ENGSE611 **การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่** **3 (1-4-4)** ****วิชาชีพเลือก**

Modern Web Technology Development

Lecture05

พื้นฐาน JavaScript

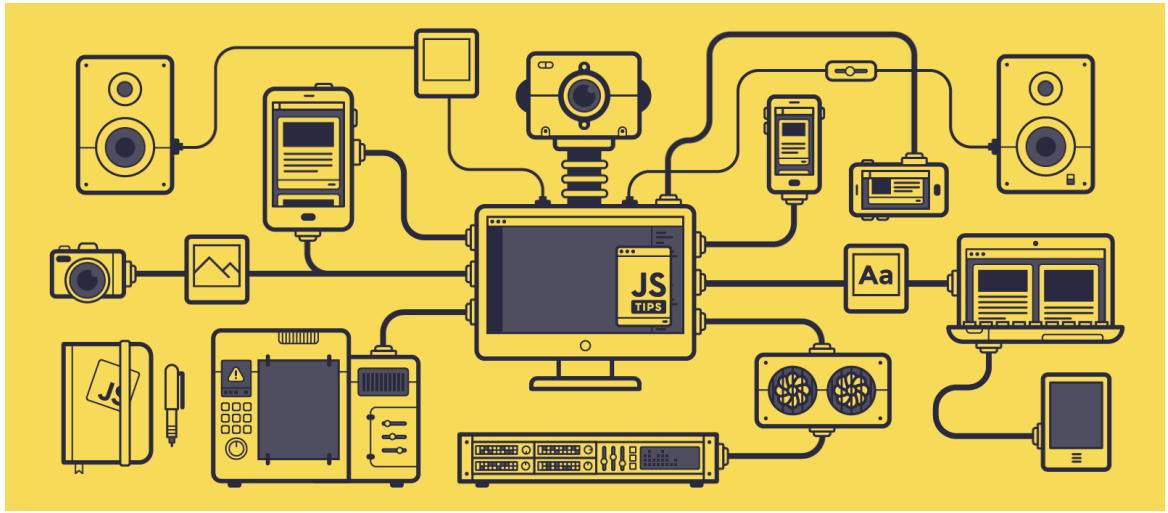
สอนโดย อ.ธนิต เกตุแก้ว

หลักสูตรวิศวกรรมซอฟต์แวร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา เชียงใหม่

วัตถุประสงค์การเรียนรู้

- เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับภาษา JavaScript
- เพื่อให้ผู้เรียนประยุกต์ใช้งาน JavaScript กับภาษามาร์กอัป HTML เพื่อพัฒนาเว็บไซต์เบื้องต้นได้



credit: http://wikiwebpedia.com/javascript-introduction

หัวข้อการเรียนรู้

การโปรแกรมฝั่งลูกข่ายด้วย JavaScript

หัวข้อเรียนรู้ :

1. พื้นฐาน JavaScript
2. การจัดการ DOM (Document Object Model)
3. การจัดการ Event ในหน้าเว็บ

กิจกรรม : การเขียน JavaScript เพื่อสร้างปฏิสัมพันธ์บนหน้าเว็บ

งานที่ต้องทำ : สร้างหน้าเว็บที่ตอบสนองต่อการคลิกและการพิมพ์ข้อมูล

https://www.w3schools.com/js/default.asp

วัตถุประสงค์ของ web technologies แต่ละชนิด

- **HTML** ใช้เพื่อสร้างโครงของเอกสาร(document) และข้อมูล (content)

- **CSS** ใช้เพื่อควบคุมและกำหนดการมองเห็น (visual aspect)

- **Javascript** ใช้เพื่อการโต้ตอบกับผู้ใช้(interactivity)

https://docs.google.com/presentation/d/1j6CqGnWdBiWGyxFVLZBjlADFpv0aUj65zHpsri5bMHs/edit#slide=id.p

https://www.aware.co.th/it-jobs/th/what-you-need-to-know-before-learning/

สถาปัตยกรรมของซอฟต์แวร์เบื้องต้น

ตีแผ่ แบไต๋ Front-end / Back-end สั้น ง่าย ได้ใจความ

ที่มา: https://swiftlet.co.th/

WEB APPLICATION ARCHITECTURE

USERS

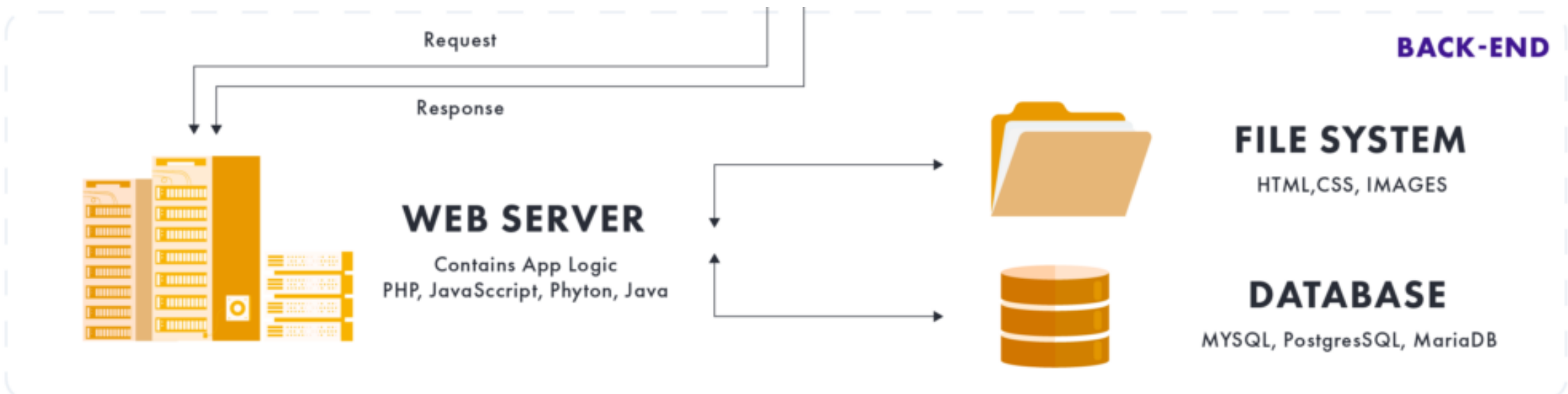
Collect Data

Display Results

FRONTEND

What the User Sees
& Interacts with
HTML, CSS, JavaScript

WEB APPLICATION ARCHITECTURE



WEB APPLICATION ARCHITECTURE

USERS

Collect Data

Display Results

FRONTEND

What the User Sees
& Interacts with
HTML, CSS, JavaScript

Request

Response

BACK-END

WEB SERVER

Contains App Logic
PHP, JavaSccript, Phyton, Java

FILE SYSTEM

HTML,CSS, IMAGES

DATABASE

MYSQL, PostgresSQL, MariaDB

FRONT-END

Web Browser / Desktop App

Mobile App

- ✓ คือส่วนติดต่อผู้ใช้ (User interface)
- ✓ การทำงานที่ผู้ใช้สามารถเข้าถึงได้บนหน้าจอ การคลิก การกดปุ่มต่างๆ
- ✓ เน้นไปทางด้านความสวยงาม ฟังก์ชันที่ทันสมัย ทุกอย่างที่เราเห็นบนหน้าจอ
- ✓ การออกแบบนั้นต้องรองรับได้กับทุกอุปกรณ์
- ✓ ทักษะพื้นฐานที่ต้องมีก็คือ ต้องเขียนภาษา HTML / CSS / JavaScript

Web Framework ที่นิยมใช้

- HTML / CSS / JavaScript - jQuery
- React js
- Vue js

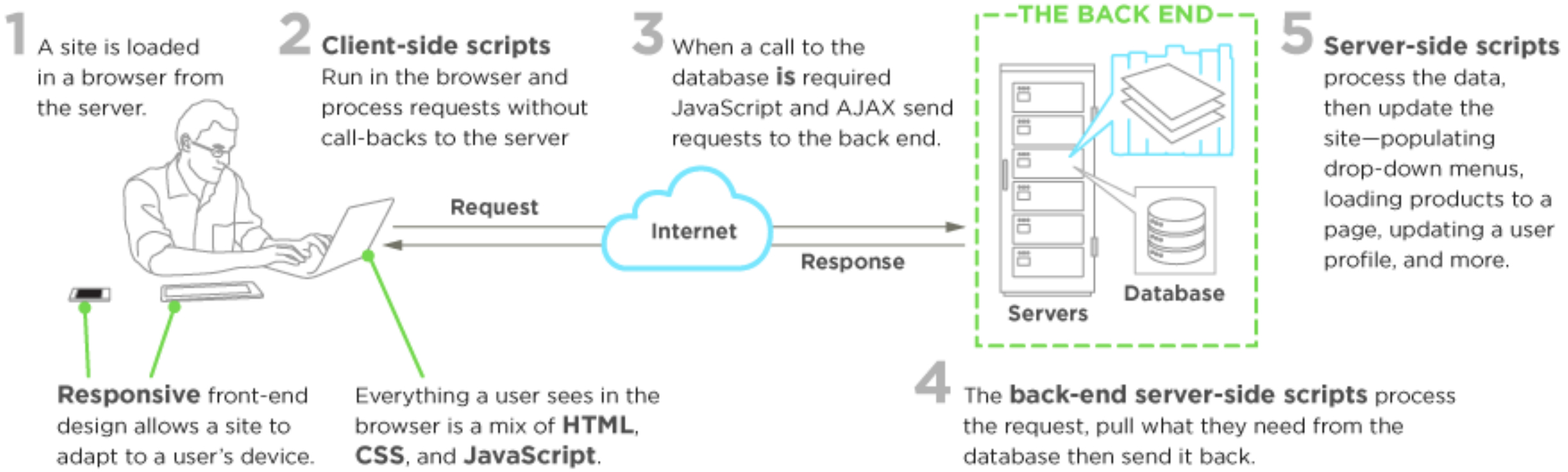
Mobile Framework ที่นิยมใช้

- iOS – Swift
- Android – Java/Kotlin
- Flutter (Hybrid)
- React Native (Hybrid)

Desktop Framework ที่นิยมใช้

- Java / C# / C++
- Electron js (Hybrid desktop apps)

ภาพรวมการทำงานของ Front-end Developer



A front-end **developer architects** and **develops websites** and **web applications** using web technologies

เช่น HTML, CSS, and JavaScript

Baseline Web Technologies Employed by Front-End Developers

Front-End Developers Develop For...

ทำงานบน OS

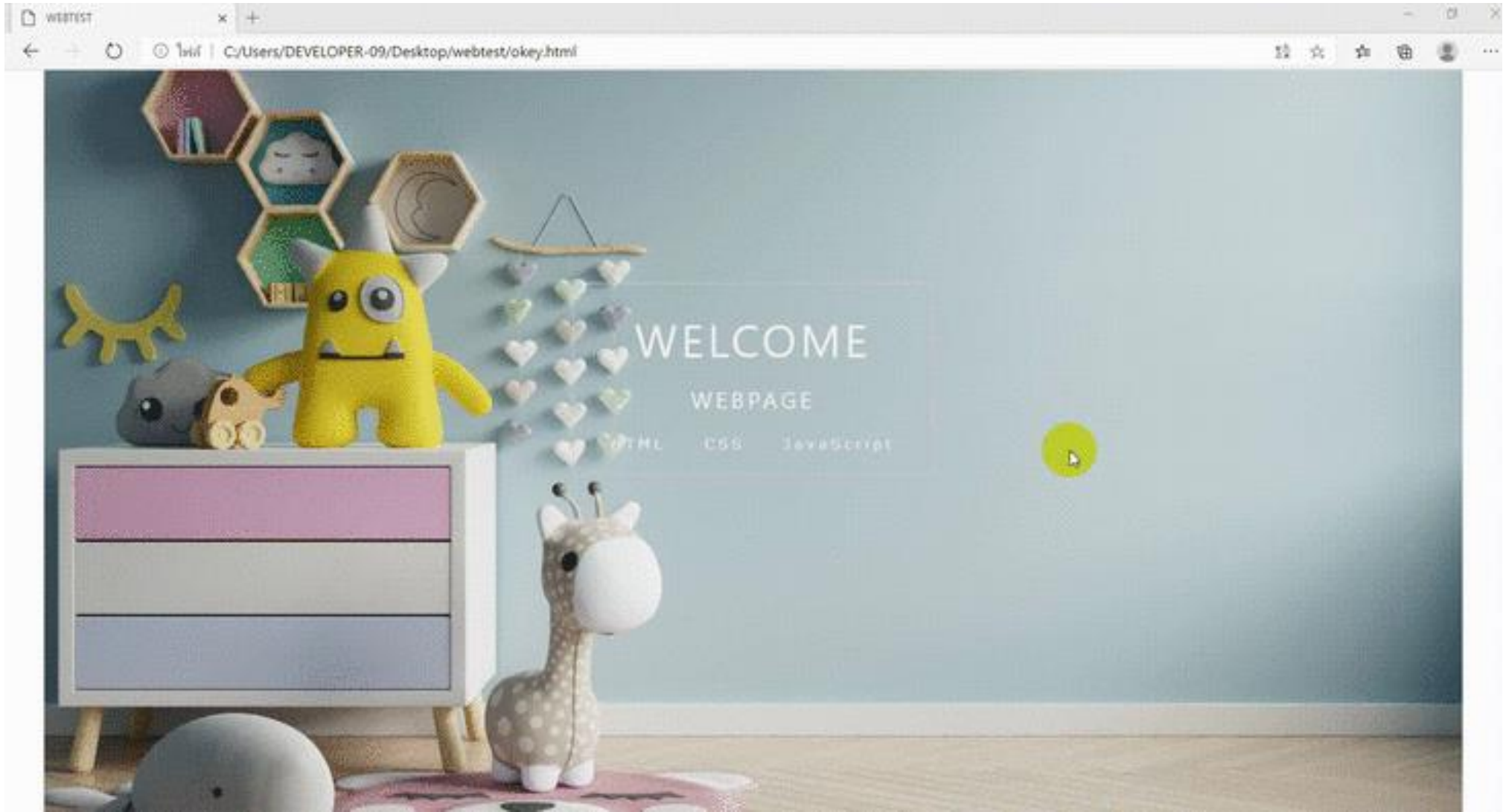
- Android
- Chromium
- iOS
- OS X (i.e. MacOS)
- Ubuntu (or some flavor of Linux)
- Windows

