

みんなのPython勉強会 #26

2017.7.12

# JavaScriptつてなに？

辻 真吾

@tsjshg

---

# 自己紹介

---

- ❖ 1975年8月生まれ
- ❖ 都内のとある大学で研究やってることになってます
  - ❖ 3月で辞めるとか言っておいて、辞めてない・・・
  - ❖ 研究室のテーマは癌とゲノム、最近は創薬にも興味があります
  - ❖ 実際にやっていることPythonでデータ解析
- ❖ データ解析の仕事が多いですが、時々Webアプリ作ったりするので、たまにJavaScriptも書きます（が、ふつうに初心者です）
- ❖ <http://www.tsjshg.info/>

# JavaScriptの歴史



# ここに書いてあります



Page **Discussion**

## A Short History of JavaScript

JavaScript, not to be confused with [Java](#), was created in 10 days in May 1995 by [Brendan Eich](#), then working at [Netscape](#) and now of [Mocha](#), a name chosen by [Marc Andreessen](#), founder of Netscape. In September of 1995 the name was changed to LiveScript, then in December 1995 JavaScript was adopted. This was somewhat of a marketing move at the time, with Java being very popular around then.

In 1996 - 1997 JavaScript was taken to [ECMA](#) to carve out a standard specification, which other browser vendors could then implement based on. This eventually led to the official release of ECMA-262 Ed.1: ECMAScript is the name of the official standard, with JavaScript being the most well known implementation of ECMAScript, with extensions (see below).

The standards process continued in cycles, with releases of ECMAScript 2 in 1998 and ECMAScript 3 in 1999, which is the baseline for modern JavaScript (then of Netscape, now at Google) started in 2000 and at first, Microsoft seemed to participate and even implemented some of the proposals.

Over time it was clear though that Microsoft had no intention of cooperating or implementing proper JS in IE, even though they had no competition on the .NET server side. So by 2003 the JS2/original-ES4 work was mothballed.

The next major event was in 2005, with two major happenings in JavaScript's history. First, Brendan Eich and Mozilla rejoined Ecma as a not-for-profit organization. Microsoft employees at BEA (originally acquired as Crossgain). This led to working jointly with Macromedia, who were implementing E4X in ActionScript.

So, along with Macromedia (later acquired by Adobe), work restarted on ECMAScript 4 with the goal of standardizing what was in AS3 and implementing it in a new engine named Tamarin, as an open source project. But Tamarin and AS3 were too different from web JavaScript to converge, as was realized by the community.

Alas, there was still turmoil between the various players; Doug Crockford — then at Yahoo! — joined forces with Microsoft in 2007 to oppose the ES4 effort.

While all of this was happening the open source and developer communities set to work to revolutionize what could be done with JavaScript. In 2005, in a white paper in which he coined the term Ajax, and described a set of technologies, of which JavaScript was the backbone, used to create web applications for full page reloads and resulting in more dynamic applications. This resulted in a renaissance period of JavaScript usage spearheaded by the community.

- [Main page](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)
- [Help](#)
- ▼ [Tools](#)
  - [What links here](#)
  - [Related changes](#)
  - [Special pages](#)
  - [Printable version](#)
  - [Permanent link](#)
  - [Page information](#)
  - [Browse properties](#)

# 復習：インターネットとWeb

1960

1970

1980

1990

2000

2010

Web (HTTP)

電子メール (SMTP)

ARPAnet

インターネット



# 誕生

- ❖ 1995年、Brendan Eich (Netscape→Mozilla) によって開発される
- ❖ 当初MochaやLiveScriptという名前だったが、当時勢いのあるJavaにあやかり、JavaScriptに改名
- ❖ 1996年の段階から、ECMA (European Computer Manufacturers Association) で標準化
  - ❖ ECMAScript



