

# 変数 & 条件分岐

---

# Contents

---

- JavaScript概要
- 変数と計算
- 乱数の作成
- 条件分岐(おみくじ作成)
- jQueryを用いたwebアプリ
- おみくじアプリ演習
- 課題発表 -> P2Pタイム
- 写経のススメ(?)

# rules...

---

- 授業中は常にエディタを起動！
- 考えたことや感じたことはzoomチャットでガンガン発信！
- 質問はslackへ！ 他の人の質問にも目を通そう！（同じ質問があるかも）
- 演習時，できた人はスクショなどslackに貼ってアウトプット！
- まずは打ち間違いを疑おう！
  - `{'";` など
- 書いたら保存しよう！（よく忘れる！）
  - `command + s`
  - `ctrl + s`

# 本日のゴール

---

- プログラミングに慣れる！
- 「変数」「条件分岐」を扱う！
- 課題に取り組み始める！

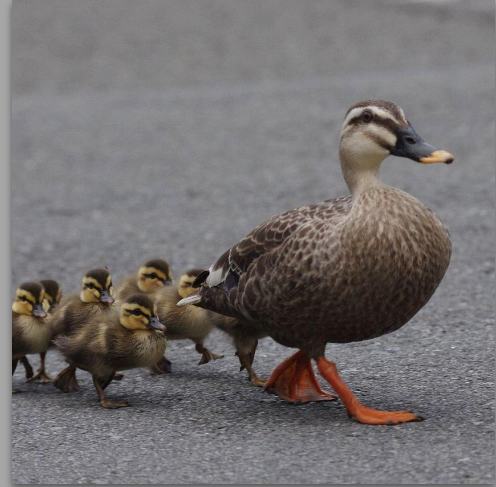
# JavaScript概要

- 
- **html**(マスター済み)
    - コンテンツの指定
    - タイトル, 文章, 画像などの記述
  - **css**(マスター済み)
    - コンテンツの装飾
    - 色, 大きさ, 配置などの指定
  - **JavaScript**
    - ユーザー操作, イベント発生による動きを実現
    - (わりとなんでもできる)

