

 Monaca で学ぶ

HTMLxCSSxJavaScript入門

本日の研修について

超入門

- HTMLxCSSxJavaScriptのポイントを60分に詰め込んで解説します
 - └ 実習時間を短縮いたします
 - └ 実際の授業では3時間以上が目安です
- 教材は「Monacaのテキスト」と連動スライドを利用します
 - └ 連動スライドの2～4章を改造
- 書籍のサポートページ
 - └ 新サポートページを利用します
 - <https://edu.monaca.io/book/a1>
 - サンプルプログラムをダイレクトインポートできます
 - 書籍のPDF(1～4章)も掲載されています

アジェンダ

アジェンダ

- HTML超入門
 - └ HTMLとは
 - └ リンク
 - 解説と実習1
 - └ 画像の表示
 - 解説と実習2
- CSS超入門
 - └ CSSとは
 - └ セレクタの種類と実習1
 - └ プロパティの種類と実習2
- JavaScript超入門
 - └ JavaScriptの書き方
 - └ データの扱いと実習1
 - └ 実習2

第1章 アプリ開発入門

※本日は省略します

第1章 アプリ開発入門

Monacaのアカウント登録

1) 教育版公式サイト URL にアクセス

<https://edu.monaca.io/>



アプリ開発によるプログラミング教育

ログイン アカウント作成

Monaca Education

事例 教材 料金 FAQ 記事

スマホアプリ開発で
楽しくプログラミング学習

お問い合わせ

お知らせ

2019年3月13日 お知らせ
IPSJ第81回全国大会にブース出展します

2019年3月13日 お知らせ

記事

2019年2月14日 記事
第2回専門学校HTML5作品アワード結果速報

2019年2月12日 記事

2) 『アカウント作成』 から移動して必要事項を記入。

3) 仮登録完了メールを確認



アカウント作成

メールアドレス **【必須】**

パスワード **【必須】**

アカウント作成ボタンをクリックすると、利用規約に同意したとみなされます。

アカウント新規作成



4) プラン選択

■ Education

└ 有料プランです。利用にはアクティベーションコードが必要です。

■ Freeプラン

└ 無料で使えますが容量や機能に制限があります。



Monacaをはじめる

[アクティベーションコードをお持ちですか？](#)

利用プラン選択 [プランを比較する](#)

アクティベーションコードを使う

アクティベーションコード

XXXX-XXXX-XXXX-XXXX

次に進む

第1章 アプリ開発入門

Monaca ダッシュボードとプロジェクト

ログインとダッシュボードの表示

- 開発中のアプリはプロジェクト単位で管理を行う
 - └ 画面左側に一覧表示される

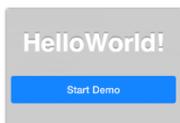


新規プロジェクト作成

1. 『新しいプロジェクトを作る』 ボタンをクリック
2. テンプレートから 『最小限のテンプレート』 を選択
3. プロジェクト名を設定 (例 : はじめてのプログラム)
4. 『作成』 ボタンをクリック

⊗ 新しいプロジェクトを作る

1 テンプレート

 <p>Start Demo</p> <p>HelloWorld</p> <p>HelloWorldを表示するだけのアプリです。</p>	 <p>SCORE: 0</p> <p>ブロック崩し</p> <p>pixi.jsを用いたブロック崩しゲームです。難易度やシナリオはプレイ後に自由に調整してください。</p>	 <p>This is a template for Monaca app.</p> <p>最小限のテンプレート</p> <p>フレームワークを使用しない空のテンプレートです。</p>
---	--	---

