

JavaScript



JavaScript for Beginner

Introduction to JavaScript #1

Topic

- ทำความรู้จัก JavaScript
- คู่มือ และ Spec
- แนะนำ เครื่องมือใช้เขียน Code
- การใช้งาน Dev mode/Console

ทำความรู้จัก JavaScript

JavaScript คืออะไร?

- *JavaScript* สร้างขึ้นเพื่อให้ “เว็บไซต์ ดูมีชีวิตโต้ตอบ กับ user ได้ หรือ effect ต่างๆ”.
- เป็นภาษา *scripts*. โดยเราสามารถเขียน เข้าไปใน HTML และ สามารถ **run** ได้อัตโนมัติ เมื่อ **page loads**.
- **Scripts** สามารถ **Run** ได้เลย.
- โดยไม่ต้องมีขั้นตอนในการ **compilation (Interpreters)**
- ไม่ใช่ภาษาเดียวกันกับ ภาษา Java.

ทำไมเรียกว่า JavaScript?

- ตอนสร้าง JavaScript, แรกชื่อ : “LiveScript”.
 - ช่วงเวลาดังกล่าว ภาษา Java ค่อนข้างเป็นที่นิยม
 - เลยเปลี่ยนชื่อ.จนเป็น JavaScript หรือเรียกว่า ECMAScript (ES6 2015, ES15 2024)
- และก็ไม่ได้เกี่ยวกับ ภาษา Java เลย

