

# Introduction to JavaScript

แนะนำภาษา JavaScript และการเขียนโปรแกรมเบื้องต้น

อ.อนิราช มิงขวัญ (Anirach Mingkhwan)

Anirach.m@fitm.kmutnb.ac.th

# โครงสร้างเนื้อหา (Course Outline)

## บทนำ (Introduction)

ทำความรู้จักกับ JavaScript ประวัติความเป็นมา และความสำคัญ

## การตั้งค่า (Setup)

การติดตั้งเครื่องมือและการเขียนโปรแกรม Hello World แรกของคุณ

## JavaScript ในเบราว์เซอร์

การทำงานของ JS ใน Web Browser และการจัดการ DOM

## ไวยากรณ์พื้นฐาน

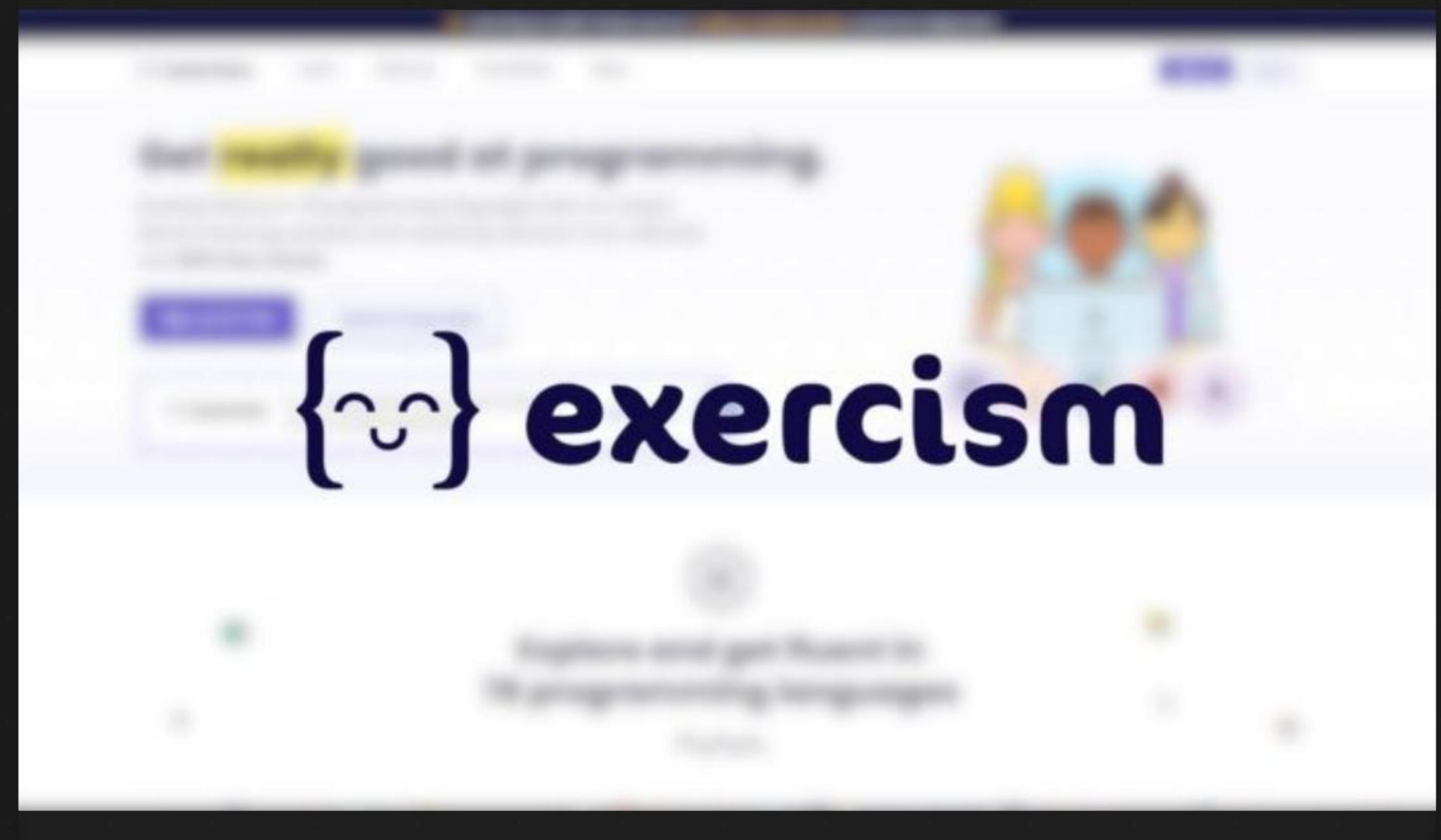
ตัวแปร ชนิดข้อมูล ตัวดำเนินการ และโครงสร้างควบคุม

# เกณฑ์การให้คะแนน (Grading Criteria)

หมวดหมู่ (Category)	คะแนน (Points)	รายละเอียด
การมีส่วนร่วม (Participation)	20%	การเข้าเรียนและกิจกรรมในห้อง
การบ้าน (Homework)	25%	5 ครั้ง (ครั้งละ 5 คะแนน)
โปรเจกต์ (Project)	10%	โครงการท้ายวิชา
สอบกลางภาค (Midterm)	15%	สอบข้อเขียน/ปฏิบัติ
สอบปลายภาค (Final)	30%	สอบข้อเขียน/ปฏิบัติ

