



# Khoá học Front End

Javascript từ cơ bản đến nâng cao

---

Mentor: Trần Công Lực

April 11, 2019

Email: [luctc.developer1902@gmail.com](mailto:luctc.developer1902@gmail.com)

Facebook: <https://facebook.com/congluc1902>

# Nội dung chương 1 - Javascript Basic

1. Giới thiệu về Javascript
2. Biến và kiểu dữ liệu (Variables and Data Types)
3. Cấu trúc điều khiển, rẽ nhánh (Control flow)
4. Hàm, thủ tục (Function, Procedure)
5. Mảng và Đối tượng (Array and Object)
6. Cấu trúc lặp (For, While Loop)

# **Giới thiệu về Javascript**

---

# Javascript là gì?

- Javascript là một ngôn ngữ lập trình thông dịch. Hỗ trợ OOP, đa nền tảng và là một ngôn ngữ nhỏ và nhẹ (small and lightweight)

# Javascript là gì?

- Javascript là một ngôn ngữ lập trình thông dịch. Hỗ trợ OOP, đa nền tảng và là một ngôn ngữ nhỏ và nhẹ (small and lightweight)
- Là một trong ba kỹ năng cốt lõi của Web Development

# Javascript là gì?

- Javascript là một ngôn ngữ lập trình thông dịch. Hỗ trợ OOP, đa nền tảng và là một ngôn ngữ nhỏ và nhẹ (small and lightweight)
- Là một trong ba kỹ năng cốt lõi của Web Development
- Ngày nay Javascript xuất hiện ở nhiều nơi:

# Javascript là gì?

- Javascript là một ngôn ngữ lập trình thông dịch. Hỗ trợ OOP, đa nền tảng và là một ngôn ngữ nhỏ và nhẹ (small and lightweight)
- Là một trong ba kỹ năng cốt lõi của Web Development
- Ngày nay Javascript xuất hiện ở nhiều nơi:
  - Client-side: Javascript thuần sử dụng trong các browser

# Javascript là gì?

- Javascript là một ngôn ngữ lập trình thông dịch. Hỗ trợ OOP, đa nền tảng và là một ngôn ngữ nhỏ và nhẹ (small and lightweight)
- Là một trong ba kỹ năng cốt lõi của Web Development
- Ngày nay Javascript xuất hiện ở nhiều nơi:
  - Client-side: Javascript thuần sử dụng trong các browser
  - Server-side: Node.js



# Javascript là gì?

- Javascript là một ngôn ngữ lập trình thông dịch. Hỗ trợ OOP, đa nền tảng và là một ngôn ngữ nhỏ và nhẹ (small and lightweight)
- Là một trong ba kỹ năng cốt lõi của Web Development
- Ngày nay Javascript xuất hiện ở nhiều nơi:
  - Client-side: Javascript thuần sử dụng trong các browser
  - Server-side: Node.js
- Dùng Javascript để làm gì?

# Javascript là gì?

- Javascript là một ngôn ngữ lập trình thông dịch. Hỗ trợ OOP, đa nền tảng và là một ngôn ngữ nhỏ và nhẹ (small and lightweight)
- Là một trong ba kỹ năng cốt lõi của Web Development
- Ngày nay Javascript xuất hiện ở nhiều nơi:
  - Client-side: Javascript thuần sử dụng trong các browser
  - Server-side: Node.js
- Dùng Javascript để làm gì?
- React, Angular, Vue dựa trên 100% Javascript.

