書込
M0	ビット	1	書込

デバイス名を選択し、× ボタンを表示します。

2 D11 (例) をクリア

デバイス名	データ型	値	
D10	16bit 整数	3	書込
	32bit 整数	10	書込
M0	ビット	1	書込

× ボタンをクリックし、デバイス名を削除します。

3 D20 を入力

デバイス名	データ型	値	
D10	16bit 整数	3	書込
D20	32bit 整数	10	書込
M0	ビット	1	書込

4 値10 (例) のセルをクリック

デバイス名	データ型	値	
D10	16bit 整数	3	書込
D20	32bit 整数	10	書込
M0	ビット	1	書込

値を選択し、× ボタンを表示します。

5 値10 (例) をクリア

デバイス名	データ型	値	
D10	16bit 整数	3	書込
D11	32bit 整数		書込
M0	ビット	1	書込

× ボタンをクリックし、値を削除します。

6 55 を入力

デバイス名	データ型	値	
D10	16bit 整数	3	書込
	32bit 整数	55	書込
M0	ビット	1	書込

7 書込みボタンをクリック

デバイス名	データ型	値	
D10	16bit 整数	3	書込
D20	32bit 整数	55	書込
M0	ビット	1	書込

シーケンサのD20に55が書き込まれます。

関連ページ

CGI部品を使ったデバイスモニタのWebページ作成方法は、下記を参照してください。

- 扱えるデバイスの仕様
6.3 共通仕様
- デバイス書込みWeb画面の作成手順
6.5 デバイス書込みWeb画面の作成

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読み出しWeb画面の作成

デバイス書き込みWeb画面の作成

6.3 共通仕様

CGI部品で指定するデータ

▶ デバイス名

デバイス読み出しCGI/デバイス書き込みCGIでアクセス可能なデバイスを示します。

種別	デバイス
ビットデバイス*1	X, Y, M, L, B, F, SB, S, TS, TC, STS, STC, CS, CC, LCS, LCC, SM
ワードデバイス*2	T(現在値), ST(現在値), C(現在値), D, W, SW, SD, U□¥G□, Z, R
ダブルワードデバイス	LC(現在値), LZ

*1: 8進数のデバイス名(X***, Y***)を指定するときは、16進数で指定してください。(例: X20を指定する場合、CGIではX10と指定する。)

*2: U□¥G□のデバイス名を指定する時は、U□¥G□と「¥」を重ねて指定してください。1つ目の¥(エスケープシーケンス)が特殊文字を意味する記号のため、このようになります。

▶ デバイスサイズ

デバイスによって、使用可能なデバイスサイズが異なります。デバイス番号の表記は、それぞれのデバイスに応じて8進数/10進数/16進数に分けて指定します。

○: デバイス読み出しCGI/デバイス書き込みCGIの両方で使用可能、×: 使用不可

デバイス		表記	デバイスサイズ		
			ビット	ワード	ダブルワード
ユーザデバイス					
入力 (X)		8進数*1	○	×	×
出力 (Y)		8進数*1	○	×	×
内部リレー (M)		10進数	○	×	×
ラッチリレー (L)		10進数	○	×	×
リンクリレー (B)		16進数	○	×	×
アナンシェータ (F)		10進数	○	×	×
リンク特殊リレー (SB)		16進数	○	×	×
ステップリレー (S)		10進数	○	×	×
タイマ (T) *2	TS (接点)	10進数	○	×	×
	TC (コイル)	10進数	○	×	×
	TN (現在値)	10進数	×	○	○
積算タイマ (ST) *2	STS (接点)	10進数	○	×	×
	STC (コイル)	10進数	○	×	×
	STN (現在値)	10進数	×	○	○
カウンタ (C) *2	CS (接点)	10進数	○	×	×
	CC (コイル)	10進数	○	×	×
	CN (現在値)	10進数	×	○	○
ロングカウンタ (LC) *2	LCS (接点)	10進数	○	×	×
	LCC (コイル)	10進数	○	×	×
	LCN (現在値)	10進数	×	×	○
データレジスタ (D)		10進数	×	○	○
リンクレジスタ (W)		16進数	×	○	○
リンク特殊レジスタ (SW)		16進数	×	○	○
システムデバイス					
特殊リレー (SM)		10進数	○	×	×
特殊レジスタ (SD)		10進数	×	○	○
ユニットアクセスデバイス (U□¥G□)					
G (U□¥G□)		10進数	×	○	○
インデックスレジスタ					
インデックスレジスタ (Z)		10進数	×	○	○
ロングインデックスレジスタ (LZ)		10進数	×	×	○
ファイルレジスタ					
ファイルレジスタ (R)		10進数	×	○	○

*1: CGIでは16進数として扱われます。

*2: T、ST、C、LCを指定した場合、現在値のデバイス(TN、STN、CN、LCN)として扱われます。

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読出しWeb画面の作成

デバイス書き込みWeb画面の作成

▶ デバイス値

CGI 部品で扱うデバイス値の表記は下記としてください。

値の扱い	デバイス値は16進数表記のString型(文字列)で扱います。
JavaScriptによる変換	Webページ上で8進数、10進数または実数の値を使用する場合、JavaScript(FUserWebLib.js)を使用して、16進数への変換を行ってください。
16進数の指定方法	先頭に0xを付加する必要はありません。また、0補間の必要もありません。 例えば0x012Fという値を書き込む場合、12Fと記述します。 読み出した値も同様に0x付加や0補間は行われず、12Fと表示します。
	16進数の英字の大文字/小文字は区別しません。 例えば0x012Fという値を書き込む場合、12F, 12fのどちらでも表記します。 読み出した値は、大文字で表示します。

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読出しWeb画面の作成

デバイス書込みWeb画面の作成

デバイス読出しCGI仕様

レイアウト方法

HTML作成例

6.4 デバイス読出しWeb画面の作成

デバイス読出しCGI仕様

名称	機能
デバイス読出しCGI	指定したデバイスに対して現在値の読み出しを要求する

▶ アクセスメソッドとアクセス情報

CGIとの通信方式には、主にGETとPOSTがあります。Webサーバ機能は、POSTに対応しています。

項目内容	内容
アクセス方式	method="POST"
アクセス先情報 (URL)	/cgi/RdDevRnd.cgi

▶ リクエスト仕様

リクエストで使用するパラメータを示します。パラメータはクエリストリング形式で指定します。

DEV(n)とTYP(n)は連番で指定します。連番で指定しない場合はエラーになります。

パラメータ名	データ型	内容	設定範囲	参照節
NUM	string	読出しデバイス点数 16進数 (n: 1~20) の文字列	Web ページ1ページあたりの読出し / 書込みに指定するデバイス点数の合計が32点以内となるように設定します。	次ページ
DEV1	string	1点目のデバイス名	英数字16文字以内 (大文字 / 小文字は区別しません。間接指定、ビット指定、桁指定、インデックス修飾はできません。)	6.3
TYP1	string	1点目のデバイスサイズ	B: ビット W: ワード D: ダブルワード	6.3
⋮				
DEV(n)	string	n点目のデバイス名	上記パラメータ名 [DEV1] と同じ	6.3
TYP(n)	string	n点目のデバイスサイズ	上記パラメータ名 [TYP1] と同じ	6.3

例 D0、M100 … SD0のデバイス10点を読み出す場合

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

用語説明

クエリストリング形式とは、Webサーバにデータ (パラメータ) を渡すためのフォーマットです。URLの後ろに、[&パラメータ名=値] の形で付与します。複数のパラメータを渡す場合は [&] でつなぎます。
例: http://www.melsec/iq-f¶m1=0¶m2="Sample"

