

# 変数 & 条件分岐



G's ACADEMY  
FUKUOKA



# アジェンダ

- JavaScriptの役割, できること, 概要
- 変数
- 条件分岐 (if文)
- 乱数生成でおみくじ作成 (Mathオブジェクト)
- jQuery
- おみくじアプリ演習
- 課題発表→チュータリング(演習)タイム
- 写経のススメ(?)

# 授業のルール

- 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

## 今日のゴール

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

# JavaScript概要

# 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ってどんな言語？？

# JavaScript $\neq$ Java



# Webアプリケーションの構造

## フロントエンド



Webブラウザで動く

JavaScript

アクセス

データ

## サーバサイド



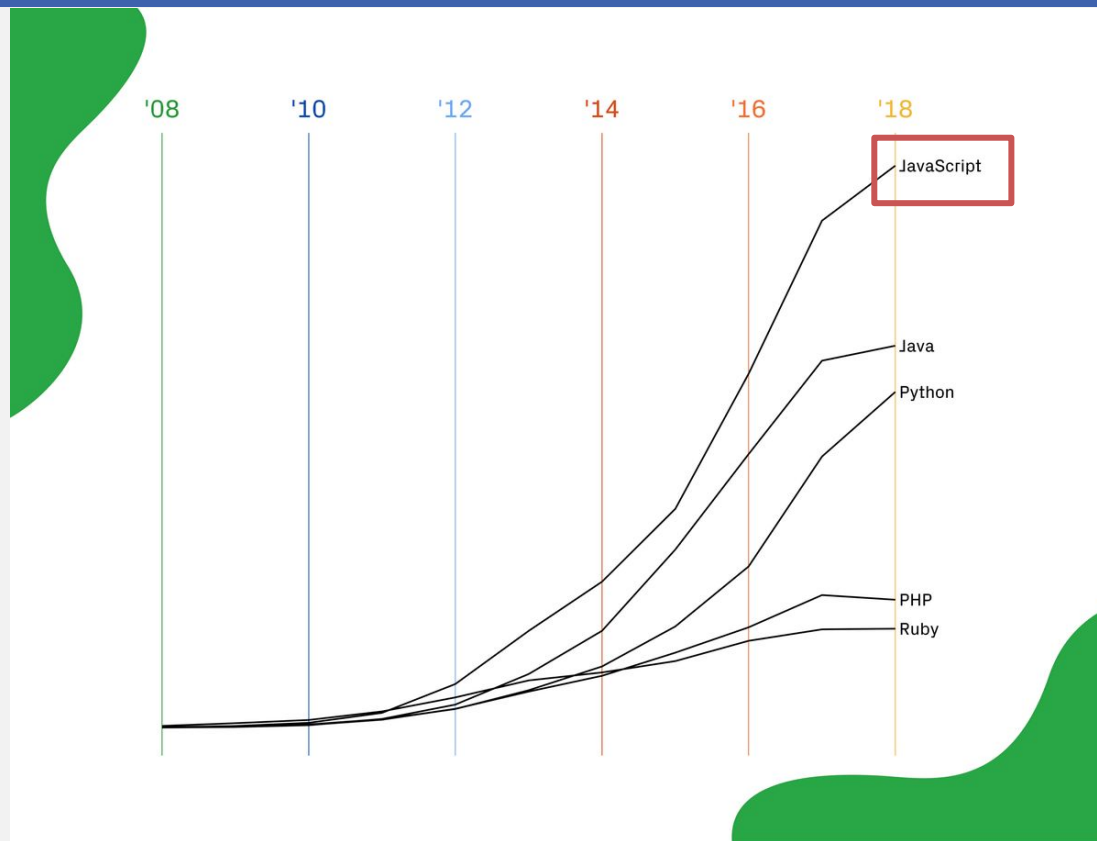
サーバで動く

PHP, Python  
Ruby, JAVA, ...