2 プロジェクトの情報

⊗ 新しいプロジェクトを作る

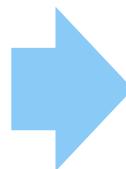
テンプレート
最小限のテンプレート

2 プロジェクトの情報

プロジェクト名
はじめてのプログラム

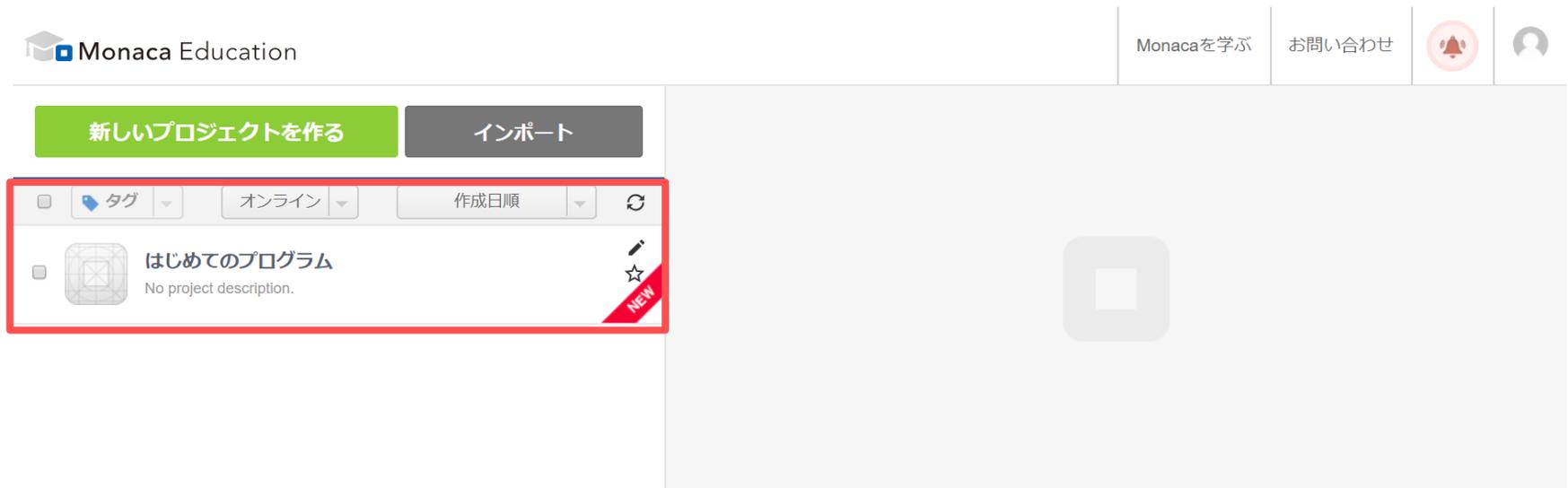
説明

作成



プロジェクト一覧に表示されれば成功

- プロジェクト名
 - └ 自由に設定可能
 - └ 後で管理しやすいように気をつける



第1章 アプリ開発入門

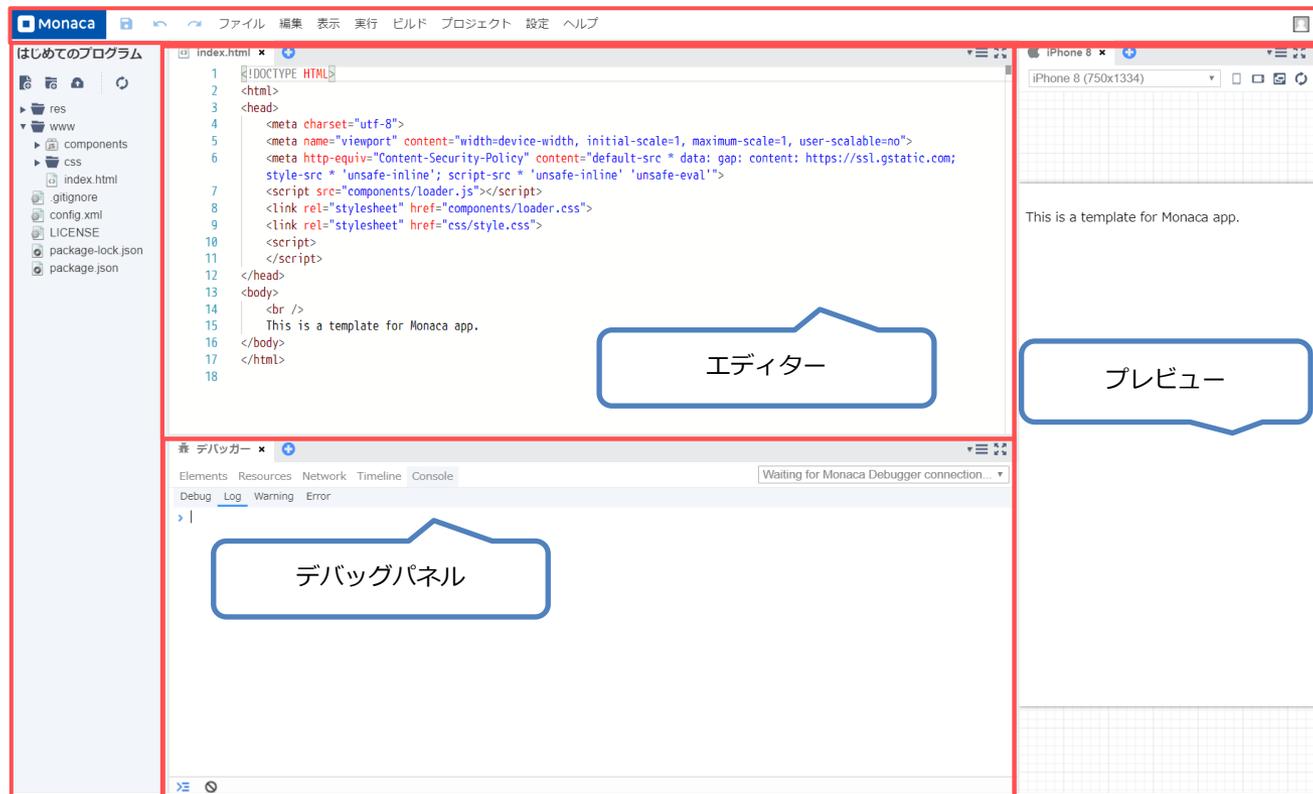
プログラムを記述する

Monaca IDE

- IDE(Integrated Development Environment)は統合開発環境の意味

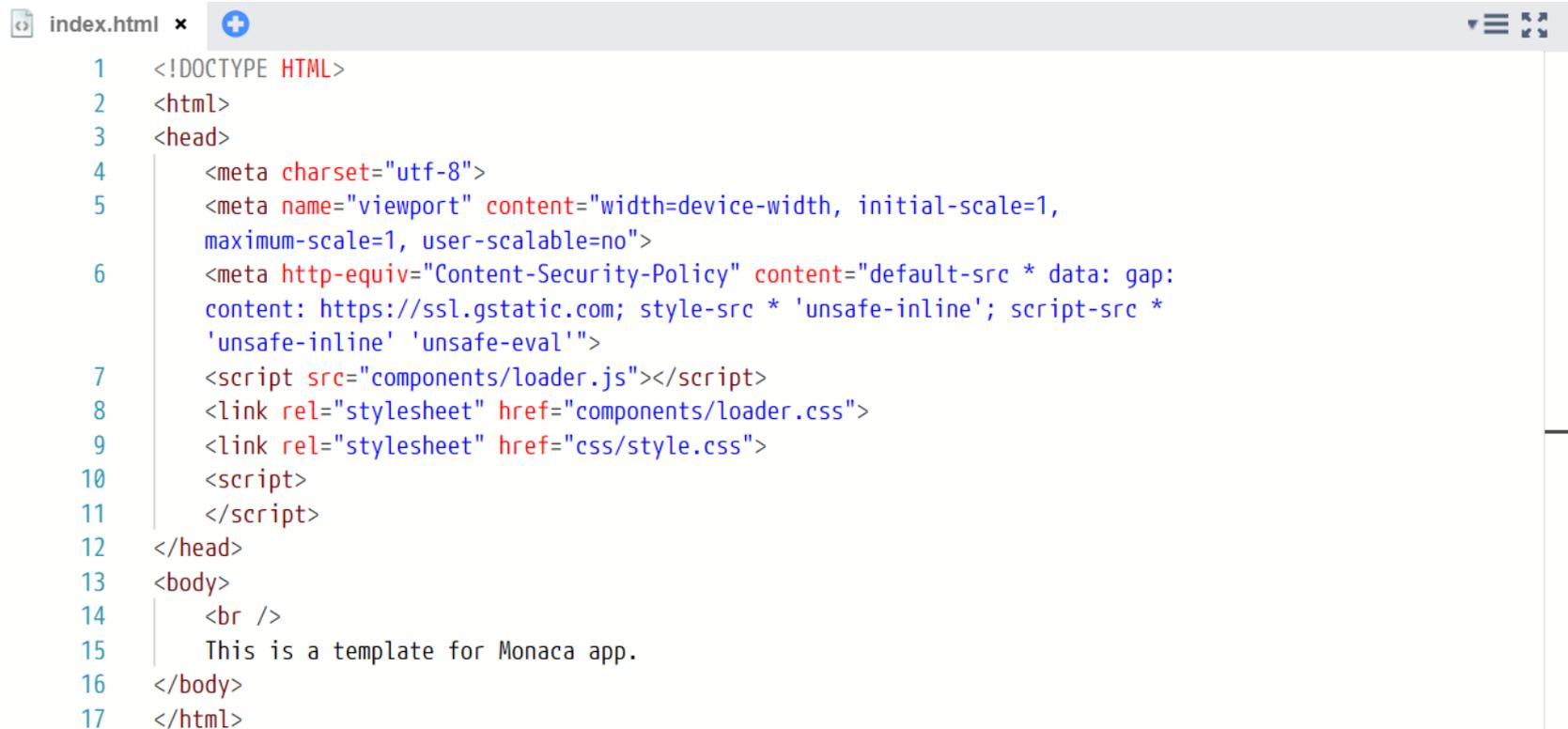
メニューバー

プロジェクトパネル



エディター

■ プログラムを記述するためのパネル



```
index.html × +
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1,
6     maximum-scale=1, user-scalable=no">
7   <meta http-equiv="Content-Security-Policy" content="default-src * data: gap:
8     content: https://ssl.gstatic.com; style-src * 'unsafe-inline'; script-src *
9     'unsafe-inline' 'unsafe-eval'">
10  <script src="components/loader.js"></script>
11  <link rel="stylesheet" href="components/loader.css">
12  <link rel="stylesheet" href="css/style.css">
13  <script>
14  </script>
15  </head>
16  <body>
17    <br />
18    This is a template for Monaca app.
19  </body>
20 </html>
```