# แหล่งเรียนรู้ออนไลน์: Exercism

- Exercism เป็นแพลตฟอร์มสำหรับฝึกฝนการเขียนโค้ด
- มี Track สำหรับ JavaScript ให้ฝึกฝนพรี
- มีระบบ Mentor ช่วยแนะนำโค้ด
- เป็น Open Source และไม่มีค่าใช้จ่าย



# แหล่งเรียนรู้อื่นๆ (Other Resources)



## Channel 9

Beginner's Series to: JavaScript  
โดย Microsoft



## Codecademy

Learn JavaScript Course แบบ  
Interactive



## JavaScript.info

บทเรียน Modern JavaScript ที่  
ละเอียดมาก

# JavaScript คืออะไร?

## ภาษาของเว็บ

JavaScript เป็นภาษาโปรแกรมที่ออกแบบมาให้ทำงานบน Web Browser เป็นหลัก

## การโต้ตอบ (Interactive)

ช่วยให้เว็บไซต์มีการโต้ตอบกับผู้ใช้ได้ไม่ใช่แค่แสดงผลนิ่งๆ (Static)

"เป็น 1 ใน 3 เทคโนโลยีหลักของเว็บ: HTML (โครงสร้าง), CSS (รูปลักษณ์), JavaScript (การทำงาน)"

# ประวัติและการกำเนิด

---



- พัฒนาโดย **Brendan Eich** จากบริษัท Netscape
- ชื่อเดิมคือ **Mocha**
- เปลี่ยนชื่อเป็น **LiveScript** ก่อนเปิดตัว
- เปลี่ยนเป็น **JavaScript** เพื่อผลทางการตลาด (ภาษากระแส Java)

# Java vs JavaScript



ภาษา OOP ที่เข้มงวด (Strict) ต้อง Compile ก่อนรัน ใช้สร้างแอปพลิเคชันขนาดใหญ่



ภาษาสคริปต์ (Scripting) ที่ยืดหยุ่น รันบนเบราว์เซอร์ได้ทันที ไม่เกี่ยวข้องกับทางเทคนิค

*"Car กับ Carpet ต่างกันฉันใด Java กับ JavaScript ก็ต่างกันฉันนี้"*

# มาตรฐาน ECMAScript (ES)

## ECMA

องค์กรที่กำหนดมาตรฐานของ  
ภาษา JavaScript

## ES6 (2015)

การอัปเดตครั้งใหญ่ที่สุด เพิ่ม  
ฟีเจอร์ modern เช่น let, const,  
arrow function

## ESNext

เวอร์ชันใหม่ๆ ที่ออกมาทุกปี  
(ES2016, ES2017, ...)

# JavaScript Engine

เบราว์เซอร์แต่ละตัวมี "Engine" ในการประมวลผลโค้ด JS ที่ต่างกัน

## V8 Engine

ใช้ใน Google Chrome และเป็นพื้นฐานของ Node.js

## SpiderMonkey

ใช้ใน Mozilla Firefox

# Browser vs Node.js

---

## Browser (Front-End)

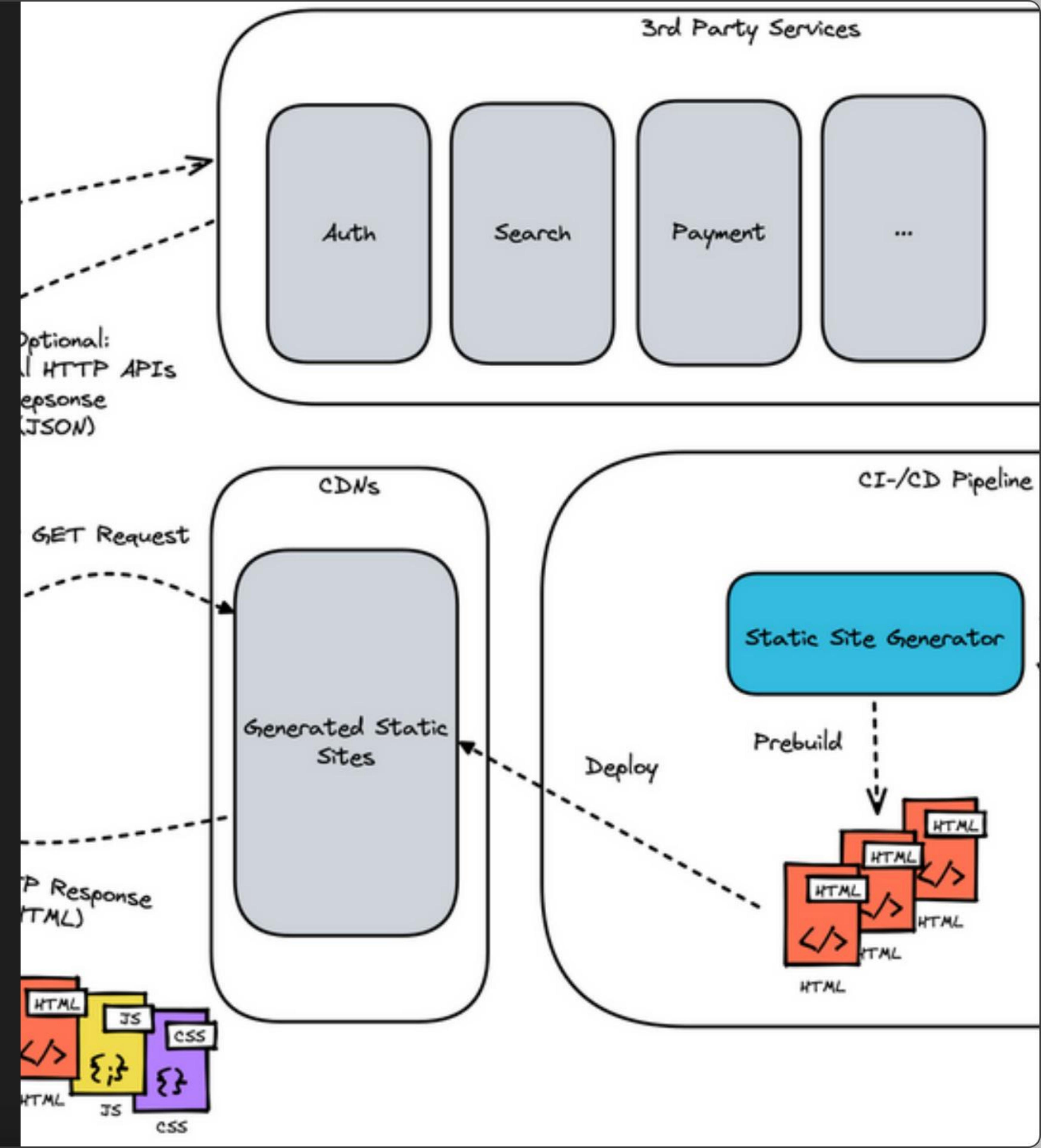
JavaScript ทำงานในเบราว์เซอร์เพื่อจัดการหน้าเว็บ กราฟิก และการโต้ตอบกับผู้ใช้