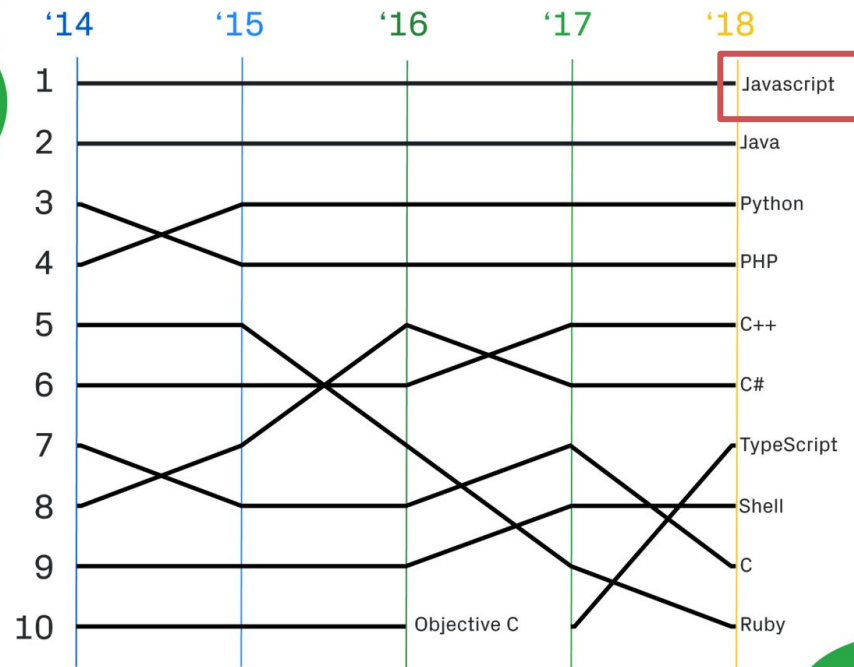
Webアプリに欠かせない存在！！

で、実際JavaScriptってどうなの??

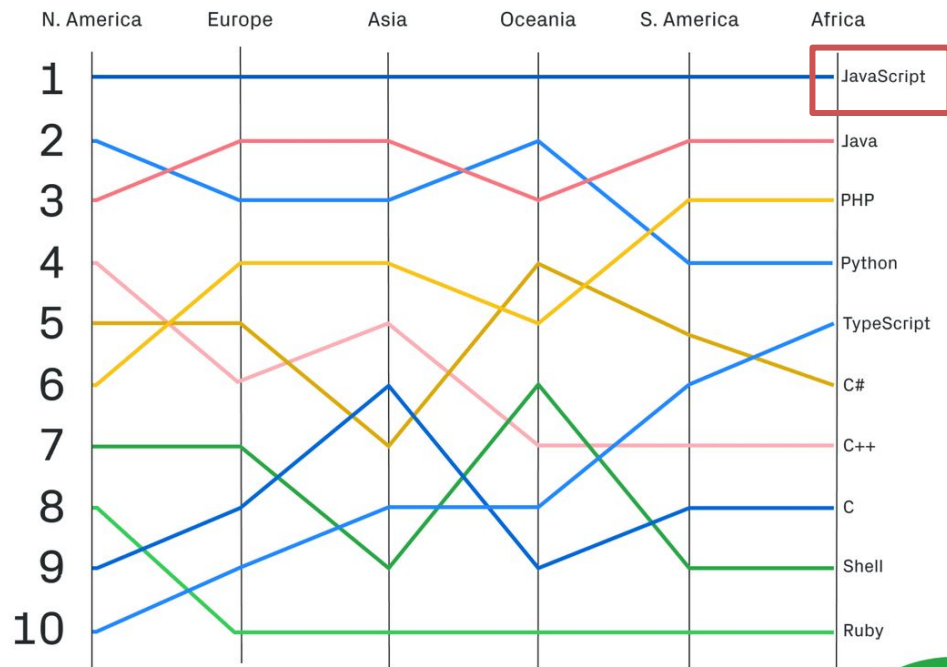
# Githubのリポジトリ数



# プログラミング言語の人気ランキング



# 地域別の人気ランキング



# JavaScriptに関する技術







- 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！



- 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)



- 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基礎

# JavaScriptの記述場所

- 書き方

- `<script></script>`の間に処理を記述

- どこに書くの??

- **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.