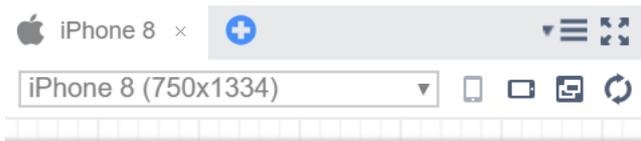
メニューバー

- 各種機能呼び出す
 - └ 保存ボタン
 - └ 元に戻す
 - └ やり直し
 - └ ビルドなど



プレビュー

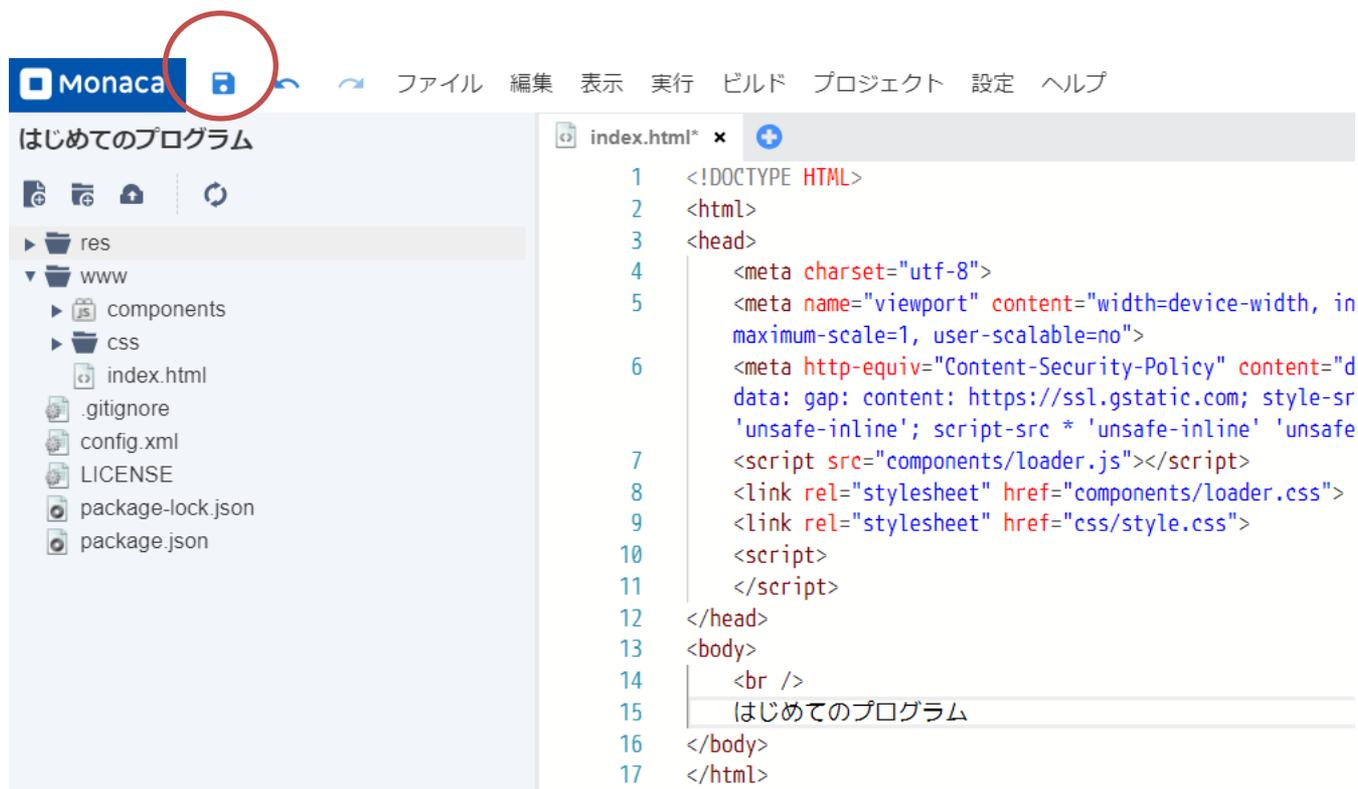
- プログラムの実行結果が表示されます。



This is a template for Monaca app.

はじめてのプログラム

- 「This is a template for Monaca app.」を削除
- 「はじめてのプログラム」と記述
- メニューバーの [保存] アイコンをクリック



プレビューの更新

- [保存] と連動してプレビュー画面が自動更新される
- 自動で更新されない場合
 - └ プレビュー画面右上の円状の矢印ボタンをクリック



JavaScript で簡単な命令を実行させる

- HTML内にJavaScriptを記述する方法
 - └ <script> で囲まれた部分に記述

```
<script>  
  alert("こんにちは");  
</script>
```

