



Javascript cơ bản

Bùi Văn Hiên

hien@techmaster.vn

Kiểu dữ liệu boolean



Kiểu Boolean

- Giá trị của kiểu dữ liệu này chỉ có thể là **true** hoặc **false**
- **Ví dụ**
 - ❖ let a = **true**;
 - ❖ let b = **false**;
 - ❖ let c = **!a**;
 - ❖ let d = **!b**;



Xác định giá trị Boolean

- Để xác định giá trị boolean cho biến, biểu thức,... ta sử dụng hàm **Boolean()**
- Ví dụ:

```
let a = 5;  
Boolean(a);  
  
let b = 6;  
let c = 10;  
Boolean(b < c)
```



Truthy, Falsy value

- **Truthy value** là những giá trị mà khi ép kiểu về Boolean thì sẽ cho ra giá trị là **true**.
- Ngược lại, **Falsy value** là những giá trị mà khi ép kiểu về Boolean thì cho ra giá trị là **false**
- **Note:**
 - Có **6** giá trị sau được coi là **falsy** : **false, 0, NaN, '', null, undefined**
 - Các giá trị còn lại, ngoài các giá trị trên được gọi là **truthy**

Câu lệnh điều kiện



Câu lệnh if

- Thực hiện câu lệnh trong **if** nếu điều kiện cho trước là đúng.
- Cú pháp :

```
if(điều kiện) {  
    // Khôi lệnh thực thi nếu điều kiện đúng  
}
```

Ví dụ

```
let hour = 6;  
if(hour < 10) {  
    console.log('Good morning');  
}
```

Câu lệnh if/else

- Nếu **điều kiện đúng** thực hiện câu lệnh trong **if**, ngược lại nếu **điều kiện sai** thì hiện câu lệnh trong **else**.
- Cú pháp :

```
if(điều kiện) {  
    // Khôi lệnh thực thi nếu điều kiện đúng  
} else {  
    // Khôi lệnh thực thi nếu điều kiện sai  
}
```

Ví dụ

```
let hour = 14;  
if(hour < 12) {  
    console.log('Good morning');  
} else {  
    console.log('Good afternoon');  
}
```


Câu lệnh if/else if/else

- Chỉ định một điều kiện mới nếu điều kiện đầu tiên là sai.
- Cú pháp :

```
if(điều kiện 1) {  
    // Khôi lệnh thực thi nêu điều kiện đúng  
} else if(điều kiện 2) {  
    // Khôi lệnh thực thi nêu điều kiện 1 sai và điều kiện 2 đúng  
} else {  
    // Khôi lệnh thực thi nêu điều kiện 1 và điều kiện 2 sai  
}
```

Ví dụ

```
let hour = 10;  
if(hour < 12) {  
    console.log('Good morning');  
} else if(hour ≥ 12 && hour < 18) {  
    console.log('Good afternoon');  
} else {  
    console.log('Good evening');  
}
```

Bài tập if/else



Bài tập

- Viết function nhập vào biến **mark** có giá trị từ 0 -> 100. Kiểm tra giá trị của biến **mark** và in ra nội dung sau
 - "A" nếu **mark** ≥ 85
 - "B" nếu **70** \leq **mark** $<$ **85** \longleftrightarrow **(70 \leq mark && mark $<$ 85)**
 - "C" nếu **40** \leq **mark** $<$ **70** \longleftrightarrow **(40 \leq mark && mark $<$ 70)**
 - Các trường hợp còn lại in ra "D"
- Viết function truyền vào 2 số a, b. In ra màn hình số có giá trị lớn hơn



Bài tập

3. Viết function nhập vào 1 số. Kiểm tra số đó là số âm, số dương hay là số 0.
4. Viết function nhập vào 1 số. Kiểm tra số đó là số chẵn hay số lẻ.
5. Viết function nhập vào 1 số. Kiểm tra số đó có **đồng thời** chia hết cho 3 và 5 không.
6. Viết function nhập vào 3 số a, b, c. Kiểm tra xem c có bằng $a + b$ không?