JavaScript ≠ JAVA

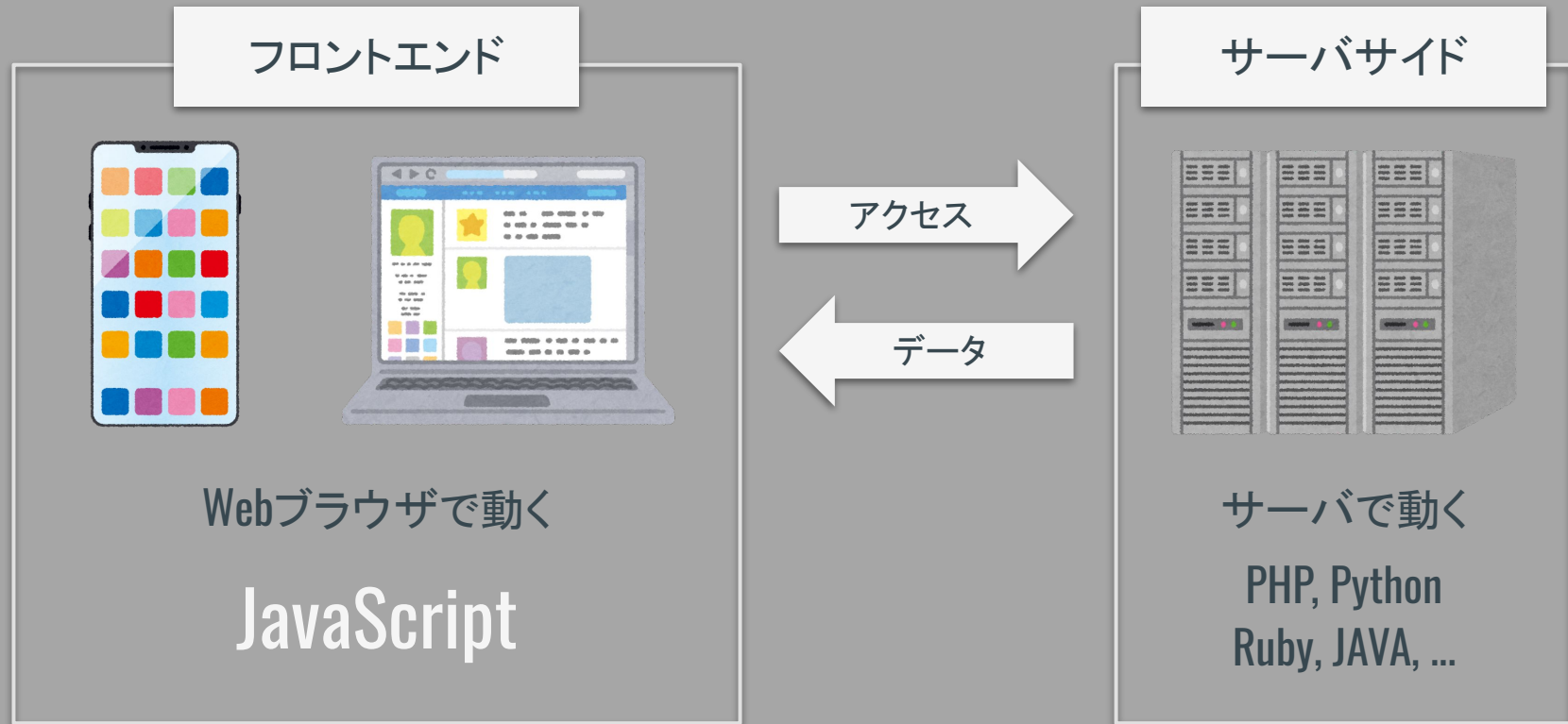


≠



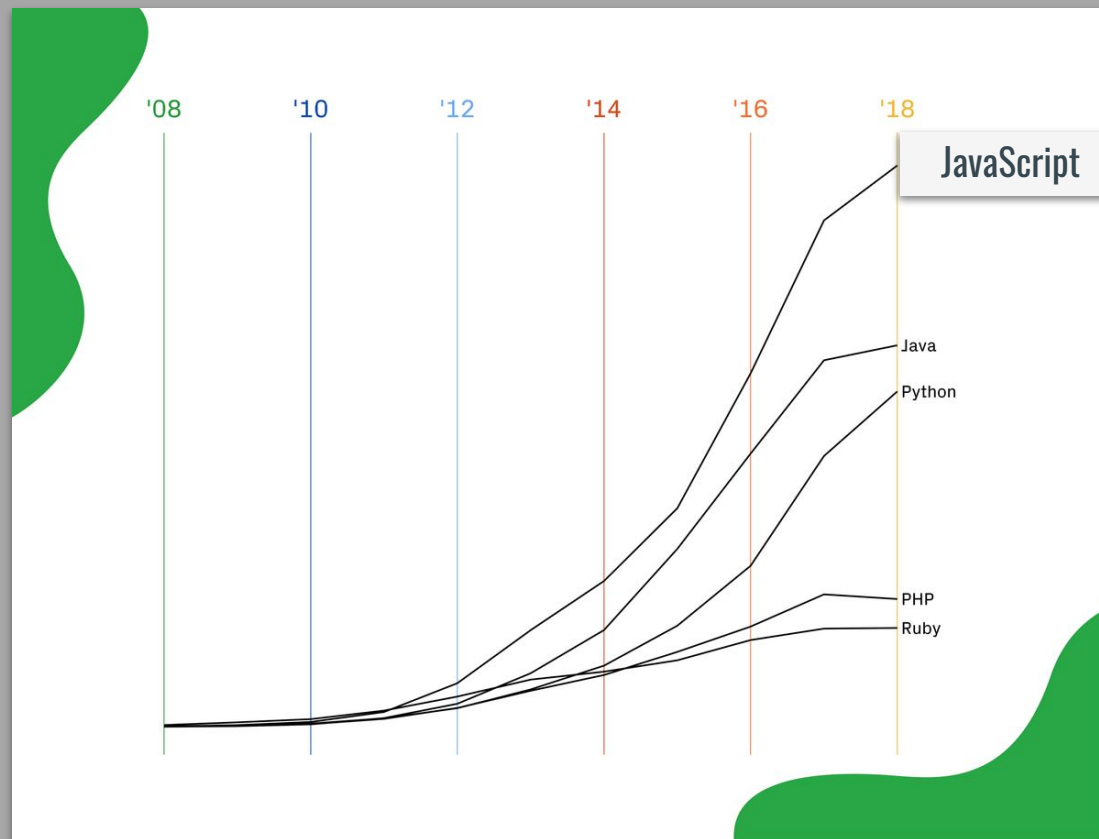


# Webアプリケーションの構造

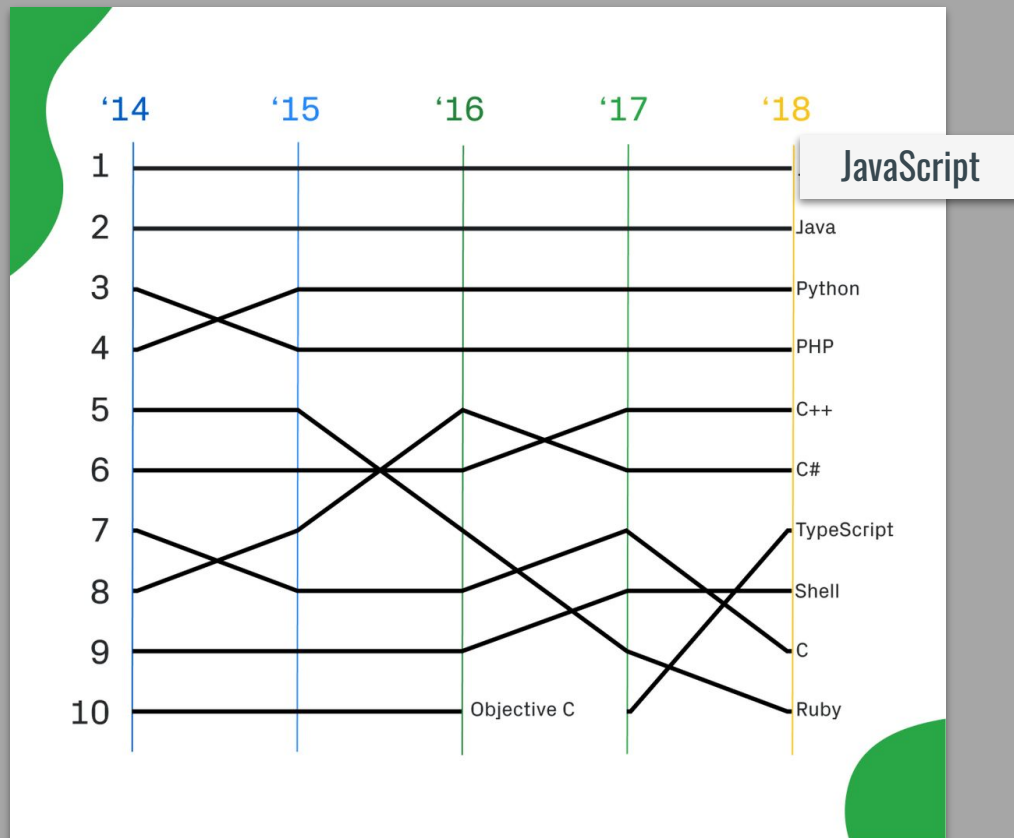


JavaScriptはwebアプリに欠かせない！

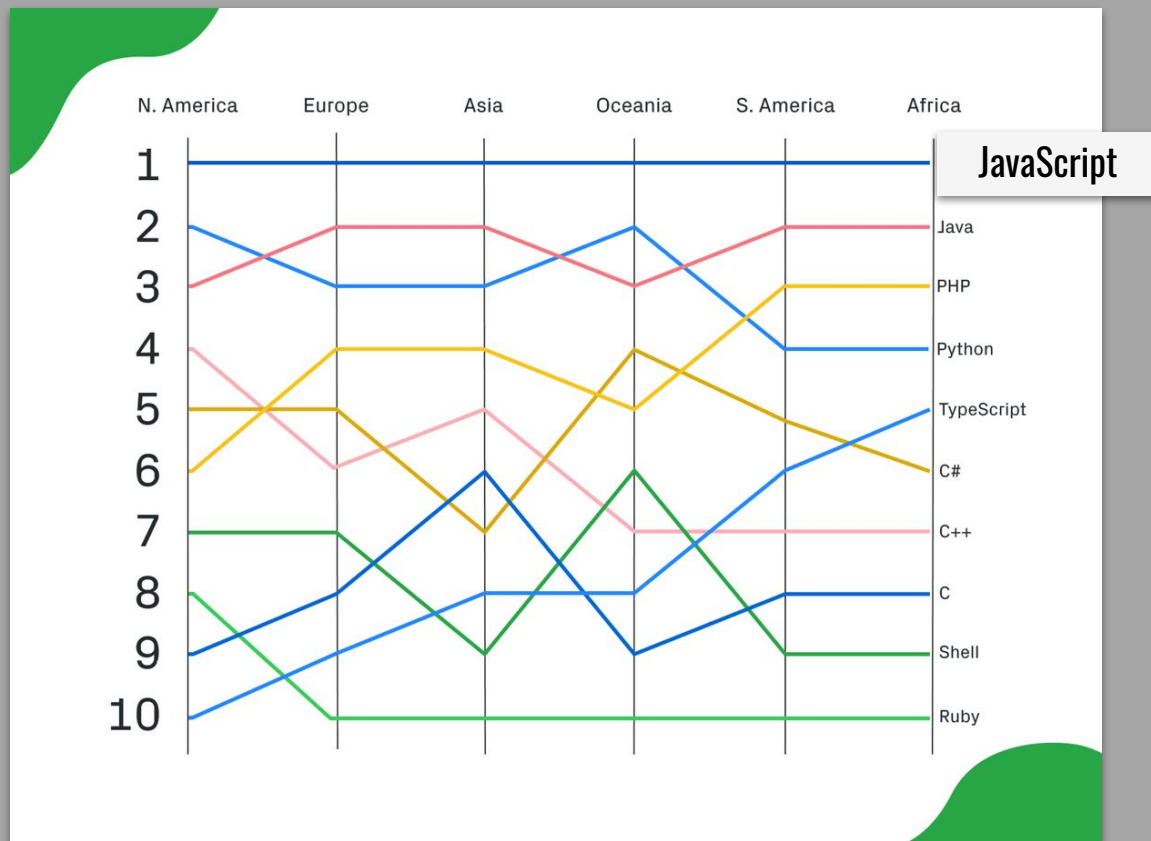
# Githubのリポジトリ数



# 言語別人気ランキング



# 地域別人気ランキング



# JavaScript周辺の技術



# よく使われる技術の紹介

---



- JavaScriptのライブラリ.
- ユーザ操作イベントやアニメーションを実現