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読みWeb画面の作成

デバイス書き込みWeb画面の作成

デバイス読みCGI仕様

レイアウト方法

HTML作成例

- Web ページ1ページあたりで利用できるデバイス点数は最大32 点以内

例 Web ブラウザ表示画面 合計: 10 点

読み出し

デバイス名	データ型	値
D10	16bit 整数	
D11	16bit 整数	
D12	16bit 整数	
D13	16bit 整数	
D14	16bit 整数	

読み出し

5 点

書き込み

デバイス名	データ型	値	
D10	16bit 整数	25	書き込み
D11	16bit 整数	3	書き込み
D12	16bit 整数	30	書き込み
D13	16bit 整数	55	書き込み
D14	16bit 整数	8	書き込み

5 点

レスポンス仕様

レスポンスで使用するパラメータを示します。レスポンスデータはJSON形式です。

パラメータ名	データ型	内容	
		値	実行結果 (16 進数の文字列) の内容
RET	string	0000	正常
		0001	未ログオン
		0005	要求元 (Referer) 不正
		4005	点数オーバ
		4030	デバイス種別不正
		4031	デバイス範囲外
		4041	指定バッファメモリ番号 + 指定転送点数がバッファメモリ領域外エラー
		4043	未存在ユニット指定エラー
		4080	CGIパラメータの異常
DATA	string	読み出し値 (配列), 16 進数の文字列	

例 デバイス読み CGI のレスポンスデータ

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

左記のレスポンスは、伝文上では下記形で転送します。

正常時 {"RET":"0", "DATA":["100", ..., "FABC"]}

異常時 {"RET":"4031"}

用語説明

JSONとは、“JavaScript Object Notation”の略称です。

データの構造を記述するための言語で、簡易なデータベースとして使用されてます。コンピュータで扱いやすく、使用するユーザにも見やすい形を目指して作成されました。

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読みWeb画面の作成

デバイス書き込みWeb画面の作成

デバイス読みCGI仕様

レイアウト方法

HTML作成例

▶ Webページ上にデバイス値を実数形式で表示する場合



デバイス名	値
D0	B36FH
D1	35A5H

D0とD1に単精度実数で値が格納されています。
図は各デバイス値を16進数でモニタした時の値です。



デバイス名	値
DATA[0]	"35a5b36f"

デバイスサイズにD:ダブルワードを指定してリクエストします。



デバイス名	データ型	値
D0	単精度実数	1.234568E-06

読み出したデータをJavaScriptで実数形式に変換します。

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読み出しWeb画面の作成

デバイス書き込みWeb画面の作成

デバイス読み出しCGI仕様

レイアウト方法

HTML作成例

Webブラウザ表示レイアウト方法

デバイス読み出しCGIでは<table>タグで表組みを作成しています。

セル内に項目名やモニタするデバイス名などを<input>タグを使用して設定しています。

Webブラウザの表示

デバイス名	データ型	値
D10	32bit整数	
SD0	16bit整数	
M0	bit	

読出

上記の表のHTML本文は次のとおりとなります。

デバイス名	データ型	値
D10	32bit整数	
SD0	16bit整数	
M0	bit	

読出

HTMLがどの部分を作成しているか色分けで示します。

HTML文書

```
<form id="devform" name="readdev" method="post">
```

フォーム

```
<table id="devtbl" class="devtbl" border="1">
```

表組み

```
<tr>
```

```
<th>デバイス名</th><th>データ型</th><th>値</th>
```

```
</tr>
```

```
<tbody>
```

```
<tr>
```

```
<td><input type="text" id="DEV1" name="DEV1" class="input" value='D10' /></td>
```

D10
の行

```
<td><input type="text" id="TYP1" name="TYP1" class="input" value='32bit整数' /></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>
```

SD0
の行

```
<td><input type="text" id="TYP2" name="TYP2" class="input" value='16bit整数' /></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>
```

M0
の行

```
<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="読出" class="input" onclick="ReadDeviceRandomTbl(devtbl)"/>
```

```
</form>
```

関連ページ

フォームや表組の詳細は下記を参照してください。

7.2 HTMLタグ リファレンス

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読み出しWeb画面の作成

デバイス書き込みWeb画面の作成

デバイス読み出しCGI仕様

レイアウト方法

HTML作成例

▶ サンプルHTML

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charsetの設定※Webサーバ設定がUTF-8のためUTF-8を設定する -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- タイトル名を設定する -->
    <title>デバイス読み出し CGI サンプル</title>
    <!-- ユーザのJavaScript をここに書く -->
    <script>
      // CGI リクエスト用の関数
      function ReadDeviceRandomTbl(devtblid) {
        var devtblitem = document.getElementById(devtblid);
        var i, devitem, typitem;
        var tblrows = devtblitem.rows.length;
        var param;

        // デバイス点数の設定
        param = "NUM=" + (tblrows - 1) + '&';
        for( i=1; i < tblrows; i++) {
          // デバイス名のパラメータ設定
          devitem = document.getElementById(devtblitem.rows[i].cells[0].childNodes[0].id);
          param += devitem.name + "=" + devitem.value + '&';

          // デバイスサイズのパラメータ設定
          typitem = document.getElementById(devtblitem.rows[i].cells[1].childNodes[0].id);
          if( "ビット" == typitem.value ) {
            param += typitem.name + "=" + 'B';
          }
          else if( "16bit 整数" == typitem.value ) {
            param += typitem.name + "=" + 'W';
          }
          else if( "32bit 整数" == typitem.value ) {
            param += typitem.name + "=" + 'D';
          }
          else {
            param += typitem.name + "=" + 'Q';
          }
          if( i < (tblrows - 1) ) param += '&';
        }
        // CGI へのリクエスト要求
        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); }; // レスポンス解析用関数の設定
        xhr.onreadystatechange = FUNC;
        xhr.send(param);
      }
    </script>
  </head>
  <body>
    <div id="tbl">
      <table>
        <tr>
          <td>デバイス名</td>
          <td>サイズ</td>
        </tr>
        <tr>
          <td>デバイス1</td>
          <td>ビット</td>
        </tr>
        <tr>
          <td>デバイス2</td>
          <td>16bit 整数</td>
        </tr>
        <tr>
          <td>デバイス3</td>
          <td>32bit 整数</td>
        </tr>
        <tr>
          <td>デバイス4</td>
          <td>不明</td>
        </tr>
      </table>
    </div>
  </body>
</html>
```

次のページへつづく

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読み出しWeb画面の作成

デバイス書き込みWeb画面の作成

デバイス読み出しCGI仕様

レイアウト方法

HTML作成例



前のページからのつづき

```
// レスポンス解析用の関数
function ReadDeviceRandomTbl_Response(xhr, devtblid) {
    // XMLHttpRequest クライアント状態のチェック
    // 0:UNSENT 1:OPENED 2:HEADERS_RECEIVED 3:LOADING 4:DONE
    if (4 != xhr.readyState) {
        // ステータス4のDONE(操作完了)以外の場合は、処理を終了する。
        return;
    }

    // HTTP レスポンスコードのチェック
    if (200 != xhr.status) {
        // 「200 OK」以外の場合は、エラーのダイアログ表示をする。
        alert("HTTP STATUS ERROR=" + xhr.status);
        return;
    }

    var i, dataitem;
    var devtblitem = document.getElementById(devtblid);
    var tblrows = devtblitem.rows.length; // テーブルの行数を取得 (ヘッダ含む)
    var res = JSON.parse(xhr.response); // JSON 文字列の解析処理

