Web Frontend Day 1

Web ページの構成要素とチュートリアル

伊藤 龍之介

https://comame.xyz

Slack: @comame

- 121 回生
- 競プロと C++ はサボっていた
  - AtCoder のレーティングは 47
- 最近は Web 周りの開発

やること

Day 1

- Web ブラウザに表示されているものの構成要素
  - HTML, JavaScript, CSS
- 構成要素の記述方法

Day 2

- もう少し本格的に動くものを作ってみる
  - HTML 5
  - JavaScript, DOM, Web API

2 日分で 1 講座だと思いこんで作ってしまいました 🤦

やらないこと

- Web サーバとの通信を使った何か
- HTTP の仕組み
- ブラウザがページを読み込む仕組み

Web ページの構成要素

Web ページは何でできている？

- HTML (HyperText Markup Language)
- CSS (Cascading Style Sheets)
- JavaScript

そもそも Web ページって？

リンク付きの文書

\ クリックすると例が開くよ /

HyperText Markup Language

- HyperText を Markup する言語
- HyperText = リンク付き文書 (Wikipedia とか分かりやすいよね)
- Markup = コンピュータが分かる形式に書き起こす

HyperText Markup Language

```
<p>
    HTML は XML のような構文で文書構造を表現します。
    開始タグと閉じタグ、属性などを使って表現していきます。
</p>

<h1>
    H1 タグで囲われた部分は見出しを表します
</h1>
<p some-attr=”属性はこんな感じで書きます”></p>
```

Cascading Style Sheets

- 味気ない文書に見た目を付けるもの
- 最近の Web アプリケーションでは職人技により変態的な働きをしている
- 「セレクタ」というのを使って、見た目をつける対象の部分を絞り込む

Cascading Style Sheets

```
selector {
    property-name: property-value;
}

/* p タグで表現された全ての部分のフォントサイズを 12 ピクセルにする */

p {
    font-size: 12px;
}
```

JavaScript

- 文書に動的な要素を付けるためのプログラミング言語
- 動的 ≠ 動き (アニメーション)

書いてみよう

まずは HTML から

「おいしいカレーの作り方」を HTML で書こう

材料と手順を含めてください

HTML: タグ

タグを使って構造を表す
原則的に開始タグと終了タグが必要

```
<tag-name>
    <another-tag-name>
        You can nest tags!
    </another-tag-name>
</tag-name>
```

HTML: 属性 (Attributes)

タグに対して属性を付与できる

```
<tag-name attr-1=”value” attr-2=”value”>
    Lorem ipsum
</tag-name>
```

HTML: よく使うタグ

- `<h1>` 一番大きい見出し
- `<h2>` 2 番目に大きい見出し
- `<h6>` 一番小さい見出し

- `<ul>` 箇条書きリスト
- `<ol>` 番号付きリスト
- `<li>` リストの項目

- `<p>` パラグラフ、本文
- `<br>` 改行 *(閉じタグ不要)*

- `<a>` リンク

HTML: 見出しをつける

```
<h1>見出し 大</h1>
<h2>見出し  </h2>
<h3>見出し  </h3>
<h4>見出し  </h4>
<h5>見出し  </h5>
<h6>見出し 小</h6>
```

HTML: 箇条書き・リストを作る

```
<ul>
    <li>item-1</li>
    <li>item-2</li>
<ul>

<ol>
    <li>number-1</li>
    <li>number-2</li>
</ol>
```

HTML: 本文を記述する

```
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt.<br>
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.
</p>
```

HTML: リンクをつける

- Wikipedia の「カレー」へのリンクを付けてみよう
- <a> タグと href 属性

- `<a href="https://example.com">リンクテキスト</a>`
- 上の例はこんな感じになります: リンクテキスト

HTML: 例

https://gist.github.com/comame/3f021ad257bd094a219e6ac189923522

CSS で見た目をつけてみよう

「おいしいカレーの作り方」の見た目を CSS で整えよう

CSS: はじめに

```
<link rel=”stylesheet” href=”path/to/style.css”>
```

あるいは

```
<!-- 今回はこっちで書きます -->
<style>
    p {
        font-size: 12px;
    }
</style>
```

