



JavaScript



Buổi 1: Làm quen với JavaScript.

Nội dung buổi 1

- JavaScript là gì? JavaScript Ecosystem.
- Giá trị và biến. let, const và var.
- Kiểu dữ liệu primitive.
- Các toán tử cơ bản.
- Type Conversion and Coercion.
- Template Literals.
- Mệnh đề điều kiện if/else.
- Giá trị Truthy và Falsy

JavaScript là gì?

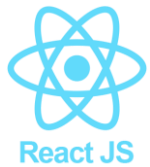


JavaScript là gì?

- Là ngôn ngữ lập trình **bậc cao, hướng đối tượng** và **đa mô hình**.
- Được tích hợp và nhúng vào HTML giúp website trở nên sống động hơn.
- Là một trong 3 ngôn ngữ chính trong lập trình web:
 - **HTML:** Cung cấp cấu trúc cơ bản, hỗ trợ trong việc xây dựng layout, thêm nội dung dễ dàng trên website.
 - **CSS:** Được sử dụng để kiểm soát và hỗ trợ việc định dạng thiết kế, bố cục, style, màu sắc,...
 - **JavaScript:** Tạo nên những nội dung “động” trên website.

JavaScript Ecosystem

Front End



Back End



Mobile Applications



Desktop Applications



Sử dụng JavaScript trong HTML

CÁCH 1:

Vị trí viết code JavaScript



```
<body>
  <p>Hello world!</p>

  <script>
    const message = "Nice to meet you all!";
    console.log(message);
  </script>
</body>
```

Chú ý: Vị trí đặt thẻ `<script>` nằm ở dòng cuối cùng trong thẻ `<body>`

Sử dụng JavaScript trong HTML

CÁCH 2:

Đính kèm đường dẫn đến file chứa code JavaScript

```
<body>
  <p>Hello world!</p>
  <script src="./script.js"></script>
</body>
```

File JavaScript có định dạng là .js

Chú ý: Vị trí đặt thẻ `<script>` nằm ở dòng cuối cùng trong thẻ `<body>`

Giá trị và biến



Giá trị và biến

- Trong ngôn ngữ lập trình nói chung và JavaScript nói riêng, mọi dữ liệu được khai báo và sử dụng trong môi trường lập trình được gọi là các giá trị (value) và khi chúng ta muốn lưu trữ lại các giá trị này thì cần phải gán chúng với một biến (variable).
- Biến là một thành phần đại diện cho giá trị mà chúng được gán để sử dụng lặp đi lặp lại hoặc đơn giản là để chia nhỏ các giá trị thành các thành phần đơn giản và dễ hiểu hơn.

Cách đặt biến

- Sử dụng let, const hoặc var trước tên biến.
- Tên biến phải bắt đầu bằng một chữ cái, dấu gạch dưới (_) hoặc biểu tượng \$.
- Không được đặt tên biến trùng với các từ khóa đã được thiết lập sẵn trong JavaScript (Khi bị trùng, JavaScript sẽ báo lỗi).
- Cách đặt tên biến thông dụng:
 - Cú pháp camel case: userData, generateString, fetchProducts...
 - Cú pháp underscore: student_list, product_quantity, total_price...

let vs. const vs. var

Sử dụng let khi cần gán lại giá trị mới cho biến



```
let thisMonth = "October";  
thisMonth = "November";  
// Giá trị cuối cùng của biến "thisMonth" là "November"
```

Sử dụng const khi giá trị gán cho biến là một giá trị không thay đổi.



```
const thisYear = 2022;  
thisYear = 2023;  
// Báo lỗi ❌
```

- Có thể sử dụng var để áp dụng cho cả 2 trường hợp trên.
- Tuy nhiên, cú pháp khai báo biến bằng var là cách viết cũ. Sử dụng trong phiên bản ES5 (2009) trở về trước. Cách viết này không tối ưu, dễ gây nhầm lẫn cho lập trình viên trong quá trình phát triển.