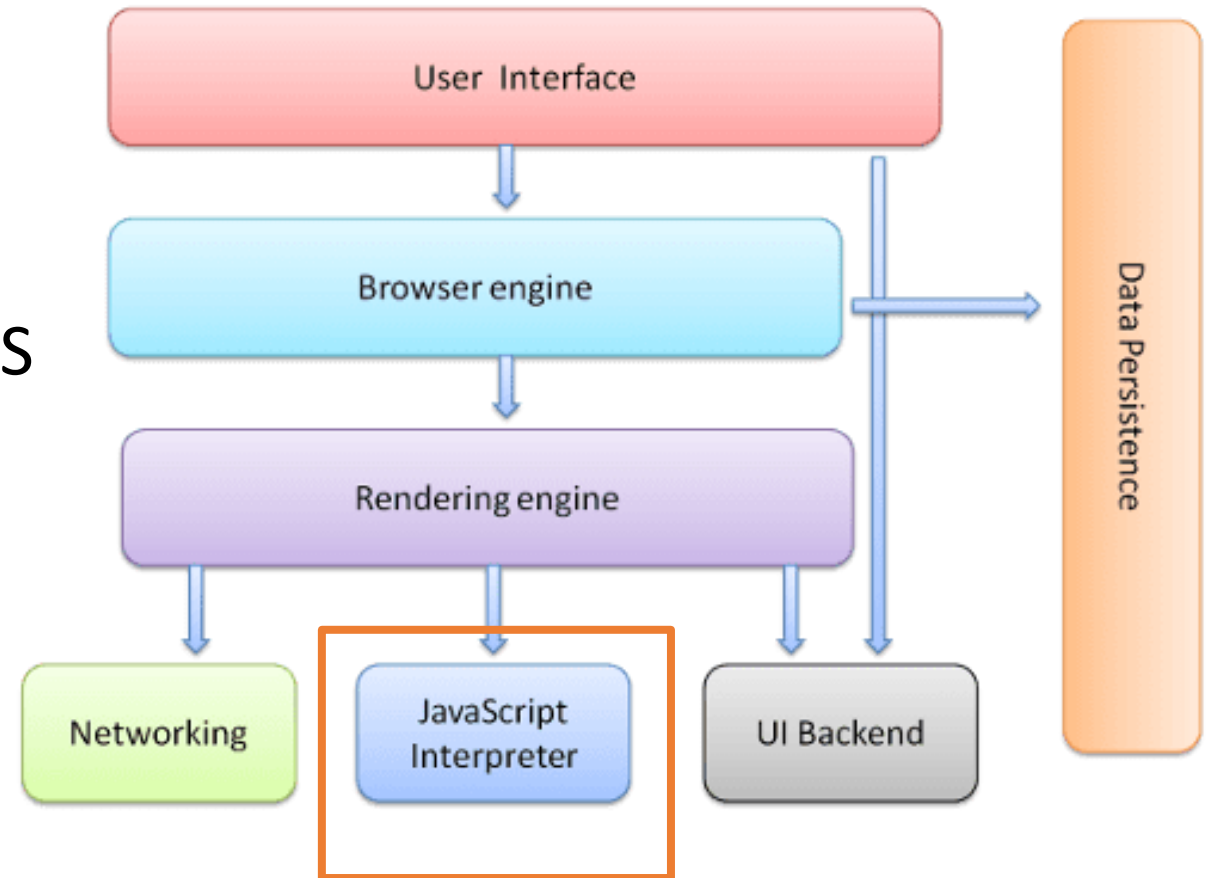
JavaScript Engine

- ทุกวันนี้เราสามารถ Run JavaScript ได้ ไม่เฉพาะบน Browser เท่านั้น แต่ยังสามารถ Run บน the server หรืออุปกรณ์ ต่างๆ ที่มี [JavaScript engine](#).
- ใน Browser จะมี Engine อยู่เรียกว่า “JavaScript virtual machine”.
- ชื่อของ JavaScript Engine มีหลายตัว ได้แก่:
 - [V8](#) – ใน Chrome, Opera and Edge.
 - [SpiderMonkey](#) – ใน Firefox.
 - “Squirrelfish” ใน Safari, etc.



Browser Component

- **User Interface** แถบ address, ปุ่ม Bookmark, ปุ่มย้อนกลับ ,เมนู
- **Browser Engine**
- **Rendering engine** แสดงผล HTML, CSS
- **Networking** จัดการ http request
- **JavaScript Interpreter**
- **UI Backend** หน้าจอ กำหนดค่า
- **Data persistence** คุกกี้ localStorage, IndexedDB, WebSQL



การทำงานของ JavaScript Engine (Client)

1. Parsing

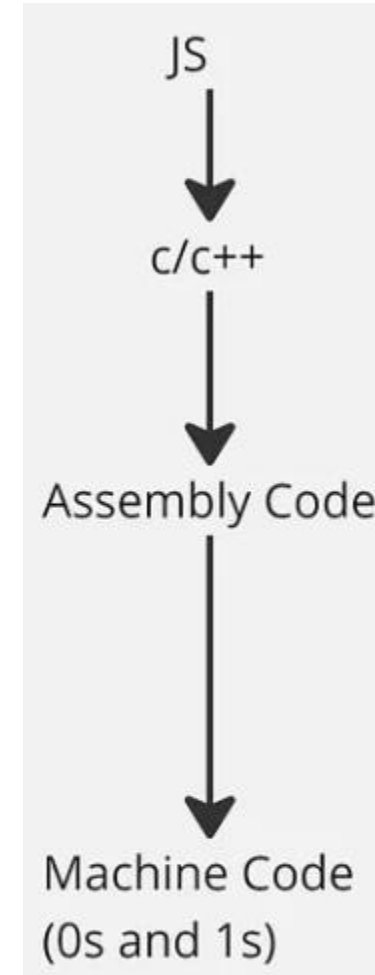
(การแปลงโค้ดเป็น AST)

2. Compilation

(การคอมไพล์ Byte Code แบบ Just-in-Time)

3. Execution (การประมวลผล)

4. Optimization



V8 JavaScript Engine (ของ Google) ทำงานดังนี้:

1. Parsing

V8 รับโค้ด JavaScript แล้วแปลงเป็น **Abstract Syntax Tree (AST)** ผ่านขั้นตอนการ **Parsing** โดย **Parser** จะทำการวิเคราะห์โค้ด