    // CGI からの結果判断
    if (res.RET != "0000") {
        // 異常の場合エラーのダイアログを表示する。
        alert("ERROR=" + res.RET);
    }
    else {
        // 正常の場合は取得した値をテーブルに反映する。
        for (i = 1, m = 0; i < tblrows; i++, m++) {
            dataitem = document.getElementById(devtblitem.rows[i].cells[2].childNodes[0].id);
            // ここでテーブルの値に読み出し結果を設定 (16 進数文字列を数値に変換)
            dataitem.value = parseInt(res.DATA[m], 16);
        }
        alert("読出完了");
    }
}

</script>
</head>
<body>
    <form>
        <table id="devtbl" class="devtbl" border="1">
            <tr>
                <th>デバイス名</th>
                <th>データ型</th>
                <th>値</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="16bit 整数"/></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="32bit 整数"/></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="ビット"/></td>
                    <td><input type="text" id="DATA3" name="DATA3" class="read-input" /></td>
                </tr>
            </tbody>
        </table>
        <input type="button" value="読出" class="input" onclick="ReadDeviceRandomTbl('devtbl')"/>
    </form>
</body>
</html>
```

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読出しWeb画面の作成

デバイス書込みWeb画面の作成

デバイス書込みCGI仕様

レイアウト方法

HTML作成例

6.5 デバイス書込みWeb画面の作成

デバイス書込みCGI仕様

名称	機能
デバイス書込みCGI	指定したデバイスに対して指定した値の書き込みを指令する

▶ アクセス方式とアクセス情報

CGIとの交信方式は、主にGETとPOSTがあります。Webサーバ機能は、POSTに対応しています。

項目内容	内容
アクセス方式	method="POST"
アクセス先情報 (URL)	/cgi/WrDev.cgi

▶ リクエスト仕様

リクエストで使用するパラメータを示します。パラメータはクエリストリング形式で指定します。

パラメータ名	データ型	内容	設定範囲	参照節
NUM	string	書込みデバイス点数 (1)	1を設定します。 (Web ページ1ページあたりの読出し/書込みに指定するデバイス 点数の合計が32点以内となるようにしてください。)	次ページ
DEV1	string	デバイス名	英数字10文字以内 (大文字/小文字は区別しません。間接指定、ビット指定、桁指定、 インデックス修飾はできません。)	6.3
TYP1	string	デバイスサイズ	B: ビット W: ワード D: ダブルワード	6.3
DATA1	string	書込み値	16進数の文字列	—

例 D0にFFFFを書き込む場合

NUM=1&DEV1=D0&TYP1=W&DATA1=FFFF

用語説明

クエリストリング形式とは、Webサーバにデータ (パラメータ) を渡すためのフォーマットです。
URLの後ろに、[&パラメータ名=値] の形で付与します。複数のパラメータを渡す場合は [&] でつなぎます。
例: http://www.melsec/iq-f¶m1=0¶m2="Sample"

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読出しWeb画面の作成

デバイス書込みWeb画面の作成

デバイス書込みCGI仕様

レイアウト方法

HTML作成例

- Webページ1ページあたりで利用できるデバイス点数は最大32点以内

例 Webブラウザ表示画面 合計: 10点

読出し

デバイス名	データ型	値
D10	16bit 整数	
D11	16bit 整数	
D12	16bit 整数	
D13	16bit 整数	
D14	16bit 整数	

読出

5点

書込み

デバイス名	データ型	値	
D10	16bit 整数	25	書込
D11	16bit 整数	3	書込
D12	16bit 整数	30	書込
D13	16bit 整数	55	書込
D14	16bit 整数	8	書込

5点

レスポンス仕様

レスポンスで使用するパラメータを示します。レスポンスデータはJSON形式です。

パラメータ名	データ型	内容	
		値	実行結果(16進数の文字列)の内容
RET	string	0000	正常
		0002	権限なし(デバイス書込み許可権限がないユーザがCGI実行)
		0001	未ログオン
		0005	要求元(Referer)不正
		4005	点数オーバー
		4030	デバイス種別不正
		4031	デバイス範囲外
		4041	指定バッファメモリ番号+指定転送点数がバッファメモリ領域外エラー
		4043	未存在ユニット指定エラー
		4080	CGIパラメータの異常
DATA	string	書き込んだデバイスの読出し値(配列),16進数の文字列	

例 デバイス書込みCGIのレスポンスデータ

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

左記のレスポンスは、伝文上では下記形で転送します。

正常時 {"RET":"0","DATA":["100"]}

異常時 {"RET":"4031"}

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読み出しWeb画面の作成

デバイス書き込みWeb画面の作成

デバイス書き込みCGI仕様

レイアウト方法

HTML作成例

▶ Webページ上にデバイス値を実数形式で表示する場合



デバイス名	データ型	値
D0	単精度実数	1.234568E-06

入力したデータをJavaScriptで16進数に変換します。

デバイス名	値
DATA1	"35a5b36f"

デバイスサイズにD:ダブルワードを指定してリクエストします。



デバイス名	値
D0	B36FH
D1	35A5H

D0とD1に単精度実数で値が格納されます。
図は各デバイス値を16進数でモニタした時の値です。

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読みWeb画面の作成

デバイス書き込みWeb画面の作成

デバイス書き込み CGI 仕様

レイアウト方法

HTML 作成例

Webブラウザ表示レイアウト方法

デバイス書き込みは<table>タグで表組みを作成しています。

セル内に項目名やモニタするデバイス名などを<input>タグを使用して設定しています。

Webブラウザの表示

デバイス名	データ型	値	
D10	32bit 整数	3	書込

上記の表のHTML本文は次のとおりとなります。

デバイス名	データ型	値	
D10	32bit 整数	3	書込

HTMLがどの部分を作成しているか色分けで示します。

HTML文書

```
<form id="devform" name="readdev" method="post">
```

```
<table id="devtbl" class="devtbl" border="1">
```

```
<tr>
```

```
<th>デバイス名</th><th>データ型</th><th>値</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='32bit 整数'/></td>
```

```
<td><input type="text" id="DATA1" name=" DATA1" class=" input"value='3'/></td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

```
<input type="button" value="書込" class="input" onclick=" WriteDeviceBlockTbl(devbl,1,1)"/>
</form>
```

フォーム

表組み

6. CGIによるデバイスモニタ画面を作成しよう

デバイス書込み Web 画面の作成

デバイス書込み CGI 仕様

レイアウト方法

HTML 作成例

HTML作成例

下記の手順により、デバイス書込みWeb画面を作成します。

STEP1. 次ページのHTMLをメモ帳で作成します。

[illegible]

HTMLに関しては、MELSEC iQ-F FX5ユーザズマニュアル (Ethernet通信編) [JY997D55101] にも記載があります。

操作>> [ファイル] ▶ [名前を付けて保存] を選択

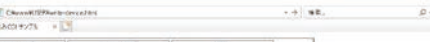
▶ファイル名 (例: write-device.html) を入力

▶文字コードに (UTF-8) を選択 ▶[保存] をクリック

Point

文字コードにUTF-8以外を指定した場合は、Webブラウザで表示時に文字化けすることがあります。

STEP3. 作成したファイルをダブルクリックしWebブラウザで表示します。



The screenshot shows a web browser window with the address bar displaying "http://localhost:8080/". The page title is "デバイス管理". Below the title, there is a table with the following data:

