



JavaScript for Beginner

Introduction to JavaScript #2

Fundamentals

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

Topic

- โครงสร้าง ของการเขียน Code Structure
- Modern mode “Use Strict”
- ตัวแปร Variable
- ประเภทข้อมูล Data Type

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>
```

โครงสร้าง Code Structure

1. <script>

2. alert('Hello'); alert('World');

3.

4. alert('Hello');

5. alert('World');

6.

7. alert('Hello')

8. alert('World')

9. </script>

```
<html>
<head>

<title>Learn JavaScript</title>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>

</head>
<body>

    <h1 id="topic">JavaScript</h1>
    <p id="demo">Content</p>
    <button onclick="changeText()">Click me</button>
    <script>
        function changeText() {
            document.getElementById("topic").innerHTML = "Hello!";
        }
    </script>

</body>
</html>
```

โครงสร้าง ตำแหน่ง การวาง script

- Head

```
<html>
<head>

<title>Learn JavaScript</title>

<script>
    function myFunction() {
        document.getElementById("demo").innerHTML = "Paragraph changed.";
    }
</script>
</head>
<body>
```

- Body

```
    <h1 id="topic">JavaScript</h1>
    <p id="demo">Content</p>
    <p id="demo1">Content</p>
    <button onclick="changeText()">Click me</button>
    <button onclick="demo1()">External</button>

    <script>
        function changeText() {
            document.getElementById("topic").innerHTML = "Hello!";
        }
    </script>
```

- External

```
    <script src="myScript1.js"></script>

</body>
</html>
```

Code Comment

```
// one line comment
```

```
alert('Hello');
```

```
/*    Multiple line  
      of comment
```

```
*/
```

```
alert('Hello');
```

The modern mode, “use strict”

ใช้ในการประกาศว่า จะใช้งาน **strict mode** ให้มีการ ประกาศ “use strict”; ไว้ ด้านบน ตั้งแต่ **ECMA Script 5** เป็นต้นมา ก็จะทำงานใน **strict mode**

```
1  "use strict";  
2  
3  // this code works the modern way  
4  ...
```

- **Check** ตัวแปร ว่ามีประกาศ ไว้ไหม
- ช่วยให้ **JavaScript optimize Code** ให้สามารถ **Run** ได้ เร็วกว่า
- ป้องกัน การ แปลง ตัวแปร ผิดพลาด, การ **assign** ค่าให้ ค่าคงที่
- กำหนดค่าให้ **object** ที่ยังไม่สร้าง

ตัวอย่าง การใช้ use strict

```
"use strict";  
x = 3.14;           // Error x not declare
```

```
"use strict";  
myFunction();  
function myFunction(){  
    y = 3.14;        // Error y not declare  
}
```

ตัวแปร Variable

- ตัวแปร คือ ชื่อของ ที่เก็บข้อมูล ที่เราจะใช้งานใน โปรแกรม หรือ ใน Script

//สร้าง ตัวแปร โดยใช้ keyword **let**

```
let message;
```

// กำหนดค่าให้ตัวแปร โดยใช้ operator =

```
let message;
```

```
message='Hello' ;
```

```
Console.log(message) ;
```

ตัวอย่าง การกำหนด ค่า ตัวแปร

```
let user = 'John', age = 25, message = 'Hello';
```

```
let user = 'John';  
let age = 25;  
let message = 'Hello';
```

```
let user = 'John',  
    age = 25,  
    message = 'Hello';
```

การกำหนด ค่าตัวแปร ด้วย var

ในยุค ก่อน จะมีการใช้ keyword **var** ในการกำหนด ค่าตัวแปร

```
var message = 'Hello';
```

ซึ่ง การใช้ **var** จะมีการทำงาน คล้ายๆ กับ **let**

แต่ **var** จะ สามารถใช้งาน ตัวแปร ในลักษณะ **global**

****** การใช้งาน **var** ก็ จะ ทำงาน ได้ แต่อาจจะพบปัญหา เกี่ยวกับ **scope** ของตัวแปร ได้
เนื่องจาก ตัวแปรที่ประกาศด้วย **var** จะสามารถเข้าถึงได้ จาก ทุก **scope**

ตัวอย่าง การใช้งาน ตัวแปร