# JavaScriptの記述場所

```
<!doctype html>  
<html lang="ja">  
<head>.....</head>
```

```
<body>  
  <h1 id="echo">js練習</h1>  
  <script>  
    // ここにjavascriptの処理を記述！  
  </script>  
</body>  
  
</html>
```

</body>のすぐ上！

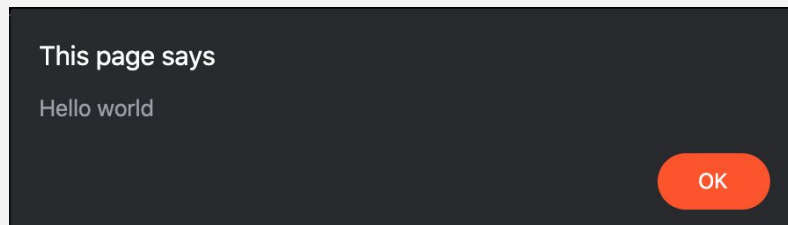
## まずは書いて動かそう！！

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

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

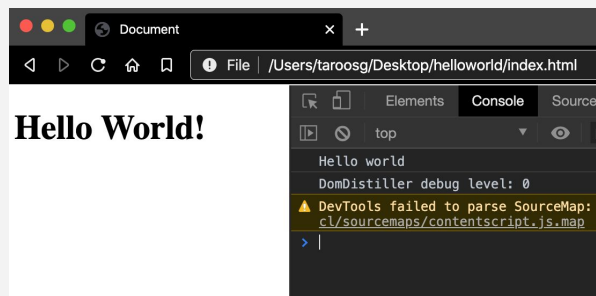
- 前ページを参考にalert()とconsole.log()を動かそう！

- alert()



リロードすると表示される！

- console.log()



consoleに表示される！



# 変数

# 変数とは？？(最初のハマリポイント)

- 数学で出てきたアレ

$$x = 2$$

$$y = 4$$

$$z = 2x + y$$

この $x, y, z$ が変数！

のとき,  $x + y + z$  の値を求めよ！

## 変数とは?? (最初のハマリポイント)

```
// - プログラミングの変数
//   - 数字以外のものにも使用できる.
//   - 例
<script>
  const text1 = 'hello'; // 文字列 (''などで囲む)
  const text2 = 'world'; // 文字列
  const number = 100;    // 数値
</script>
```

「=」は「等しい」ではない  
「numberに100を代入」の意味



## 変数の定義(最初に変数を使うときの話)

// 変数を使うときは最初に定義が必要！

// 宣言は3種類！

<script>

const number1 = 10; // 一度入れたら中身を変更できない箱

number1 = 20; // エラー

let number2 = 100; // 中身を入れ替えられる箱

number2 = 200; // OK！

var number3 = 300; // 古い書き方（非推奨 / 出番なし！）

</script>

まずは「const」を使って宣言！

→ 実行時エラーが出たら適宜letに変更！

# ダメなやつ

```
// - NGパターン  
//   - 「"」 「'」 の混在  
//   - 変数名の先頭が数字  
// - ↓↓↓これはまずい！↓↓↓
```

```
<script>  
  let text1 = "hello";  
  let text2 = world;  
  let 1number = 100;  
</script>
```

# 変数を使う意味

- 数値や文字列を再利用できる

```
const a = 3;
```

九九の3の段を計算！

```
const x = a * 1;
```

```
const y = a * 2;
```

```
const z = a * 3;
```

4の段に変更しようとした場合,

- 変数を使わないと全部書き換えが必要...
- 変数を使えば, 上の「a」の値を変更するだけでOK！

# 演算子と計算

```
// - 演算子  
//   - 「+」とか「-」のこと.
```

```
// - 例
```

```
<script>  
  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>
```



// - 数値や文字列の計算

// - 例

<script>

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;

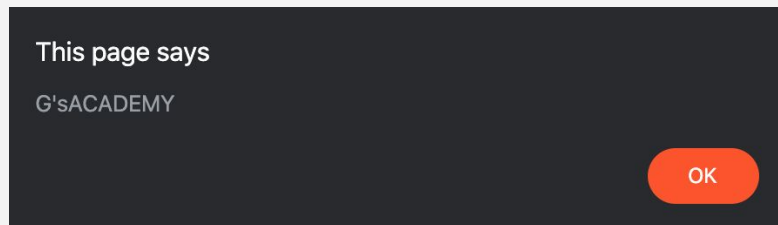
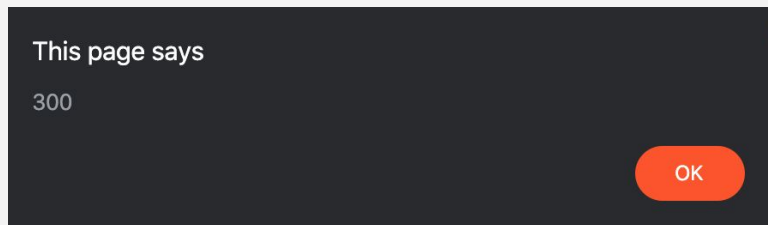
// helloworld (文字列)