# Javascript là gì?



Nội dung



Trình bày



Hiệu ứng động

# Javascript là gì?



Nội dung



Trình bày



Hiệu ứng động

Danh từ	Tính từ	Động từ
<code>&lt;p&gt; &lt;/p&gt;</code>	<code>p { color: red; }</code>	<code>p.hide()</code>
Nghĩa là "một đoạn văn"	Nghĩa là "đoạn văn màu đỏ"	Nghĩa là "ẩn đi paragraph"

# Các phiên bản Javascript

- Javascript được phát triển cho Netscape. Netscape 2 là trình duyệt đầu tiên sử dụng Javascript
- Microsoft ra mắt JScript để cạnh tranh. JScript được thêm vào IE 3.0 và ra mắt năm 1996
- Netscape chuyển Javascript đến tổ chức tiêu chuẩn cho các hệ thống thông tin và truyền thông (ECMA International) để viết đặt tả và làm công tác chuẩn hoá.
- Từ năm 2015, ECMAScript được đặt tên theo từng năm (ECMAScript 2015 -> ES6)



# **Biến và kiểu dữ liệu (Variables and Data Types)**

---

# Kiểu dữ liệu cơ bản (Primitive Data Types)

## Kiểu dữ liệu cơ bản trong Javascript

1. **Number:** Số thực dấu chấm động, số thập phân, số nguyên.
2. **String:** Một dãy các kí tự, được dùng cho đoạn *text*.
3. **Boolean:** Kiểu dữ liệu luận lý. Có thể là *true* or *false*.
4. **Undefined:** Kiểu dữ liệu cho một biến chưa được gán giá trị.
5. **Null:** Có thể hiểu là không tồn tại, phải được gán có chủ đích.

# Kiểu dữ liệu cơ bản (Primitive Data Types)

## Kiểu dữ liệu cơ bản trong Javascript

1. **Number:** Số thực dấu chấm động, số thập phân, số nguyên.
2. **String:** Một dãy các kí tự, được dùng cho đoạn *text*.
3. **Boolean:** Kiểu dữ liệu luận lý. Có thể là *true* or *false*.
4. **Undefined:** Kiểu dữ liệu cho một biến chưa được gán giá trị.
5. **Null:** Có thể hiểu là không tồn tại, phải được gán có chủ đích.

### Lưu ý:

Javascript có kiểu dữ liệu động (**dynamic typing**):  
Các kiểu dữ liệu được tự động tiên đoán cho một biến khi gán giá trị



# Đột biến giá trị & Ép kiểu

## Ép kiểu (Type coercion)

```
var firstName = 'John';  
var age = 28;  
  
console.log(firstName + ' ' + age);  
  
.  
.  
.
```

# Đột biến giá trị & Ép kiểu

## Ép kiểu (Type coercion)

```
var firstName = 'John';  
var age = 28;  
  
console.log(firstName + ' ' + age);  
  
var job, isMarried;  
job = 'teacher';  
isMarried = false;
```

# Đột biến giá trị & Ép kiểu

## Ép kiểu (Type coercion)

```
var firstName = 'John';  
var age = 28;  
  
console.log(firstName + ' ' + age);  
  
var job, isMarried;  
job = 'teacher';  
isMarried = false;
```

## Đột biến giá trị (Variable mutation)

```
age = 'twenty eight';  
job = 'driver';  
//alert(firstName + ' is a ' + age + ' year old ' + job + '.  
    Is he married? ' + isMarried);  
//prompt('message')
```

# Toán tử gán (Assignment Operators)

**Table 1:** Các toán tử phép cán cơ bản thường dùng

Tên	Viết tắt	Ý nghĩa
Toán tử gán	$x = y$	$x = y$
Cộng xong rồi gán	$x += y$	$x = x + y$
Trừ xong rồi gán	$x -= y$	$x = x - y$
Nhân xong rồi gán	$x *= y$	$x = x * y$
Chia xong rồi gán	$x /= y$	$x = x / y$
Chia lấy dư xong rồi gán	$x \% y$	$x = x \% y$
Luỹ thừa y xong rồi gán	$x ** y$	$x = x ** y$
Cộng một xong rồi gán	$x++$	$x = x + 1$
Trừ một xong rồi gán	$x--$	$x = x - 1$