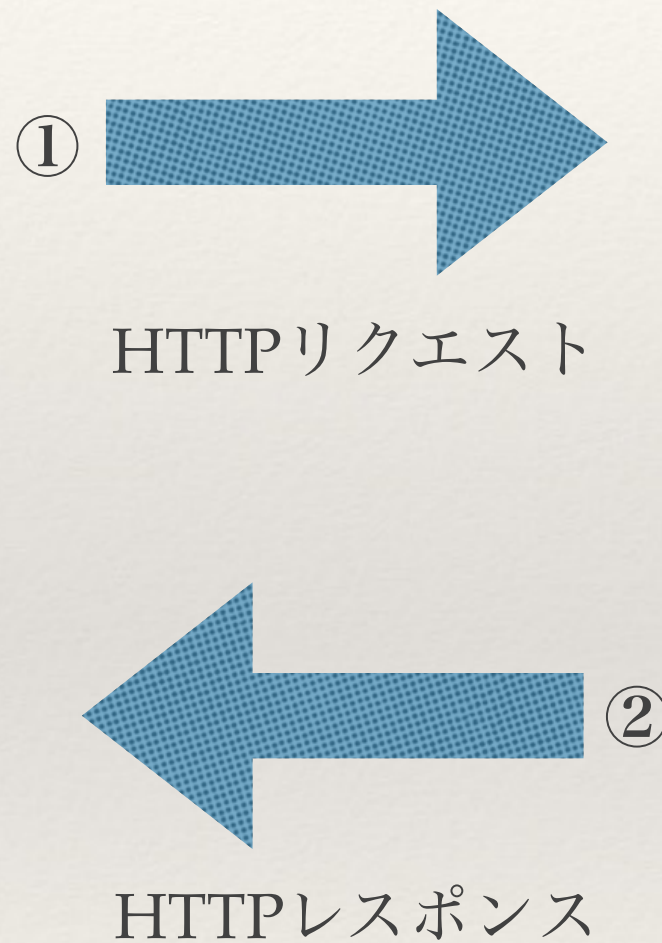
なぜ、ブラウザの中で動く  
言語が必要だったか？



# Webは超シンプル



Webブラウザ  
(クライアント)



HTTPサーバ  
(サーバ)



---

# 泥沼の00年代

---

- ❖ 動いた方が楽しいということで、Webとともに普及
- ❖ しかし、MicrosoftのIEの身勝手、Macromedia Flashの台頭、セキュリティホールが相次ぐ報告など、マイナスイメージが先行
- ❖ 2005年、JavaScriptのその後の運命を変える出来事が・・・