</script>

# 使用できない単語

- 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！！
- 引がかかる場合は少ないので今は気にしなくてOK！

- 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)



# 条件分岐

# 条件分岐

// - 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！

// - 例

```
<script>  
  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  
  }  
</script>
```

# 条件分岐

// - 複数の条件で処理を分岐させることもできる！

// - 例

```
<script>
```

```
  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
```

```
  }
```

```
</script>
```

【Point】

先に「if」「else if」などの枠をつくらうまくいやすい！

# 条件分岐

// - 条件を決める条件式の例

// - 例

```
<script>
```

```
  const number = 10;
```

```
  if(number == 10){
```

// numberの値が10と等しい (true) とき

```
    alert('値は10です！');
```

```
  } else {
```

```
    alert('値は10以外です！');
```

```
  }
```

```
</script>
```

# 条件分岐

## - 比較演算子の種類

`==` 等しければtrue

「=」ではなく「==」です！！

`!=` 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



# 条件分岐

// - 条件の組み合わせ

// - 例

```
<script>
```

```
  const number = 10;
```

```
  if(number >= 5 && number <= 15){
```

```
    // numberが5以上15以下のときに実行する処理
```

```
  } else if(number == 100 || number == 200) {
```

```
    // numberが100または200のときに実行する処理
```

```
  } else {
```

```
    // 上記2つを満たさないときに実行する処理
```

```
  }
```

```
</script>
```

# Mathオブジェクト

```
// - Mathオブジェクトは数字の操作ができる関数のセット！  
// - （最初から準備されているので書くだけでOK）
```

```
// - 例
```

```
<script>
```

```
    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>
```

- 以下の数をランダムで発生させてalert();で表示させよう！
  - ① 0から9のどれか
  - ② 1から9のどれか
  - ③ 5から10のどれか
  - ④ 50から99のどれか

# 演習1

### - 仕様

- ランダムに「大吉・中吉・小吉・凶・大凶」をalert()で表示！

### - ヒント

- Math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！

おみくじをwebアプリに！

# DOM



# HTML = document

htmlに記述されている各要素のこと(document object modelでググろう)

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



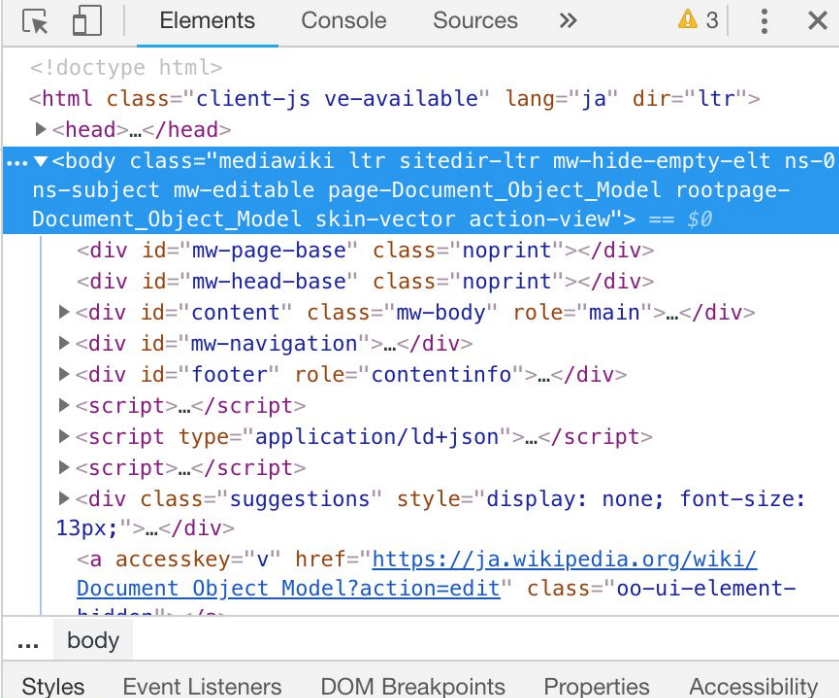
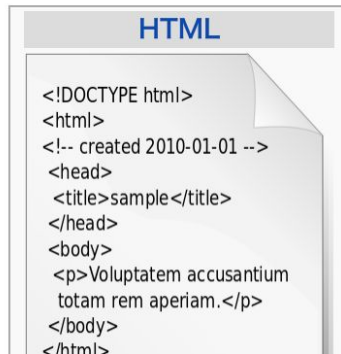
## Document Object Model