実行結果

- 次のダイアログが表示される



第1章 アプリ開発入門

Monaca デバッガーの利用

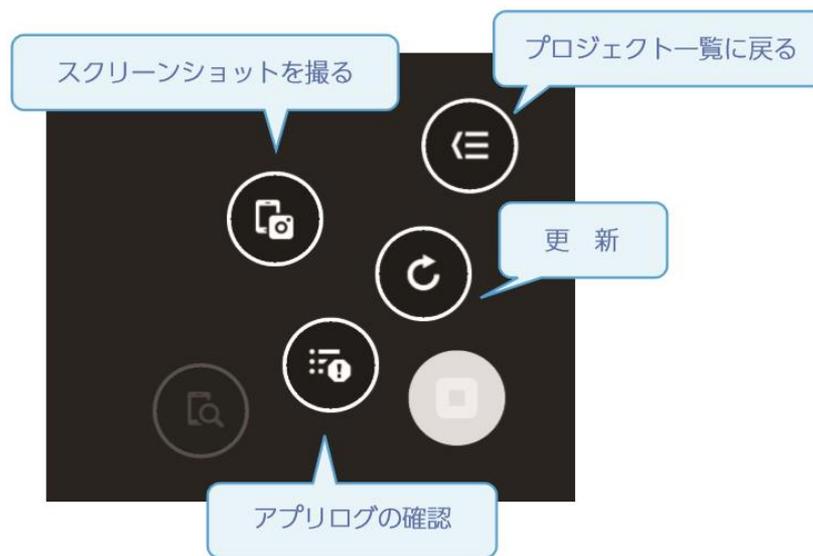
Monaca デバッガーとは

- アプリの動作を確認するツール
- プレビューより高度な動作確認が可能
 - └ カメラやコンパスなどのハードウェア機能



Monaca デバッガーのメニュー

- プロジェクトを開きアプリの動作を確認
- 丸いMonacaボタンはメニュー
 - └ 更新やログの確認が可能



第2章 HTML入門

第2章 HTML入門

HTMLとは

HTML (Hyper Text Markup Language) とは

- マークアップ言語の1つ
- 文書をタグで囲んで記述

HTMLで実現可能なこと

- 文章の構造化
- 文章同士のリンク
- 画像や動画の参照

HTMLタグの書き方

- 「開始タグ」と「終了タグ」にはタグ名が入る
 - └ タグは全部で100個以上存在する

文法 タグの記述方法と名称

```
<開始タグ>内容</終了タグ>
```

- 例では段落を表すPタグを利用

例 タグの記述例

```
<p>これは段落です。 </p>
```

空要素の記述方法

- スラッシュはつけなくても良い
- 例では改行を表すBRタグを利用



文法 空要素の記述方法

```
<開始タグ>  
または  
<開始タグ />
```



例 空要素の記述例

```
<br>  
または  
<br />
```

属性について

- タグには属性と属性値を設定できる
- 例ではAタグにhref属性とリンク先の属性値を付加
- 属性値はダブルクォートかシングルクォートで囲む



文法 属性の記述

```
<開始タグ 属性1="値" 属性2="値">内容</終了タグ>
```



例 属性の記述例

```
<a href="top.html">TOPページへ</a>
```

HTMLの例

- HTML文章はタグを入れ子にして記述する
 - └ この章ではBODYタグに入るタグを重点的に学習する
- BODYの外のタグを誤って消さないように注意する

index.html

```
<!DOCTYPE HTML> .....①
<html> .....②
<head> .....③
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1, maximum-scale=1, user-scalable=no"> .....④
  <meta http-equiv="Content-Security-Policy" content="default-src
*; style-src * 'unsafe-inline'; script-src * 'unsafe-inline'
'unsafe-eval'">
  <script src="components/loader.js"></script> .....⑤
  <link rel="stylesheet" href="components/loader.css"> .....⑥
  <link rel="stylesheet" href="css/style.css">
  <script>
    alert("こんにちは"); } .....⑦
  </script>
</head>
<body> .....⑧
  <br /> .....⑨
  はじめてのプログラム
</body>
</html>
```

第2章 HTML入門

<body>タグ内に記述するタグの種類

終了タグのあるタグ

タグ名	概要
h1	見出しを定義します。h1～h6まであり、h1が最も高レベル、h6が最も低レベルな見出しです。 例：<h1>見出し</h1>
p	文章の段落を定義します。 例：<p> 文章の段落を定義します。 </p>
div	特に意味を持たないタグです。複数のタグをまとめて扱うときや、四角い枠を描画したいときに使います。 例：<div> <h1>見出し</h1> <p>段落</p> </div>
a	リンクを定義します。 href属性・・・リンク先のURLを指定します。 例：TOPへ
button	ボタンを定義します。 例：<button>ボタン</button>

空要素（終了タグのないタグ）

タグ名	概要
img	<p>画像を参照します。</p> <p>src属性：画像の参照先を指定します。</p> <p>alt属性：画像が何らかの理由で表示できなかった場合に、画像の変わりに表示する文字列を指定します。</p> <p>例：</p>

すべてのタグにつけられる属性

タグ名	概要
id	要素を識別するためのIDです。文書内で重複する値を指定することはできません。 例 : <div id="header">…</div>
class	CSSのクラス名を指定します。(→第3章) 例 : <div class="container">…</div>

第2章 HTML入門

リンク

リンクの設定

- Aタグを使用し、リンク先はパスをhref属性の値で指定
- 絶対パス指定
 - └ パスを全て記述する方法
- 相対パス指定
 - └ 現在のファイルからの相対的なパスを記述する方法

