

たつこデラックス

# 1回目の制作物

## G's ACADEMY TOKYO

LAB



スタート

12

3

22

26

33

11

15

40

20

24

6

28

16

34

18

8

17

1

19

13

21

2

23

10

25

5

27

30

29

36

31

4

7

32

35

38

37

14

39

9

# 1回目の制作物

## 「G's 席替えシステム」

ボタンを押すと、ランダムで授業時の各受講生の座席をランダムに割り振りして表示。

- ・ LABかDEVかを選んで、選んだらコースに応じた人数分だけランダムに座席が割り振られる。
- ・ 実際に割り振りが表示されるのはスタートボタンを押した時。スタートボタンを連続で押した場合は、改めてランダムな割り振りが表示される。



# 理想を言えば

福岡DEVを追加しても、少ない  
コード変更で仕様成立するように！！  
(福岡DEVは30名)



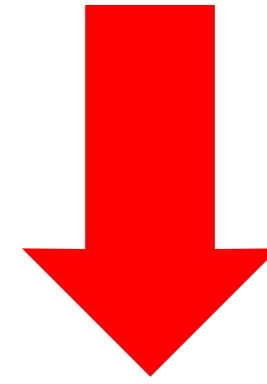
まず、はじめに。。。。

18人もいると思ってなかった・・・！！！！



# 出てきそうな懸念事案

人によって求めるものの振れ幅がある・・・



流れの組み方に悩みました。。。結果

# 基本的な考え方

- DEPLOYフェーズでやった「たつこの部屋」のやり方を基本的には踏襲。
- Vue.js講義を重複してとっている方が8割強いるので、その辺も意識。
- jQueryは使用しない → Vanilla.jsで進める。
- まずは完成を目指す！！！！→最後の方、余力があったら新しい概念も。



jQueryばかりでVanilla JS怖い・・・という方へ

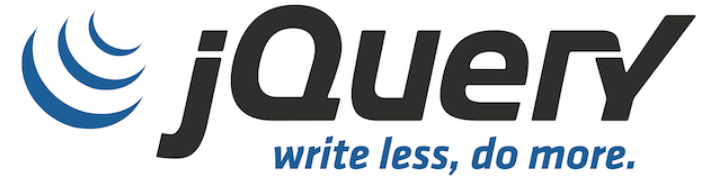




# Vanilla JSとjQueryとの大きな違い

基本的には「コード量」

# 例



`$('#app')`

`.html()`

`.addClass("className")`

`.on("click, ...")`



`document.querySelector("#app")`

`innerHTML`

`classList.add("className")`

`addEventListener("click", ...)`

作る前に・・・知っておきたい・その1

ES2015? 2017? ってそもそもなんだっけ



# ES・ECMA Scriptとは

## JavaScriptの標準仕様が定義されたもの

2015,2016,2017,2018と、近年毎年のように  
アップデートが図られているもの。

・ ESの内容がアップデートされていることによって以前使用できなかったものが近年のJSではたくさん使用可能になっています。



# 近年JavaScriptで使えるようになったもの

const, let . . . ES2015以降  
includes . . . ES2016以降  
padEnd . . . ES2017以降

【詳しくは下記を参照】

<http://kangax.github.io/compat-table/es6/>



JavaScriptまじ便利じゃん！・・・？

JS便利になっている～わ～いわ～い\ ^o^/



立ちはだかるあいつ・・・



**IEの壁！！！！！！**

**(2015年10月まで)**



		98%	Compilers/polyfills							Desktop browsers								
Feature name	Current browser	Traceur	Babel 6 + core-js 2	Babel 7 + core-js 2	Babel 7 + core-js 3	Closure 2018.11	Type-Script + core-js 2	Type-Script + core-js 3	es6-shim	Kong 4.14 <sup>[1]</sup>	IE 11	Edge 17	Edge 18	Edge 19 Preview	FF 60 ESR	FF 63	FF 64	FF 65 Beta
Optimisation																		
proper tail calls (tail call optimisation)		0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2
Syntax																		
default function parameters		7/7	4/7	4/7	4/7	4/7	5/7	5/7	5/7	0/7	0/7	0/7	7/7	7/7	7/7	7/7	7/7	7/7
rest parameters		5/5	4/5	3/5	3/5	3/5	2/5	4/5	4/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5
spread syntax for iterable objects		15/15	15/15	13/15	13/15	13/15	11/15	14/15	14/15	0/15	0/15	0/15	15/15	15/15	15/15	15/15	15/15	15/15
object literal extensions		6/6	6/6	6/6	6/6	6/6	5/6	6/6	6/6	0/6	0/6	0/6	6/6	6/6	6/6	6/6	6/6	6/6
for..of loops		9/9	9/9	9/9	9/9	9/9	6/9	9/9	9/9	0/9	0/9	0/9	9/9	9/9	9/9	9/9	9/9	9/9
octal and binary literals		4/4	2/4	4/4	4/4	4/4	2/4	4/4	4/4	2/4	0/4	0/4	4/4	4/4	4/4	4/4	4/4	4/4
template literals		5/5	4/5	4/5	4/5	4/5	3/5	3/5	3/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5
RegExp "y" and "u" flags		6/6	4/6	4/6	4/6	4/6	0/6	0/6	0/6	0/6	0/6	0/6	6/6	6/6	6/6	6/6	6/6	6/6
destructuring, declarations		22/22	20/22	21/22	21/22	21/22	20/22	21/22	21/22	0/22	0/22	0/22	22/22	22/22	22/22	22/22	22/22	22/22
destructuring, assignment		24/24	23/24	24/24	24/24	24/24	22/24	24/24	24/24	0/24	0/24	0/24	24/24	24/24	24/24	24/24	24/24	24/24
destructuring, parameters		24/24	19/24	21/24	21/24	21/24	20/24	21/24	21/24	0/24	0/24	0/24	23/24	24/24	24/24	24/24	24/24	24/24
Unicode code point escapes		2/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2	2/2	2/2
new.target		2/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2	2/2	2/2
Bindings																		
const		18/18	16/18	16/18	16/18	16/18	16/18	16/18	16/18	0/18	2/18	14/18	18/18	18/18	18/18	18/18	18/18	18/18
let		14/14	12/14	12/14	12/14	12/14	12/14	12/14	12/14	0/14	0/14	12/14	14/14	14/14	14/14	14/14	14/14	14/14
block-level function declaration <sup>[17]</sup>	Ⓢ	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Functions																		
arrow functions		13/13	11/13	9/13	9/13	9/13	10/13	9/13	9/13	0/13	0/13	0/13	13/13	13/13	13/13	13/13	13/13	13/13
class		24/24	17/24	19/24	19/24	19/24	14/24	19/24	19/24	0/24	0/24	0/24	24/24	24/24	24/24	24/24	24/24	24/24
super		8/8	7/8	4/8	4/8	4/8	7/8	7/8	7/8	0/8	0/8	0/8	8/8	8/8	8/8	8/8	8/8	8/8
generators		27/27	24/27	24/27	24/27	24/27	19/27	22/27	22/27	0/27	0/27	0/27	27/27	27/27	27/27	27/27	27/27	27/27