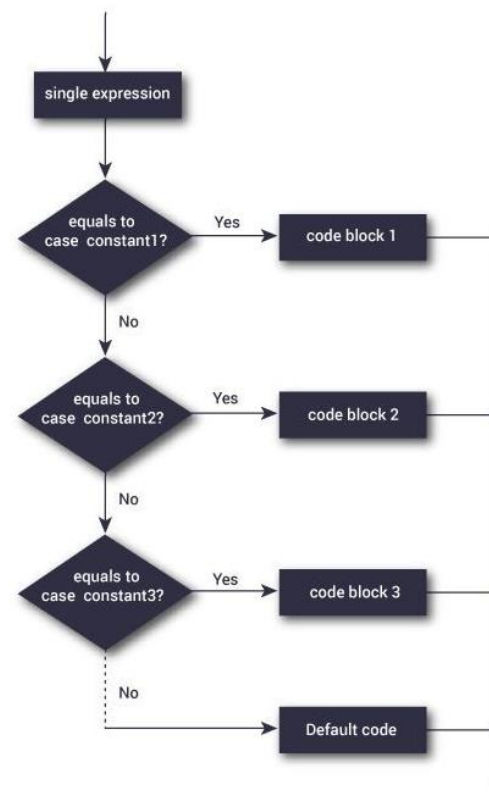
Câu lệnh **switch-case**

- Câu lệnh **switch case** có công dụng giống như lệnh **if else** đó là đều dùng để rẽ nhánh chương trình, ứng với mỗi nhánh sẽ có một điều kiện cụ thể và điều kiện đó phải sử dụng **toán tử so sánh bằng**

Cú pháp switch-case

```
switch (variable)
{
    case value_1 : {
        // do some thing
        break;
    }
    case value_2 : {
        // do some thing
        break;
    }
    default : {
        // do something
        break;
    }
}
```

Sơ đồ làm việc



Ví dụ về switch-case

- Giả sử khi đến một quán nước, ở đó có một cái menu giống như bên dưới và trong tay bạn chỉ có đúng 10.000 VND.

MENU		
Cà phê sữa	12.000đ
Cà phê đá	10.000đ
Sting dâu	8.000đ
Trà đá	2.000đ

- **Câu hỏi:** Nếu yêu cầu bạn chọn một món nước uống có giá bằng đúng với số tiền mà bạn đang có, thì món nước uống đó là món gì ?
- **Trả lời:** Cà phê đá.

Bài tập switch/case





Bài tập 1

Tạo biến **day** có giá trị từ 0 -> 6, là các ngày trong tuần

Trong đó: (**0** - chủ nhật, **1** - thứ 2, ..., **6** - Thứ 7)

- Sử dụng **switch-case** để in ra ngày tương ứng với giá trị của biến **day**
- **Ví dụ:** day = 6 → "Hôm nay là thứ 7"



Bài tập 2

- Tạo biến **month** có giá trị từ 1 -> 12, là các tháng trong năm
- Sử dụng **switch-case** để in ra mùa tương ứng với giá trị của biến **month**
 - 1, 2, 3 : Mùa xuân
 - 4, 5, 6 : Mùa hạ
 - 7, 8, 9 : Mùa thu
 - 10, 11, 12 : Mùa đông

Ví dụ: month = 9 → “Mùa thu”

Vòng lặp



Vòng lặp là gì

- Vòng lặp được sử dụng để thực thi một đoạn code lặp đi lặp lại một số lần nhất định
- Một số loại vòng lặp trong Javascript:
 - For
 - While
 - Do/while
 - ForEach
 - ...



For Loop

- Cú pháp:

```
for(bien_khoy_tao; dieu_kien_thuc_thi, buoc_nhay) {  
    // Code thực thi của từng vòng lặp  
}
```

- Trong đó:

- **bien_khoy_tao**: giá trị khởi tạo ban đầu của vòng lặp.
- **dieu_kien_thuc_thi**: điều kiện mà vòng lặp được phép chạy (true -> chạy tiếp, false -> dừng lại)
- **buoc_nhay**: là khoảng đệm nhảy của mỗi vòng lặp.