```
let message;
```

```
message = 'Hello'; // Assign value
```

```
message = 'world'; // Change value
```

```
alert(message);
```

ตัวอย่าง การใช้งาน ตัวแปร

```
let hello = 'Hello world';
```

```
let message;
```

```
message = hello; // Copy value from hello
```

```
alert(hello);
```

```
alert(message);
```

ตัวอย่าง การใช้งาน ตัวแปร

```
let message = 'Orange';
```

// ใช้ let กำหนดค่า ตัวแปร ซ้ำกันได้

```
let message = 'Apple';
```

```
// Syntax Error: already been declared
```

การตั้งชื่อ ตัวแปร

1. ชื่อประกอบด้วย ตัวอักษร ตัวเลข หรือ สัญลักษณ์ \$ หรือ _
2. ชื่อห้ามขึ้นต้นด้วย ตัวเลข
3. ไม่แนะนำให้ใช้ ตัวอักษร ภาษาอื่นๆ ควรใช้ ภาษาอังกฤษ
4. ห้ามตั้งชื่อใช้ **Reserve words**

Reserved words (คำสงวน)

- break
- case
- catch
- class
- const
- continue
- debugger
- default
- delete
- do
- else
- export
- extends
- false
- finally
- for
- function
- if
- import
- in
- instanceof
- new
- null
- return
- super
- switch
- this
- throw
- true
- try
- typeof
- var
- void
- while
- with
- let
- const
- static
- yield
- await

ตัวอย่าง การ ตั้งชื่อ ตัวแปร

```
let userName; // camel case  
let test123;
```

```
let $ = 1; // ตัวแปร ชื่อ "$"  
let _ = 2; // ตัวแปร ชื่อ "_"
```

```
alert($ + _); // 3
```

ตัวอย่าง ที่ผิด

```
let 1a; // ไม่สามารถ เริ่มต้นด้วย ตัวเลขได้  
let my-name; // มี hyphens '-' ซึ่งไม่อนุญาต ให้ใช้เป็นชื่อตัวแปร  
let 我 = '...'; // non-english
```

Constant ค่าคงที่ (ค่าที่ไม่มีการเปลี่ยนแปลง)

```
const myBirthday = '18.04.1982';
```

```
myBirthday = '01.01.2001';
```

```
// error, can't reassign the constant!
```

Uppercase constants

```
const COLOR_RED      = "#F00";  
const COLOR_GREEN    = "#0F0";  
const COLOR_BLUE     = "#00F";  
const COLOR_ORANGE   = "#FF7F00";
```

```
// ... เมื่อต้องการเลือกสี
```

```
let color = COLOR_BLUE;  
alert(color); // #00F ได้สี น้ำเงิน
```

Data Type

- **Number** สามารถคำนวณ บวก ลบ คูณ หาร และคำนวณ ทาง คณิตศาสตร์ ได้
 - ทศนิยม floating point 1.25 , 0.256
 - จำนวนเต็ม integer -1, 0, 256, 1000
 - Infinity ค่า อนันต์
 - NaN not a number
 - BigInt สามารถเก็บค่า ที่มากกว่า 2^{53} ได้
- **String** ตัวอักษร, ข้อความต่างๆ อยู่ใน เครื่องหมาย “ab” ‘c’ quote, `d` backtick
- **Boolean** true, false
- “null”/ undefined value ค่า null เหมือน nothing, empty
- **Object, Symbol** ค่าตัวแปร ที่ซับซ้อน

Example Data type Number

```
let n = 123;  
n = 12.345;
```

```
console.log( 1 / 0 );    // Infinity  
console.log( Infinity ); // Infinity
```

```
console.log( "not a number" / 2); // NaN,  
console.log( NaN + 1 );           // NaN,  
console.log( 3 * NaN );           // NaN,
```

```
//Big int    larger than (253-1)  
console.log(9007199254740991 + 1); // 9007199254740992  
console.log(9007199254740991 + 2); // 9007199254740992  
const bigInt = 1234567890123456789012345678901234567890n;
```

Example Data type String

```
let str = "Hello";
```

```
let str2 = 'Single quotes are ok too';
```

```
let phrase = `can embed another ${str}`;
```

```
let name = "John";
```

```
// embed a variable
```

```
console.log( `Hello, ${name}!` ); // Hello, John!
```

```
console.log( `the result is ${1 + 2}` ); // result 3
```

Example Data type Boolean, null

Boolean

```
let isGrater=  
(5>4);
```

Null / Undefined

```
let age;  
let maxValue=null;
```


Object

```
let user = new Object(); // "object constructor" syntax
```

```
let user = {}; // "object literal" syntax
```

```
let user = { // an object  
  name: "John", // by key "name" store value "John"  
  age: 30 // by key "age" store value 30  
};
```

```
console.log( user.name ); // John
```

```
console.log( user.age ); // 30
```

การใช้ operator `typeof` เพื่อเช็คประเภทของข้อมูล

```
typeof undefined; // "undefined" typeof Symbol("id"); // "symbol"
```

```
typeof 0; // "number"
```

```
typeof Math; // "object" (1)
```

```
typeof 10n; // "bigint"
```

```
typeof null; // "object" (2)
```

```
typeof true; // "boolean"
```

```
typeof alert; // "function" (3)
```

```
typeof "foo"; // "string"
```



Summary

- โครงสร้าง ของการเขียน Code Structure

;

/**/ //

Case sensitive

- “use strict”

- ตัวแปร Variable

- let
- var
- const

- ประเภทข้อมูล Data Type

- Number
- String
- Boolean
- Undefined
- Null
- Object