ที่ทำงานบน Devices

- Desktop computer
- Laptop / netbook computer
- Mobile phone
- Tablet
- TV
- Watch
- Things (i.e., anything you can imagine, car, refrigerator, lights, thermostat, etc.)

ตัวอย่างเว็บเพจที่สร้างจากภาษา HTML, CSS และ JavaScript

- สำหรับในส่วนของตัวอักษรบนหน้าเว็บเพจ รวมถึง ปุ่ม SEND กำหนดด้วย**ภาษา HTML**
- ส่วนพื้นหลัง และสีของกรอบข้อความต่างๆ **ใช้ CSS**
- สำหรับช่องกรอกข้อมูลต่างๆ จะ**ใช้ JavaScript** เข้ามาช่วย ถ้าหากกรอกข้อมูลไม่ครบก็จะมีข้อความขึ้นเตือน



ข้อดี และ ข้อเสีย ของ ภาษา HTML, CSS และ JavaScript

HTML	CSS	JavaScript
สามารถนำไฟล์เอกสารอื่นๆ มาใช้ได้ เช่น ไฟล์ .css .js	สามารถกำหนดการตกแต่งพร้อมกันทีเดียวได้	สามารถทำให้หน้าเว็บเพจเปลี่ยนแปลง และตรวจสอบข้อมูลที่ผู้ใช้งานกรอกได้
การแก้ไขรูปแบบการตกแต่งค่อนข้างยากต้องแก้ทีละจุด	มีรูปแบบการนำมาใช้งานที่หลากหลาย	การใช้คำสั่งยากกว่า HTML และ CSS ซึ่งอาจจะค่อนข้างยากสำหรับมือใหม่
เป็นภาษาที่ง่ายสำหรับนักพัฒนาเว็บมือใหม่	ฟอนต์ของ CSS ใช้งานไม่ได้กับทุกเครื่อง	เว็บเบราว์เซอร์ส่วนใหญ่จะต้องมีการใช้ JavaSript เพื่อให้เว็บมีความสมบูรณ์มากขึ้น

BACK-END

Database Server

Web Server

File Server

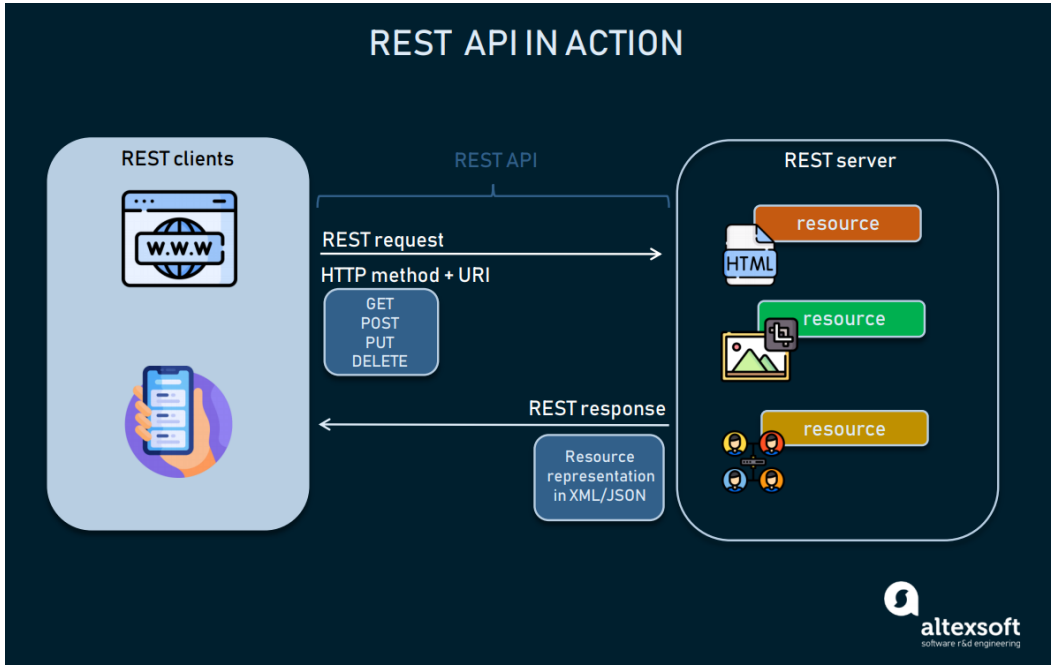
- ✓ เบื้องหลังที่คอยเชื่อมโยงกับทาง Front-end ทำให้ทำงานได้อย่างมีประสิทธิภาพ
- ✓ เป็นรากฐานของระบบทั้งหมดเป็นส่วนด้านในผู้ใช้ทั่วไป
- ✓ ไม่สามารถเข้าถึงหน้านี้ได้โดยจะมีผู้ดูแลระบบคอยจัดการ
- ✓ จะเกี่ยวข้องกับโครงสร้างเว็บไซต์ การเขียนโค้ดควบคุม XML, text file, JAVA, PHP, C#, C++

Language / Framework ที่นิยมใช้

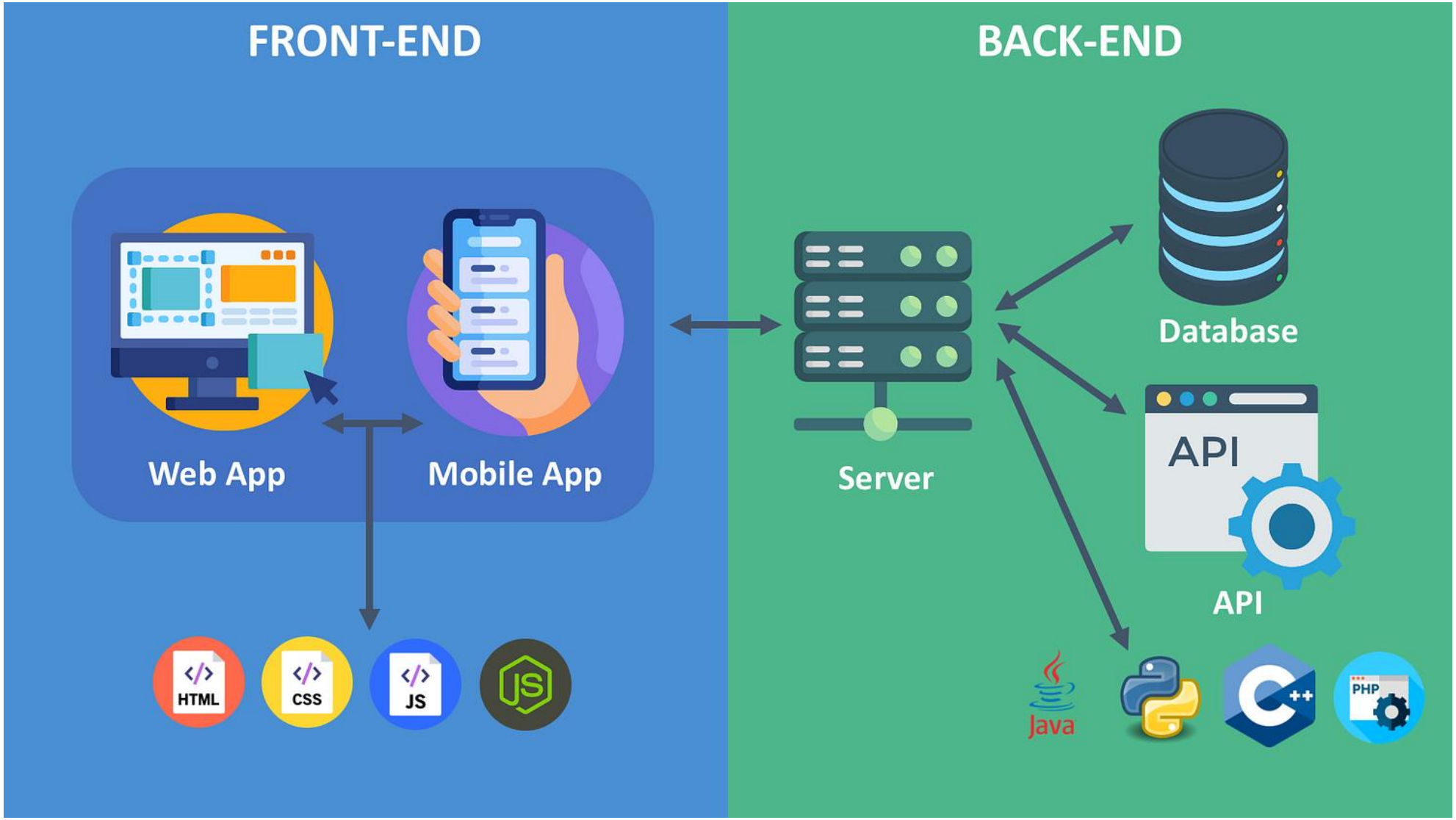
- PHP
- JAVA
- java script - Nodejs

Back-end Application ที่นิยมใช้

RESTful API



สรุปการทำงานของ Front-end / Back-end



The best language to consider in

the different the Application Areas

ภาษาคอมพิวเตอร์ที่นิยมใช้สูงที่สุดในปัจจุบัน

ปี 2025

The best language to consider in the different phases of the development process

Application Area	Programming Languages
Front-end Web Development	JavaScript, Elm, TypeScript
Back-end Web Development	JavaScript, Scala, Python, Go, Ruby
Mobile Application	Swift, Java, Objective C, JavaScript
Game Development	Unity, TypeScript
Desktop Application	Scala, Go, Python
System Programming	Go, Rust

See more at: Most popular technologies

Source: https://distantjob.com/blog/programming-languages-rank/

Top Programming Languages: Rankings In Comparison

Index	1st	2nd	3rd	4th	5th
IEEE Spectrum	Python	Java	C	C++	JavaScript
GitHut 2.0 (Pull Requests)	JavaScript	Python	Java	Go	Ruby
GitHut 2.0 (Pushes)	JavaScript	Python	Java	C++	PHP
RedMonk	JavaScript	Python	Java	PHP	C++/C#
PYPL	Python	Java	Javascript	C#	C/C++
TIOBE	C	Python	Java	C++	C#

Source: https://distantjob.com/blog/programming-languages-rank/

JavaScript คืออะไร

JavaScript คืออะไร

คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่ได้รับความนิยมอย่างสูง

JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช่ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น

JavaScript เป็นภาษาสคริปต์ที่มีลักษณะดังนี้

1. เป็นภาษาสคริปต์แบบ lightweight programming language (ภาษาสคริปต์แบบสั้นๆ)
2. JavaScript สามารถใช้ร่วมกับ HTML และ CSS เพื่อสร้างและพัฒนาเว็บไซต์ ที่**ใช้ได้** กับ web browser และทำงานโดย **ไม่ใช้** web browser ก็ได้

   อีกทั้งยังใช้ได้ทั้ง PCs, laptops, tablets, smart phones อีกด้วย

JavaScript ใช้ทำอะไร?