## Node.js (Back-End)

นำ V8 Engine ออกมารันข้างนอก ทำให้ JS เขียนโปรแกรมฝั่ง Server ได้เหมือน PHP หรือ Python

# Web App Architecture

- **Front-End:** ส่วนที่ผู้ใช้เห็น (HTML, CSS, JS)
- **Back-End:** ส่วนประมวลผลและฐานข้อมูล (Node.js, Python, SQL)
- **Full Stack:** การพัฒนาทั้งสองส่วนด้วย JavaScript



# เครื่องมือที่ต้องเตรียม (Setup Tools)



## Node.js

Runtime environment (แนะนำเวอร์ชัน LTS)



## Git

ระบบจัดการเวอร์ชัน (Version Control)



## VS Code

Code Editor ยอดนิยม

# การติดตั้ง Git และ GitHub

- ดาวน์โหลด Git จาก [git-scm.com](http://git-scm.com)
- สมัครบัญชี GitHub เพื่อฝากไฟล์โค้ด
- สร้าง **Repository** ใหม่สำหรับเก็บโค้ดวิชานี้ (เช่นตั้งชื่อ JavaXcript)
- ใช้สำหรับส่งงานและทำงานร่วมกัน



# VS Code Extensions ที่แนะนำ

---

- **Prettier:** ช่วยจัดรูปแบบโค้ด (Format) ให้สวยงามอัตโนมัติ
- **Live Server:** จำลอง Server และรีเฟรชหน้าเว็บทันทีเมื่อแก้โค้ด
- **HTML Boilerplate:** สร้างโครงสร้าง HTML พื้นฐานได้อย่างรวดเร็ว

Tip: กด Extensions Icon (สีเหลือง) ใน VS Code เพื่อค้นหาและติดตั้ง

# ໄວຍາກຮັດໜີ້ພືນຫຼານ

(Basic Syntax)

# การประกาศตัวแปร (Variables)

JavaScript มีวิธีการประกาศตัวแปร 3 แบบ:

`var`

แบบดั้งเดิม (Old School)

`let`

แบบใหม่ (แนะนำให้ใช้)

`const`

ค่าคงที่ (Constant)

# รู้จักกับ var

- เป็นการประกาศตัวแปรยุคแรกของ JS
- ข้อเสีย: เป็น Function Scope (ทະลຸ Block ໄດ້)
- Hoisting: ถูกยกขึ้นไปประกาศข้างบนสุดเสมอ อาจทำให้ สับสน
- ปัจจุบันไม่แนะนำให้ใช้

```
var name = "John"; if (true) { var name =  
"Doe"; } console.log(name); // "Doe" (ทັບຄ່າ  
ເດີມ)
```

# รู้จักกับ let (แนะนำ)

- เริ่มใช้ในมาตรฐาน ES6
- เป็น **Block Scope** (ทำงานเฉพาะในปีกกา {})
- ปลอดภัยกว่า ป้องกันการประกาศตัวแปรซ้ำใน Scope  
เดียวกัน

```
let name = "John"; if (true) { let name =  
  "Doe"; console.log(name); // "Doe" }  
console.log(name); // "John" (ค่าเดิมยังอยู่)
```

# รู้จักกับ const

- ย่อมาจาก Constant (ค่าคงที่)
- ใช้สำหรับตัวแปรที่ไม่ควรถูกเปลี่ยนค่า
- ต้องกำหนดค่าทันทีที่ประกาศ
- เป็น Block Scope เหมือน let

```
const pi = 3.14; pi = 5; // Error! const  
user = { id: 1 }; user.id = 2; // ได้ (แก้ไขในได้)
```