2. Ignition (Interpreter)

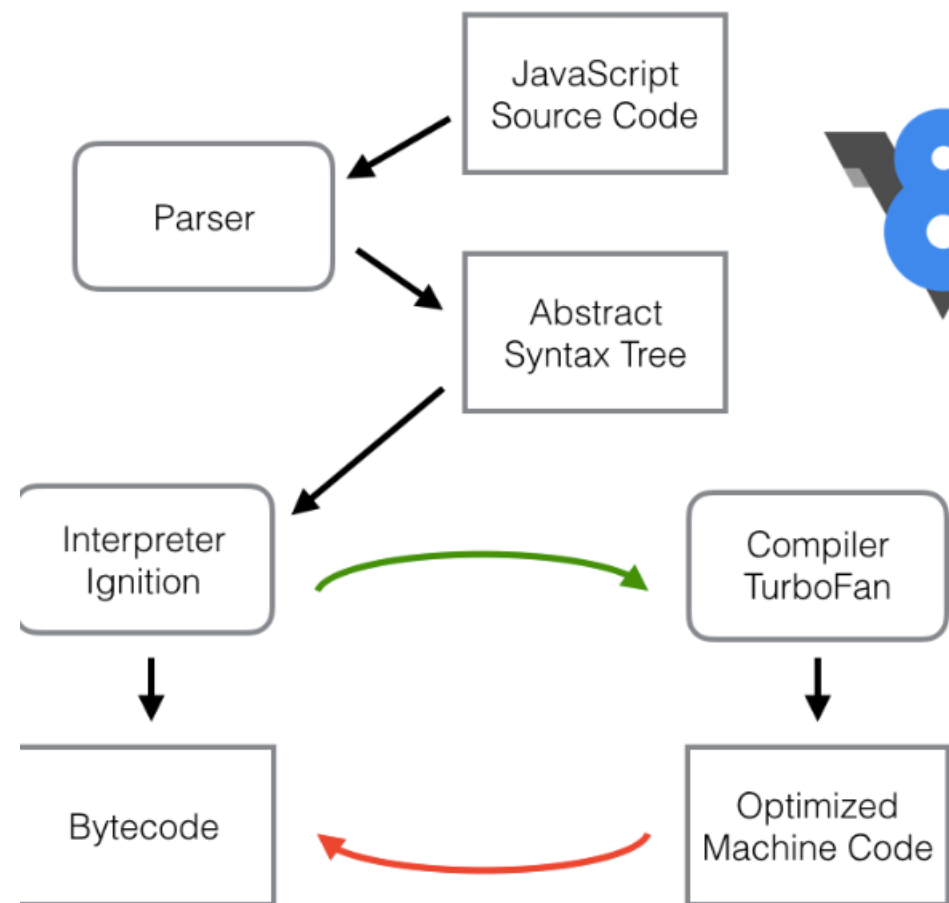
AST จะถูกแปลงเป็น **Bytecode** โดย Ignition ซึ่งเป็น **Interpreter** ของ V8 ทำให้โค้ดสามารถรันได้ในทันที

3. Turbofan (JIT Compiler)

ในขณะที่ Bytecode รัน, Turbofan จะวิเคราะห์โค้ดที่รันบ่อยๆ และคอมไพล์เป็น **Machine Code** ผ่านกระบวนการ **Just-In-Time (JIT)** เพื่อเพิ่มความเร็วในการรัน

4. Garbage Collection

V8 จัดการหน่วยความจำโดยใช้ระบบ **Garbage Collector** เพื่อลบข้อมูลที่ไม่ใช้งานแล้ว



JavaScript ฝั่ง Server



- NodeJS

- สามารถ Run JavaScript นอก Browser ได้ โดยที่ต้องติดตั้ง Node.js ซึ่ง
- ทั้งนี้ Node.js จะใช้ JavaScript Engine V8 ของ Google ในการทำงาน
- โดยสามารถเขียน Function ฝั่ง Server เช่น การเข้าถึง Database, Business Logic หรือ Event ต่างๆของ Server ได้, ตลอดจนทำงานเป็น Web Server ได้ด้วย เป็นต้น