デバイス名	データ型	値	
D10	16bit 整数	3	表示
I11	32bit 整数	10	表示
E40	ビット	1	表示

STEP4. 作成したHTMLファイルをSDメモ리카ードに格納し、シーケンサのWebサーバ機能で表示します。

6. CGIによるデバイスモニター画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読み出しWeb画面の作成

デバイス書き込みWeb画面の作成

デバイス書き込みCGI仕様

レイアウト方法

HTML作成例

▶ サンプルHTML

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- charsetの設定※Webサーバ設定がUTF-8のためUTF-8を設定する -->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <!-- タイトル名を設定する -->
    <title>デバイス書き込みCGI サンプル</title>
    <!-- ユーザのJavaScriptをここに書く -->
    <script>
      function WriteDeviceBlockTbl(devtblid, row) {
        var dataitem;
        var xhr;
        var devtblitem = document.getElementById(devtblid);
        // デバイス点数の設定(1点固定)
        var param = 'NUM=1&';
        // デバイス名のパラメータ設定
        var devitem = document.getElementById(devtblitem.rows[row].cells[0].childNodes[0].id);
        param += 'DEV1=' + devitem.value + '&';

        // デバイスサイズのパラメータ設定
        var typitem = document.getElementById(devtblitem.rows[row].cells[1].childNodes[0].id);
        if('ビット' == typitem.value) {
          param += 'TYP1=' + 'B';
        }
        else if('16bit 整数' == typitem.value) {
          param += 'TYP1=' + 'W';
        }
        else if('32bit 整数' == typitem.value) {
          param += 'TYP1=' + 'D';
        }
        else {
          param += 'TYP1=' + 'Q';
        }
        param += '&';

        // データのパラメータ設定
        var dataitem = document.getElementById(devtblitem.rows[row].cells[2].childNodes[0].id);
        param += 'DATA1=' + parseInt(dataitem.value).toString(16);

        // CGIへのリクエスト要求
        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); }; // レスポンス解析用関数の設定
        xhr.onreadystatechange = FUNC;
        xhr.send(param);
      }
    </script>
  </head>
  <body>
    <table id="devtbl">
      <tr>
        <td>デバイス名</td>
        <td>デバイスサイズ</td>
        <td>デバイスデータ</td>
      </tr>
    </table>
  </body>
</html>
```



次のページへつづく

6. CGIによるデバイスモニタ画面を作成しよう

CGIとは

CGIの概要と機能

共通仕様

デバイス読み出しWeb画面の作成

デバイス書き込みWeb画面の作成

デバイス書き込みCGI仕様

レイアウト方法

HTML作成例

前のページからのつづき

```
// レスポンス解析用の関数
function WriteDeviceBlockTbl_Response(xhr, typitem, dataitem) {
    // XMLHttpRequest クライアント状態のチェック
    // 0:UNSENT 1:OPENED 2:HEADERS_RECEIVED 3:LOADING 4:DONE
    if (4 != xhr.readyState) {
        // ステータス4 のDONE (操作完了) 以外の場合は、処理を終了する。
        return;
    }

    // HTTPレスポンスコードのチェック
    if (200 != xhr.status) {
        // 「200 OK」以外の場合は、エラーのダイアログ表示をする。
        alert("HTTP STATUS ERROR=" + xhr.status);
        return;
    }

    var value;
    var res = JSON.parse( xhr.response ); // JSON 文字列の解析処理
    // CGI からの結果判断
    if (res.RET != "0000") {
        // 異常の場合エラーのダイアログを表示する。
        alert("ERROR=" + res.RET);
    }
    else {
        // 正常の場合は、書き込み結果値の値を反映する。
        dataitem.value = parseInt(res.DATA[0],16);
        alert("書き込み完了");
    }
}
</script>
</head>

<body>
<form>
<table id="devtbl" class="devtbl" border="1">
<tr>
<th>デバイス名</th>
<th>データ型</th>
<th>値</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="16bit 整数"/></td>
<td><input type="text" id="DATA1" name="DATA1" class="input" value="3"/></td>
<td><input type="button" value=" 書込 " 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="32bit 整数"/></td>
<td><input type="text" id="DATA2" name="DATA2" class="input" value="10"/></td>
<td><input type="button" value=" 書込 " 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="ビット"/></td>
<td><input type="text" id="DATA3" name="DATA3" class="input" value="1"/></td>
<td><input type="button" value=" 書込 " class="input" onclick="WriteDeviceBlockTbl('devtbl',3)"/></td>
</tr>
</tbody>
</table>
</form>
</body>
</html>
```

7. リファレンス

Style Sheet

HTMLタグ

カラーネーム・カラーコード

プロパティ一覧

解説

使用例による説明

7.1 Style Sheetリファレンス

ここでは、Style Sheetの主要なプロパティの働きを簡単に解説します。
Style Sheetの使用方法などの詳細は、市販の参考書籍をご覧ください。

プロパティ一覧

・ユーザWebページのライブラリファイル (UserWebStyle.css) に使用しているプロパティの動作について説明します。

No.	プロパティ	内容
1	position:	要素の位置を決める
2	z-index:	要素の重なり順序を指定
3	width:	要素の横幅
4	height:	要素の高さ
5	margin-top:	要素の外側上余白

No.	プロパティ	内容
6	margin-left:	要素の外側左余白
7	background-color:	要素の背景色
8	border:	要素の線枠
9	text-align:	文字の位置指定
10	font-weight:	文字の太さ
11	cursor:	カーソル形状

解説

▶ No. 1 [position:] … 要素の位置を決める

値	内容
static	配置方法を指定しない (初期値)
absolute	絶対位置への配置 (基準位置: Web ブラウザ (ウィンドウ全体) の左上)
relative	相対位置への配置
fixed	絶対位置への配置 (基準位置: Web ブラウザ (現在表示) の左上)

HTML

```
<div class="parent">
  <p>要素①</p>
  <p id="absolute">要素②</p>
</div>
```

Style Sheet

```
.parent {
  padding: 20px;
  position: relative;
  /* 親要素は relative に */
}

#absolute {
  position: absolute;
  top: 0;
  left: 0;
}

/* 装飾は省略 */
```

ブラウザ表示

要素②

要素①

7. リファレンス

Style Sheet

HTML タグ

カラーネーム・カラーコード

プロパティ一覧

解説

使用例による説明

▶ No. 2 [z-index:] … 要素の重なり順を指定

値	内容
整数値	重なり順を整数で指定 (0を基準として、値が大きいものほど上側になる)
auto	親要素と同じ階層になります。(初期値)

HTML

```
<div id="example">
  <p class="one">1 番目 </p>
  <p class="two">2 番目 </p>
  <p class="three">3 番目 </p>
</div>
```

Style Sheet

```
/* 親要素を relative に */
#example { position: relative; }

/* 3 つとも absolute に */
.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; }

/* 色付けなどは省略 */
```

z-index を指定しないと、後に記述された要素が上に重なります。



▶ No. 3 [width:] … 要素の横幅

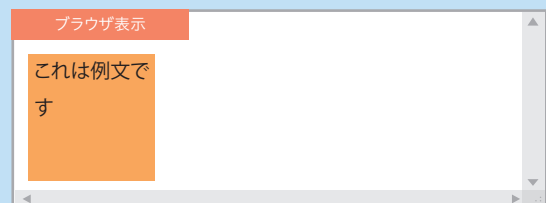
▶ No. 4 [height:] … 要素の高さ

HTML