# ชนิดข้อมูล (Data Types)

- **Primitive (พื้นฐาน):** เก็บค่าเดียว เช่น String, Number, Boolean
- **Non-primitive (อ้างอิง):** เก็บโครงสร้างข้อมูล เช่น Object, Array

## Data Types in Javascript

Data types

Non-pr

Non-Numeric Type

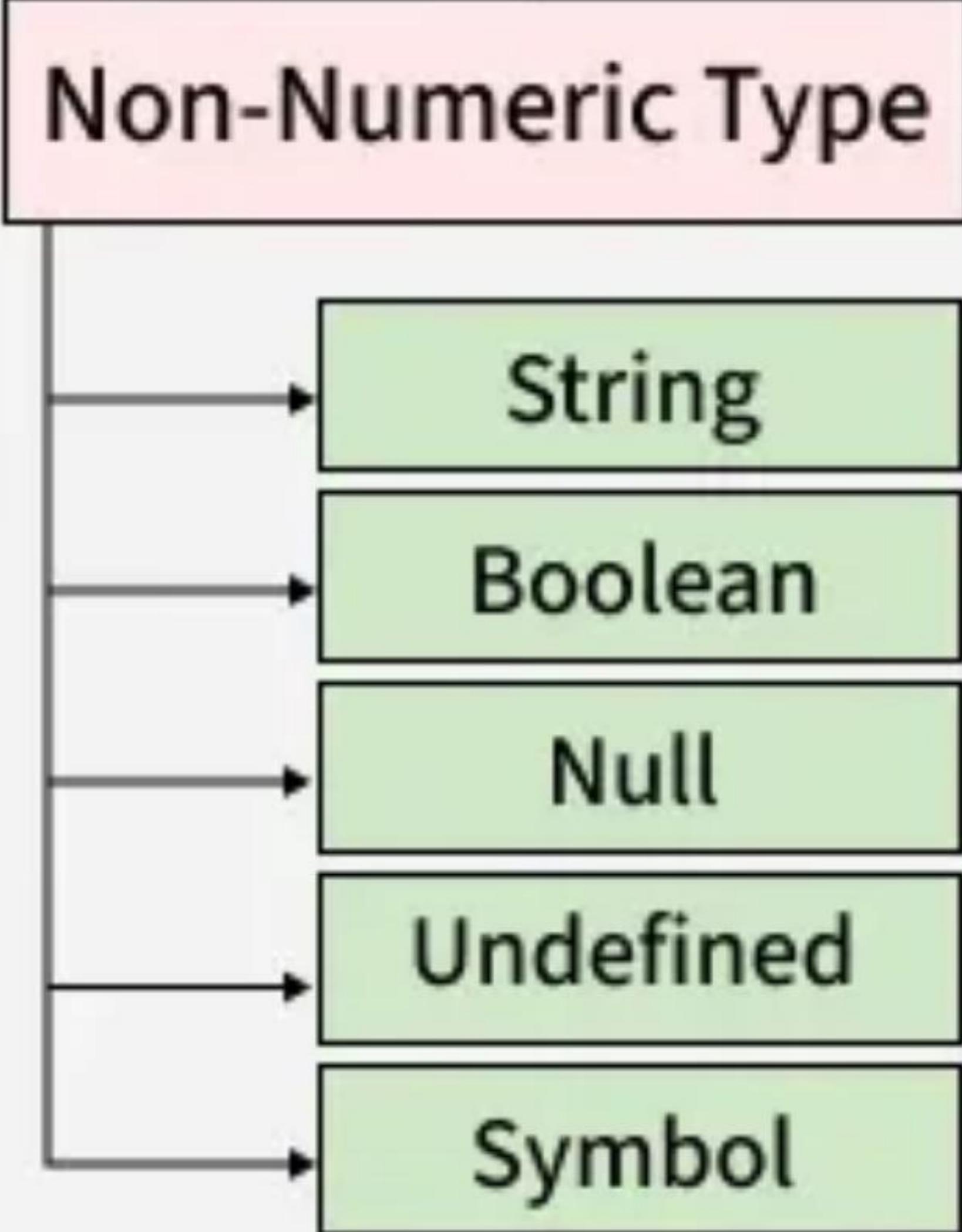
String

Boolean

Null

Undefined

Symbol



# Primitive Data Types (1)

## String

เก็บข้อความ

```
"Hello", 'World'
```

## Number

เก็บตัวเลข (ทั้งเต็มและทศนิยม)

```
42, 3.14
```

## Boolean

ค่าความจริง

```
true, false
```

# Primitive Data Types (2)

## Null & Undefined

**Null:** ค่าว่างเปล่า (เรากำหนดเอง)

**Undefined:** ยังไม่ได้กำหนดค่า (โปรแกรมบอก)

## Symbol

ค่าที่ไม่ซ้ำกัน (Unique) ใช้เป็น Identifier (เริ่มใน ES6)

# Object (Non-primitive)

---

- ใช้เก็บข้อมูลหลายค่าในรูปแบบ **Key-Value pair**
- เปรียบเสมือนวัตถุที่มีคุณสมบัติ (Property)
- สร้างด้วยปีกกา { ... }

```
let student = { name: "John", age: 20,  
    isActive: true };
```