- ✓ 生JSと比較して短くかける.
- ✓ webアプリケーションで広く普及している.
- ✓ 導入が簡単.
- ✓ 学習コストが低い.
- ▲ 難しいことをやろうとすると複雑になる.
- まずはここから！



# よく使われる技術の紹介

---



- JavaScriptのライブラリ.
- ✓ モダンなwebアプリケーション(SPA)を実現
- ✓ 高速！
- ✓ スマホアプリも見据えた開発が可能.
- ▲ 学習コスト / 環境構築がややハードル.
- (私はReactが好き)

# よく使われる技術の紹介

---



- サーバサイドでJavaScriptを動かす技術.
- ✓ フロントもサーバもJavaScriptで書ける... !
- ▲ 難しいことをやろうとすると複雑になる.

JavaScriptを書く！

# 早速書く！

---

- 書き方
  - `<script></script>`の間に処理を記述
- 書く場所
  - htmlファイルの`</body>`のすぐ上に書こう！
  - ほかにいくつか書ける場所があります.
  - 別にファイルを作るやり方もあります.

# 早速書く！

---

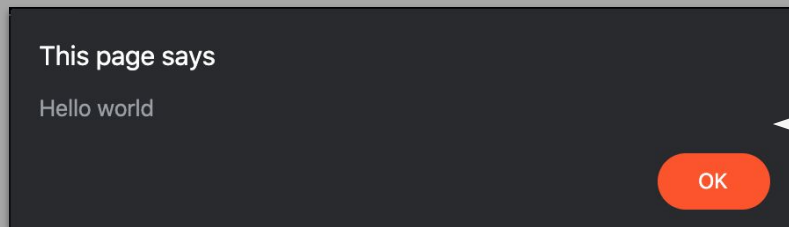
```
<script>
  // alert();でポップアップ表示！
  alert('Hello world');    // 文字列は「'」か「"」で囲む.

  // console.log(); ブラウザで検証ツール→consoleで確認！
  console.log('Hello world');
</script>
```