```
<p>これは例文です</p>
```

Style Sheet

```
p { width: 100px;
    height: 100px;
    background: orange;
}
```



7. リファレンス

Style Sheet

HTML タグ

カラーネーム・カラーコード

プロパティ一覧

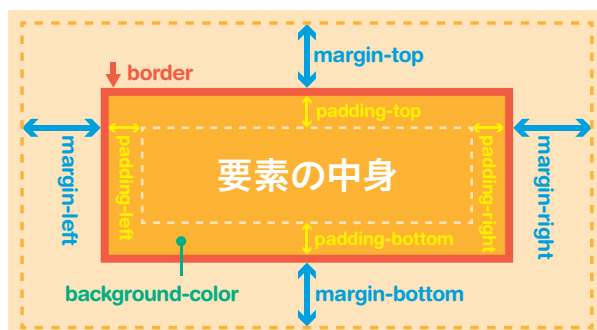
解説

使用例による説明

▶ No. 5 [margin-top:] … 要素の外側上余白

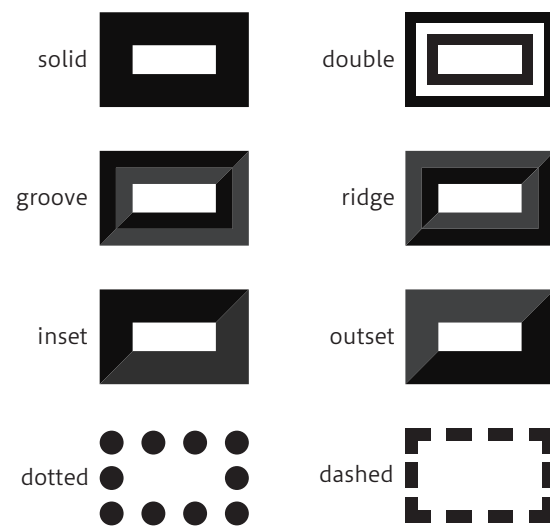
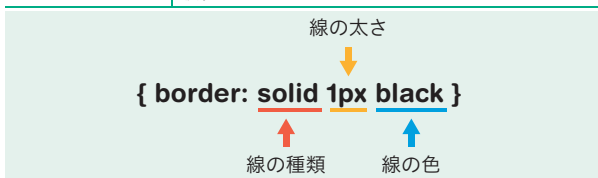
▶ No. 6 [margin-left:] … 要素の外側左余白

▶ No. 7 [background-color:] … 要素の背景色



▶ No. 8 [border:] … 要素の線枠

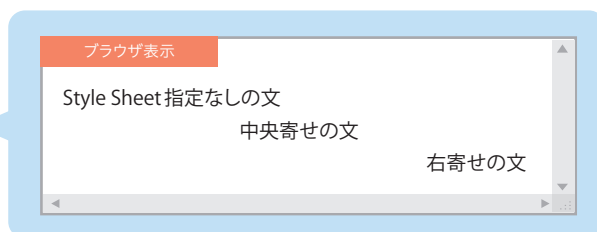
値	内容
none	線なし (初期値)
hidden	線なし
solid	実線 (一本線)
double	二重線 (二本線)
groove	凹型の線
ridge	凸型の線
inset	内側凹型の線 (上左が影)
outset	内側凸型の線 (下右が影)
dotted	点線
dashed	破線



▶ No. 9 [text-align:] … 文字の位置指定

値	内容
left	左寄せ (初期値)
center	中央寄せ
right	右寄せ

HTML
<code><p>Style Sheet 指定なしの文</p></code> <code><p class="center">中央寄せの文</p></code> <code><p class="right">右寄せの文</p></code>
Style Sheet
<code>.center { text-align: center; }</code> <code>.right { text-align: right; }</code>



▶ No. 10 [font-weight:] … 文字の太さ

値	内容
normal	標準の太さ
bold	一般的な太字の太さ
lighter	相対的に一段階細く
bolder	相対的に一段階太く

7. リファレンス

Style Sheet

HTML タグ

カラー・ネーム・カラー・コード

プロパティ一覧

解説

使用例による説明

▶ No. 11 [cursor:] … カーソル形状

値	内容	形状
auto	初期値 (状況に応じて自動選択)	—
default	デフォルト	
pointer	リンク	
crosshair	十字	
context-menu	コンテキストメニュー	
cell	セル選択	
help	ヘルプ	

〇〇 {cursor: 値;}
〇〇には「何にカーソルをのせたときに形状を

変えるか」ということを示すセレクタを記述します。

カーソルの形状は、ユーザ Web ページを表示する端末の OS により異なります。

使用例による説明

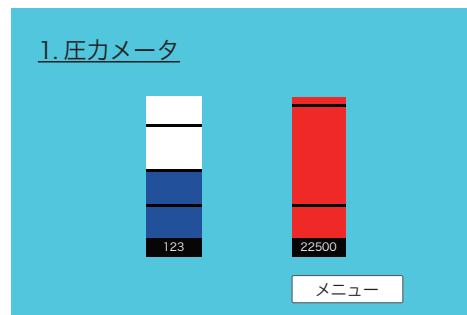
「3. カスタマイズで簡単にユーザ Web ページを作成しよう」で使用した、“メニュー” ボタンの Style Sheet のプロパティの動作について説明します。

No.	プロパティ	内容
A	position:	座標の絶対、相対の指定
B	left:	Web ブラウザの左からの座標位置
C	top:	Web ブラウザの上からの座標位置

HTML