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



**DOM = 要素**

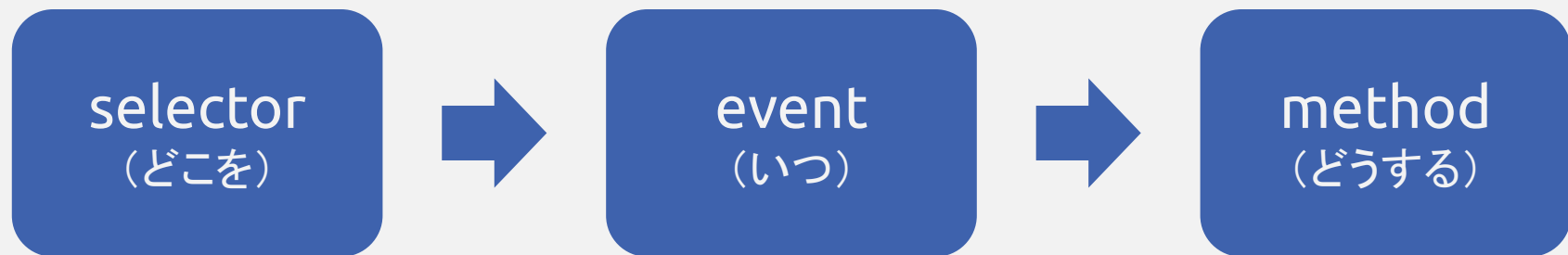
# DOMになにかの操作を行ったときに処理を実行！

- 要素って？？
  - <div>とか<p>とかのこと.
  - タグで囲われたものを要素と呼ぶ
- 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(おみくじアプリ)
  - 「idがbutton」の要素を「クリック」したら「大吉-大凶のどれかを表示」

# 基本的な考え方

# どこを, いつ, どうする?

- 基本の3要素!



なんだけど...

# JavaScriptでDOM操作はなかなかつらい...



# jQueryライブラリ

# jQueryとは

## - 特徴

- 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
- 1行追加するだけで導入が簡単(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/)





web サイト 用途なら今でも全然使える

- ・ web サイト内の要素をグリグリ動かすなど、用法用量さえ間違えなければこれほど便利なライブラリは他にない（現在進行形）
- ・ オワコンと言っているのは、より複雑な web アプリを作っているごく一部の『酔狂な人々』であり、彼らの多くは件の web サイト制作のことは一切視野に入れていない

【参考】<https://speakerdeck.com/wakamsha/rmp-enziniabutokiyantu-2017-youkoso-web-hurontoendofalseshi-jie-he?slide=53>

**まずはたくさん書いてなれることが大事！**

# jQueryの準備

## jQueryの準備(超重要)

// まず思考停止で以下を記述 (サンプルファイルには記述済み)

```
<script
```

```
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