# Array (Non-primitive)

---

- ใช้เก็บข้อมูลหลายค่าในรูปแบบ ลำดับ (Sequence)
- เข้าถึงข้อมูลด้วย Index (เริ่มที่ 0)
- สร้างด้วยวงเล็บเหลี่ยม [ ... ]

```
let colors = ["Red", "Green", "Blue"];
console.log(colors[0]); // "Red"
```

# ตัวดำเนินการทางคณิตศาสตร์

+

บวก (Addition)

-

ลบ  
(Subtraction)

\*

คูณ  
(Multiplication)

/

หาร (Division)

%

หารเอาเศษ  
(Modulus)

# การเปรียบเทียบและตรรกะ

Operator	ความหมาย	ตัวอย่าง
<code>== / ===</code>	เท่ากับ (Strict แนะนำ <code>===</code> )	<code>5 === '5' // false</code>
<code>!= / !==</code>	ไม่เท่ากับ	<code>a !== b</code>
<code>&amp;&amp;</code>	และ(AND)	<code>true &amp;&amp; false // false</code>
<code>  </code>	หรือ(OR)	<code>true    false // true</code>

# นิพจน์ (Expressions)

---

การรวม ตัวแปร และ ตัวดำเนินการ เพื่อให้ได้ผลลัพธ์

```
let result = (a + b) * 2;
```

# คำสั่งเงื่อนไข (Conditionals)

ใช้ตัดสินใจการทำงานของโปรแกรมตามเงื่อนไข (True/False)

- if
- else if
- else
- switch

```
if (age < 18) { console.log("Minor"); }  
else { console.log("Adult"); }
```

# การวนรอบ: For Loop

---

ใช้ทำซ้ำชุดคำสั่งเมื่อ รู้จำนวนรอบที่แน่นอน

```
for (let i = 0; i < 5; i++) {  
  console.log(i); } // Output: 0, 1, 2, 3, 4
```

# การวนรอบ: While Loop

ใช้ทำซ้ำชุดคำสั่ง ตราบเท่าที่เงื่อนไขยังเป็นจริง

```
let count = 0; while (count < 5) {  
    console.log(count); count++; }
```

# ฟังก์ชัน (Functions)

---

บล็อกของโค้ดที่ทำงานเฉพาะอย่างและนำกลับมาใช้ซ้ำได้  
(Reusable)

```
function greet(name) { return "Hello, " +  
  name; } let msg = greet("Alice");
```