ตัวอย่าง ที่ JavaScript สามารถ ทำได้ใน Browser

- เพิ่ม **HTML** ไปยัง **Page** เปลี่ยนเนื้อหา หรือ **Style**
- ตอบกับ **User** ที่ทำ **action** ต่างๆ เช่น **mouse click**, เลื่อน **pointer** ไปวาง หรือ กดปุ่ม
- ส่ง **Request** ไปยัง **Server** เพื่อ **Download, Upload File** ต่างๆ โดยสามารถใช้เทคโนโลยี เช่น **AJAX, COMET** เป็นต้น
- **Get/Set Cookies** ส่งคำถามให้กับผู้เข้า **web** หรือแสดง ข้อความ ต่างๆ
- เก็บข้อมูล ไว้ที่ **Client side (local storage)**

สิ่งที่ JavaScript ไม่สามารถทำใน Browser ได้

- Read/Write File บน Hard disk หรือ สั่ง Run โปรแกรม โดยตรง
 - Browser ใหม่ๆ จะยอมให้ เฉพาะการ เลือกไฟล์ ผ่าน input tag
- Access Different tabs/ Windows เรียกว่า Same Origin Policy
 - Tab1 : <http://gmail.com>
 - Tab2 : <http://site1.com>
- JavaScript สามารถ Communicate กับ Server ต่างๆ ผ่าน Network ได้
หลายๆ sites/domain แต่ต้องได้รับการยินยอม จากฝั่ง Remote ด้วยเพื่อ ความปลอดภัย

คู่มือ และ Spec

- [ECMAScript® 2025 Language Specification \(tc39.es\)](https://tc39.es/ecma262/)
 - <https://tc39.es/ecma262/>
- [GitHub - tc39/proposals: Tracking ECMAScript Proposals](https://github.com/tc39/proposals)
 - <https://github.com/tc39/proposals>
- [JavaScript Tutorial \(w3schools.com\)](https://www.w3schools.com/js/)
 - <https://www.w3schools.com/js/>
- [JavaScript | MDN \(mozilla.org\)](https://developer.mozilla.org/en-US/docs/Web/JavaScript)
 - <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [The Modern JavaScript Tutorial](https://javascript.info/)
 - <https://javascript.info/>
- [JavaScript | web.dev](https://web.dev/javascript) <https://web.dev/javascript>