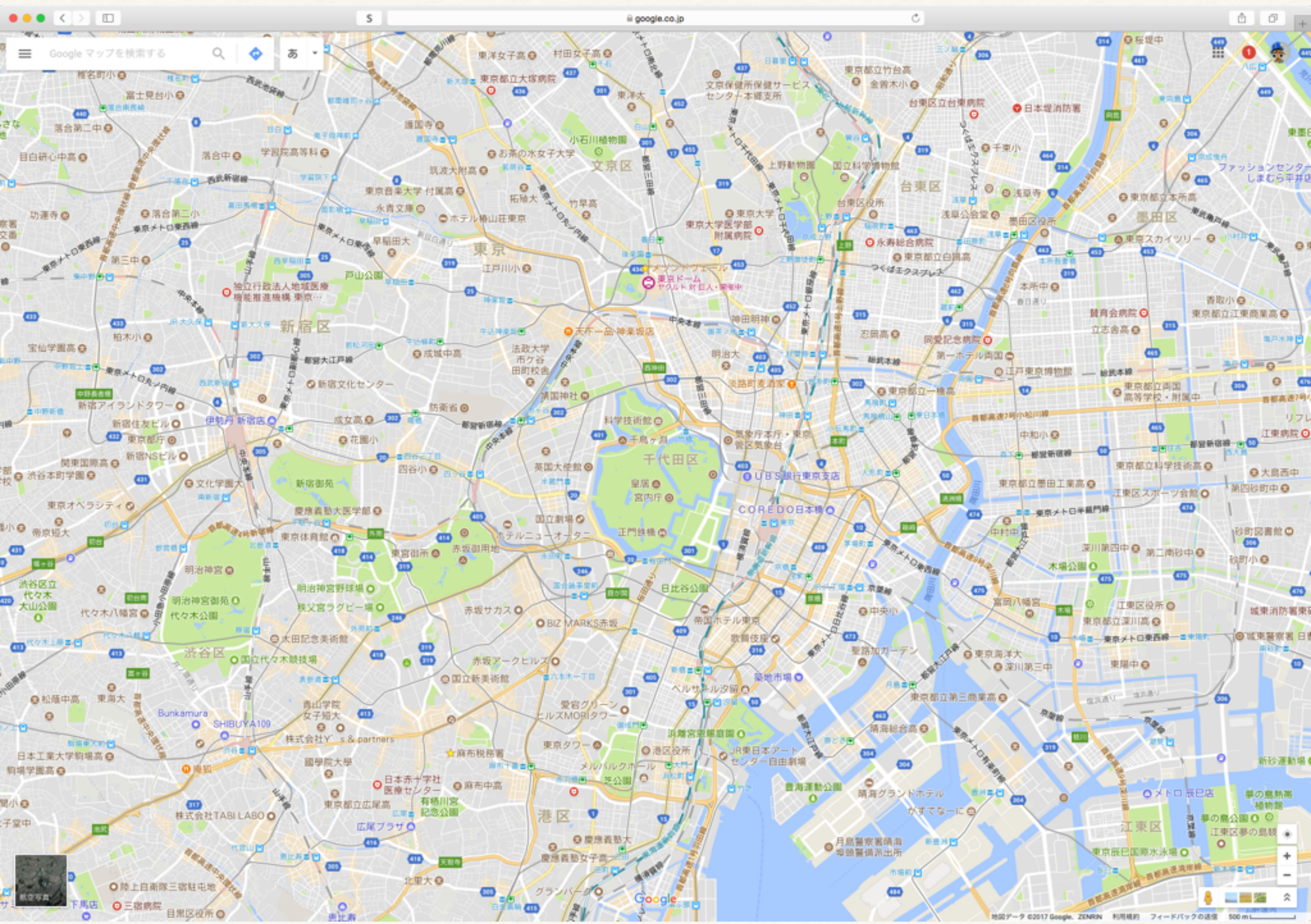
---

# 救世主 Ajax

---

- ❖ Asynchronous JavaScript + XML
- ❖ WebブラウザとHTTPサーバが非同期 (asynchronous) に通信して、情報を取得し、Webページの内容をダイナミックに変更