# Toán tử so sánh (Comparison and Logical Operator )

## Toán tử so sánh

So sánh các toán hạng với nhau và trả về **giá trị logic (boolean)**

>, <, >=, <=, ==, ===, !=, !==

# Toán tử so sánh (Comparison and Logical Operator )

## Toán tử so sánh

So sánh các toán hạng với nhau và trả về **giá trị logic (boolean)**

>, <, >=, <=, ==, ===, !=, !==

## Grouping

```
var number1 = 50;  
var number2 = 150;  
var average = (number1 + number2) / 2;
```

# Toán tử so sánh (Comparison and Logical Operator )

## Toán tử so sánh

So sánh các toán hạng với nhau và trả về **giá trị logic (boolean)**

>, <, >=, <=, ==, ===, !=, !==

## Grouping

```
var number1 = 50;  
var number2 = 150;  
var average = (number1 + number2) / 2;
```

## Multiple assignments

```
var x, y;  
x = y = (3 + 5) * 4 - 6;
```

# Thử thách cơ bản số 1

Mark và John muốn so sánh chỉ số BMI với nhau (Body Mass Index - Chỉ số khối cơ thể). Hãy giúp Mark và John tính toán chỉ số này bằng công thức bên dưới:

$$BMI = mass/height^2 = KhoiLuong/ChieuCao^2$$



# Thử thách cơ bản số 1

Mark và John muốn so sánh chỉ số BMI với nhau (Body Mass Index - Chỉ số khối cơ thể). Hãy giúp Mark và John tính toán chỉ số này bằng công thức bên dưới:

$$BMI = mass/height^2 = KhoiLuong/ChieuCao^2$$

1. Lưu mass và height của Mark và John vào các biến.
2. Thực hiện phép tính giá trị BMI của cả hai người.
3. Tạo một biến luận lý chứa thông tin liệu rằng BMI của Mark có lớn hơn John hay không?
4. In kết quả: "BMI của Mark lớn hơn BMI của John? true".

## **Solution - Thử thách cơ bản 01**

## **Cấu trúc điều khiển, rẽ nhánh (Control flow)**

---

# Toán tử logic cơ bản (AND, OR, NOT)

		var A	
var B	AND	TRUE	FALSE
	TRUE	TRUE	FALSE
	FALSE	FALSE	FALSE

- AND (&&) => Trả về **true** nếu tất cả các điều kiện đều đúng.
- OR (||) => Trả về **true** chỉ cần thoả mãn một trong số nhiều điều kiện.
- NOT (!) => Phủ định **true/false** value

		var A	
var B	OR	TRUE	FALSE
	TRUE	TRUE	TRUE
	FALSE	TRUE	FALSE

# Toán tử logic cơ bản (AND, OR, NOT)

		var A	
var B	AND	TRUE	FALSE
	TRUE	TRUE	FALSE
	FALSE	FALSE	FALSE

		var A	
var B	OR	TRUE	FALSE
	TRUE	TRUE	TRUE
	FALSE	TRUE	FALSE

- AND (&&) => Trả về **true** nếu tất cả các điều kiện đều đúng.
- OR (||) => Trả về **true** chỉ cần thoả mãn một trong số nhiều điều kiện.
- NOT (!) => Phủ định **true/false** value

```
var age = 16;  
age >= 20; // => false  
age < 30; // => true  
!(age < 30); // false  
  
age >= 20 && age < 30; // =>  
age >= 20 || age < 30; // =>
```

# Câu lệnh If/Else (If/Else Statement)

- Điều kiện là biểu thức luận lý, trả về **true** hoặc **false**.
- **Nếu** trời mưa **thì** tôi sẽ đi học.
- ...