```
<script>
```

```
$(function(){
```

```
    //ここに自分の処理を書く！
```

```
});
```

```
</script>
```

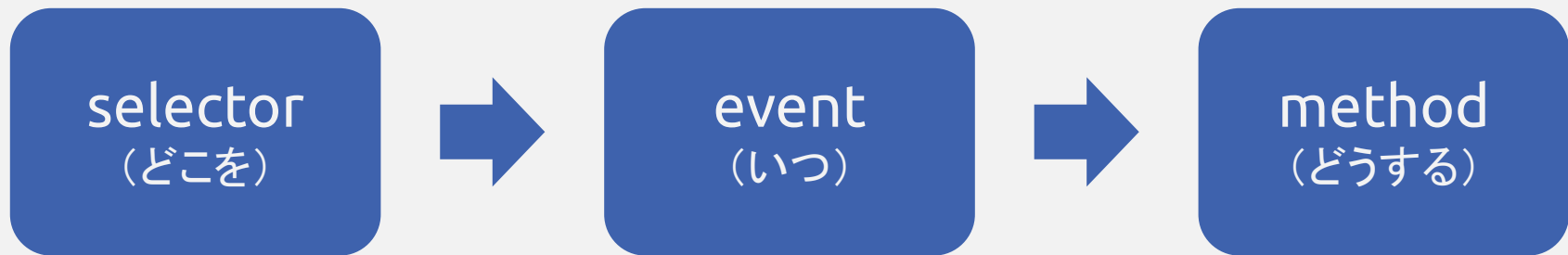
// 記述場所は</body>のすぐ上！

// 忘れたら「jQuery google」でググる！

# jQueryの基本

# jQueryの文法(覚えよう)

## - 基本の3要素！



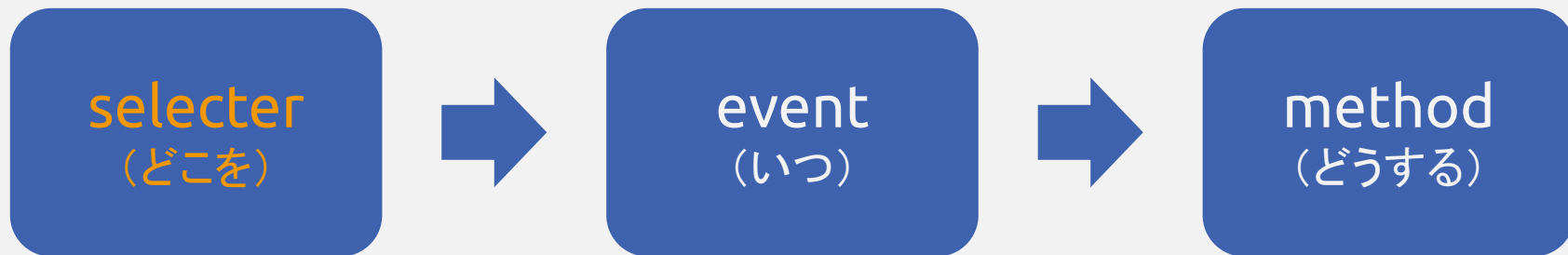
## - 例(基本の形はだいたいこれ！)

- idが「button」の要素をクリックしたらアラートを表示

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

# selector

- 基本の3要素！（javascriptと同じ！）



- 参考

<http://www.hp-stylelink.com/news/2013/11/20131122.php>

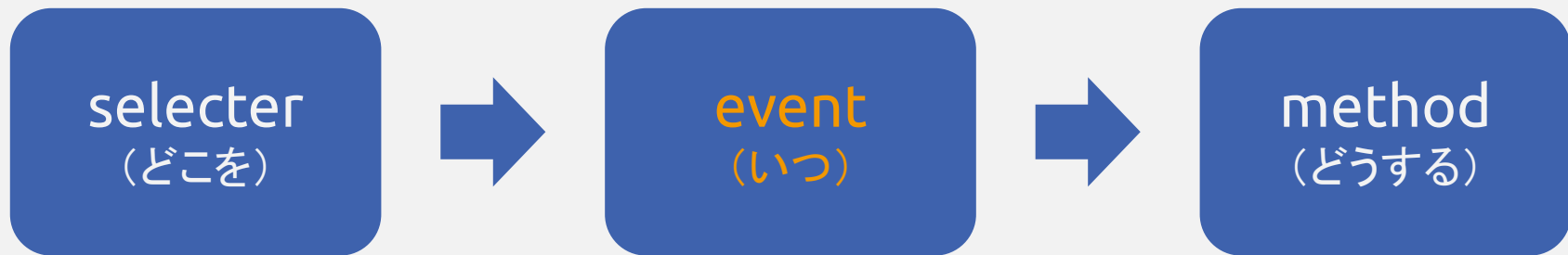
- cssで設定するときと同様

css	jQuery
#id	<code>\$('#id')</code>
.class	<code>\$('.class')</code>
Element (タグ名)	<code>\$('Element')</code>
[name=a]	<code>\$('[name=a]')</code>

- ※「#id」などは「"」か「'」で囲む！
- ※どちらを使っても良いが統一しよう！！



- 基本の3要素！（javascriptと同じ！）



# event

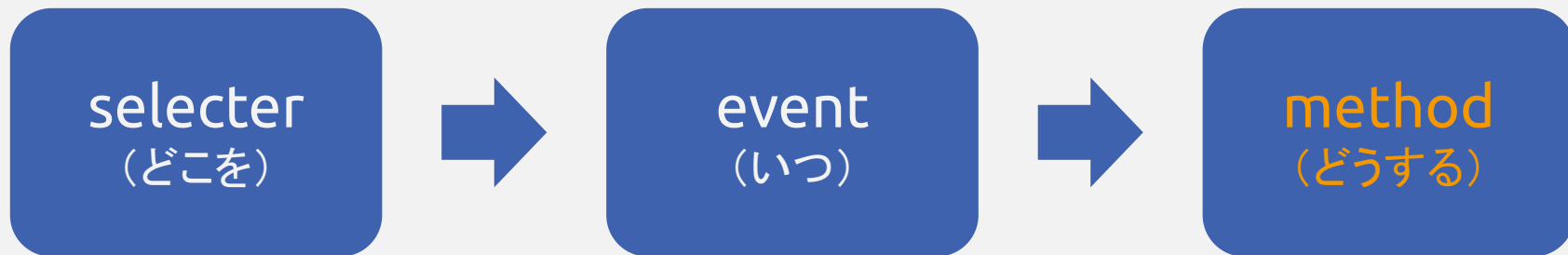
- 例: クリック時に「{}」内を実行！