# 早速書く！

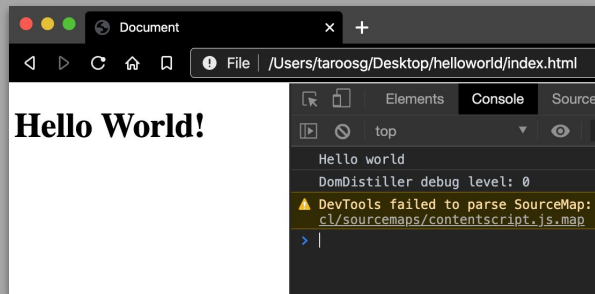
- 前ページを参考にalert()とconsole.log()を動かそう！

- alert();



リロードすると表示される！

- console.log();



consoleに表示される！

# 変数と計算

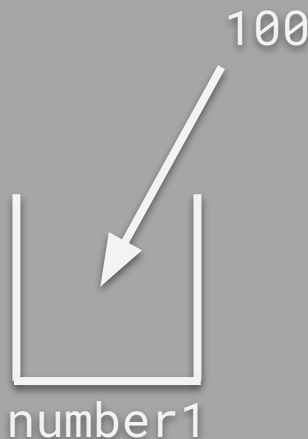
# 変数とは？？

```
<script>
```

```
// 「変数」は文字列や数値を入れるための箱のようなもの  
// 箱に名前をつけておき、あとから再度利用できる！
```

```
const text1 = 'hello';    // 文字列（' 'などで囲む）  
const text2 = 'world';    // 文字列  
const number1 = 100;      // 数値  
const number2 = 200;      // 数値  
const 3number = 1000;     // 名前の先頭が数字はダメ
```

```
</script>
```





# 計算

---

```
<script>
```

```
    // 数学と同様に計算できる！
```

```
    const number1 = 1 + 9;           // 10
```

```
    const number2 = 1 - 5;           // -4
```

```
    const number3 = 2 * 4;           // 8
```

```
    const number4 = 10 / 2;          // 5
```

```
    const number5 = 10 % 3;          // 1
```

```
</script>
```

# 計算

---

```
<script>
```

```
    // 文字列は結合される
```

```
    const number1 = 100;
```

```
    // 数値
```

```
    const number2 = 200;
```

```
    // 数値
```

```
    const text1 = "hello";
```

```
    // 文字列
```

```
    const text2 = "world";
```

```
    // 文字列
```

```
    const sum1 = number1 + number2;
```

```
    // 300 (数値)
```

```
    const sum2 = text1 + text2;
```

```
    // helloworld (文字列)
```

```
</script>
```

# NGワード

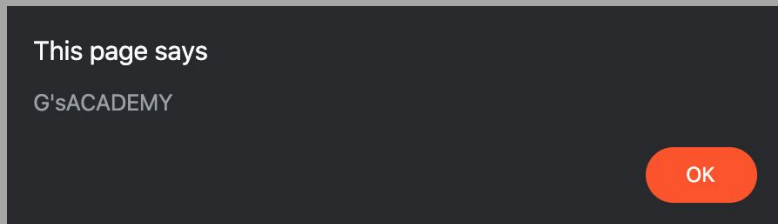
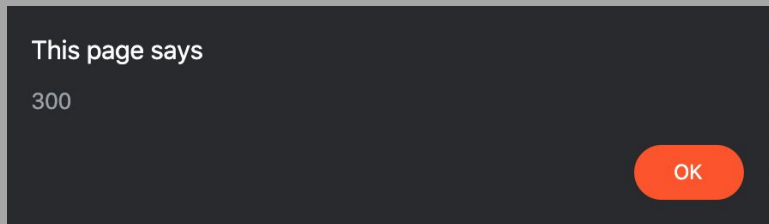
---

- 変数名には使用できない単語が存在する
  - 構文で使用するもの, 将来的に使われるもの, など
  - 「if」「for」など
- 予約語とキーワード
  - 「MDN javascript 予約語」で検索！！
  - 引っかかる場合は少ないので今は気にしないでOK！

# 練習(js\_practice.html)

---

- 以下の処理をJavaScriptで実装しよう！
  - 「100」と「200」の数値をそれぞれ変数に入れ、加算した結果をalert().
  - 「G's」と「ACADEMY」の文字列をそれぞれ変数に入れ、つなげてalert().
- 動作確認(こんな感じに出てくる)