```
<a href="index.html" class="menu" style="position: absolute; left: 350px; top: 550px;" />メニュー</a>
```

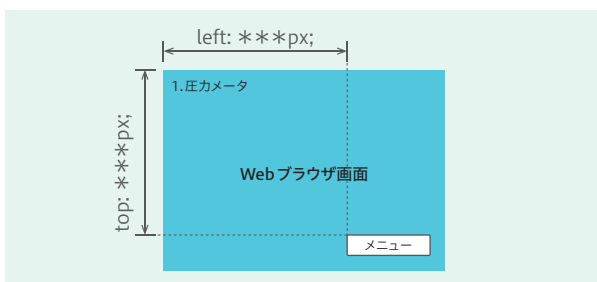
HTML<a>タグに記載した Style Sheet のパラメータ部分



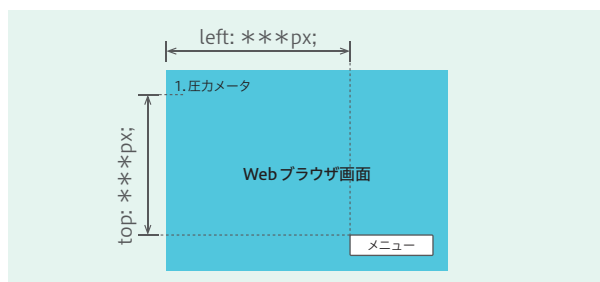
▶ No. A [position:] … 表示座標の指定方法の選択

値	内容	備考
static	配置方法を指定しない	top,bottom,left,right は適用されず、初期値に配置される。
absolute	絶対位置への配置	Web ブラウザ (ウィンドウ全体) の左上が基準位置となり、top,bottom,left,right の座標値を適用する。
relative	相対位置への配置	position プロパティで static を指定した場合に表示される位置が基準位置となり、top,bottom,left,right の座標値を適用する。
fixed	絶対位置への配置	absolute と同じ。ただし、現在表示している Web ブラウザの左上が基準位置になる。(スクロールしても追従するように絶対位置で要素を表示する)

・ position: absolute を設定時 (本書で使用した設定)



・ position: relative を設定時



7. リファレンス

Style Sheet

HTMLタグ

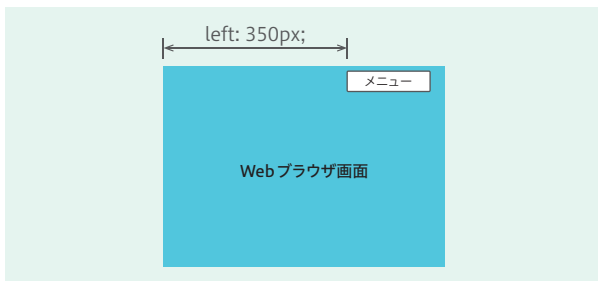
カラーネーム・カラーコード

プロパティ一覧

解説

使用例による説明

▶ No. B [left:] … Webブラウザの左からの座標位置



▶ No. C [top:] … Webブラウザの上からの座標位置



7.2 HTMLタグ リファレンス

タグ（不等号記号: "<" ">" で囲まれた）指定により、文字、画像表示などさまざまな指定を行います。
タグには、入れ子または単独で使用するものがあります。

タグ一覧

本書で使用しているタグの働きについて簡単に説明します。
ユーザWebページのライブラリファイル (index.html) には使用していませんが、「3.カスタマイズで簡単にユーザWebページを作成しよう」で使用したタグを含みます。

値	タグ	内容
HTML 定義	<meta>	宣言
文字列修飾	 	改行
	<p>	段落
	<i>	斜体
	<u>	下線
画像		画像ファイルの画像表示
リンク	<a> ~ 	別 Web ページへのリンク
テーブル	<table> ~ </table>	表組み

HTML 定義

機能	始まり	終わり	HTML 文書	Web ブラウザ 表示
宣言	<meta>	なし	<meta charset="Shift_JIS">	—

要素

charset	漢字コード
ISO-2022-JP	JIS コード
Shift_JIS	Shift JIS コード
EUC-JP	EUC コード

7.リファレンス

Style SheetHTMLタグカラーネーム・カラーコード

タグ一覧

HTML定義

文字列修飾

画像

リンク

テーブル

文字列修飾

▶ テキスト

タグ			HTML 文書	Web ブラウザ 表示
機能	始まり	終わり		
改行 Break	 	—	1 行目を表示します。 2 行目を表示します。	1 行目を表示します。 2 行目を表示します。
段落 Paragraph	<p>	</p>	1 行目を表示します。 2 行目を表示します。	1 行目を表示します。 2 行目を表示します。

▶ フォント

タグ			HTML 文書	Web ブラウザ 表示
機能	始まり	終わり		
斜体	<i>	</i>	<i>1 行目を表示します。</i>	1 行目を表示します。
下線	<u>	</u>	2 行目を<u>表示します。</u>	2 行目を表示します。

画像

タグ			HTML 文書	Web ブラウザ 表示
機能	始まり	終わり		
画像表示		—		

要素	内容	画像例  sample0.png サイズ:30 × 30
src="画像ファイル名"	画像ファイルの格納場所とファイル名を設定	
alt="代替えテキスト"	画像の代わりに表示される（読み上げられる）テキストを指定	
border="0"	画像の周りの枠線をピクセルで指定（0: 枠線非表示）	
width="300"	横幅指定	
height="10"	高さ指定	

7. リファレンス

Style Sheet

HTMLタグ

カラーネーム・カラーコード

タグ一覧

HTML定義

文字列修飾

画像

リンク

テーブル

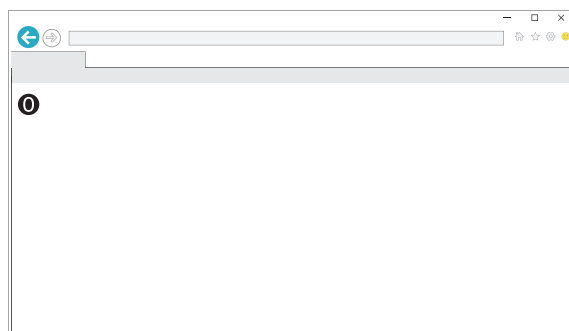
参考

ユーザWebページのライブラリの中にある、画像ファイル (sample0.png) を例として説明します。

STEP1. 画像を新規作成し、そのファイル (例: sample0.gif) をパソコンに保存します。

STEP2. 作成した画像のサイズを確認します。

画像をWebブラウザヘドラッグし、表示します。



STEP3. 表示した画像の上で右クリックし、プロパティを表示します。

1/1で表示するとき、大きさ: に表示されているサイズ (pixel) を入力します。

設定例

```
width="30"
height="30"
```

例えば、30pixelを15pixelに変更した時は、1/4に縮小したサイズで表示します。
画像ファイルのサイズを100%として、%値により表示サイズの変更も可能です。



STEP4. HTMLの本文<body>～</body>間に下記HTMLを追加します。

設定例