# Arrow Function (ES6)

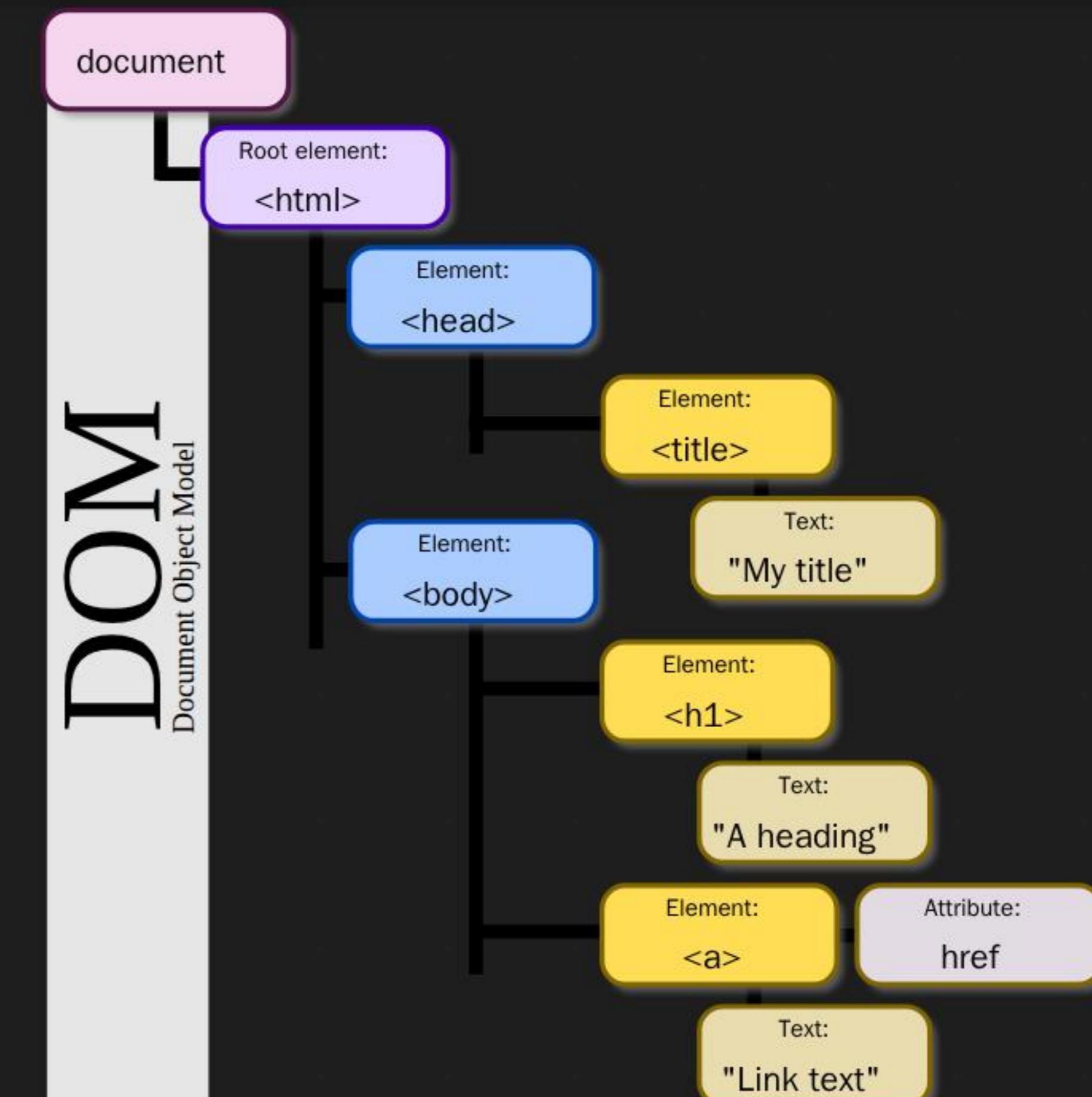
---

รูปแบบการเขียนฟังก์ชันแบบย่อ ช่วยให้โค้ดกระชับขึ้น

```
// Old way const add = function(a, b) {  
  return a + b; }; // Arrow Function const  
add = (a, b) => a + b;
```

# Document Object Model (DOM)

- DOM คือโครงสร้างต้นไม้ (Tree) ที่แสดงเอกสาร HTML
- JavaScript ใช้ DOM เพื่อเข้าถึง แก้ไข และลบ Element ในหน้าเว็บ



# การจัดการ DOM (Manipulation)

ตัวอย่างการเลือกและเปลี่ยนเนื้อหา HTML:

```
// เลือก Element โดย ID let el =  
document.getElementById("demo"); // เปลี่ยน  
เนื้อหา el.innerHTML = "Hello World"; //  
เปลี่ยน Style el.style.color = "red";
```

# การจัดการเหตุการณ์ (Event Handling)

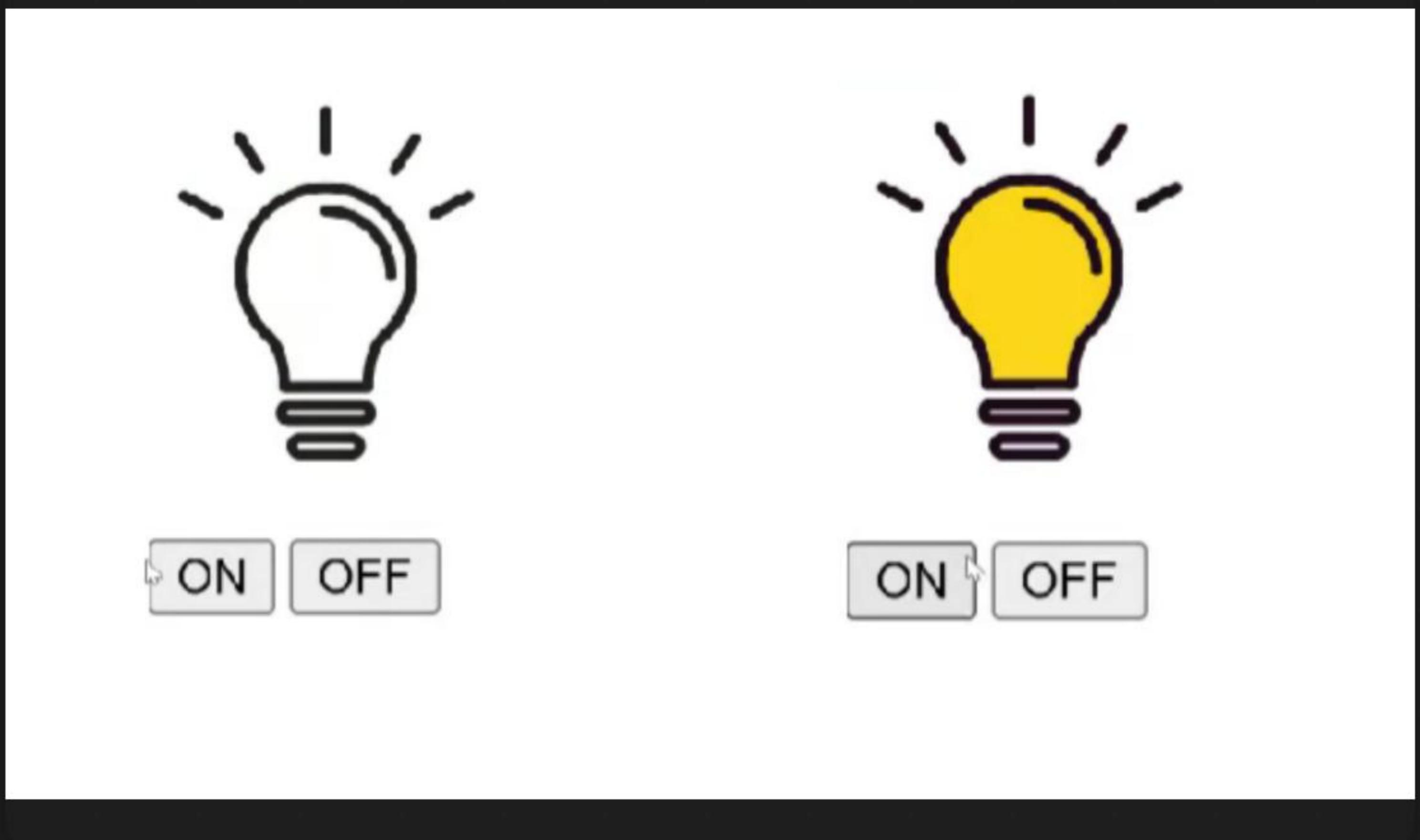
ดักจับการกระทำของผู้ใช้ เช่น คลิก (click), พิมพ์ (keypress)

```
let btn =  
  document.querySelector("button");  
  btn.addEventListener("click", function() {  
    alert("Button Clicked!");  });
```