## Câu lệnh if/else đầy đủ

```
if(condition1?) {  
    // TO DO ...  
} else if (condition2?) {  
    // TO DO ... condition1 == false  
} else {  
    // TO DO ... condition1 == false && condition2 == false  
}
```

# Câu lệnh Switch (Switch Statement)

- Về cơ bản giống if/else, cũng là câu điều kiện.
- Biểu thức điều kiện chỉ được đánh giá một lần.
- Giá trị của biểu thức được so sánh với giá trị trong các trường hợp.
- Nếu bằng nhau, khối lệnh trong *block* được thực thi.

## Câu lệnh switch đầy đủ

```
switch(expression) {  
    case x:  
        // code block  
        break;  
    case y:  
        // code block  
        break;  
    default:  
        // code block  
}
```

# Truthy và Falsy trong Javascript

- Thông thường trong Javascript các biểu thức so sánh sẽ trả về kiểu luận lý (**Boolean**) là **true** hoặc **false**



# Truthy và Falsy trong Javascript

- Thông thường trong Javascript các biểu thức so sánh sẽ trả về kiểu luận lý (**Boolean**) là **true** hoặc **false**

## Tình huống gặp phải

```
var varA;  
// TO DO ...  
if (varA) {  
    // Condition varA ???  
}
```

# Truthy và Falsy trong Javascript

## Falsy

Là những giá trị khi ép kiểu về **Boolean**, bản thân nó sẽ trả về giá trị là false khi dùng trong các biểu thức điều kiện.

**Null, Undefined, NaN, 0, "", false**

## Truthy

Là những giá trị khi ép kiểu về **Boolean**, bản thân nó sẽ trả về giá trị là true khi dùng trong các biểu thức điều kiện.

**Ngoài các giá trị Falsy**

# Truthy và Falsy trong Javascript

## Falsy

Là những giá trị khi ép kiểu về **Boolean**, bản thân nó sẽ trả về giá trị là false khi dùng trong các biểu thức điều kiện.

**Null, Undefined, NaN, 0, "", false**

## Truthy

Là những giá trị khi ép kiểu về **Boolean**, bản thân nó sẽ trả về giá trị là true khi dùng trong các biểu thức điều kiện.

**Ngoài các giá trị Falsy**

So sánh '==' và '==='?

## Thử thách cơ bản số 2

John, Mark và Mary cùng chơi bóng rổ trong ba đội khác nhau. Cuối trận, điểm số đội của John lần lượt là **89**, **120** và **103**. Điểm số đội của Mark lần lượt là **116**, **94** và **123**. Trong khi đó điểm số đội của Mark lần lượt là **97**, **134** và **105**.

## Thử thách cơ bản số 2

John, Mark và Mary cùng chơi bóng rổ trong ba đội khác nhau. Cuối trận, điểm số đội của John lần lượt là **89**, **120** và **103**. Điểm số đội của Mark lần lượt là **116**, **94** và **123**. Trong khi đó điểm số đội của Mark lần lượt là **97**, **134** và **105**.

1. Tính điểm trung bình của mỗi đội.
2. Đội thắng là đội có điểm trung bình cao nhất. In ra màn hình console đội thắng cuộc và số điểm trung bình của đội đó.
3. Đừng quên kiểm tra các đội hoà nhau.

## **Solution - Thử thách cơ bản 02**

## Hàm, thủ tục (Function, Procedure)

---

# Hàm, thủ tục là gì?

- **Hàm, Thủ tục** dùng để gom một nhóm đoạn code phục vụ cho một chức năng nào đó. Được sử dụng lại và gọi ở nhiều nơi khác nhau với cùng một thao tác giống nhau.



# Hàm, thủ tục là gì?

- **Hàm, Thủ tục** dùng để gom một nhóm đoạn code phục vụ cho một chức năng nào đó. Được sử dụng lại và gọi ở nhiều nơi khác nhau với cùng một thao tác giống nhau.

=> Tránh phải copy liên tục những đoạn code giống nhau về mặt logic.