```
src="./img/sample0.png"
```

HTML

```

```

7. リファレンス

Style Sheet

HTMLタグ

カラーネーム・カラーコード

タグ一覧

HTML定義

文字列修飾

画像

リンク

テーブル

リンク

タグ			HTML文書	Webブラウザ表示
機能	始まり	終わり		
リンク	<code><a></code>	<code></code>	<code>2. 温度推移グラフ</code>	青字と下線表示でリンクを表す 2. 温度推移グラフ

HTML

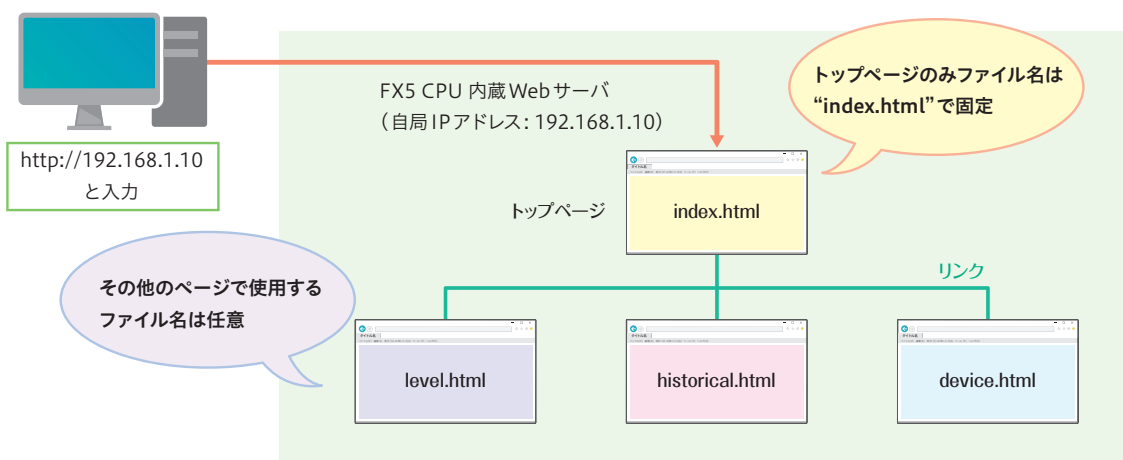
```
<p>
<a href="level.html">1. 圧力メータ</a>
</p>
<p>
<a href="historical.html">2. 温度推移グラフ</a>
</p>
<p>
<a href="device.html">3. デバイスモニタ</a>
</p>
```

ブラウザ表示

1. 圧力メータ

2. 温度推移グラフ

3. デバイスモニタ



7. リファレンス

Style Sheet

HTMLタグ

カラーネーム・カラーコード

タグ一覧

HTML定義

文字列修飾

画像

リンク

テーブル

テーブル

タグ			HTML文書	Webブラウザ表示
機能	始まり	終わり		
表	<code><table></code>	<code></table></code>	<code><table></code> 装置名 <code></table></code>	表全体を <code><table></code> ~ <code></table></code> タグで囲む 装置名
表の行	<code><tr></code>	<code></tr></code>	<code><table border="1"></code> <code><tr><td>装置-1</td><td>装置-2</td></tr></code> <code></table></code>	各行を <code><tr></code> ~ <code></tr></code> タグで作成する 装置-1 装置-2
表の見出し	<code><th></code>	<code></th></code>	<code><table border="1"></code> <code><tr><th>装置名</th><th>生産数</th></tr></code> <code></table></code>	各行を <code><th></code> ~ <code></th></code> タグで作る文字が強調される 装置名 生産数
表の内容	<code><td></code>	<code></td></code>	<code><table border="1"></code> <code><tr><td>装置-1</td></tr></code> <code><tr><td>装置-2</td></tr></code> <code></table></code>	各列を <code><td></code> ~ <code></td></code> タグで作成する 装置-1 装置-2

属性一覧

本書は、一部の属性のみ記載しています。

属性の設定を行っていない場合は、0として扱われます。

▶ 主な属性一覧

機能	属性	HTML文書	Webブラウザ表示
枠線	border	<code><table border="1"></code>	装置名
セル色	bgcolor	<code><table bgcolor="green"></code>	装置名
列(横方向)の結合を指定する属性	colspan	<code><td colspan="結合するセルの数"></code>	colspan="1" A B C 装置名 colspan="3" 装置名
行(縦方向)の結合を指定する属性	rowspan	<code><td rowspan="結合するセルの数"></code>	rowspan="1" rowspan="3" 1 装置名 2 3 装置名

1

HTMLヘタケV作成の基礎

2

パソコンで簡単なHTML文書を作成し表示してみよう

3

カスタマイズで簡単にユーザーWebページを作成しよう

4

Style Sheet(CSS)の働きを知ろう

5

JavaScript(JS)で表示できる部品の働きを知る

6

CGIによるデバイスモニタ画面を作成しよう

7

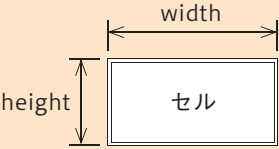
リファレンス

7.リファレンス

Style SheetHTMLタグカラーネーム・カラーコード

タグ一覧HTML定義文字列修飾画像リンクテーブル

セルのサイズ指定

機能	属性	HTML文書	Webブラウザ表示
セルの幅	width	<td width="200" height="100"> 装置名</td>	
セルの高さ	height		

文字表示位置指定

	機能	属性	HTML文書	Webブラウザ表示
横方向	文字揃え (左寄せ) (tdタグの初期値)	left	<td align="left">	装置名
	文字揃え (中央) (thタグの初期値)	center	<td align="center">	装置名
	文字揃え (右寄せ)	right	<td align="right">	装置名
縦方向	文字揃え (上寄せ)	top	<td valign="top">	装置名
	文字揃え (中央) (初期値)	middle	<td valign="middle">	装置名
	文字揃え (下寄せ)	bottom	<td valign="bottom">	装置名

表作成例

	A	B	C	D	
1行目	A-1		C-1	D-1	文字 中央揃え
2行目	A-2	B-2	C-2	D-2	文字 左揃え (初期値)
3行目	A-3	B-3	C-3		文字 左揃え (初期値)

HTML

	<table border="1">
	<tr bgcolor="green"><th width="100">A</th><th width="100">B</th><th width="100">C</th><th width="100">D</th></tr>
1行目	<tr align="center" bgcolor="white"><td colspan="2">A-1</td><td>C-1</td><td>D-1</td></tr>
2行目	<tr bgcolor="white"><td>A-2</td><td>B-2</td><td>C-2</td><td rowspan="2">D-2</td></tr>
3行目	<tr bgcolor="white"><td>A-3</td><td>B-3</td><td>C-3</td></tr>
	</table>

7. リファレンス

Style Sheet

HTMLタグ

カラーネーム・カラーコード

16色 (基本色)

Web Safe Color 216色

Web color table 147色

7.3 カラーネーム・カラーコード

16色 (基本色)

HTMLで定義されている基本色 (16色) です。

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

1

HTMLヘタケV作成の基礎

2

パソコンで簡単なHTML文書を作成し表示してみよう

3

カスタマイズで簡単にユーザーWebページを作成しよう

4

Style Sheet (CSS) の働きを知る

5

JavaScript (JS) で表示できる部品の働きを知る

6

CGIによるデバイスモニタ画面を作成しよう

7

リファレンス

7. リファレンス

Style Sheet

HTMLタグ

カラーネーム・カラーコード

16色(基本色)

Web Safe Color 216色

Web color table 147色

Web Safe Color 216色

Web Safe Colorとは、Red (赤色) Green (緑色) Blue (青色) をそれぞれ6段階 216色 (6×6×6) に分けて定義している色の集合体です。Web Safe Colorで配色するメリットは、216色の限られた範囲ですがコンピューターの性能に大きく左右されず、ほぼ同じ色を再現できるメリットがあります。

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. リファレンス

Style Sheet

HTMLタグ

カラーネーム・カラーコード

16色 (基本色)

Web Safe Color 216色

Web color table 147色

Web color table 147色

Color Name で指定できる色数は147色あります。

Color	HTML / CSS Color Name	Hex Code #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 Color Name	Hex Code #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.リファレンス

Style Sheet

HTMLタグ

カラーネーム・カラーコード

16色(基本色)

Web Safe Color 216色

Web color table 147色

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	#f0ffff
	mintcream	#f5fffa
	azure	#f0ffff
	aliceblue	#f0f8ff
	ghostwhite	#f8f8ff
	whitesmoke	#f5f5f5
	seashell	#fff5ee
	beige	#f5f5dc
	oldlace	#fdf5e6
	floralwhite	#fffaf0
	ivory	#fffff0
	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	#ffebed
	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

保証について

- ご使用に際しましては、下記の製品保証内容をご確認いただきますよう、よろしくお願いいたします。
- ・ MELSEC iQ-F FX5U ユーザーズマニュアル (ハードウェア編) マニュアル番号: JY997D54201 「保証について」
 - ・ MELSEC iQ-F FX5UC ユーザーズマニュアル (ハードウェア編) マニュアル番号: JY997D61301 「保証について」