Primitive Value



Primitive Value

- Các kiểu giá trị primitive cơ bản: **Number, String, Boolean, Undefined, Null.**

```
const number = 2; // number
const string = "This is a string"; // string
const booleanTrue = true; // boolean
const booleanFalse = false; // boolean
const undefinedVar = undefined; // undefined
const nullVal = null; // null
```

- Kiểm tra kiểu giá trị bằng cú pháp **typeof**

```
const x = 2;
typeof x; // Trả về giá trị "number"
```

Các toán tử cơ bản

Các toán tử cơ bản

Phép toán cơ bản:

```
2 + 2; // 4
3 - 2; // 1
2 * 3; // 6
4 / 2; // 2
2 ** 3; // 8
(2 + 3) * 6 - 6; // 24
```

Cộng chuỗi:

```
"a" + "b"; // "ab"
"with" + " " + "a" + "space"; // "with a space"
```


Các toán tử cơ bản

Toán tử so sánh hơn kém: >, <, >=, <=

Toán tử so sánh bằng: == và ===

```
"1" == 1; // true  
1 == "1"; // true  
0 == false; // true  
true == 1; // true
```

```
"1" === 1; // false  
"1" === "1"; // true  
1 === 1; // true  
0 === false; // false  
true === 1; // false
```

Các toán tử cơ bản

Toán tử so sánh:

```
2 + 2; // 4
```

```
3 - 2; // 1
```

```
2 * 3; // 6
```

```
4 / 2; // 2
```

```
2 ** 3; // 8
```

```
(2 + 3) * 6 - 6; // 24
```

Bài thực hành số 1



Bài thực hành số 1

- Viết công thức tính chỉ số chiều cao cân nặng theo BMI.
 - Tạo một biến cho giá trị chiều cao.
 - Tạo một biến cho giá trị cân nặng.
 - Tạo một biến cho kết quả.
 - Log kết quả ra cửa sổ console.
 - Công thức tính BMI: cân nặng (kg) / bình phương chiều cao (m).

Type Conversion – Type Coercion

Conversion:

```
Number("23"); // 23  
String(23); // "23"  
Number("abc"); // NaN
```

Coercion:

```
"2" * 3; // 6  
3 - "1"; // 2  
"4" / "2"; // 2  
  
2 + "2"; // "22"  
"12" + 3; // "123"
```

Type Conversion – Type Coercion

Một số dạng Coercion đặc biệt:

```
"" == 0; // true  
true == 1; // true  
false == 0; // true  
null == 0; // true  
"abc" - 2; // NaN  
undefined - 2; // NaN
```

Template Literals

```
const studentName = "Minh";
const age = 24;

/**
 * Cần in ra một đoạn giới thiệu có mẫu là:
 * "Xin chào, tôi là Minh. Tôi 24 tuổi."
 */

// Sử dụng phép cộng chuỗi
const intro1 = "Xin chào, tôi là " + studentName + ". Tôi " + age + "tuổi.";

// Sử dụng Template Literals
const intro2 = `Xin chào, tôi là ${studentName}. Tôi ${age} tuổi.`;
```

Bài thực hành số 2



Bài thực hành số 2

Lệnh nào sẽ được hiển thị ra màn hình?

```
let a = 1;  
  
let b = 22;  
  
let name = "John";
```

1. `alert(b + a);`
2. `alert(c);`
3. `alert(a + name);`
4. `alert(a + name + b);`
5. `alert(a + b + name);`
6. `alert(name + a);`
7. `alert(name + a + b);`
8. `alert(name + (a + b));`

9. `alert(`Hello ${name}`);`
2. `alert(`${name} + 1`);`
3. `alert(`${name + 1}`);`
4. `alert(`${name + a}`);`
5. `alert(`a + b = ${a + b}`);`
6. `alert(`a + b = ${1 + 2}`);`
7. `alert(`1 + 2 = ${1 + 2}`);`
8. `alert(`1 + 2 = ${a + b}`);`

Bài thực hành số 2

Kết quả của những biểu thức sau?