---

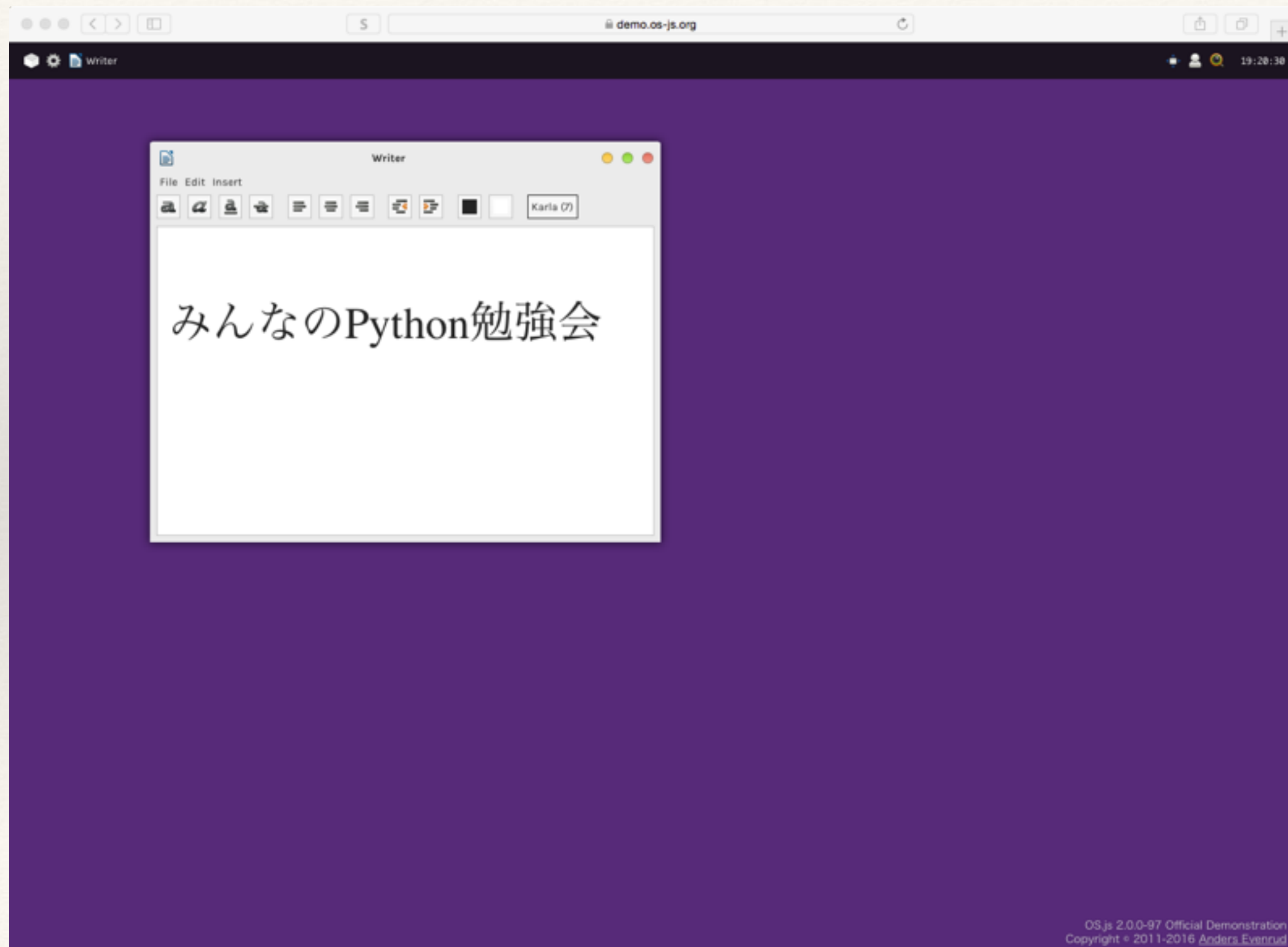
# JavaScriptの今

---

- ❖ Pythonは豊富な標準ライブラリを備えているが、JavaScriptは外部ライブラリと一緒に使う前提（な気がする）
  - ❖ jQuery, AngularJS, Google Web Toolkitなど
- ❖ Webブラウザの枠を越え、サーバサイドでも利用
  - ❖ Node.js
- ❖ 数々の困難を乗り越え、Webには必要不可欠な技術に