# Hàm, thủ tục là gì?

- **Hàm, Thủ tục** dùng để gom một nhóm đoạn code phục vụ cho một chức năng nào đó. Được sử dụng lại và gọi ở nhiều nơi khác nhau với cùng một thao tác giống nhau.

⇒> Tránh phải copy liên tục những đoạn code giống nhau về mặt logic.

- Một nhóm chức năng thực hiện các thao tác tính toán, xử lý logic và sau đó có trả về (**return**) giá trị, gọi là **hàm**.
- Một nhóm chức năng chỉ thực hiện các thao tác tính toán, xử lý logic. Không trả về giá trị gì cả, gọi là **thủ tục**.

# Khai báo hàm, thủ tục

```
// Cách 1
function calAge(year) {
    return ...
}

// Cách 2
var calAge2 = function(year) {
    return ...
}

// Cách 3 => Arrow function
var calAge3 = (year) => {
    return ...
}
```

# Mảng và Đối tượng (Array and Object)

---

# Mảng (Array) trong Javascript

## Tình huống

Một lớp học có 50 học sinh. Một hệ thống cần phải lưu trữ thông tin tên của 50 học sinh đó.

# Mảng (Array) trong Javascript

## Tình huống

Một lớp học có 50 học sinh. Một hệ thống cần phải lưu trữ thông tin tên của 50 học sinh đó.

## Hiện thực code

```
var student_1 = 'Nguyen Van A';  
var student_2 = 'Nguyen Van C';  
var student_3 = 'Nguyen Van D';  
...  
var student_50 = 'Nguyen Van Z';
```

# Mảng (Array) trong Javascript

## Tình huống

Một lớp học có 50 học sinh. Một hệ thống cần phải lưu trữ thông tin tên của 50 học sinh đó.

## Hiện thực code

```
var student_1 = 'Nguyen Van A';  
var student_2 = 'Nguyen Van C';  
var student_3 = 'Nguyen Van D';  
...  
var student_50 = 'Nguyen Van Z';
```

=> Không tối ưu về mặt mã nguồn. Xử lý logic phức tạp, code dài.

# Khai báo Array

Một Mảng sẽ có nhiều phần tử. Bao gồm hai thông tin cần phải lưu ý là giá trị (value) và chỉ số (index) của mỗi phần tử trong một mảng. Chỉ số (index) của mảng bắt đầu từ "0".

## Cách 1

```
var arrStudents = ['NV A', 'NV B', 'NV C', 'NV D', 'NV D'];  
  
console.log(arrStudents[0]) // NV A  
console.log(arrStudents[4]) // NV D  
console.log(arrStudents.length) // 5  
console.log(arrStudents[ arrStudents.length - 1 ]) // NV D
```

## Cách 2

```
var arrStudents = new Array('NV A', 'NV B', 'NV C', 'NV D');
```



# Các phương thức cơ bản (Array Methods)

## Methods

```
var arrStudents = ['NV A', 'NV B', 'NV C', 'NV D', 'NV D'];

// arrStudents.push(...items: any[]): number;
// arrStudents.unshift(...items: any[]): number;

// arrStudents.pop(): any;
// arrStudents.shift(): any;

// arrStudents.indexOf(searchElement: any): number;
// arrStudents.lastIndexOf(searchElement: any): number;
```

# Các phương thức cơ bản (Array Methods)

## Methods

```
var arrStudents = ['NV A', 'NV B', 'NV C', 'NV D', 'NV D'];

// arrStudents.push(...items: any[]): number;
// arrStudents.unshift(...items: any[]): number;

// arrStudents.pop(): any;
// arrStudents.shift(): any;

// arrStudents.indexOf(searchElement: any): number;
// arrStudents.lastIndexOf(searchElement: any): number;
```

Truy xuất vào một phần tử có *index* không hợp lệ?