เนื่องจาก JavaScript ถูกใช้ในการพัฒนาเว็บ และสามารถใช้สำหรับการพัฒนาทั้งส่วนหน้า (front-end) และส่วนหลัง (Back-end)

เราจึงจำเป็นต้องรู้จัก framework ของ JavaScript เพื่อศึกษาและพัฒนาต่อยอดด้วย ซึ่ง framework ยอดนิยมแยกประเภทได้ดังนี้

- **Front End:** React, AngularJS, Vue.js
- **Back End:** Node.js, Next.js, Express.js, Nuxt
- **Testing:** Jest, Mocha, Storybook, Cypress, Jasmine, Playwright

JavaScript ใช้ทำอะไร?

พัฒนาเว็บไซต์

JavaScript เป็นภาษาที่จำเป็นอย่างมากสำหรับการพัฒนาเว็บไซต์ทั้งในระบบหน้าบ้านหรือ Front End (ส่วนของผู้เข้าชมเว็บไซต์) และระบบหลังบ้านหรือ Back End (ส่วนเซิร์ฟเวอร์ของผู้ให้บริการ)

ระบบหน้าบ้านหรือ Front End

พัฒนาระบบหน้าบ้านจะมี HTML ที่มีหน้าที่วางโครงสร้างข้อมูลบนเว็บ CSS ซึ่งมีหน้าที่ตกแต่งข้อมูลจาก HTML และ JavaScript ที่ทำให้เว็บไซต์สามารถแสดงผลตอบสนองต่อการกระทำของผู้ใช้งานการใช้งาน ได้ทันที

ระบบหลังบ้านหรือ Back End

สามารถนำไปช่วยสร้างเซิร์ฟเวอร์ได้ผ่านการใช้ Node.js ที่เหมาะกับการสร้างเว็บไซต์ที่ต้องการการตอบสนองทันทีที่มีการกระทำ และยังมีโค้ดจำนวนมากที่มีผู้ใช้งานคนอื่นเคยพัฒนาไว้ก่อนแล้ว นอกจากนี้การที่ระบบหน้าบ้านกับระบบหลังบ้านถูกพัฒนาโดยใช้ภาษาเดียวกัน ทำให้การพัฒนาได้รวดเร็ว

JavaScript ใช้ทำอะไร?

พัฒนาแอปพลิเคชันบนมือถือ

JavaScript มี Framework ที่เปรียบเป็นโครงสร้างพื้นฐานที่ทุกบ้านต้องมี ตัวช่วยทำให้การพัฒนาซอฟต์แวร์ง่ายและรวดเร็วยิ่งขึ้นตัวอย่าง Framework ของ JavaScript คือ React Native โดยโค้ดเดียวกันสามารถนำไปพัฒนาได้ในหลายแพลตฟอร์ม ทำให้เราสามารถพัฒนาแอปพลิเคชันบนมือถือทั้ง Android และ iOS ไปพร้อมกัน เช่น Facebook, Airbnb, Instagram

พัฒนาเกม

เกมที่ใช้ JavaScript พัฒนาจะเป็นเกมสามารถเล่นบนเบราว์เซอร์หรือโทรศัพท์มือถือ ซึ่งเหมาะสำหรับคนที่อยากลองฝึกพัฒนาเกมง่าย ๆ

การเรียน JavaScript มีข้อดีอย่างไร?

- เรียนรู้ได้ง่าย เนื่องจากมีโครงสร้างภาษาที่ทำความเข้าใจได้ง่าย
- สร้างอินเทอร์เฟซที่น่าดึงดูด ช่วยเพิ่มลูกเล่นและสร้างความตื่นตาตื่นใจให้กับผู้ใช้
- มีความหลากหลายในการใช้งาน และสามารถทำงานร่วมกับภาษาโปรแกรมอื่น ๆ ได้ดี
- มีผู้ใช้งาน JavaScript จำนวนมาก ทำให้มีแหล่งข้อมูลจำนวนมากที่เผยแพร่โค้ดสำเร็จรูป ช่วยลดระยะเวลาการพัฒนา

JavaScript – The language of the Web

HISTORY AND VERSIONS

JAVASCRIPT VERSIONS

Brendan Eich

- **JAVASCRIPT (December 4th 1995)** Netscape and Sun press release
- **ECMAScript Standard Editions:** https://www.ecma-international.org/ecma-262/
  - **ES1 (June 1997)** Object-based, Scripting, Relaxed syntax, Prototypes
  - **ES2 (June 1998)** Editorial changes for ISO 16262
  - **ES3 (December 1999)** Regexps, Try/Catch, Do-While, String methods

10 yrs

  - **ES5 (December 2009)** Strict mode, JSON, .bind, Object mts, Array mts
  - **ES5.1 (June 2011)** Editorial changes for ISO 16262:2011

Main target

  - **ES6 (June 2015)** Classes, Modules, Arrow Fs, Generators, Const/Let, Destructuring, Template Literals, Promise, Proxy, Symbol, Reflect — Also: ES2015
  - **ES7 (June 2016)** Exponentiation operator (**) and Array Includes — Also: ES2016

ES9, ES10, ...

  - **ES8 (June 2017)** Async Fs, Shared Memory & Atomics — Also: ES2017

https://www.slideshare.net/slideshow/javascript-editions-es7-es8-and-es9-vs-v8/75998257

JavaScript versions

- **ECMAScript** (also called ES) is the official name of JavaScript (JS) standard
- ES6, ES2015, ES2016 etc. are implementations of the standard
- All browsers used to run ECMAScript 3
- ES5, and ES2015 (=ES6) were huge versions of JavaScript
- Then, yearly release cycles started

– By the committee behind JS: TC39, backed by Mozilla, Google, Facebook, Apple, Microsoft, Intel, PayPal, SalesForce, etc.

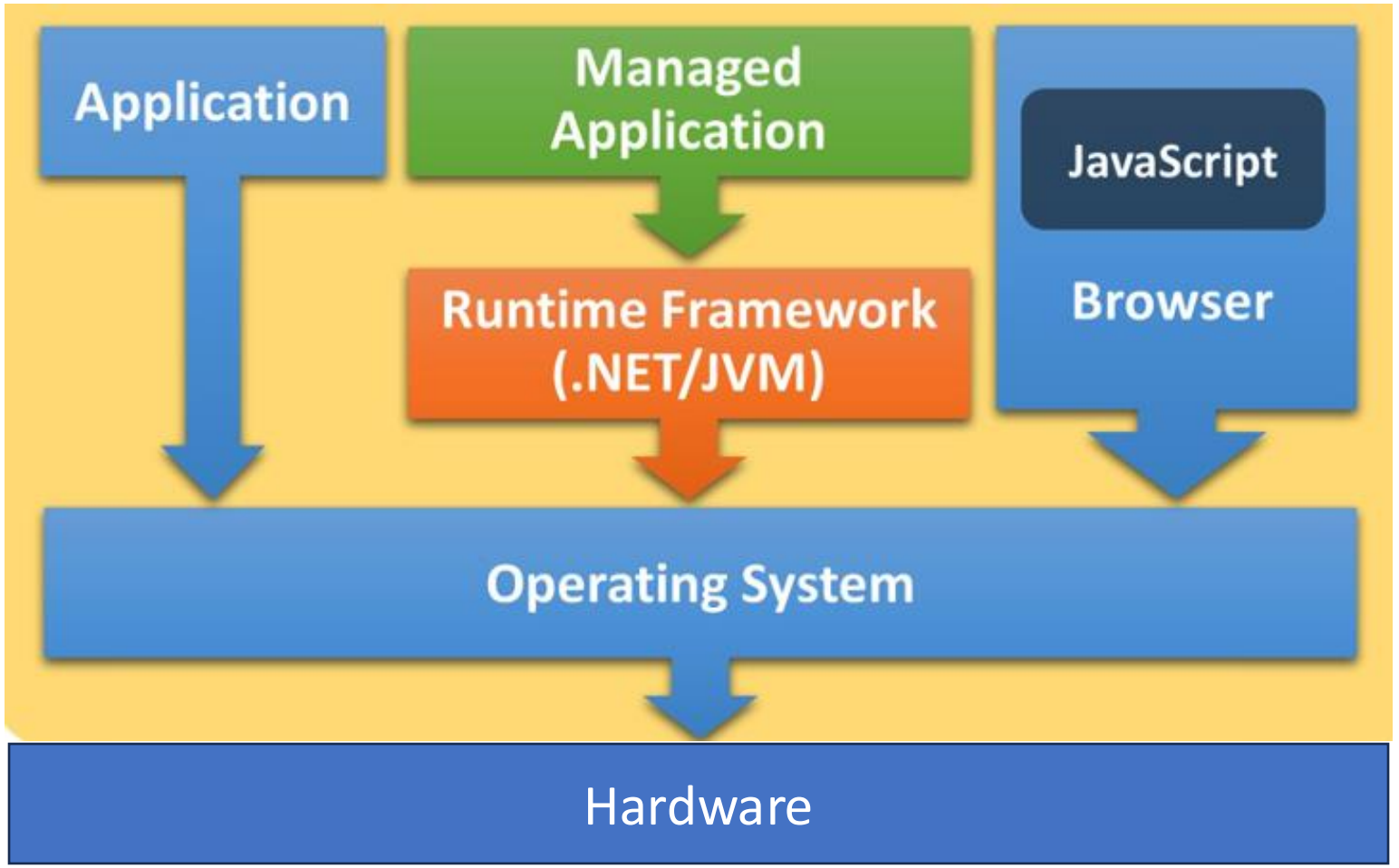
- **ES2015 (=ES6) is covered in this course**

JavaScript Engines

- V8 (Chrome V8) by Google
  - used in Chrome/Chromium, Node.js and Microsoft Edge
- SpiderMonkey by Mozilla Foundation
  - Used in Firefox/Gecko
- ChakraCore by Microsoft
  - it was used in Edge
- JavaScriptCore by Apple
  - used in Safari