# ตัวอย่าง: เปิด-ปิดหลอดไฟ

ใช้ JavaScript เปลี่ยนค่า Attribute src ของรูปภาพ

```
// Turn ON img.src = 'pic_bulbon.gif'; // Turn OFF  
img.src = 'pic_bulboff.gif';
```



# วิธีใส่ JS ใน HTML

---

## Internal

เขียนในไฟล์ HTML

```
</div> console.log("Hi");
```

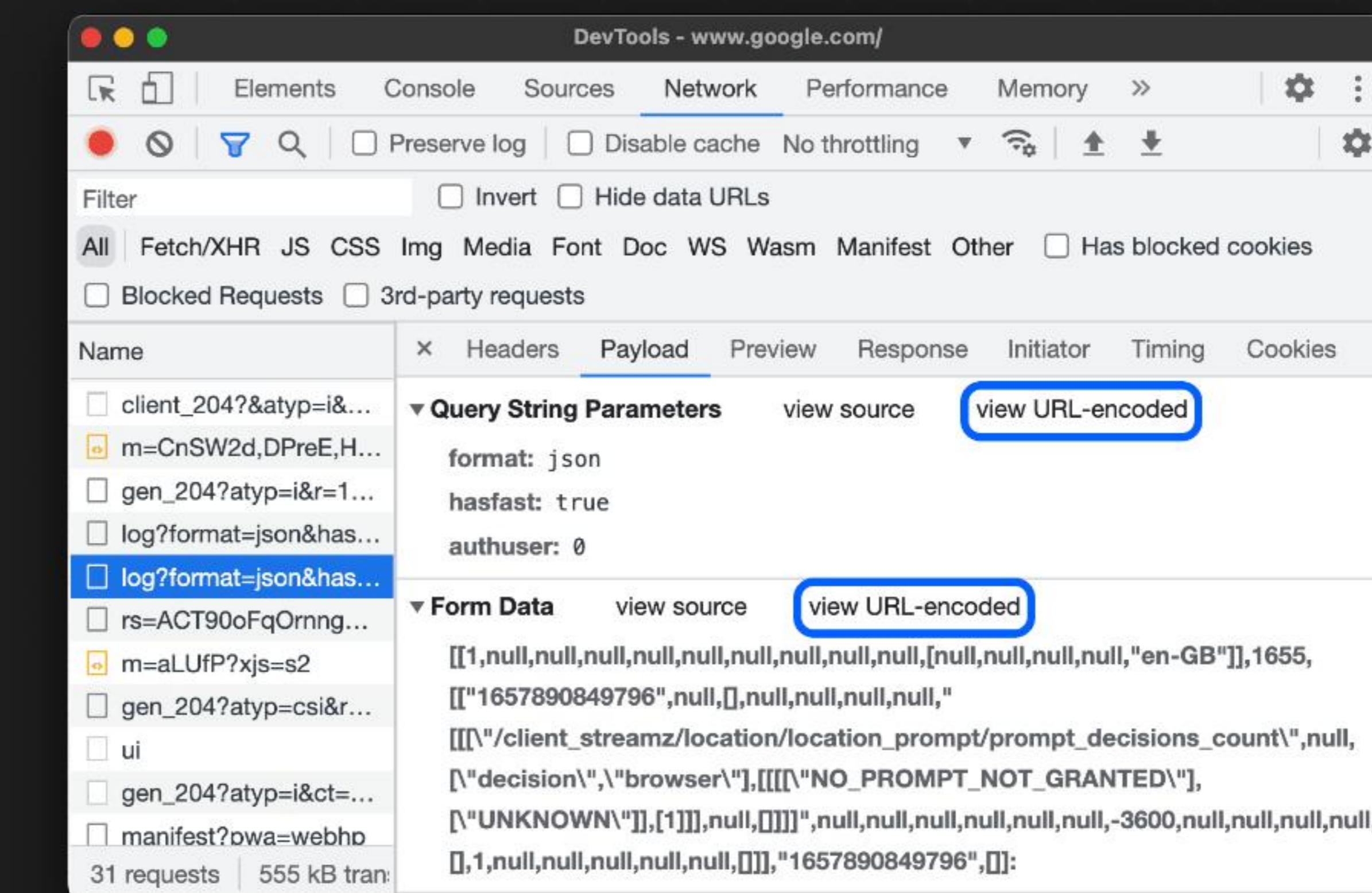
## External (แน่นำ)

แยกไฟล์ .js ต่างหาก

```
src="app.js">
```

# Console & Debugging

ใช้ `console.log()` เพื่อแสดงค่าและตรวจสอบการทำงานของ  
โปรแกรมใน Developer Tools (กด F12)



# สรุปบทเรียน

- JavaScript เป็นหัวใจสำคัญของการทำเว็บ Interactive
- เข้าใจพื้นฐาน: Variables, Types, Operators, Control Flow
- DOM Manipulation คือกุญแจสำคัญในการควบคุมหน้าเว็บ
- **Next Step:** ฝึกฝนและเริ่มสร้างโปรเจกต์จริง!

