

# JavaScript 基礎教學

# JavaScript 是什麼？

- JavaScript 是一種腳本程式語言
- 主要用於網頁開發，使網頁具有互動性
- 與 HTML 和 CSS 並稱為網頁開發三大核心技術
- 可以直接在瀏覽器中執行

# JavaScript 引入方式

## 1. 内部 JavaScript

```
<script>  
    alert("Hello World!");  
</script>
```

## 2. 外部 JavaScript

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

# JavaScript 語法

## 1. 數據類型

```
// 單行註解
/* 多行註解 */

// 基本類型
let str = "字符串";           // 字符串
let num = 42;                 // 數字
let bool = true;              // 佈爾值
let empty = null;             // 空值
let undef;                    // 未定義 (undefined)

// 引用類型
let arr = [1, "二", true];    // 數組
let obj = {                   // 對象
  name: "小明",
  age: 25
};
```

# JavaScript 語法

## 2. 運算符

```
// 算術運算符
let sum = 5 + 3;    // 加法
let diff = 10 - 4;  // 減法
let prod = 6 * 7;   // 乘法
let quot = 20 / 5;  // 除法
let power = 2 ** 3; // 2的3次方
let remainder = 10 % 3; // 10除以3的餘數

// 比較運算符
let isEqual = 5 == 5;    // 等於
let isNotEqual = 5 != 10; // 不等於
let isGreater = 10 > 5;  // 大於
let isLess = 3 < 8;      // 小於
```

```
console.log(5 == "5"); // true (值相等)
console.log(5 === "5"); // false (值和類型都相等)

// 邏輯運算符
console.log(true && false); // false (與)
console.log(true || false); // true (或)
```

# JavaScript 語法

## 3. 函數進階

// 函數錶達式

```
const square = function(x) {  
    return x * x;  
};
```

// 箭頭函數 (ES6)

```
const cube = x => x * x * x;
```

// 立即執行函數(IIFE)

```
(function() {  
    console.log("立即執行");  
})();
```

# JavaScript 語法

## 4. 字符串操作

```
let text = "JavaScript教程";

// 模闆字符串 (ES6)
console.log(`當前主題: ${text}`);

// 常用方法
console.log(text.length);           // 9
console.log(text.substring(0,4)); // "Java"
console.log(text.includes("教程")); // true
```

# JavaScript 語法

## 5. 數組操作

```
let fruits = ["Apple", "Banana", "Orange"];
```

```
// 新增元素
```

```
fruits.push("Mango");    // 末尾添加  
fruits.pop();           // 刪除末尾元素  
fruits.shift();          // 刪除開頭元素  
fruits.unshift("Grape"); // 在開頭添加元素
```

```
// 遍曆數組
```

```
fruits.forEach(function(item, index) {  
    console.log(index, item);  
});
```

```
// 高階函數
```

```
const lengths = fruits.map(fruit => fruit.length);  
const longFruits = fruits.filter(fruit => fruit.length > 2);
```



# JavaScript 語法

## 6. 條件判斷

```
// 條件判斷
if (age >= 18) {
    console.log("成年人");
} else {
    console.log("未成年人");
}

// 迴圈
for (let i = 0; i < 5; i++) {
    console.log(i);
}
```

# JavaScript 語法

## 7. 迴圈

```
// while 迴圈
while (count < 5) {
    console.log(count);
    count++;
}
```

# JavaScript 語法

## 8. 物件

```
const person = {  
  name: "John",  
  age: 30,  
  greet: function() {  
    console.log("Hello, I'm " + this.name);  
  }  
};  
  
person.greet(); // 呼叫方法
```

# DOM 操作

```
// 選取元素
const btn = document.getElementById("myButton");
const items = document.querySelectorAll(".item");

// 事件監聽
btn.addEventListener("click", function() {
    alert("按鈕被點擊了！");
});

// 修改內容
document.getElementById("demo").innerHTML = "新內容";

// 修改樣式
document.getElementById("demo").style.color = "red";
```

# 錯誤處理

```
try {  
  // 可能出錯的代碼  
  let result = riskyOperation();  
} catch (error) {  
  // 錯誤處理  
  console.error("操作失敗:", error.message);  
} finally {  
  // 無論是否出錯都會執行  
  console.log("操作完成");  
}
```

# 異步編程

```
// Promise 示例
fetch('https://api.example.com/data')
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error('請求失敗:', error));

// Async/Await (ES8)
async function getData() {
  try {
    const response = await fetch('https://api.example.com/data');
    const data = await response.json();
    console.log(data);
  } catch (error) {
    console.error('請求失敗:', error);
  }
}
```

# JavaScript 保留關鍵字不可以用作變量、標籤或者函數名。

abstract	arguments	boolean	break	byte
case	catch	char	class*	const
continue	debugger	default	delete	do
double	else	enum*	eval	export*
extends*	false	final	finally	float
for	function	goto	if	implements
import*	in	instanceof	int	interface
let	long	native	new	null
package	private	protected	public	return
short	static	super*	switch	synchronized
this	throw	throws	transient	true
try	typeof	var	void	volatile
while	with	yield		

- 標記的關鍵字是 ECMAScript5 中新添加的。

# JavaScript 對象、屬性和方法

您也應該避免使用 JavaScript 內置的對象、屬性和方法的名稱作為 Javascript 的變量或函數名：

Array	Date	eval	function	hasOwnProperty
Infinity	isFinite	isNaN	isPrototypeOf	length
Math	NaN	name	Number	Object
prototype	String	toString	undefined	valueOf



# HTML 事件句柄

除此之外，您還應該避免使用 HTML 事件句柄的名稱作為 Javascript 的變量及函數名。

實例：

onblur	onclick	onerror	onfocus
onkeydown	onkeypress	onkeyup	onmouseover
onload	onmouseup	onmousedown	onsubmit

# 實用資源

- 菜鳥教程 JavaScript [🔗](#)
- W3Schools JavaScript 教程 [🔗](#)