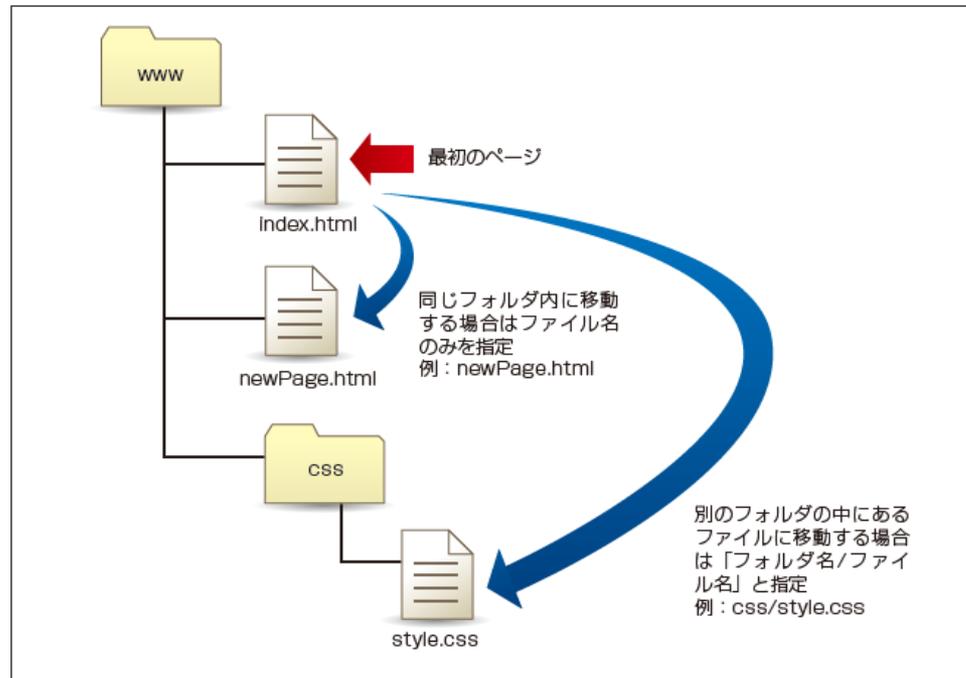


文法 リンクの設定

```
<a href="リンク先のパス">リンク文字列</a>
```

相対パス指定

- 対象ファイルまでの位置を相対的に指定する方法
 - └ 下位のフォルダを指定するときは「/」を使用
 - └ 上位のフォルダを指定するときは「..」と「/」を使用



第2章 HTML入門

実習

リンクの配置

```
<body>  
  <a href="https://edu.monaca.io/">Monacaへ</a>  
</body>
```

実行結果



※リンク先によってはプレビュー機能で動かない場合がある。
その場合はデバッガーアプリで確認。

準備

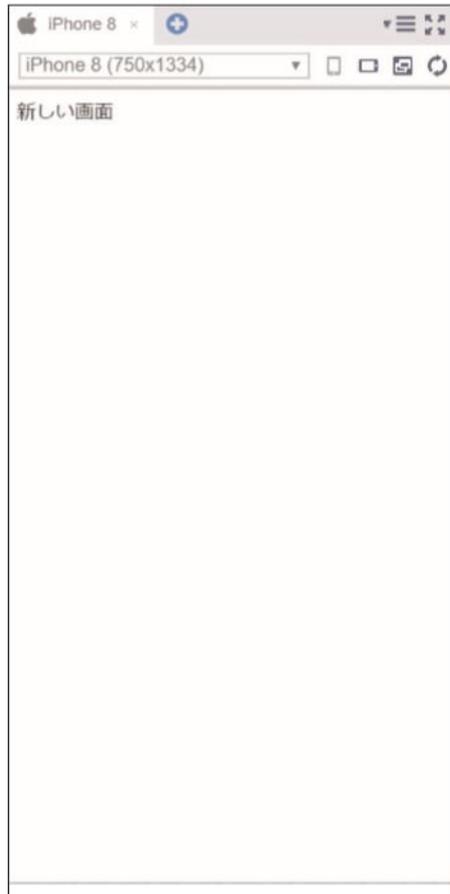
- 別ファイル「newPage.html」を確認

実習

- 相対パスのリンクを追記

```
<a href="newPage.html">次の画面へ</a>
```

実行結果



第2章 HTML入門

画像の表示

画像を表示する方法

- imgタグを記述
- src属性でパスを指定する



文法 画像の表示

```

```

- 例

```

```

第2章 HTML入門

実習

準備

- 画像ファイル「monaca.jpg」を確認

実習

- imgタグを追記

```

```

実行結果



第3章 CSS入門

第3章 CSS入門

CSSとは

CSS (Cascading Style Sheets) とは

- HTML文章を装飾するための技術
- 色やサイズなどを変更できる
- カスケーディングという特徴がある

CSSをHTMLファイルに読み込む方法

- 外部ファイル
 - └ linkタグで外部ファイルを指定して読み込む
- styleタグ
 - └ styleタグの中に記述する
- style属性
 - └ style属性の中に記述する

CSSをHTMLファイルに読み込む方法（外部ファイル）

- linkタグを記述
- href属性でパスを指定する

文法 CSSファイルの読み込み

```
<link rel="stylesheet" href="CSSファイルのパス">
```

- 例

```
<link rel="stylesheet" href="css/style.css">
```

CSSの書き方 (1)

- セレクタ
 - └ 対象となる要素を指定する
- プロパティ
 - └ どのようなスタイルを適応するかを指定する

文法 セレクタとプロパティの記述方法

```
セレクタ {  
    プロパティ: 値;  
    プロパティ: 値;  
    }  
}
```

CSSの書き方 (2)