# Image Sources

---



[https://byothe.fr/wp-content/uploads/2025/09/Exercism.org\\_-696x392.jpg](https://byothe.fr/wp-content/uploads/2025/09/Exercism.org_-696x392.jpg)

Source: [byothe.fr](https://byothe.fr)

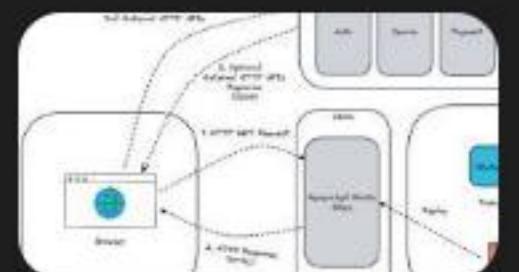
---



[https://upload.wikimedia.org/wikipedia/commons/thumb/d/d1/Brendan\\_Eich\\_Mozilla\\_Foundation\\_official\\_photo.jpg/1200px-Brendan\\_Eich\\_Mozilla\\_Foundation\\_official\\_photo.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/d/d1/Brendan_Eich_Mozilla_Foundation_official_photo.jpg/1200px-Brendan_Eich_Mozilla_Foundation_official_photo.jpg)

Source: [en.wikipedia.org](https://en.wikipedia.org)

---



<https://www.workingsoftware.dev/content/images/size/w1200/2022/12/Untitled-2022-05-21-1141.png>

Source: [workingsoftware.dev](https://workingsoftware.dev)

---



[https://mp-cdn.elgato.com/media/5377d183-f8ca-4bea-a9ca-f5b05a252441/Visual\\_Studio\\_Code\\_Mac-thumbnail-optimized-badc1ee0-a11a-49c3-89de-f0ba93f441b1.png](https://mp-cdn.elgato.com/media/5377d183-f8ca-4bea-a9ca-f5b05a252441/Visual_Studio_Code_Mac-thumbnail-optimized-badc1ee0-a11a-49c3-89de-f0ba93f441b1.png)

Source: [marketplace.elgato.com](https://marketplace.elgato.com)

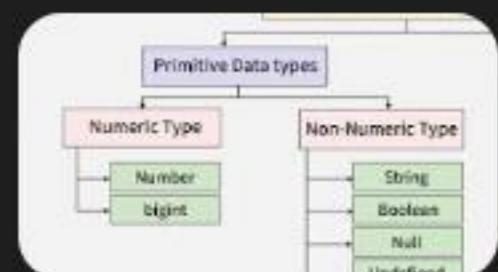
---



[https://miro.medium.com/1\\*mtsk3fQ\\_BRemFidhkel3dA.png](https://miro.medium.com/1*mtsk3fQ_BRemFidhkel3dA.png)

Source: [medium.com](https://medium.com)

---

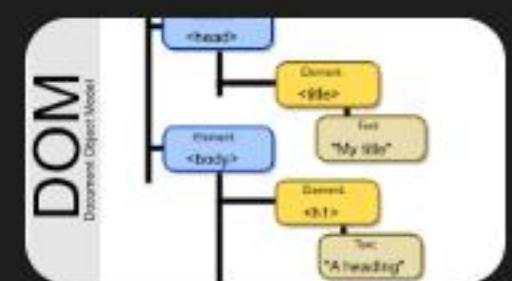


[https://media.geeksforgeeks.org/wp-content/uploads/20250728150910891820/data\\_types\\_in\\_javascript.webp](https://media.geeksforgeeks.org/wp-content/uploads/20250728150910891820/data_types_in_javascript.webp)

Source: [www.geeksforgeeks.org](https://www.geeksforgeeks.org)

# Image Sources

---



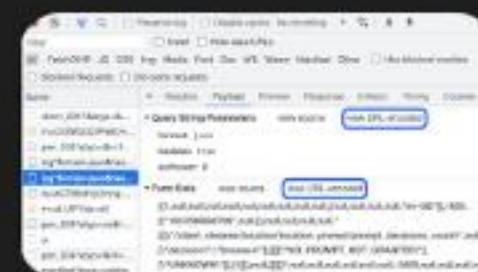
<https://upload.wikimedia.org/wikipedia/commons/5/5a/DOM-model.svg>

Source: [wikipedia.org](https://en.wikipedia.org)



<https://spaceship.roadtocode.org/javascript/15/screenshot1.png>

Source: [spaceship.roadtocode.org](https://spaceship.roadtocode.org)



<https://developer.chrome.com/static/docs/devtools/network/reference/image/toggle-url-encoding-496728c6daa9d.png>

Source: [developer.chrome.com](https://developer.chrome.com)



<https://images.unsplash.com/photo-1687603917313-ccae1a289a9d?fm=jpg&q=60&w=3000&ixlib=rb-4.1.0&ixid=M3wxMjA3fDB8MHxzZWFyY2h8Mnx8cmVhY3QlMjBqc3xlbwwwfHwwfHx8MA%3D%3D>

Source: [unsplash.com](https://unsplash.com)