Code Editor

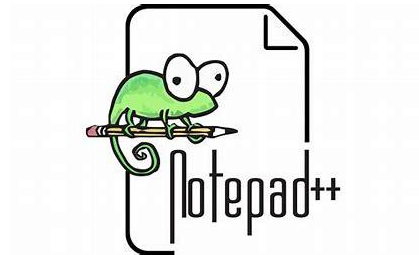
- **VS Code**



- **Sublime**



- **Notepad++**



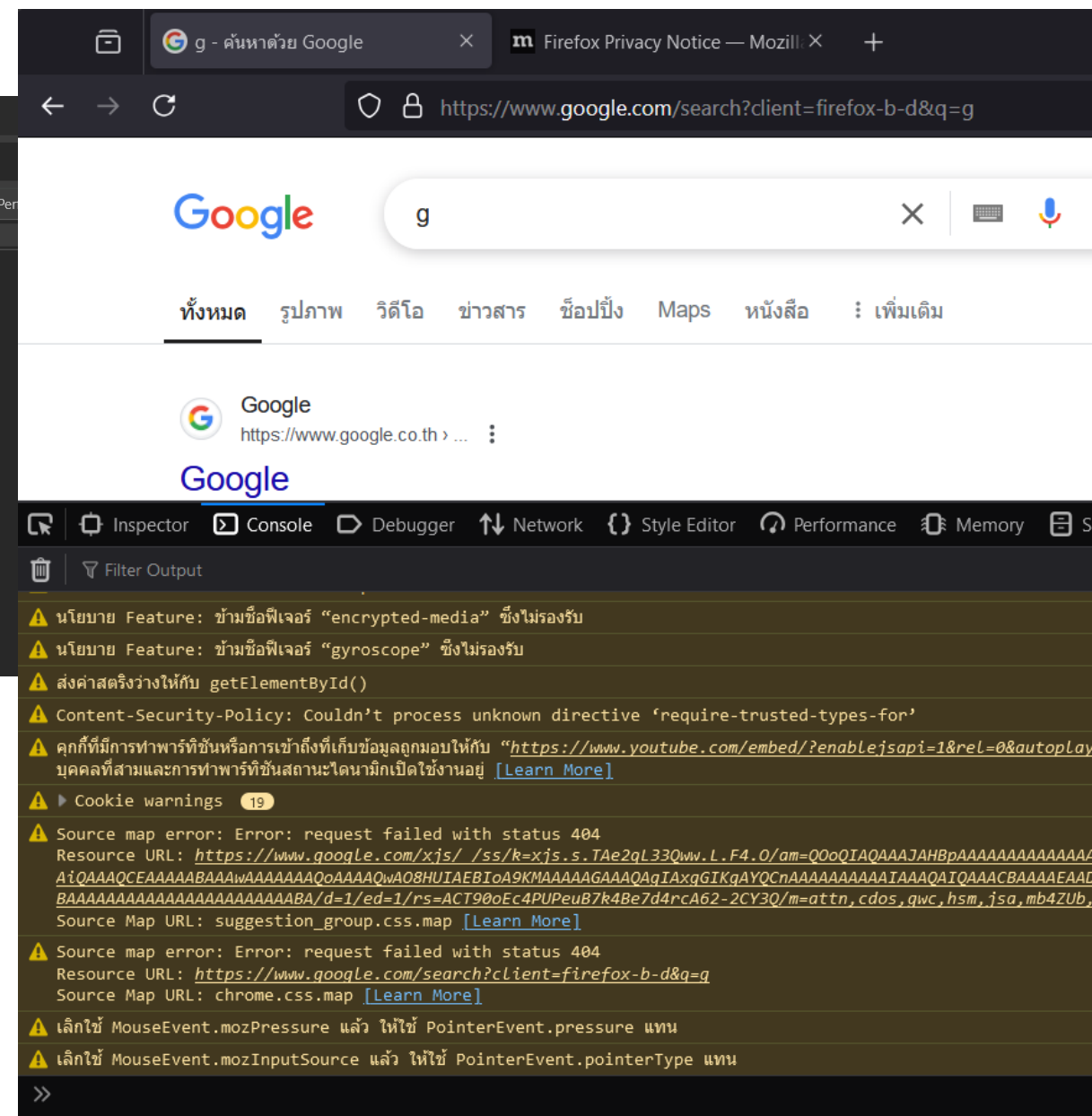
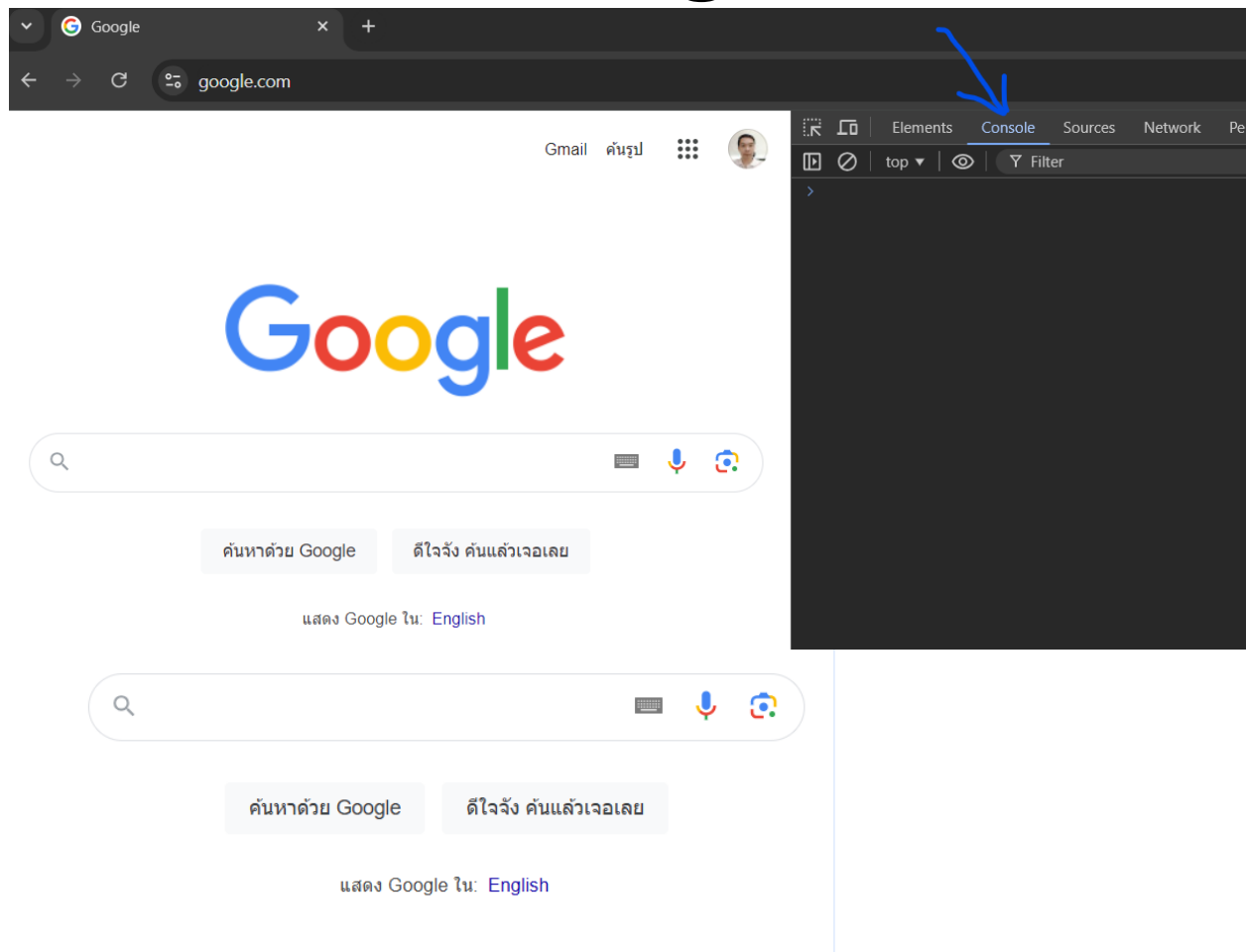
- **Vim**