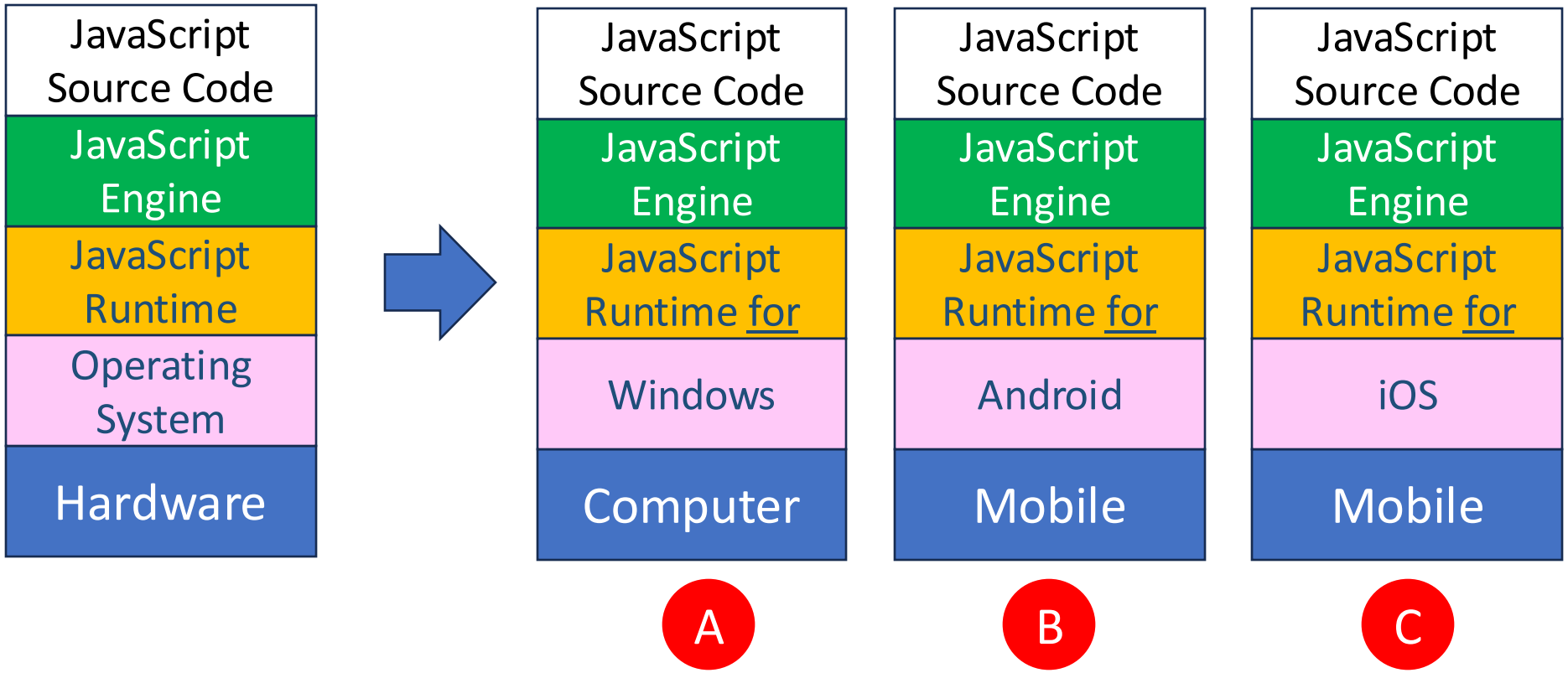
JavaScript Engine
vs
JavaScript Runtime

การทำงานทั่วไปของซอฟต์แวร์



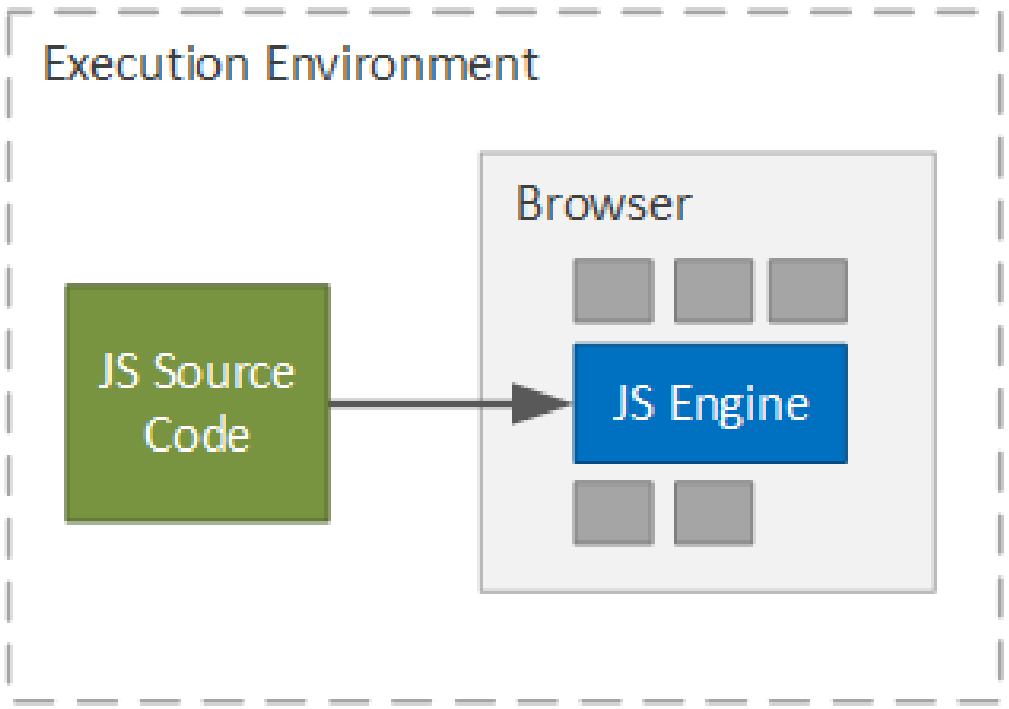
https://i0.wp.com/wentzwu.com/wp-content/uploads/2020/09/Software-Runtime-Environment.jpg?ssl=1

การทำงานระหว่าง JavaScript Engine และ JavaScript Runtime

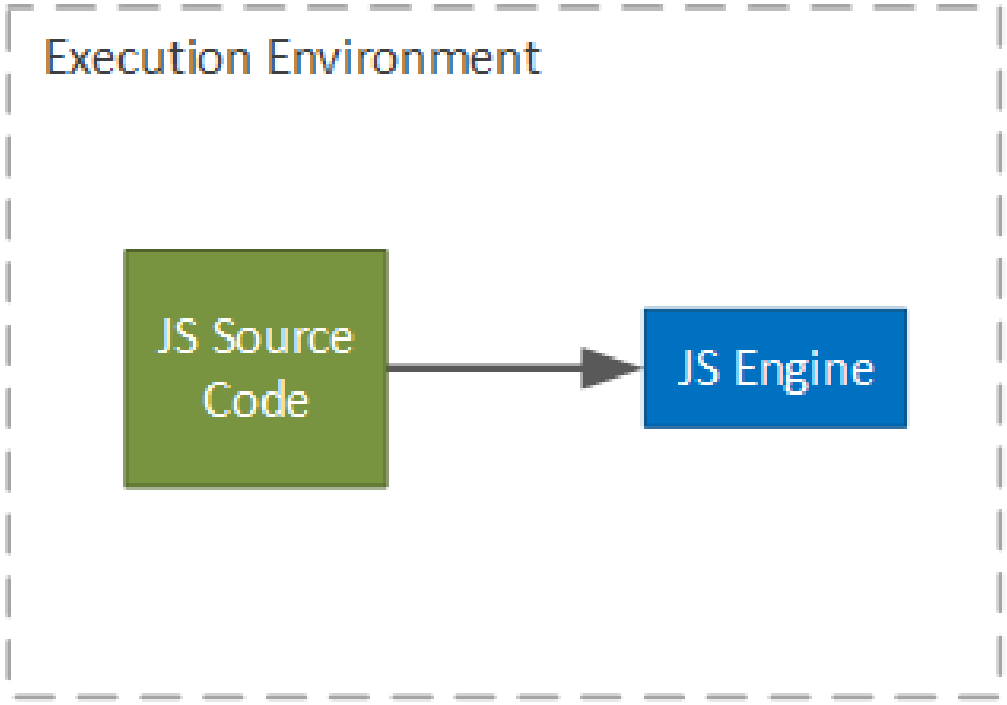


https://i0.wp.com/wentzwu.com/wp-content/uploads/2020/09/Software-Runtime-Environment.jpg?ssl=1

JavaScript Environments



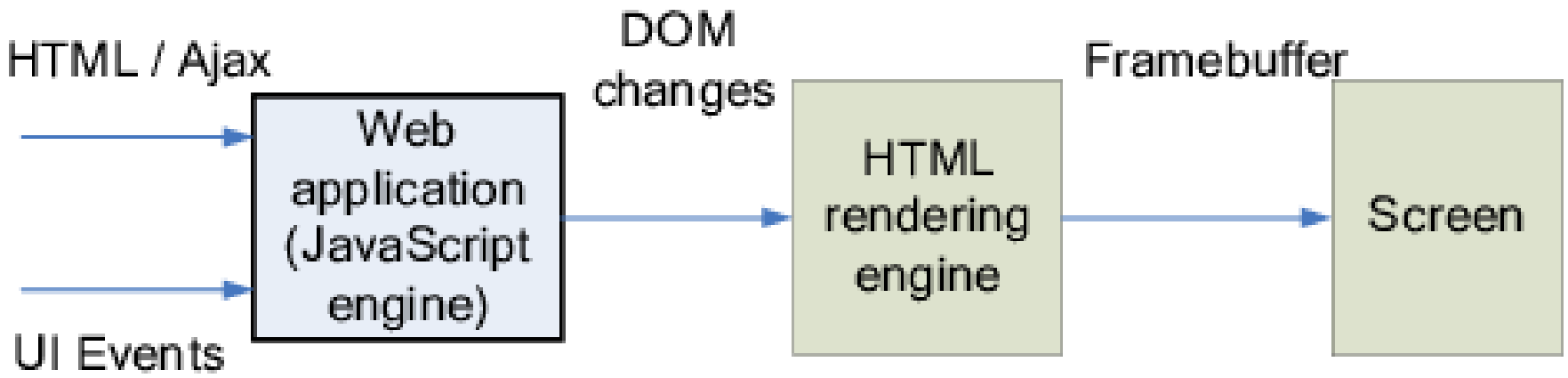
JAVASCRIPT IN A BROWSER



STANDALONE JAVASCRIPT

[Comparing Browser JS and Node.Js]