CSS: セレクタ

セレクタを使って、見た目を指定する対象を選ぶ

- タグ名
- ID
- クラス

クラスと ID は次のように指定する

```
<p id=”hoge-id”>Lorem ipsum</p>
<p class=”hoge-class”>Lorem ipsum</p>
```

CSS: セレクタの例

- `p` `<p>` タグで表現されているものすべて
- `#hoge` ID が hoge なもの
- `.hoge` クラスが hoge なもの
- `p#hoge` `<p>` タグ & ID が hoge なもの
- `p.hoge` `<p>` タグかつ & クラスが hoge なもの

CSS: プロパティ

セレクタに一致する要素に対して、プロパティを使って見た目を指定します。

```
p {
    font-size: 12px;
    color: blue;
}
```

CSS: よく使うプロパティ

- font-size: 12px;
- color: red;
- margin: 0px auto 12px auto; /* 順に上・右・下・左 */
- padding: 4px 2px 4px 2px; /* 順に上・右・下・左 */
- border: 2px solid black; /* 順に太さ、種類、色 */

他にもたくさん…

CSS: 例

https://gist.github.com/comame/3f021ad257bd094a219e6ac189923522

JavaScript で動作をつけてみよう

材料をクリックしたときに分量を表示させてみよう

JavaScript: はじめに

```
<script src=”path/to/style.css”></script>
```

あるいは

```
<!-- 今回はこっちで書きます -->
<script>
    const message = ‘Hello, world!’;
    console.log(message);
</script>
```

JavaScript: 変数定義と初期化・代入

```
// 変数の宣言
let variable;
// 代入（初期化）
variable = 3;

// 宣言と初期化は同時に可能
let variable2 = 1;

// 定数の宣言
const constantVariable = 5;
```

JavaScript: プリミティブ型

```
// number 型: 数値 (整数及び小数)
10, 3.1415

// string 型: 文字列 (1文字であっても文字列)
‘Hello, world!’, “こんにちは世界”

// boolean 型: 真偽値
true, false
```

JavaScript: 動的型付け

```
let variable;
typeof variable == ‘undefined’;

variable = 10;
typeof variable == ‘number’;

variable = true;
typeof variable == ‘boolean’;
```

JavaScript: オブジェクト

```
const obj = {
    hoge: ‘fuga’,
    pi: 3.14
};

obj.hoge == ‘fuga’;
obj[‘pi’] == 3.14;
```

JavaScript: 配列

```
const arr = [ ‘hoge’, ‘fuga’, ‘hogefuga’ ];

arr[0] == ‘hoge’;
arr[1] == ‘fuga’;
arr[2] == ‘hogefuga’;
```

JavaScript: if 文

```
const num = 2;

if (num == 1) {
    // num is 1
} else if (num == 2) {
    // num is 2
} else {
    // num is not 1 and 2; num is 3
}
```

JavaScript: while 文

```
let i = 0;

while (i < 0) {
    console.log(i);
    i += 1;
}

console.log('break');
```

JavaScript: for 文

```
for (let i = 0; i < 10; i++) {
    console.log(i);
}

console.log('break');
```

JavaScript: for..of / for...in

```
const arr = [ ‘hoge’, ‘fuga’, ‘hogefuga’ ];

for (const value of arr) {
    console.log(value);
}

for (const i in arr) {
    console.log(arr[i]);
}
```

JavaScript: 関数

```
function sayHello(name) {
    console.log(name);
}

// 関数も第一級オブジェクト
const variable = sayHello;
variable('your-name');

// こんな書き方も
const variable2 = function(name) {
    console.log(name);
}
```

JavaScript: 要素を取得する

```
<p id=”click-me”>Click me!</p>

<script>
    const clickMeElement =
        document.getElementById(”click-me”);
</script>
```

JavaScript: クリックしたときに実行する

```
clickMeElement.addEventListener(‘click’, function() {
    // クリック時にこの関数が実行される
    alert(‘clicked!’);
});
```

JavaScript: 例

https://gist.github.com/comame/3f021ad257bd094a219e6ac189923522

資料配布

Slides:

https://comame.xyz/r/pcp2019-web-day1
https://comame.xyz/r/pcp2019-web-day2

Downloads / References:

https://comame.xyz/pages/pcp2019/