⚠ 安全にお使いいただくために

- ・ 本資料に記載された製品を正しくお使いいただくためご使用前に必ず「マニュアル」をお読みください。
- ・ この製品は一般工業等を対象とした汎用品として製作されたもので、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。
- ・ 本製品を原子力用、電力用、航空宇宙用、医療用、乗用移動体用の機器あるいはシステムなど特殊用途への適用をご検討の際は、巻末記載の当社営業窓口まで照会してください。
- ・ 本製品は厳重な品質管理体制の下に製造しておりますが、本製品の故障により重大な事故または損失の発生が予測される設備への適用に際しては、バックアップやフェールセーフ機能をシステムに設置してください。
- ・ 設計上の注意、配線上の注意等に関しましては各関連マニュアルに記載の安全上のご注意をお読みください。

商標

Microsoft, Windows, および Internet Explorer は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
Google Chrome は、Google Inc. の登録商標または商標です。
SD ロゴ, SDHC ロゴは、SD-3C, LLC の登録商標または商標です。
IOS (iOS) は、Cisco Systems, Inc. およびその関連会社の米国およびその他の国における登録商標または商標であり、Apple Inc. がライセンスに基づき使用しています。
Ethernet は、富士ゼロックス株式会社の日本における登録商標です。
Oracle および JavaScript は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
本文中における会社名、システム名、製品名などは、一般に各社の登録商標または商標です。
本文中で、商標記号 (™, ®) は明記していない場合があります。

改訂履歴

＊ ドキュメント番号は、本書の裏表紙の左下に記載してあります。

改訂年月	ドキュメント番号	改訂内容
2019年9月	L (名) 08644-A	初版

本書によって、工業所有権その他の権利の実施に対する保証、または実施権を許諾するものではありません。また本書の掲載内容の使用により起因する工業所有権上の諸問題については、当社は一切その責任を負うことができません。

三菱電機マイクロシーケンサ MELSEC iQ-Fシリーズ

三菱電機株式会社 〒100-8310 東京都千代田区丸の内2-7-3(東京ビル)

お問い合わせは下記へどうぞ

本社(機器営業部).....	〒110-0016 東京都台東区台東1-30-7(秋葉原アイマークビル)	(03)5812-1450
北海道支社.....	〒060-8693 札幌市中央区北二条西4-1(北海道ビル)	(011)212-3794
東北支社.....	〒980-0013 仙台市青葉区花京院1-1-20(花京院スクエア)	(022)216-4546
関東支社.....	〒330-6034 さいたま市中央区新都心11-2(明治安田生命さいたま新都心ビル)	(048)600-5835
新潟支店.....	〒950-8504 新潟市中央区東大通2-4-10(日本生命ビル)	(025)241-7227
神奈川支社.....	〒220-8118 横浜西区みなとみらい2-2-1(横浜ランドマークタワー)	(045)224-2624
北陸支社.....	〒920-0031 金沢市広岡3-1-1(金沢パークビル)	(076)233-5502
中部支社.....	〒450-6423 名古屋市中村区名駅3-28-12(大名古屋ビルヂング)	(052)565-3314
豊田支店.....	〒471-0034 豊田市小坂本町1-5-10(矢作豊田ビル)	(0565)34-4112
関西支社.....	〒530-8206 大阪市北区大深町4-20(グランフロント大阪 タワーA)	(06)6486-4122
中国支社.....	〒730-8657 広島市中区中町7-32(ニッセイ広島ビル)	(082)248-5348
四国支社.....	〒760-8654 高松市寿町1-1-8(日本生命高松駅前ビル)	(087)825-0055
九州支社.....	〒810-8686 福岡市中央区天神2-12-1(天神ビル)	(092)721-2247

三菱電機 FA

検索

www.MitsubishiElectric.co.jp/fa

メンバー登録無料!

インターネットによる情報サービス「三菱電機FAサイト」

三菱電機FAサイトでは、製品や事例などの技術情報に加え、トレーニングスクール情報や各種お問い合わせ窓口をご提供しています。また、メンバー登録いただくとマニュアルやCADデータ等のダウンロード、eラーニングなどの各種サービスをご利用いただけます。

電話技術相談窓口 受付時間※1 月曜～金曜 9:00～19:00、土曜・日曜・祝日 9:00～17:00

対象機種		電話番号
自動窓口案内		052-712-2444
エッジコンピューティング製品	産業用 PC MELIPC	052-712-2370※2
	Edgecross 対応ソフトウェア (MTConnect データコレクタを除く)	
シーケンサ	MELSEC iQ-R/Q/L/QnAS/AnS シーケンサ (CPU 内蔵 Ethernet 機能などネットワークを除く)	052-711-5111
	MELSEC iQ-F/FX シーケンサ 全般	052-725-2271※3
	ネットワークユニット (CC-Link ファミリー / MELSECNET / Ethernet / シリアル通信)	052-712-2578
	MELSOFT シーケンサ エンジニアリングソフトウェア (MELSEC iQ-R/Q/L/QnAS/AnS)	052-711-0037
	MELSOFT 統合エンジニアリング環境 iQ Sensor Solution	052-799-3591※2
	MELSOFT 通信支援ソフトウェア ツール	
	MELSEC パソコンボード	052-712-2370※2
	Q80BD シリーズ など	
	C 言語コントローラ	052-799-3592※2
	MES インタフェースユニット / 高速データロガーユニット	
MELSEC 計装 / iQ-R/Q 二重化	プロセス CPU / 二重化機能 SIL2 プロセス CPU (MELSEC iQ-R シリーズ)	052-712-2830※2※3
	プロセス CPU / 二重化 CPU (MELSEC-Q シリーズ)	
	MELSOFT PX シリーズ	
MELSEC Safety	安全シーケンサ (MELSEC iQ-R/QS シリーズ)	052-712-3079※2※3
	安全コントローラ (MELSEC-WS シリーズ)	
電力計測ユニット / 絶縁監視ユニット	QE シリーズ / RE シリーズ	052-719-4557※2※3
対象機種		電話番号
FA センサ MELSENSOR	レーザ変位センサ ビジョンセンサ コードリーダー	052-799-9495※2
表示器 GOT	GOT2000/1000 シリーズ MELSOFT GT シリーズ	052-712-2417
SCADA MC Works64		052-712-2962※2※6
サーボ / 位置決めユニット / モーションユニット / シンプルモーションユニット / モーションコントローラ / センシングユニット / 組込み型サーボシステム コントローラ	MELSERVO シリーズ 位置決めユニット (MELSEC iQ-R/Q/L/AnS シリーズ) モーションユニット (MELSEC iQ-R シリーズ) シンプルモーションユニット (MELSEC iQ-R/iQ-F/Q/L シリーズ) モーション CPU (MELSEC iQ-R/Q/AnS シリーズ) センシングユニット (MR-MT シリーズ) シンプルモーションボード / ボジションボード MELSOFT MT シリーズ / MR シリーズ / EM シリーズ	052-712-6607
	センサレスサーボ	052-722-2182
	インバータ	052-722-2182
	三相モータ	0536-25-0900※2※4
	産業用ロボット	052-721-0100
電磁クラッチ・ブレーキ / テンションコントローラ		052-712-5430※5
	データ収集アナライザ	052-712-5440※5
低圧開閉器	MS-T シリーズ / MS-N シリーズ US-N シリーズ	052-719-4170
低圧遮断器	ノーヒューズ遮断器 / 漏電遮断器 MDU ブレーカ / 気中遮断器 (ACB) など	052-719-4559
電力管理用計器	電力量計 / 計器用変成器 / 指示電気計器 管理用計器 / タイムスイッチ	052-719-4556
省エネ支援機器	EcoServer / E-Energy / 検針システム / エネルギータラントユニット / B/NET など	052-719-4557※2※3
小容量 UPS (5kVA 以下)	FW-S シリーズ / FW-V シリーズ / FW-A シリーズ / FW-F シリーズ	052-799-9489※2※6

お問い合わせの際には、今一度電話番号をお確かめの上、お掛け間違いのないようお願い致します。

FAX 技術相談窓口 受付時間 月曜～金曜 9:00～16:00 (祝日・当社休日を除く)

対象機種	FAX 番号
電力計測ユニット / 絶縁監視ユニット (QE シリーズ / RE シリーズ)	084-926-8340
三相モータ225フレーム以下	0536-25-1258※7
低圧開閉器	0574-61-1955
低圧遮断器	084-926-8280
電力管理用計器 / 省エネ支援機器 / 小容量 UPS (5kVA 以下)	084-926-8340

三菱電機FAサイトの「仕様・機能に関するお問い合わせ」もご利用ください。

- ※1：春季・夏季・年末年始の休日を除く
- ※2：土曜・日曜・祝日を除く
- ※3：金曜は17:00まで
- ※4：月曜～木曜の9:00～17:00と金曜の9:00～16:30
- ※5：受付時間9:00～17:00(土曜・日曜・祝日・当社休日を除く)
- ※6：月曜～金曜の9:00～17:00
- ※7：月曜～木曜の9:00～17:00と金曜の9:00～16:30(祝日・当社休日を除く)



三菱電機の e-F@ctory コンセプトは FA 技術と IT 技術を活用して e-F@ctory アライアンスパートナーとのコラボレーションにより、開発費用の削減、生産性の向上および保守の改善により「一歩先を行く」ものづくりを目指すことです。