# OS.js

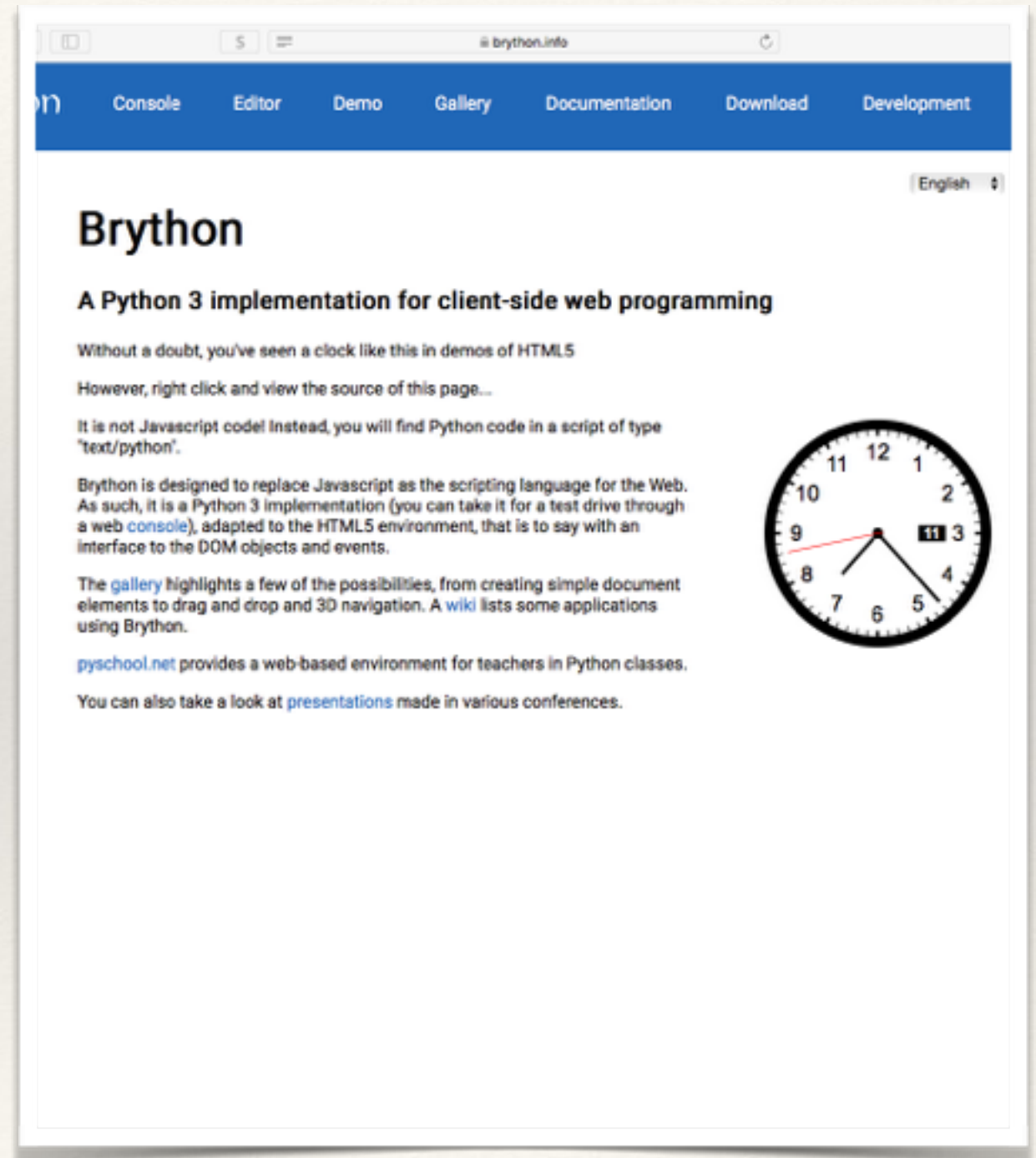


そうは言ってもPythonしか  
書きたくないあなたへ



# Brython

- ❖ トップページ動く時計はPythonで書かれている
- ❖ ページのソースを見るとびっくり
- ❖ WebブラウザでPythonを動かすために、PythonのコードをJSのコードに変換



# BeeWare

- ❖ Python製アプリをiOSやAndroidを含む多くのプラットフォームで動かすための環境
- ❖ Pythonのバイトコードを実行する仮想マシンをJSで実装





---

# JSON

---

- ❖ JavaScript Object Notation
- ❖ 軽量でシンプルなデータ交換用フォーマット
- ❖ AjaxのXはXMLですが、XMLはごちゃごちゃしているので、JSONで実装することもある
- ❖ Pythonには標準モジュールに、jsonがあります

# JSONモジュールの使い方

```
import json
```

```
# Pythonの辞書型をJSONに変換
```

```
d = {}  
for i in range(4):  
    d[i] = i
```

```
# キーは文字列、値は整数になっている
```

```
json.dumps(d)
```

```
'{"0": 0, "1": 1, "2": 2, "3": 3}'
```

```
# Fileに書き込みたければ
```

```
with open('data.json', 'w') as f:  
    json.dump(d, f)
```

```
# 読み込みはload
```

```
with open('data.json', 'r') as f:  
    data = json.load(f)  
data
```

```
{'0': 0, '1': 1, '2': 2, '3': 3}
```

---

# まとめ

---

- ❖ JavaScriptの歴史を紹介
- ❖ Ajaxの誕生から10年以上、JavaScriptはWebの中心技術の1つに
- ❖ JSの環境でPython動かそうとする人達もいます
- ❖ JSONはお手軽なので、簡単なWebアプリなどに是非



お後がよろしいようで。