JavaScript in Browser

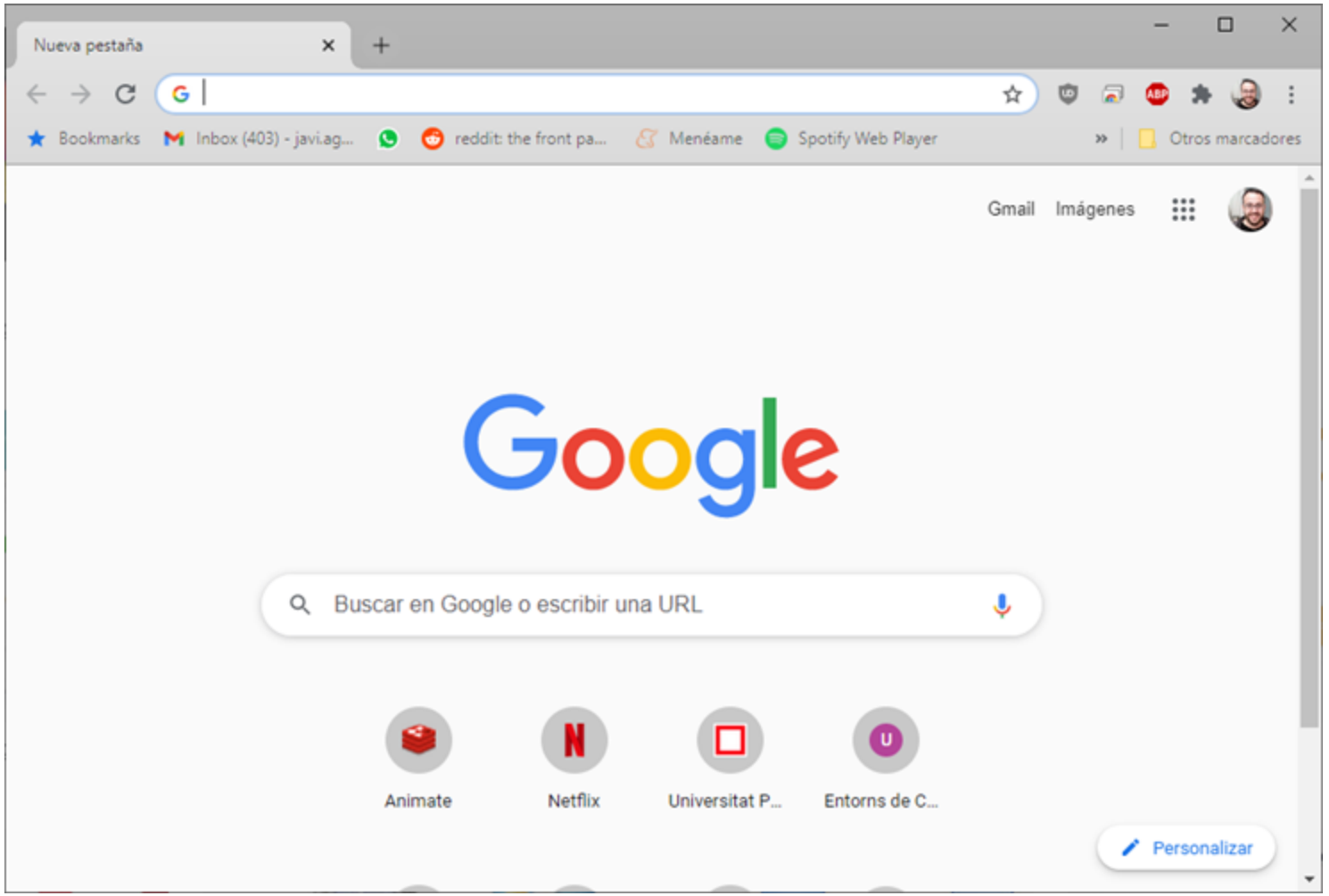


Browser model from the JavaScript engine perspective

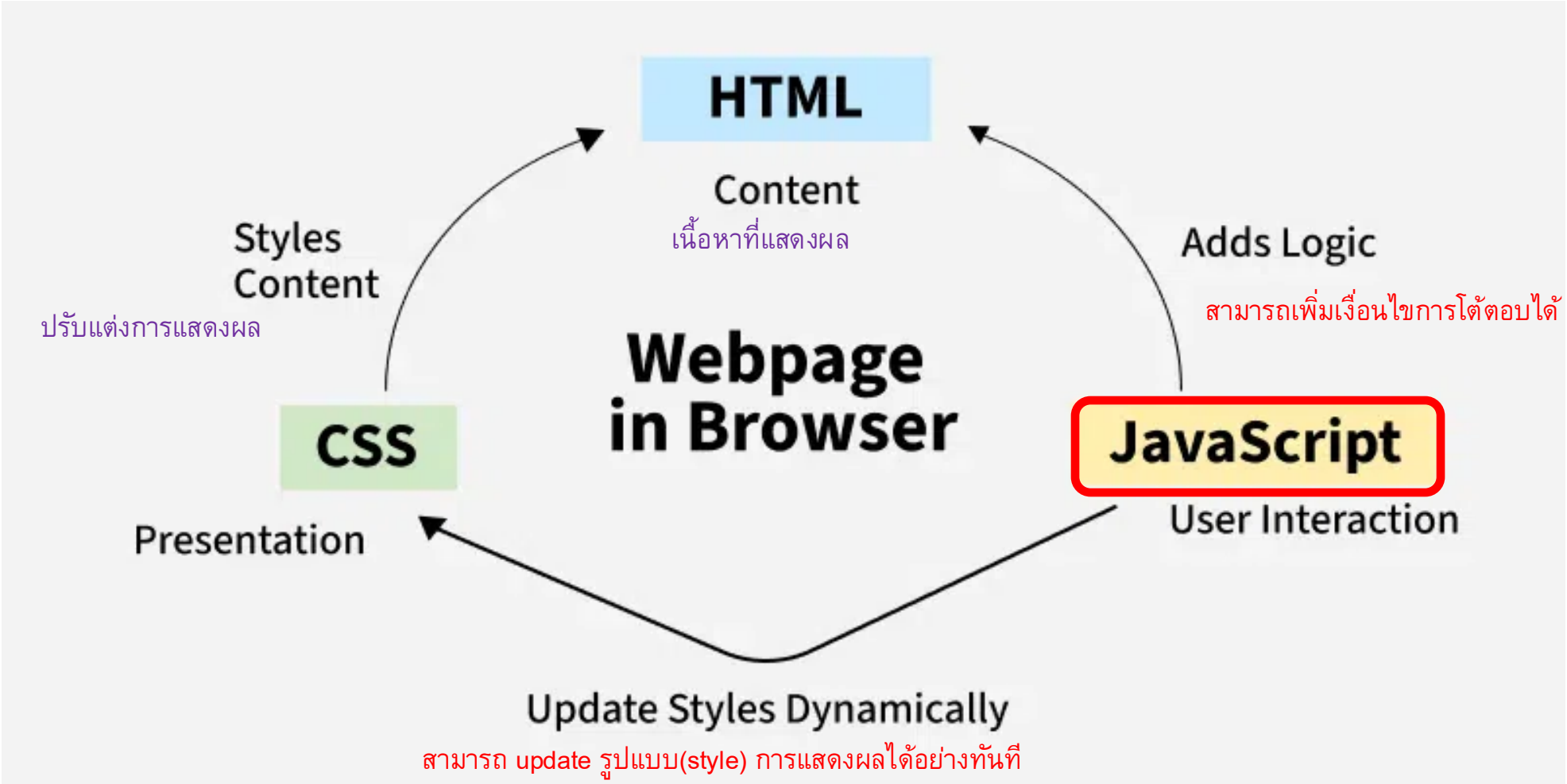
https://www.researchgate.net/publication/221022021_Co-browsing_dynamic_web_pages

Anatomy of a Browser

What is inside a Browser
about
html, CSS and JavaScript ?



Inside a browser



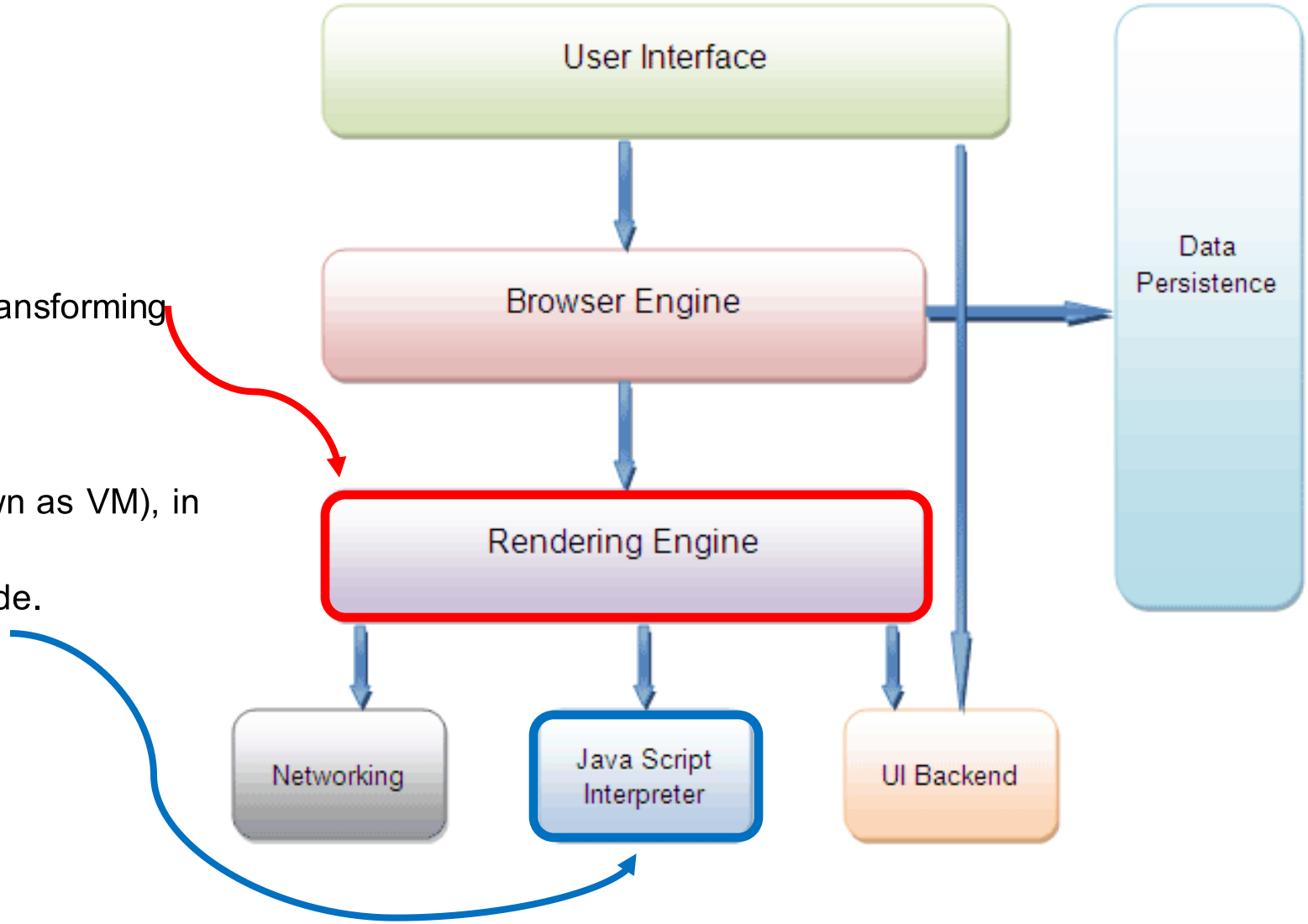
https://www.geeksforgeeks.org/javascript/

Inside a browser

Browsers have very differentiate parts.

We are interested in two of them:

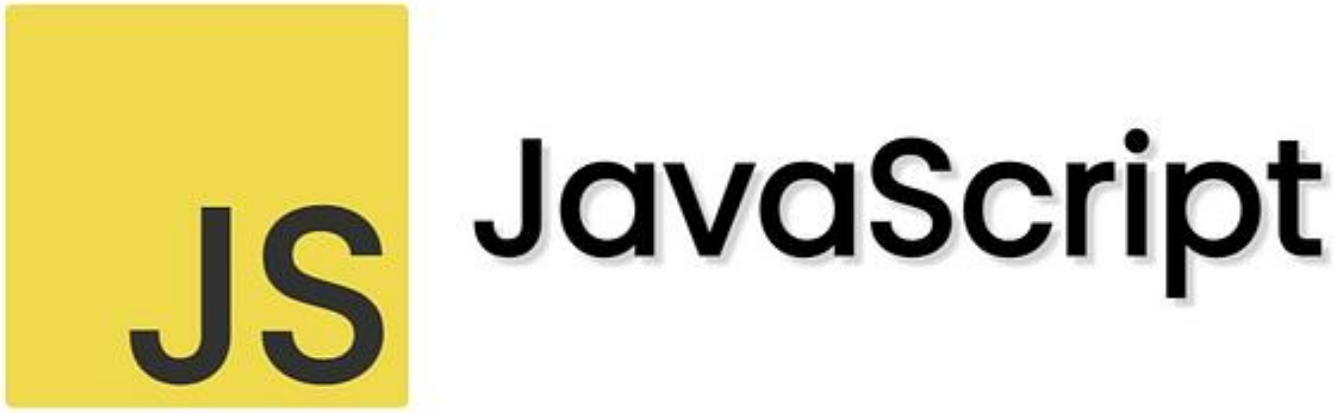
- the **Rendering Engine** (in charge of transforming our **HTML+CSS** in a visual image).
- The **JavaScript Interpreter** (also known as VM), in charge of executing the **Javascript** code.

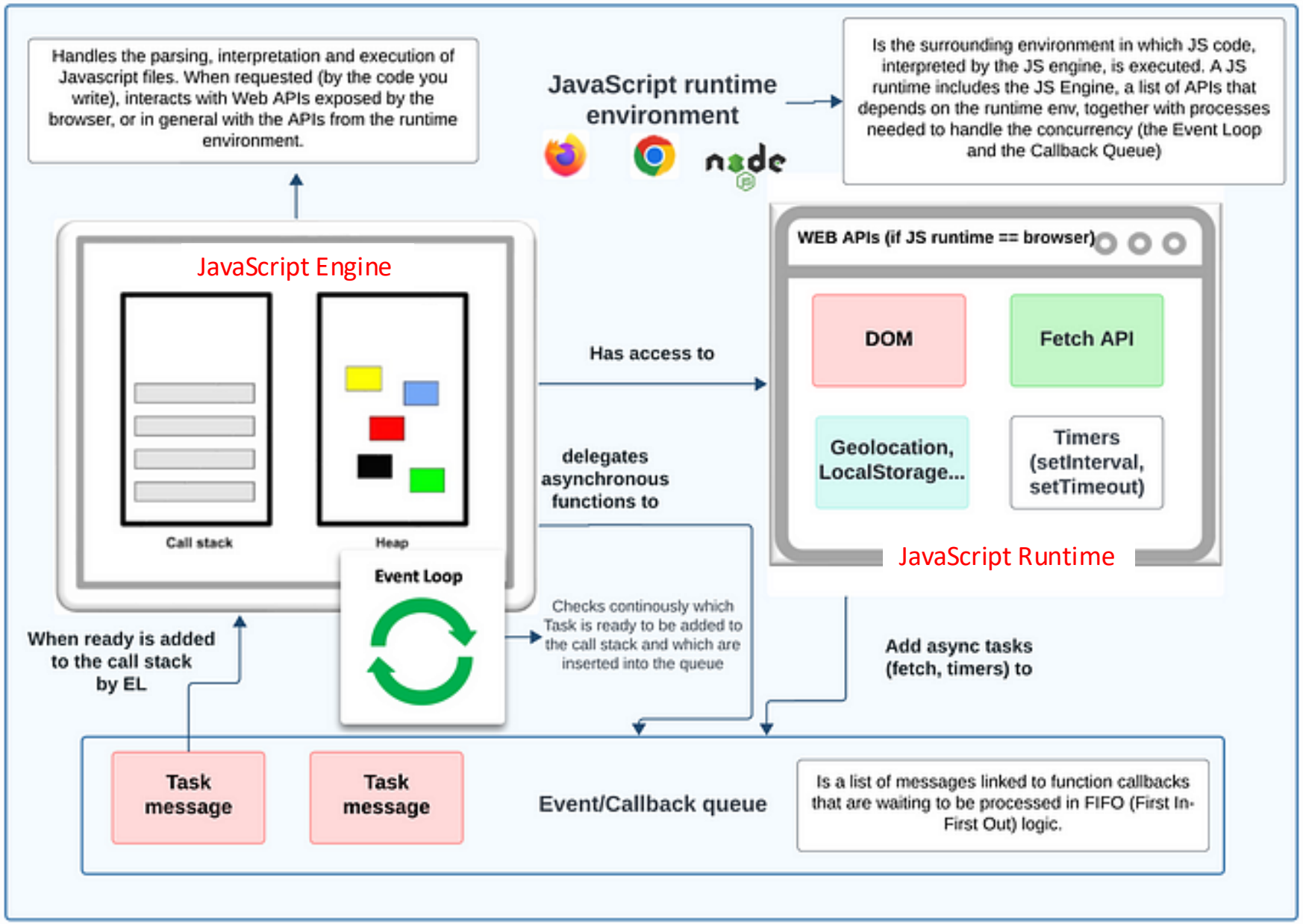


JavaScript

JavaScript เป็นภาษาโปรแกรมชนิดหนึ่งที่จัดอยู่ในกลุ่มภาษาสคริปต์
โดยภาษาในกลุ่มนี้ไม่ต้องการกระบวนการคอมไพล์หรือการแปลภาษา (Compilation Process)
ซึ่งเป็นกระบวนการที่จะแปลงสารทั้งหมดเพื่อทำให้คอมพิวเตอร์เข้าใจ

javaScript จะจัดเป็นภาษาโปรแกรมแบบ Interpreted โดยมี JIT (Just-in-time) Compiler
ช่วยแปลงสารระหว่างนักพัฒนาและคอมพิวเตอร์อยู่ตลอดเวลา