- セレクタ
 - └ P(パラグラフ)タグを対象とする
- プロパティ
 - └ 文字の色を赤くする
 - └ 文字のサイズを10pxにする

文法 セレクタとプロパティの記述例

```
p {  
    color: red;  
    font-size: 10px;  
}
```

第3章 CSS入門

セレクタの種類

タグセレクトタ

- 対象要素をタグ名で指定

IDセレクトタ

- 対象要素をID属性値で指定
 - └ セレクトタの先頭に#をつける

クラスセレクトタ

- 対象要素をクラス値で指定
 - └ セレクトタの先頭に.(ドット)をつける

第3章 CSS入門

実習

実習

- 2章のプロジェクトを開く
- cssフォルダ内のstyle.cssを編集

```
a {  
    font-size: 30px;  
}
```

実行結果



実習

- index.htmlのaタグにID属性を付与

```
<a href="https://edu.monaca.io/" id="target">Monacaへ</a>
```

- style.cssにIDセレクタを利用したスタイルを記述

```
#target {  
    color: red;  
}
```

実行結果



実習

- index.htmlのaタグにクラス属性を付与

```
<a href="newPage.html" class="bright">次の画面へ</a>
```

- style.cssにクラスセレクトを使用したスタイルを追記

```
.bright {  
    background-color: yellow;  
}
```

実行結果



第3章 CSS入門

プロパティの種類

色を指定するプロパティ

プロパティ	説明	例
color	文字色を設定します。	color: red;
background-color	背景色を設定します。	background-color: red;
border	線の色（および線種と線の太さ）を設定します。	border: solid 1px red; 線種、線の太さ、線の色の順に設定します。solidは直線を表します。

カラーコード

- 光の三原色で色を作る方法
- カラーコードでは16進数を使う

カラーコードの例

- 紫の色を作る

```
#ff00ff
```

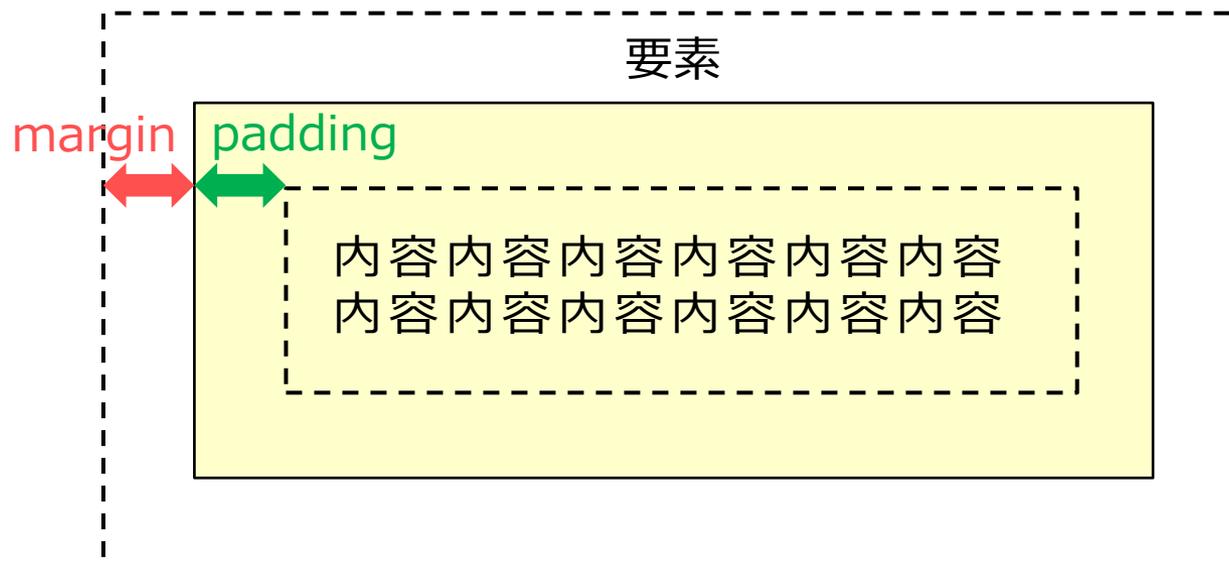
- └ 赤がff (255) 緑が00 (0) 青がff (255)
- └ 原色の赤と青を混ぜた色 = 紫になる

サイズや位置を指定するプロパティ

プロパティ	説明	例
font-size	文字のサイズを設定します。	font-size: 12px;
text-align	要素内の横方向の配置を設定します。	text-align: left; (左寄せ) text-align: right; (右寄せ) text-align: center; (中央揃え) text-align: justify; (均等割付)
width	要素の横幅を設定します。	width: 100px
height	要素の高さを設定します。	height: 300px;
margin	枠線の外側の余白を設定します。	margin: 20px;
padding	枠線の内側の余白を設定します。	padding: 10px;

marginとpadding

- どちらも余白の幅を指定するプロパティ
- marginはborder(枠線)の外側の余白
- paddingはborderの内側の余白



第3章 CSS入門

実習

実習

- style.cssを編集

```
img {  
  width: 30%;  
  border: solid 3px #0000ff;  
  margin: 10px;  
  padding: 20px;  
}
```