乱数(ランダムな数)をつくる！

# 計算

---

```
<script>

  // Math.random()を使う！（JavaScriptに最初から用意されている）

  const randomNumber1 = Math.random();
  alert(randomNumber1);    // 0から1の間でランダムな値（乱数）を表示.

  const randomNumber2 = Math.floor(Math.random() * 5);
  alert(randomNumber2);    // 0から4までのどれかが表示される！

</script>
```

# 練習！（math.html）

---

- 以下の数をランダムで発生させて`alert();`で表示させよう！
  - a. 0から9のどれか
  - b. 1から9のどれか
  - c. 5から10のどれか
  - d. 50から99のどれか

# 条件分岐(if文)



# 計算

---

```
// 条件を満たすときと満たさないときで別々の処理を実行する！
```

```
if(条件式){  
    // 条件式を満たす場合の処理  
} else {  
    // 条件式を満たさない場合の処理  
}
```

# 計算

---

```
// 複数の条件で処理を分岐させることもできる！

if(条件式1){
    // 条件式1を満たす場合の処理
} else if(条件式2) {
    // 条件式1と満たさなくて条件式2を満たす場合の処理
} else {
    // いずれの条件も満たさない場合の処理
}
```

# 条件式

- 条件の書き方(比較演算子)
  - == 等しければtrue
  - != 等しくなければtrue
  - > 左側のほうが大きければtrue
  - < 右側のほうが大きければtrue
  - >= 左側が右側以上ならtrue
  - <= 右側が左側以上ならtrue



=



==

# おみくじの処理を作ろう！(omikuji01.html)

---

- ランダムに「大吉・中吉・小吉・凶・大凶」をalert()で表示！
- ヒント！！
  - Math.rondom()関数で0から4を発生させる.
  - 出た数値に応じてif文を使って条件分岐し, 異なる内容をalert()で出力！

Webアプリを実装！

# おみくじのWebアプリ！

---

- 画面上のHTML要素(DOM)をクリックして処理を実行！
- 要素を「指定」する！
  - classやidでDOMを特定する.
  - 指定したDOMに対してJavaScriptで操作を行う！
- 例
  - 「idがbutton」の要素を「クリック」したら...
  - 「大吉-大凶のどれかをランダムに表示」！

# 【参考】DOM

## - HTMLに記述されている各要素のこと(document object model)

The screenshot shows a web browser window with the Wikipedia page for "Document Object Model". The browser's developer tools are open, displaying the DOM tree on the right side. The DOM tree shows the root element `<html>` with attributes `class="client-js ve-available"`, `lang="ja"`, and `dir="ltr"`. The `<body>` element is selected, showing its attributes and children, including `<div id="mw-page-base" class="noprint">`, `<div id="mw-head-base" class="noprint">`, `<div id="content" class="mw-body" role="main">`, `<div id="mw-navigation">`, `<div id="footer" role="contentinfo">`, `<script>`, `<script type="application/ld+json">`, `<script>`, and `<div class="suggestions" style="display: none; font-size: 13px;">`. The `<a accesskey="v" href="https://ja.wikipedia.org/wiki/Document_Object_Model?action=edit" class="oo-ui-element-...">` link is visible within the suggestions div.

ログインしていません トーク 投稿記録 アカウント作成 ログイン

ページ ノート 閲覧 編集 履歴表示 Wikipedia内を検索

## Document Object Model

出典: フリー百科事典『ウィキペディア (Wikipedia)』

**Document Object Model (DOM)** は、HTML 文書やXML文書を各種プログラムから利用するための仕組みである<sup>[1]</sup>。WHATWGがLiving Standardとして定義している。

WHATWG以前はW3Cが仕様を策定しており、Level 1からLevel 4まで勧告している。

XMLを読み込むAPIであるSAXと異なり、XML

**HTML**

```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
<head>
<title>sample</title>
</head>
<body>
<p>Voluptatem accusantium
totam rem aperiam.</p>
</body>
</html>
```