*BABEL*



# 卒制にES2015以降の仕様を取り入れる場合

就職（特にフロントエンド頑張りたい方）・・・当面はbabelやらwebpackやらついてくる

起業志望の方・・・自身がやるかはさておき、ブラウザ対応完璧にしたいならbabelやらwebpackやら、最低限知識としてでもやっぱり知っておく必要がどこかで出てくる。

※今回のお話を踏まえて、卒業制作に「取り入れない」という選択肢を取る方もいらっしゃると思います。それはそれで悪いことでもなんでもありません！

WebPackについては下記あたりを参照

<https://www.slideshare.net/ssuser46977e/webpack-why-cant-you-understand-the-webpack>

# 作る前に・・・知っておきたい・その2

esLintとか、なんとか  
(特に就転職向け?)



# ESLintとは

## JavaScriptの静的検証ツール

JSLint、JSHintなどが類似してあるが、近年1番使用されている検証ツールとのこと。

- ・ 複数人で開発しても統一された美しいコードを実装することを助けるツール。
- ・ 明らかなバグ（あるいはバグではないが未使用変数が定義されたままなど）を見つけるのを助けてくれるツール。

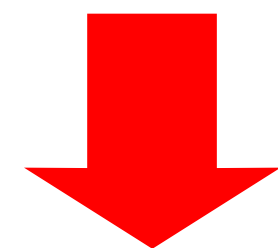


# ESLintとは

```
.addEventListener("change",
```

基本シングルクォーテーションにしてねーとか。

```
const makeArray = function(){  
[eslint] Missing space before opening  
brace. [space-before-blocks]
```



()のあとはスペース入れてねーとか。

```
const makeArray = function() {
```

# Visual Studio Codeでの設定方法

<https://qiita.com/howdy39/items/6e2c75861bc5a14b2acf>



# 今日のお題を作ってみよう

**まずは、完成させることを目指す！**

たぶんやり方めっちゃあるので！焦らず！自分のやり方でOK！



# うーん、悩むという方へ・・・

たつこの部屋の時同様、「仕様の分割」を考えてみましょう！

- 1、まずはdemo.htmlの内容をJavaScriptを使って同じ表示ができるようにする。（id=“app”のついたdivの中身をJSで表示）
- 2、座席がランダムに振り分けられるようにする。
- 3、スタートボタンがクリックされた時に2の仕様を満たす。
- 4、LAB or DEV、選んだものに応じた人数で2の仕様を満たす。



# うーん、悩むという方へ・・・

```
<div class="table-item text-center">  
  <div class="seat">22</div>  
  <div class="seat">13</div>  
  <div class="seat">6</div>  
  <div class="seat">25</div>  
  <div class="seat">24</div>  
  <div class="seat">14</div>  
</div>
```

```
<div class="table-item text-center">  
  <div class="seat">1</div>  
  <div class="seat">23</div>  
  <div class="seat">5</div>  
  <div class="seat">9</div>  
  <div class="seat">3</div>  
  <div class="seat">11</div>  
</div>
```

繰り返す



# なんとなくわかるけど設計考えたい方へ

「データ操作」と「DOM表示」を切り分けて設計してみよう。

- 座席表を表示するための関数
- 選ばれているのがLABかDEVなのかを管理する変数を操作する関数
- 選ばれているコースに応じた数の配列を作成する関数



# なんとなくわかるけど設計考えたい方へ

どんな「機能」「データ」が必要なのかを割り振りしてみましょう。

- ・HTMLにランダムな座席割りを表示する。（関数）



# 課題内容

## 「G's 席替えシステムZ」

以下の仕様を「**できる限り**」追加してほしいです。

- 1、ルーレット風の演出を行う。（ガラガラジャーン→ストップ）
- 2、欠席者がいる場合は、欠席者分を抜いた状態で座席割り振りを  
する仕組みを追加する。
- 3、加納さん山田さんにヒヤリング→欲しいと言われた機能を追加。
- 4、視力が悪い人は必ず前の3テーブルに配置されるようにする。（難）



※次回授業前日23:59までにご提出頂ければ、僕なりにチェック→コメントを返します！（少し時間はかかるかもしれませんが）