Break/Continue

- **break** được sử dụng để kết thúc vòng lặp.
- **continue** được sử dụng để bỏ qua một vòng lặp hiện tại.

Bài tập Vòng lặp





Bài tập vòng lặp

1. Tính tổng các số lẻ từ 1 -> 100
2. Tính tổng các số chia hết cho 3 và 5 trong khoảng từ 0 -> 100
3. Chạy vòng lặp từ 0 -> 20
 - Nếu là số chẵn thì in ra : **"0 là số chẵn"**
 - Nếu là số lẻ thì in ra : **"1 là số lẻ"**
 - ...

Array





Array

- **Array(mảng)** được dùng để lưu trữ nhiều phần tử
- Mỗi phần tử của mảng (*element*) có thể có các kiểu dữ liệu khác nhau
- **Khai báo mảng:** `let arr = [value1, value2, ..., valueN]`

Trong đó:

- arr - tên mảng
- value1, value2, ..., valueN : Các giá trị của mảng



Truy cập các phần tử trong mảng

- Các phần tử trong mảng được truy cập thông qua *chỉ số* (index)
- Chỉ số của mảng **bắt đầu từ 0**, kết thúc ở **arr.length - 1**
- Thông thường, chúng ta truy cập vào phần tử mảng để:
 - ❖ Lấy giá trị của phần tử mảng: **arr[index]**
 - ❖ Cập nhật giá trị cho phần tử mảng **arr[index] = new_value**

Một số phương thức của *Array*



Các phương thức thêm, xóa phần tử mảng

- ❖ `push()` : Thêm phần tử vào cuối mảng
- ❖ `unshift()`: Thêm phần tử vào đầu mảng
- ❖ `pop()`: Xóa phần tử ở cuối mảng
- ❖ `shift()`: Xóa phần tử ở đầu mảng
- ❖ `splice()`: Thêm/Xóa phần tử ở vị trí bất kỳ trong mảng



Các phương thức khác

- ❖ `concat()`: Nối nhiều mảng lại thành 1 mảng
- ❖ `slice()`: Cắt mảng thành mảng con
- ❖ `reverse()`: Đảo ngược thứ tự của các phần tử trong mảng
- ❖ `sort()`: Sắp xếp mảng
- ❖ `join()`: Chuyển từ array -> string
- ❖ ...

Duyệt mǎng





Ví dụ về duyệt mảng

- ❖ In ra giá trị của các phần tử trong mảng

```
let numbers = [2,4,6,8,10]
for(let i = 0; i < numbers.length; i++) {
    console.log(numbers[i])
}
```


Bài tập Array





Viết function thực hiện những thao tác sau

1. Kiểm tra mảng có phải là mảng tăng dần hay không
2. Kiểm tra mảng có phải là mảng giảm dần hay không
3. Kiểm tra mảng có phải là mảng số chẵn tăng dần hay không
4. Kiểm tra mảng có phải là mảng số lẻ giảm dần hay không

Bài tập về nhà





Viết function thực hiện những thao tác sau

1. Tìm số lớn nhất trong mảng
2. Tìm số nhỏ nhất trong mảng
3. Viết hàm cho phép truyền vào 1 mảng các số, kết quả trả về là 1 mảng mới với các số là số dư tương ứng khi chia mảng cũ cho 2

Vd : [4,2,5,6,2,7] => [0,0,1,0,0,1]

4. Cho 1 chuỗi, hãy viết hàm có tác dụng sao chép đó chuỗi lên 10 lần
5. Cho 1 chuỗi, hãy viết hàm có tác dụng sao chép đó chuỗi lên 10 lần, ngăn cách nhau bởi dấu gạch ngang.

Ví dụ: **repeatString('a')** => Kết quả trả về là **'a-a-a-a-a-a-a-a-a-a'**