Elements Console Sources 3

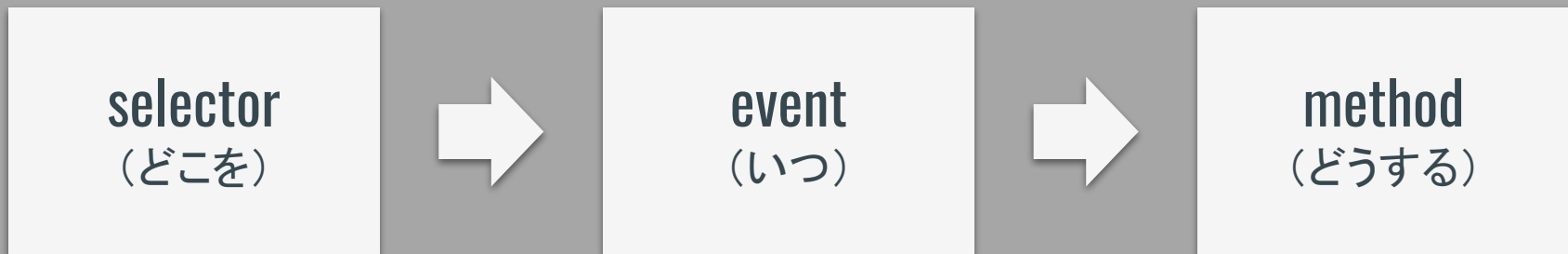
```
<!doctype html>
<html class="client-js ve-available" lang="ja" dir="ltr">
  <head>...</head>
  <body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0
ns-subject mw-editable page-Document_Object_Model rootpage-
Document_Object_Model skin-vector action-view"> == $0
    <div id="mw-page-base" class="noprint"></div>
    <div id="mw-head-base" class="noprint"></div>
    <div id="content" class="mw-body" role="main">...</div>
    <div id="mw-navigation">...</div>
    <div id="footer" role="contentinfo">...</div>
    <script>...</script>
    <script type="application/ld+json">...</script>
    <script>...</script>
    <div class="suggestions" style="display: none; font-size:
13px;">...</div>
    <a accesskey="v" href="https://ja.wikipedia.org/wiki/
Document_Object_Model?action=edit" class="oo-ui-element-
hidden">...</a>
  </body>
```

Styles Event Listeners DOM Breakpoints Properties Accessibility

# 【重要】基本の考え方

---

- 基本の3要素



なんだけど...



# JavaScriptはDOM操作が苦手...



# jQueryライブラリ



# 【重要】最初に読み込みが必要！

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
<script>  
    // jsの処理  
</script>
```

# jQueryとは？？

---

- cssと同じ要領で対象箇所を指定できる
- 素のJavaScriptよりも短く書ける！ <- 重要
- アニメーションなど手軽に設定できる.
- 書き方(順序や考え方)はJavaScriptと同様！ <- 重要
- 導入が簡単(フレームワークなどは環境構築で詰む)

jQueryはJavaScriptを短縮して書けるライブラリ

【参考】[https://webkikaku.co.jp/homepage/blog/hpseisaku/webdesign/jquery\\_start/](https://webkikaku.co.jp/homepage/blog/hpseisaku/webdesign/jquery_start/)

## 【重要】基本の考え方

---

**selector**  
(どこを)



**event**  
(いつ)



**method**  
(どうする)

```
$('#button').on('click', function () {  
    alert('Hello World!');  
});
```

## 【重要】基本の考え方

---

**selector**  
(どこを)



**event**  
(いつ)



**method**  
(どうする)

```
$(セレクト名).on(イベント名, function () {  
    実行したい処理 (メソッド)  
});
```

# セレクトタ / イベントなど

---

- たくさんあります
  - 「jQuery セレクトタ」でググる！
  - 「jQuery イベント」でググる！

まずは形の入力に慣れよう！

```
$('#id').on('click', function () {  
    // ...  
});
```



（口に出しながら書くと定着する）

だらーあいでいーおんくりっくふあんくしょんかっこかっこなみかっこえんたー..

# とにかく慣れる！！！！

---

**selector**  
(どこを)



**event**  
(いつ)



**method**  
(どうする)

```
$('#button').on('click', function () {  
    alert('Hello World!');  
});
```

演習！！！！

# 演習！！！（omikuji02.html）

---

- 仕様
  - おみくじボタンをクリックしたら以下のどれかを表示！
  - 「大吉・中吉・小吉・凶・大凶」
- ヒント