実行結果



第4章 JavaScript入門

第4章 JavaScript入門

JavaScriptの書き方

scriptタグの中に記述する

```
<script>  
  alert("こんにちは");  
</script>
```

外部ファイルに記述する

```
<script src="JavaScript ファイルのパス"></script>
```

書き方のルール

- 基本的に半角の英数字と記号のみを使用する
 - └ 「'」か「"」で括れば全角文字も利用可能
- 大文字と小文字は別の文字として扱われる
- 命令文の末尾には「;」をつける。
- ひとまとまりの命令群を波かっこ{ }で囲む
 - └ 囲まれた範囲をブロックと呼ぶ
- 改行や半角スペースは自由に挿入できる

改行やスペースの活用例

 例 プログラムA(改行とスペースを使わない)

```
for(i=0;i<10;i++){alert(i);}
```

 例 プログラムB (改行とスペースを使う)

```
for(i = 0; i < 10; i++) {  
    alert(i);  
}
```

インデントとは

- 見やすいように字下げを行うこと
 - └ 空白スペースやタブなどを用いる
 - └ 波かっこの対応関係が一目でわかるようになる

コメントとは

- メモを書いたり命令を無効化したりできる

文法 一行コメント

```
//alert("こんにちは");
```

文法 複数行コメント

```
/*  
コメントとして記述した内容は、  
スクリプトには影響しません。  
*/
```

第4章 JavaScript入門

データの扱い方

変数の作り方

- メモリ上に変数を作る作業を変数の「宣言」と言う
- 変数は名前を付けて管理できる

文法 変数宣言の書式

```
var 変数名;
```

例 xという名前の変数を作る

```
var x;
```

変数の使い方

- 変数を作った直後は空の状態
- 変数にデータを入れるには代入を行う
- 宣言と代入は1行にまとめて同時に行うこともできる

文法 変数へ値を入れる

```
変数名 = 値;
```

文法 宣言と代入を同時に行う

```
var 変数名 = 値;
```

JavaScript から画面にデータを出力する

- document.writeln()で簡単に値を画面に出力できる
 - └ ※簡易的ですが実務は推奨されない命令です

 文法 変数へ値を入れる

```
document.writeln(表示するデータ);
```

結果位置

```
<body>
  <div>
    <p>こんにちは</p>
  </div>
</div>
```

この位置に出力される

第4章 JavaScript入門

実習

実習

- index.htmlのbodyタグ内を編集
- index.htmlのscriptタグ内を編集

実習

- index.htmlのscriptタグ内を編集

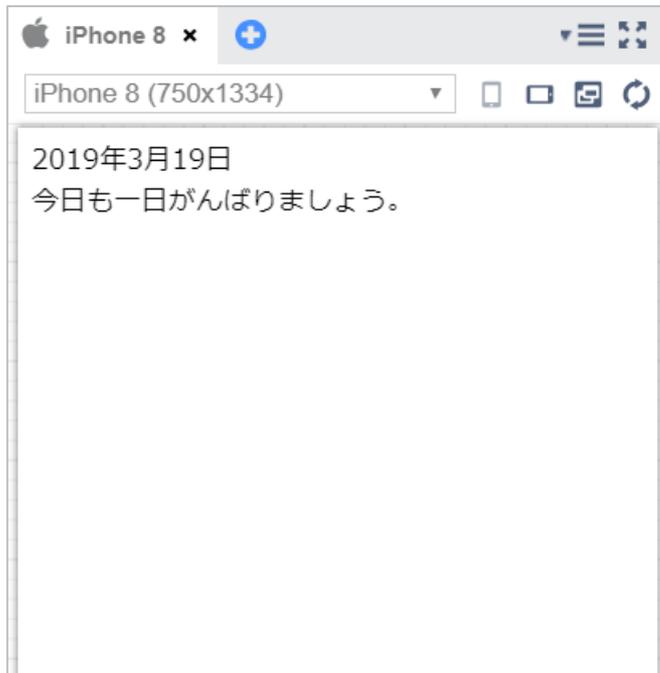
```
<script>
  var x = 10;
  var y = 20;
  var z;

  z = x * y;
  document.writeln(z);
</script>
```

第4章 JavaScript入門

追加実習

結果



今日の日付をプログラムで取得する方法

- 日付オブジェクトの準備

```
var 変数 = new Date();
```

- 年を取得する命令

```
変数.getFullYear();
```

- 月を取得する命令

```
変数.getMonth();
```

- 日を取得する命令

```
変数.getDate();
```

実習

- index.htmlのscriptタグ内を編集

```
<script>
  // 日付に関する命令を使えるようにする
  var date = new Date();
  // 年、月、日の取得
  var year = date.getFullYear();
  var month = date.getMonth() + 1;
  var day = date.getDate();
  // 日本の表記にする
  var today = year + " 年" + month + " 月" + day + " 日";
  document.writeln(today);
</script>
```