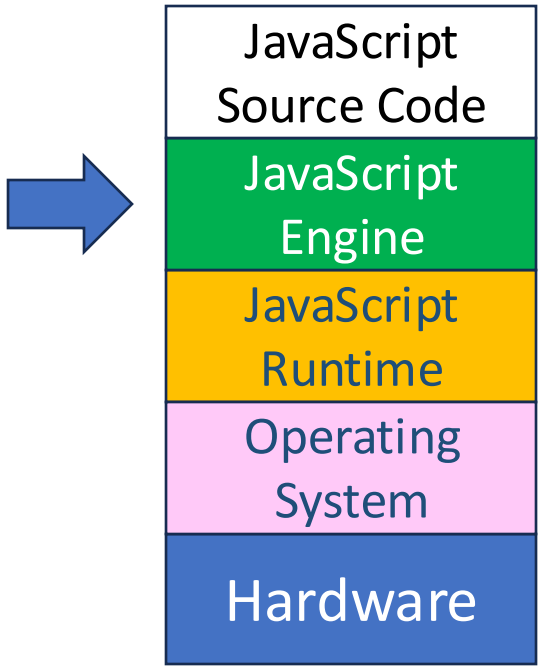




ซึ่งในปัจจุบันที่ได้รับความนิยมมากที่สุด คือ V8 JavaScript Engine ซึ่งอัปเดตถึง v9.1 แล้ว
เจ้านี่เป็น Engine ที่มีประสิทธิภาพสูง พัฒนาโดย Google ด้วยภาษา C++ ใช้ใน Chrome และ Node.js

[JavaScript in depth]

JavaScript Engine

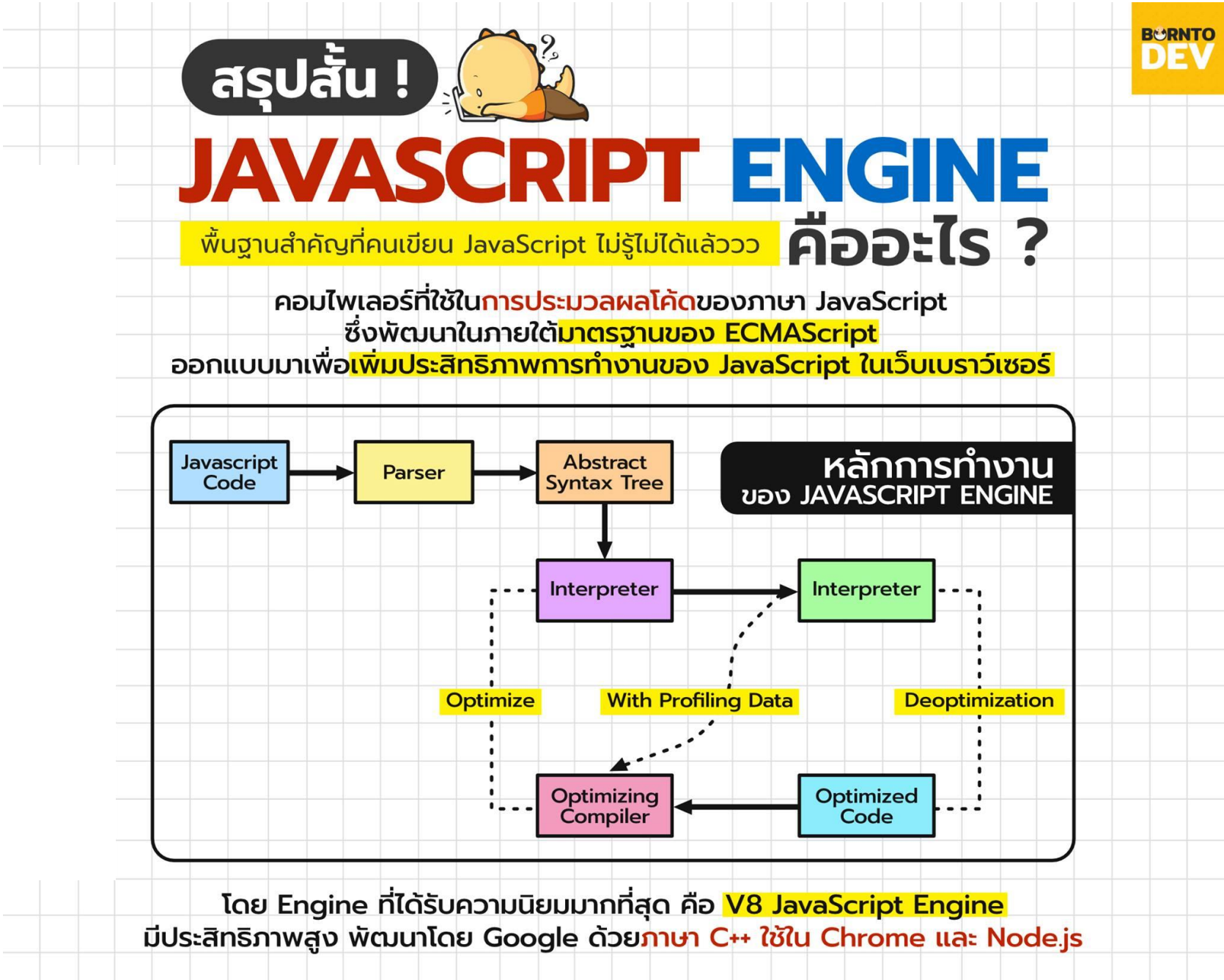


JavaScript Engine คือ

คอมไพเลอร์ที่ใช้ในการประมวลผลโค้ดของภาษา JavaScript

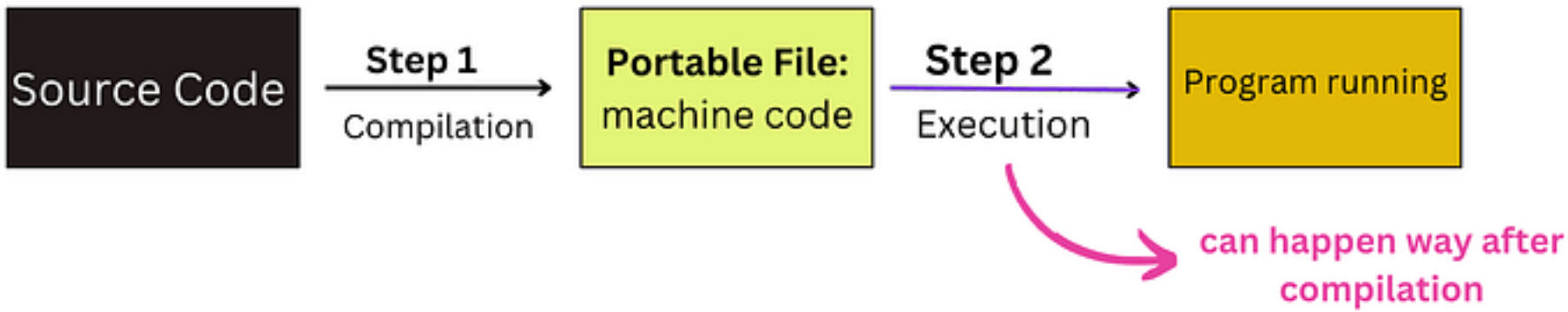
ซึ่งพัฒนาในภายใต้มาตรฐานของ ECMAScript

ออกแบบมาเพื่อเพิ่มประสิทธิภาพการทำงานของ JavaScript ในเว็บเบราว์เซอร์



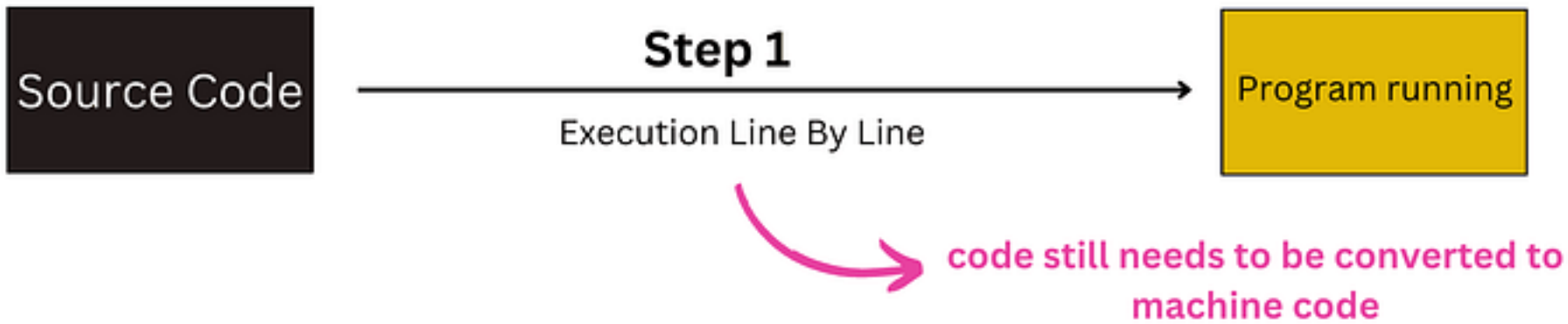
BorntoDev

Compilation



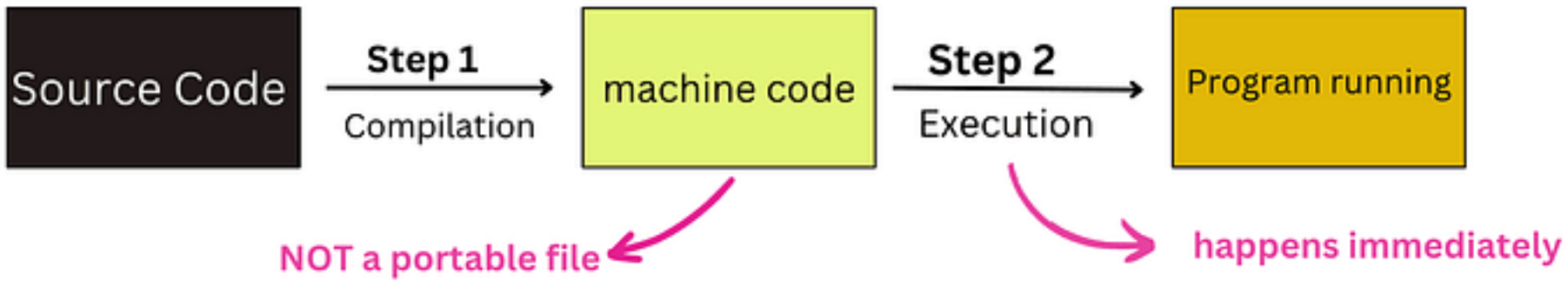
https://srivastavaankita080.medium.com/javascript-engine-runtime-ae9d392c170a