```
$('#id').on('click', function () {  
    alert('Hello World');    // 実行される処理  
    const number = 100;      // 実行される処理  
    console.log(number);    // 実行される処理  
});
```

※「click」部分をほかのものに変更することでイベントを指定できます！

'click', 'dblclick', 'mouseout', 'mousedown', 'mouseup', 'change', 'focus', 'submit',  
'resize', 'scroll', 'keydown', 'keyup', ...

- 基本の3要素！（javascriptと同じ！）



※ イベントを設定せずにメソッド単体で記述すると、読み込み時に実行される！

## method(命令)

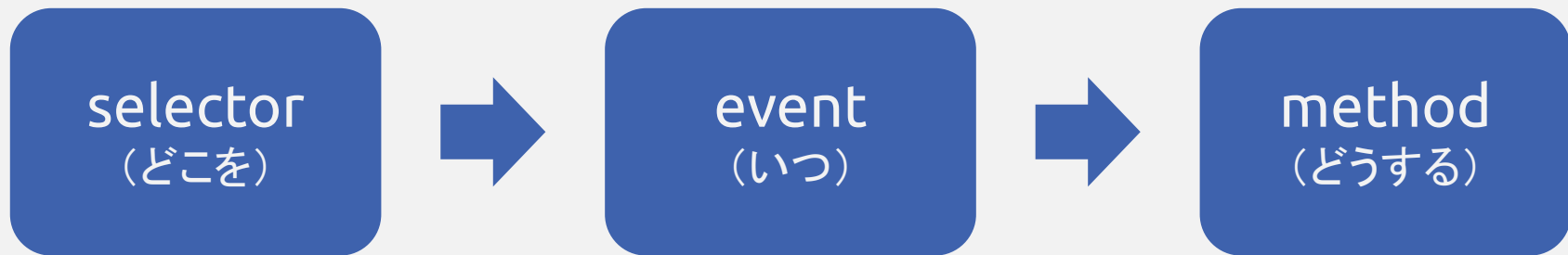
```
// cssを操作する
const elem = '<a href="#">次ページ</a>';

$('#id').css('color', 'red');           // 色を変更
$('#id').css('fontsize', '24px');       // フォントサイズを変更
$('#id').css('flex-direction', 'column'); // flexの縦横を変更
$('#id').show(4000);                    // 非表示→表示(逆はhide)
$('#id').fadeIn(1000);                  // 非表示→表示(逆はfadeOut)
$('#id').html(elem);                    // htmlを表示
$('#id').text(elem);                    // 文字列として表示
```

もう一度まとめ

# jQueryの文法(覚えよう)

- 基本の3要素！



- 例(基本の形はだいたいこれです！)

- idが「button」の要素をクリックしたらidが「result」の要素に計算結果を表示

```
$('#button').on('click', function () {  
    const total = 100 + 200;  
    $('#result').text(total);  
});
```

**まずは形を入力することに慣れよう！**

## (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つくふあんくしょん  
かっこかっこ...



# 演習2

### - 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

```
<script>  
  $('#btn').on('click', function(){.....}); // クリックの検出  
  // Math.random(), if(...){...}など  
  $('#id名').html('大吉');// 結果の表示  
</script>
```

# 課題

- 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！！

**提出は次回授業前木曜「23:59:59」まで！！**  
**詳細はあとのスライド参照！**

やばいいいいい . . .  
(` ; ω ; `)

詰んだ... どうしようもない... という方は

# 写☆経

※写経とは

誰かが書いた動作する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！！  
提出は次回授業前まで！！**

# 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！！