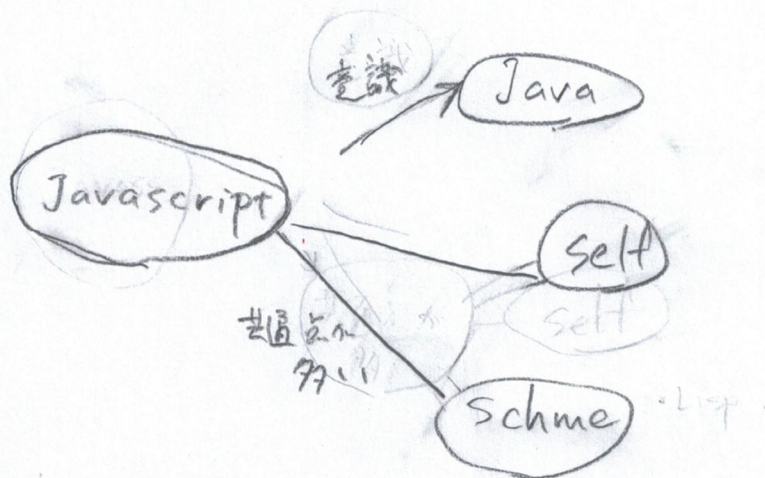


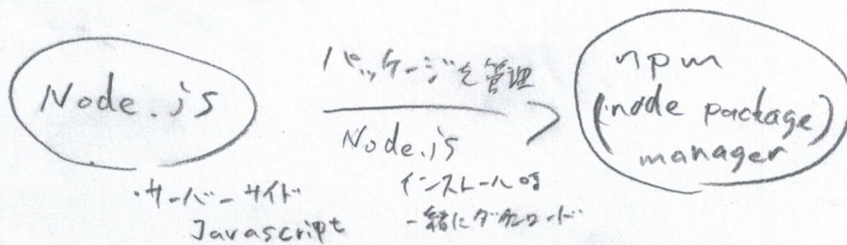
1章 歴史

JavaScriptの歴史


Brendan Eich
1995年に開発



JavaScriptの各種ツール



Babel

ES6のプログラムを
ES5に変換

Gulp

開発に必要な作業の
自動化ツール

ESLint

JavaScriptのコード中で怪しい部分を
指摘、エラー発見しやすくなる

2章

Canvas

- HTML5から導入
- さまざまな図形や線を使った画像を描く


使い方

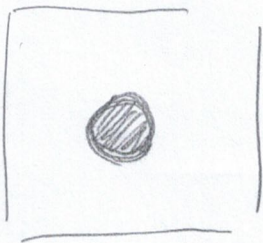
- ① paper.jsを読み込み
- ② canvasタグのidを指定
- ③ ②で指定したidでCSSを記述

画像の生成

例

```
#mainCanvas {  
  width: 400px  
  height: 400px  
  border: solid 1px black;  
}
```

⇒  ← 画像

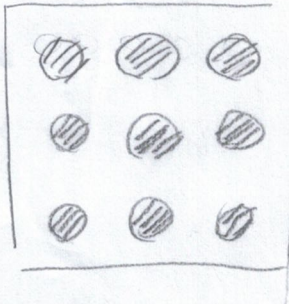


コード
 \Rightarrow `let c = shape.Circle(200, 200, 50);`
`c.fillColor = 'green';`

例: 円オブジェクトを生成

2行目: 引数 (円の中心座標(緯度), 円の中心座標(経度), 円の半径)

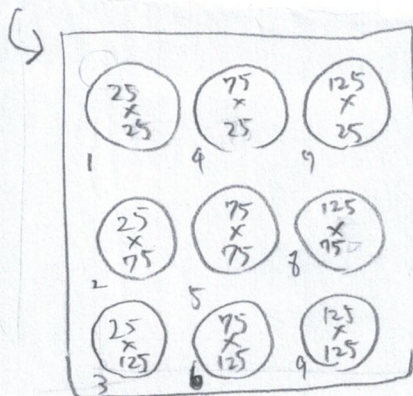
2行目: 円の色指定



\Rightarrow `let c;`
 ① `for (let x=25; x<400; x+50) {`
 ② `for (let y=25; y<400; y+50) {`
`c = shape.Circle(x, y, 20);`
`c.fillColor = 'green';`
`}`
`}`

※ 処理概要

for文で円を50px間隔で表示



★ JSは名称が同じで引数の数が違うXY.ットを定義してる

(例) 引数3 : `Shape.Circle(200, 200, 20);`
 \Rightarrow 半径20pxでX座標200, Y座標200の位置に円を表示

引数2 : `shape.Circle(event.point, 20);`

\Rightarrow クリックした座標位置に半径20pxの円を表示

event.point

① 変数

初期値を指定しない場合

let targetTempC; \Rightarrow let targetTempC = undefined;

↓
未定義「undefined」が
自動で設定された

② 定数

JSの具体的な数値や文字列を表す定数の慣習

const ROOM_TEMP_C = 26.5;

① 大文字 ② アンダーバー

③ 識別子

規則

① 1文字目

- Unicode文字
- \$
- _ (アンダーバー)

2文字目以降

- Unicode文字
- 数字
- \$
- _ (アンダーバー)

② 予約語に使えない

④ リテラル

リテラル = コードの中で直接値を指定するもの

(例) let room = 'へか1';

↓ ↓
識別子 リテラル

② プリミティブ型 と オブジェクト型

プリミティブ型

- ・ 数値 (Number)
- ・ 文字列 (String)
- ・ 論理値 (Boolean)
- ・ null
- ・ undefined
- ・ シンボル (Symbol)

オブジェクト型

- ・ 配列 (Array)
- ・ 日時 (Date)
- ・ 正規表現 (RegExp)
- ・ Map と WeakMap
- ・ Set と WeakSet

テンプレートリテラル

```
let message = "現在の温度は" + currentTemp + "℃です。";
```

↓

テンプレートリテラルで表現

```
let message = `現在の温度は ${currentTemp}℃です`;
```

複数行にわたる文字列

別な書き方

```
const multiline = "1行目\n2行目";
```

```
console.log(multiline);
```

// 実行結果

1行目
2行目

①

```
const multiline = "1行目\n2行目";
```

```
console.log(multiline);
```

// 実行結果

1行目

2行目

②

```
const multiline = `1行目\n2行目`;
```

```
console.log(multiline);
```

// 実行結果

1行目

2行目