1. `"" + 1 + 0;`

2. `"" - 1 + 0;`

3. `true + false;`

4. `6 / "3";`

5. `"2" * "3";`

6. `4 + 5 + "px";`

7. `"$" + 4 + 5;`

8. `"4" - 2;`

9. `"4px" - 2;`

10. `7 / 0;`

11. `"-9" + 5;`

12. `"-9" - 5;`

13. `null + 1;`

14. `undefined + 1;`

15. `typeof(7) + 1;`

16. `typeof "9" + 5;`

Mệnh đề điều kiện if/else

Mệnh đề điều kiện if/else

Sử dụng "if":

```
let underAge = true;
const age = 20;

if (age >= 18) {
  underAge = false;
}
console.log(underAge); // false
```

Block code
được thực thi

```
let underAge = true;
const age = 12;


if (age >= 18) {
  underAge = false;
}
console.log(underAge); // true
```

Block code
không được
thực thi

Mệnh đề điều kiện if/else

Sử dụng “if” và “else”:

```
let statement = "";  
const age = 12;  
  
if (age >= 18) {  
    statement = "Được phép thi bằng lái xe";  
} else {  
    statement = "Chưa được phép thi bằng lái xe";  
}  
console.log(statement); // "Chưa được phép thi bằng lái xe"
```



Nếu điều kiện của “if” không thỏa mãn, block code của “else” chắc chắn sẽ thực thi.

Mệnh đề điều kiện if/else

Sử dụng “if”, “else if” và “else”:

```
let statement = "";
const age = 16;

if (age >= 18) {
  statement = "Được phép thi bằng lái xe máy";
} else if (age >= 16) {
  statement = "Được phép đi xe máy điện";
} else {
  statement = "Chưa được phép lái xe gắn máy";
}

console.log(statement); // "Được phép đi xe máy điện"
```

- Câu lệnh “else if” sẽ dùng để tạo nhánh điều kiện bổ sung các trường hợp điều kiện khác.
- Nếu điều kiện phía trên chưa được thỏa mãn thì sẽ tiếp tục chạy xuống dưới và kiểm tra các nhánh điều kiện. Nếu không có nhánh điều kiện nào thỏa mãn sẽ chạy block code theo nhánh “else” (nếu có).
- Khi một nhánh điều kiện được thỏa mãn, block code của nó sẽ được thực thi và bỏ qua các nhánh điều kiện phía dưới.
- Có thể sử dụng không giới hạn nhánh điều kiện “else if”. Tuy nhiên, chỉ có thể khai báo duy nhất 1 nhánh “if” và “else”.

Mệnh đề điều kiện if/else

Phân biệt “if” và “else if”:

```
let statement = "";
const age = 20;

if (age >= 18) {
  statement = "Được phép thi bằng lái xe máy";
} else if (age >= 16) {
  statement = "Được phép đi xe máy điện";
} else {
  statement = "Chưa được phép lái xe gắn máy";
}
console.log(statement); // "Được phép thi bằng lái xe máy"
```

```
let statement = "";
const age = 20;

if (age >= 18) {
  statement = "Được phép thi bằng lái xe máy";
}

if (age >= 16) {
  statement = "Được phép đi xe máy điện";
}

console.log(statement); // "Được phép đi xe máy điện"
```

Bài thực hành số 3



Bài thực hành số 3

- Bài 1: Áp dụng mệnh đề điều kiện với bài tính chỉ số BMI và báo kết quả ra cửa sổ console về tình trạng cân nặng theo giá trị chiều cao và cân nặng đưa vào.
- Bài 2: Tạo các biến để lưu giá trị tên và tuổi của 2 người. Sử dụng mệnh đề điều kiện để so sánh tuổi của 2 người và báo kết quả ra cửa sổ console theo dạng chuỗi. Ví dụ: "Mai nhỏ hơn Tuấn 2 tuổi", "Minh bằng tuổi Quang", "Lâm lớn hơn Phong 3 tuổi".
- Bài 3: Viết điều kiện để tính tổng tiền hóa đơn của một cửa hàng ăn. Được biết với hóa đơn trong khoảng 100 - 400USD thì tiền tip tính bằng 10% hóa đơn, các trường hợp khác tiền tip tính bằng 15%. In ra cửa sổ console chi tiết của hóa đơn (Gồm tiền dịch vụ sử dụng, tip và tổng tiền cần thanh toán).

Hoàn thành JavaScript – Buổi 1

Good job! 