Developer Mode

- เมื่อ Code มี Error เราสามารถ ตรวจสอบได้ ว่า Error ที่จุดไหน และสามารถ แก้ไขได้
- Google Chrome
 - กดปุ่ม **F12** หรือ ถ้าใช้ mac กดปุ่ม **cmd+opt+J**
- Firefox, Edge
 - กดปุ่ม **F12**
- Safari
 - Preference Advance, Checkbox Show develop menu in menu bar (**cmd+opt+C**)

Chrome / Edge



JavaScript Fundamentals

1. Hello, world!
2. Code structure
3. The modern mode, "use strict"
4. Variables
5. Data types
6. Interaction: alert, prompt, confirm
7. Type Conversions
8. Basic operators, math
9. Comparisons
10. Conditional branching: if, '?'
11. Logical operators
12. Nullish coalescing operator '??'
13. Loops: while and for
14. The "switch" statement
15. Functions
16. Function expressions
17. Arrow functions, the basics

Hello, World

```
<html>
<head><title>Learn JavaScript</title></head>
<body>
  <h1 id="topic">JavaScript</h1>
  <button onclick="changeText()">Click me</button>
  <button onclick="resetText()">Click me</button>
  <script>

    function changeText() {
      document.getElementById("topic").innerHTML = "Hello, World!";
    }
    function resetText() {
      document.getElementById("topic").innerHTML = "JavaScript";
    }
  </script>
</body>
</html>
```



Summary

- JavaScript เป็น ภาษา โปรแกรมมิ่ง ทำงาน แบบ Interpreter
 - สามารถ ทำงาน บน Browser ได้
 - None Browser ผ่าน Node.js
- เรียกว่า JavaScript หรือ ECMAScript (ES6 2015- ES15 2024)
- JavaScript Engine (V8, SpiderMonkey, Squirrelfish)
- Reference/Manual
- Editor (VS Code, Sublime, NotePad++, Vim)
- Developer Mode
- Hello, World