Interpretation



https://srivastavaankita080.medium.com/javascript-engine-runtime-ae9d392c170a

JIT (just in time) compiler

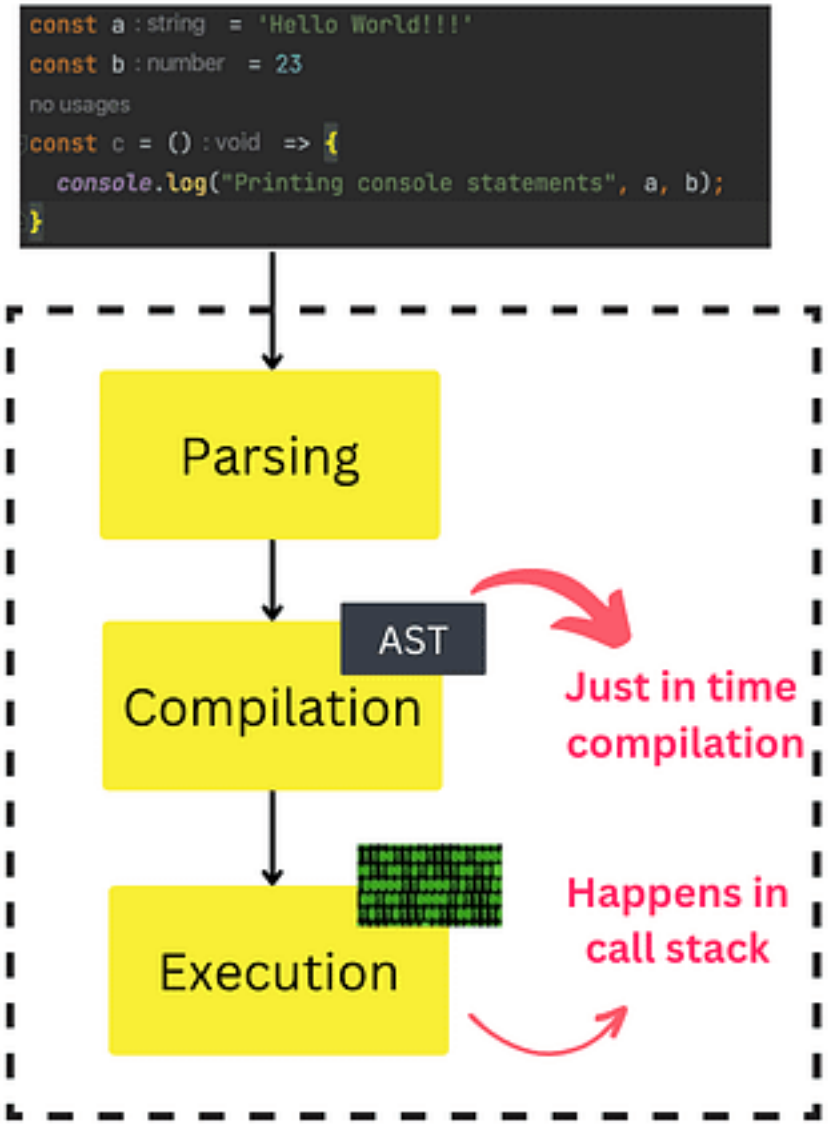


JavaScript combines both compilation and interpretation.
This is called "**Just-in-Time compilation**".

https://www.freecodecamp.org/news/how-javascript-works-behind-the-scenes/

https://srivastavaankita080.medium.com/javascript-engine-runtime-ae9d392c170a

JIT (just in time) compiler

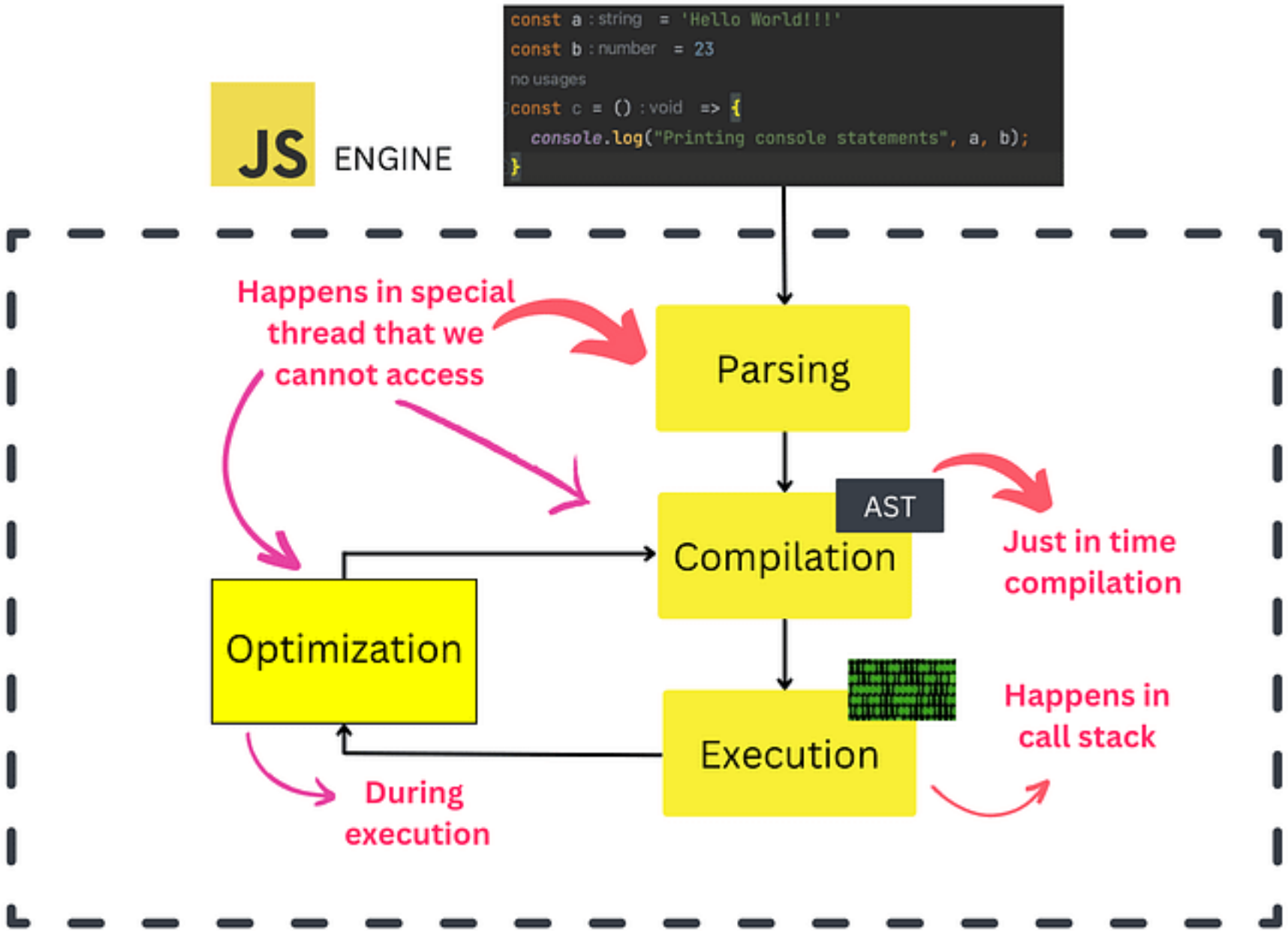


AST

```
body: [
 - VariableDeclaration {
      type: "VariableDeclaration"
      start: 0
      end: 26
    - declarations: [
       - VariableDeclarator {
            type: "VariableDeclarator"
            start: 6
            end: 26
          + id: Identifier {type, start, en
          - init: Literal = $node {
               type: "Literal"
               start: 10
               end: 26
               value: "Hello World!!!"
               raw: "'Hello World!!!'"
            }
```

https://srivastavaankita080.medium.com/javascript-engine-runtime-ae9d392c170a

JIT (just in time) with Optimize compiler



https://srivastavaankita080.medium.com/javascript-engine-runtime-ae9d392c170a

สรุป JavaScript Engine

JavaScript Engine

คือซอฟต์แวร์ที่ออกแบบมาเพื่อแปลงและรันโค้ด JavaScript ให้เครื่องคอมพิวเตอร์หรืออุปกรณ์สามารถเข้าใจและทำงานได้ โดยทำหน้าที่ดังนี้:

1.การแปลงโค้ด JavaScript:

1. เปลี่ยนโค้ด JavaScript (ที่เป็น high-level code) ให้กลายเป็น machine code ที่ CPU เข้าใจได้
2. ใช้เทคนิคเช่น **Just-In-Time (JIT) Compilation** เพื่อเพิ่มประสิทธิภาพในการรันโค้ด

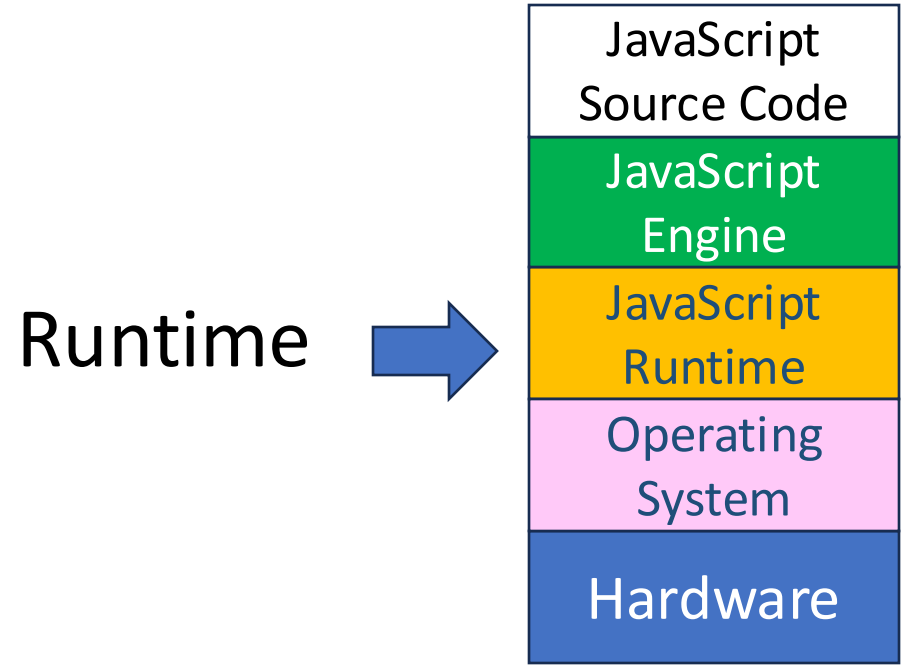
2.ส่วนประกอบหลัก:

1. **Parser**: แปลงโค้ด JavaScript ให้เป็น Abstract Syntax Tree (AST)
2. **Interpreter**: อ่านและรันโค้ดทีละบรรทัด
3. **Compiler**: แปลงโค้ดเป็น machine code (ผ่าน JIT Compiler)
4. **Garbage Collector**: จัดการหน่วยความจำอัตโนมัติ

3.ตัวอย่าง JavaScript Engine:

1. **V8**: ใช้ใน Google Chrome และ Node.js.
2. **SpiderMonkey**: ใช้ใน Mozilla Firefox
3. **JavaScriptCore**: ใช้ใน Safari
4. **Chakra**: ใช้ใน Microsoft Edge รุ่นเก่า (ก่อนเปลี่ยนมาใช้ Chromium)

JavaScript Runtime



สรุป JavaScript Runtime

JavaScript Runtime

คือสภาพแวดล้อมที่ให้เครื่องมือและ API ต่างๆ เพื่อช่วยให้โค้ด JavaScript ทำงานได้อย่างสมบูรณ์ โดยรวมถึงองค์ประกอบต่อไปนี้:

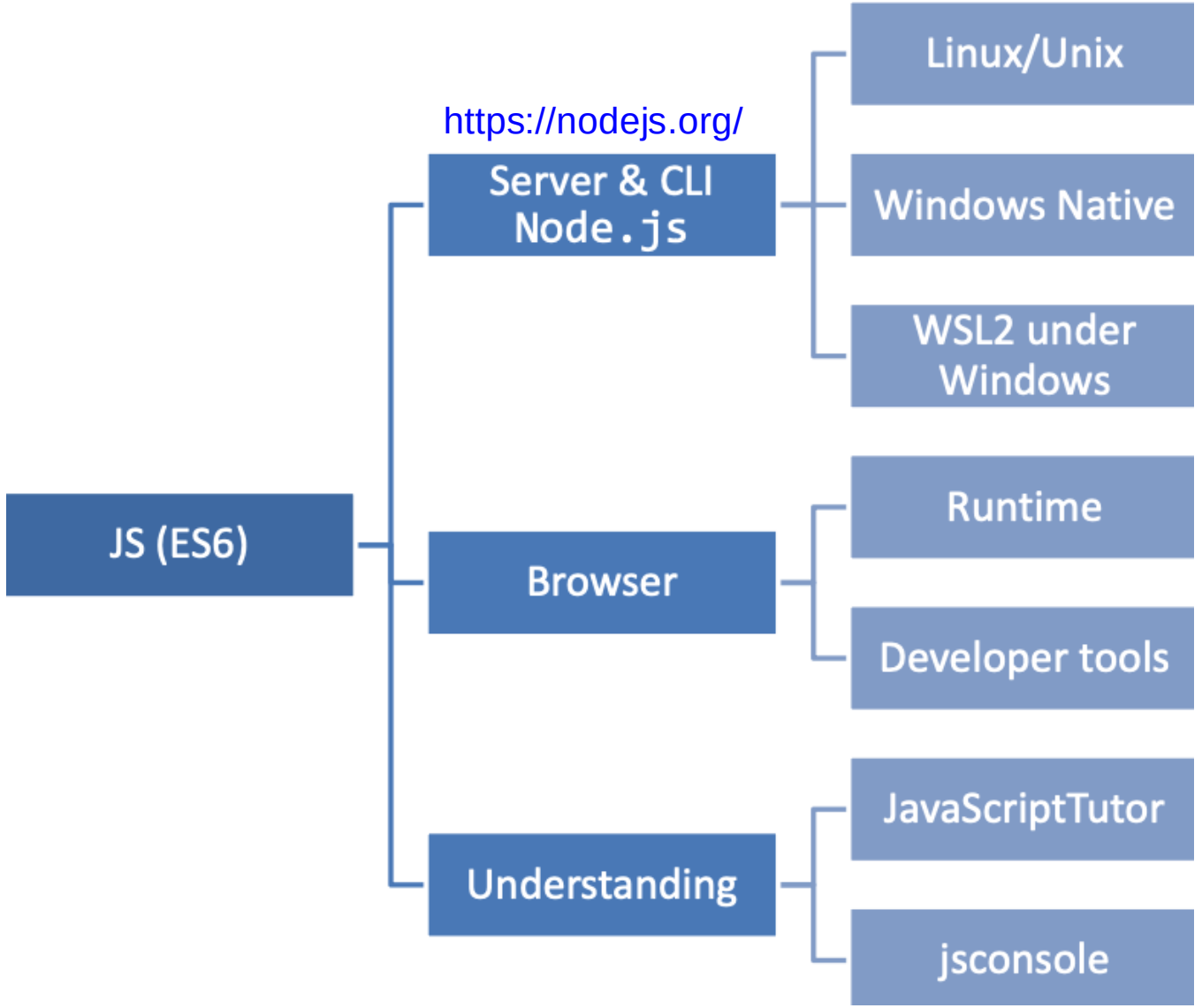
1.สิ่งที่จัดเตรียมให้โดย Runtime:

1. **JavaScript Engine**: เป็นแกนกลางในการประมวลผลโค้ด JavaScript
2. **Built-in APIs**: เช่น setTimeout, fetch, หรือ console
3. **Event Loop**: ช่วยจัดการงานที่เป็น asynchronous
4. **Callback Queue และ Task Queue**: สำหรับจัดการงานที่ต้องรอ (เช่น I/O หรือ Timer)

2.ตัวอย่างของ JavaScript Runtime:

1. **Browser Runtime**:
    1. ให้ API เช่น DOM (document.querySelector), AJAX (XMLHttpRequest), และ Web APIs อื่นๆ
2. **Node.js Runtime**:
    1. เพิ่ม API สำหรับการทำงานในฝั่งเซิร์ฟเวอร์ เช่น fs (File System), http, และ Buffer

JS Runtime (Execution) Environments



https://nodejs.org/en/download/package-manager/

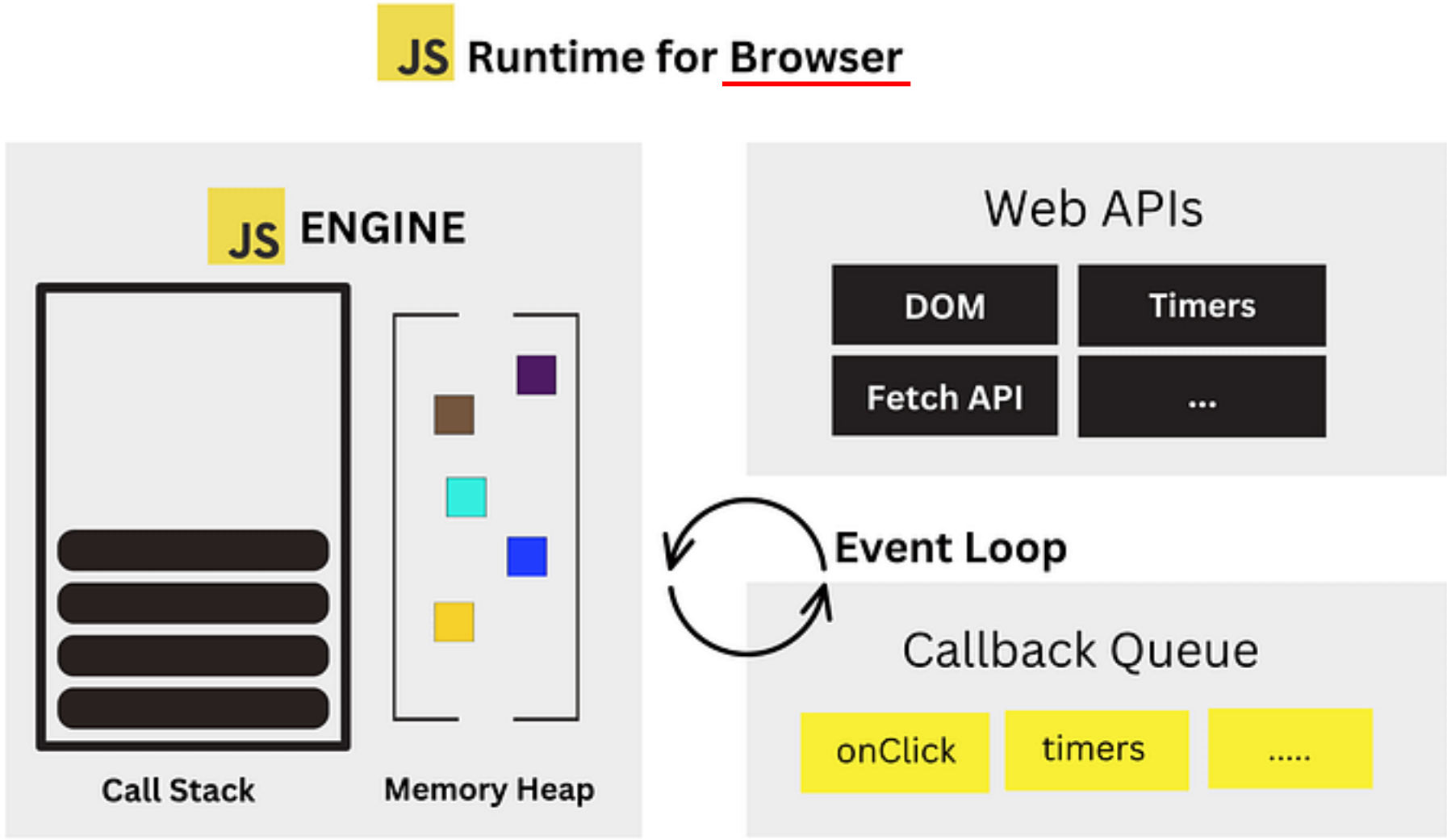
https://docs.microsoft.com/en-us/windows/nodejs/setup-on-windows

https://docs.microsoft.com/en-us/windows/nodejs/setup-on-wsl2

http://pythontutor.com/javascript.html

https://jsconsole.com/

JS Runtime for Browser



https://srivastavaankita080.medium.com/javascript-engine-runtime-ae9d392c170a

Example - Runtime for Browser

loupe help

```
1  $.on('button', 'click', function onClick() {
2      setTimeout(function timer() {
3          console.log('You clicked the button!');
4      }, 1000);
5  });
6
7  console.log("Hi!");
8
9  setTimeout(function timeout() {
10     console.log("Click the button!");
11 }, 3000);
12
```

Save + Run

Click me!

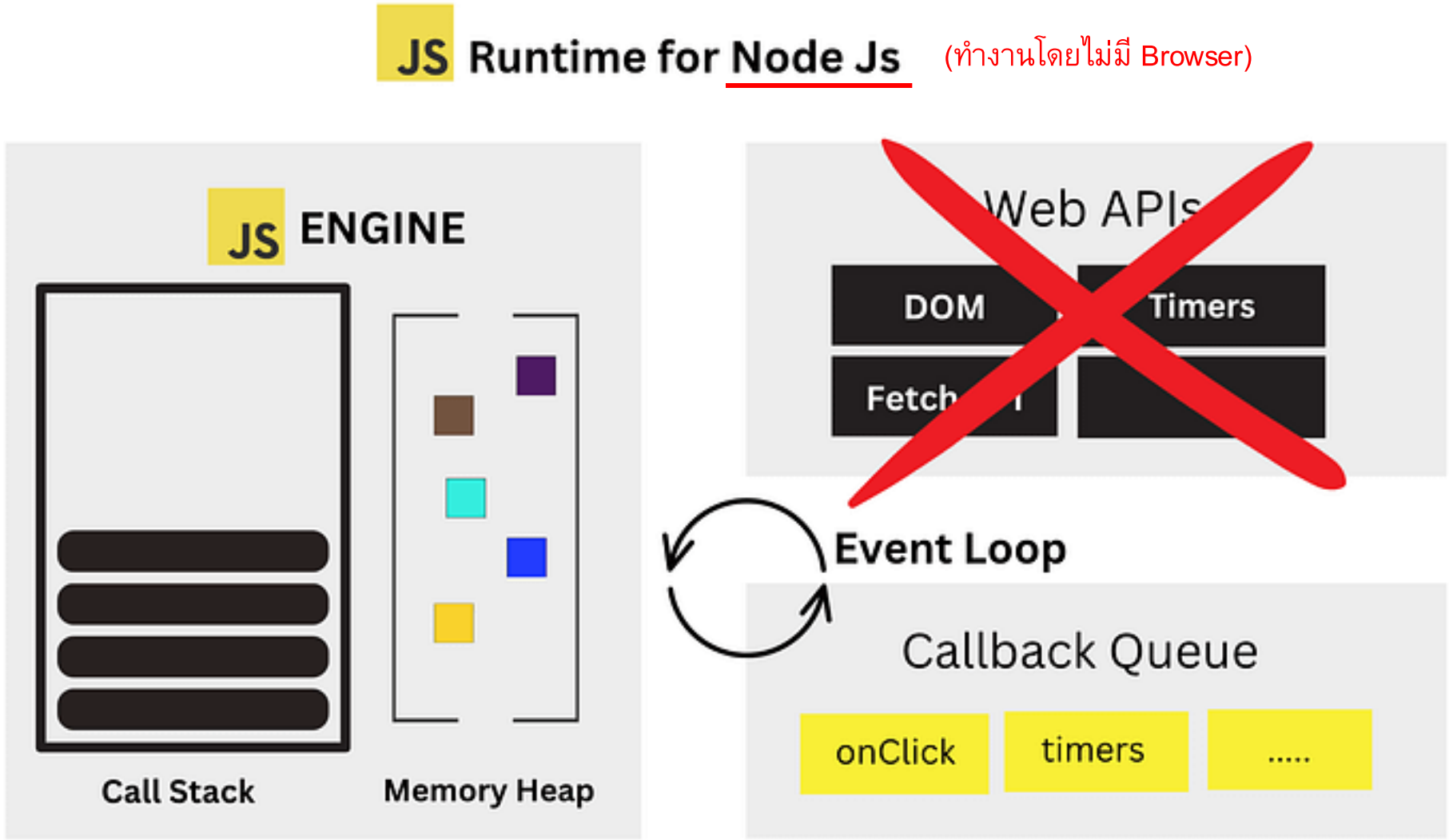
Edit

Call Stack

Web Apis

Callback Queue

JS Runtime for Node JS



https://srivastavaankita080.medium.com/javascript-engine-runtime-ae9d392c170a

สรุปความแตกต่างระหว่าง JavaScript Engine และ JavaScript Runtime

หัวข้อ	JavaScript Engine	JavaScript Runtime
บทบาท	แปลงและประมวลผลโค้ด JavaScript	สร้างสภาพแวดล้อมให้โค้ดทำงานได้เต็มที่
ตัวอย่าง	V8, SpiderMonkey, JavaScriptCore	Browser (Chrome, Firefox) หรือ Node.js
API ที่ให้	ไม่มี (เฉพาะการประมวลผลภาษา JavaScript)	มี API เสริม เช่น DOM หรือ File System
Event Loop	ไม่มี	มี เพื่อจัดการงาน asynchronous

สรุป

JavaScript Engine เป็นหัวใจที่ทำให้โค้ด JavaScript ทำงานได้ โดยแปลงโค้ดเป็น machine code

JavaScript Runtime เป็นกรอบที่จัดเตรียมสิ่งแวดล้อมและเครื่องมือสำหรับรันโค้ด รวมถึง API และ Event Loop เพื่อการทำงานที่ซับซ้อนมากขึ้น

ทั้งสองทำงานร่วมกันเพื่อให้ JavaScript ทำงานได้ทั้งในเบราว์เซอร์และบนเซิร์ฟเวอร์ (ผ่าน Node.js)

Let’s practice by your own