```
$('#button').on('click', function () {  
    // 0から4でランダムな数を作成  
    // 0だったら大吉, 1だったら中吉. . .  
    // 結果をidで指定した場所に表示  
});
```

# 課題

# じゃんけんを作ろう！！

---

- じゃんけんアプリの仕様
  - ①「グー」「チョキ」「パー」のボタンを設置
  - ②どれかをクリックしたら「コンピュータの出した手は？」を変更
    - 「コンピュータ:グー」「コンピュータ:チョキ」など
  - ③「結果は？」の箇所に
    - 「あなたの負け」「あなたの勝ち」「あいこ」のどれかを表示！
- ※上記を最低ラインとして製作
- ※これを土台にしてガンガン発展させよう！！



誰かに言われたものを作る



オリジナリティを表現する



プログラムを書く



プロダクトを創る



締切は11/12「23:59:59」  
提出方法は11/07講義で案内！

やばいいい... (´;ω;´)

完全に詰んだ．．．という方は

# 写経

※ 写経とは ※

誰かが書いた「動くコード」をひたすら書き写すこと

```
1 <!DOCTYPE html>
2 <html lang="ja">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <meta http-equiv="X-UA-Compatible" content="ie=edge">
8     <link rel="stylesheet" href="css/sample.css">
9     <title>じゃんけん</title>
10 </head>
11
12 <body>
13     <header>
14         <h1>じゃんけん</h1>
15     </header>
16
```

```
12 <body>
13   ....<header>
14     ....<h1>じゃんけん</h1>
15   ....</header>
16
17   ....<main>
18     ....<ul>
19       ....<li id="gu_btn">グー</li>
20       ....<li id="cho_btn">チョキ</li>
21       ....<li id="par_btn">パー</li>
22     ....</ul>
23     ....<div id="com_hand">コンピュータの出した手は？</div>
24     ....<div id="judgment">結果は？</div>
25   ....</main>
26
27   ....<footer>フッター</footer>
28
```

```
29 .....<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
30 .....<script>
31 .....$(function(){
32 .....    const result = ['グー', 'チョキ', 'パー'];
33 .....
34 .....    $('#gu_btn').on('click', function(){
35 .....        // 自分の手はグー
36 .....        let rand = Math.floor(Math.random() * 3);
37 .....        // 0コンピュータグー, 1コンピュータチョキ, 2コンピュータパー
38 .....        // コンピュータの手を表示
39 .....        $('#com_hand').text('コンピュータの手: ' + result[rand]);
40 .....        if (rand == 0){
41 .....            $('#judgment').text('あいこ');
42 .....        } else if (rand == 1){
43 .....            $('#judgment').text('勝ち');
44 .....        } else {
45 .....            $('#judgment').text('負け');
46 .....        }
47 .....    });
48 .....
```

```
49 ..... $('#cho_btn').on('click', function () {
50 .....     // 自分の手はチョキ
51 .....     let rand = Math.floor(Math.random() * 3);
52 .....     // 0コンピュータグー, 1コンピュータチョキ, 2コンピュータパー
53 .....     // コンピュータの手を表示
54 .....     $('#com_hand').text('コンピュータの手: ' + result[rand]);
55 .....     if (rand == 0) {
56 .....         $('#judgment').text('負け');
57 .....     } else if (rand == 1) {
58 .....         $('#judgment').text('あいこ');
59 .....     } else {
60 .....         $('#judgment').text('勝ち');
61 .....     }
62 ..... });
63
```



```
64 ..... $('#par_btn').on('click', function() {
65 ..... // 自分の手はチョキ
66 ..... let rand = Math.floor(Math.random() * 3);
67 ..... // 0コンピュータグー, 1コンピュータチョキ, 2コンピュータパー
68 ..... // コンピュータの手を表示
69 ..... $('#com_hand').text('コンピュータの手: ' + result[rand]);
70 ..... if (rand == 0) {
71 .....     $('#judgment').text('勝ち');
72 ..... } else if (rand == 1) {
73 .....     $('#judgment').text('負け');
74 ..... } else {
75 .....     $('#judgment').text('あいこ');
76 ..... }
77 ..... });
78
79 ..... });
80 ..... </script>
81
82 </body>
83
84 </html>
```

「写経」これでいける！ 締切厳守！

# P2Pタイム

まずはチーム内で解決を目指す！

訊かれた人は苦し紛れでも応える！！