## Thử thách cơ bản số 3

John và gia đình của anh ta đi nghỉ dưỡng tại Phan Thiết và đã đến thưởng thức tại 3 nhà hàng khác nhau. Hoá đơn cho mỗi nhà hàng lần lượt là \$124, \$48 và \$268.

Vì đây là nhà hàng 5 sao nên để giúp John trả tiền tip hợp lí cho nhân viên theo quy luật như sau. Tip là **20%** giá trị hoá đơn khi hoá đơn nhỏ hơn \$50. Là **15%** khi hoá đơn nằm trong khoảng \$50 đến \$200, và bằng **10%** nếu giá trị hoá đơn lớn hơn \$200.

## Thử thách cơ bản số 3

John và gia đình của anh ta đi nghỉ dưỡng tại Phan Thiết và đã đến thưởng thức tại 3 nhà hàng khác nhau. Hoá đơn cho mỗi nhà hàng lần lượt là \$124, \$48 và \$268.

Vì đây là nhà hàng 5 sao nên để giúp John trả tiền tip hợp lí cho nhân viên theo quy luật như sau. Tip là **20%** giá trị hoá đơn khi hoá đơn nhỏ hơn \$50. Là **15%** khi hoá đơn nằm trong khoảng \$50 đến \$200, và bằng **10%** nếu giá trị hoá đơn lớn hơn \$200.

1. Tạo một mảng chứa thông tin của **bills**.
2. Một *function* thực hiện chức năng tính tiền *tip*.
3. Kết quả thu được một hai **array** tiền tip và tổng tiền phải trả.

## **Solution - Thử thách cơ bản 03**

# Đối tượng (Object) trong Javascript

## Tình huống

Một lớp học có 50 học sinh. Một hệ thống cần phải lưu trữ thông tin bao gồm (Họ và tên, Năm sinh, Điểm thi Toán, Lý, Hoá, Điểm trung bình) của 50 học sinh đó.

# Đối tượng (Object) trong Javascript

## Tình huống

Một lớp học có 50 học sinh. Một hệ thống cần phải lưu trữ thông tin bao gồm (Họ và tên, Năm sinh, Điểm thi Toán, Lý, Hoá, Điểm trung bình) của 50 học sinh đó.

```
var arrName      = ['NV A', 'NV B', 'NV C'];  
var arrYear      = [1990, 1991, 1990];  
var arrMaths     = [7.5, 10, 9.2];  
var arrPhysics   = [1.0, 2, 3.3];  
var arrChemistry = [9.0, 7, 5];  
var arrAverage   = [??, ??, ??];  
// ...
```

# Đối tượng (Object) trong Javascript

- Trong hình hướng trên, một học sinh là một *đối tượng* (**object**).
- **Object** này có những *thuộc tính* (**properties**) như là họ tên, năm sinh, điểm thi.
- **Object** học sinh có *phương thức* (**method**) là tính điểm trung bình, phụ thuộc vào những giá trị lưu trong **properties** điểm toán, lý, hoá.
- **Method** có thể hiểu là *function* gắn liền với từng **object**.



# Đối tượng (Object) trong Javascript

Một **Object** có nhiều (**properties**) và (**methods**).

Hiểu đơn giản, **object** gom nhóm nhiều variables lại mô tả đặc tính riêng của đối tượng đó.

## Cách 1

```
var student = {  
  fullName: 'Nguyen Van A',  
  year: 1990,  
  score: {  
    maths: 9,  
    physics: 7.5,  
    chemistry: 5  
  },  
  calAverage: function() {  
    // TO DO ...  
    return ...;  
  }  
}
```

# Đối tượng (Object) trong Javascript

Một **Object** có nhiều (**properties**) và (**methods**).

Hiểu đơn giản, **object** gom nhóm nhiều variables lại mô tả đặc tính riêng của đối tượng đó.

## Cách 2

```
var student = new Object();
    student.fullName = 'Nguyen Van A';
    student.year = 1990;
    student.score = {
        maths: 9,
        physics: 7.5,
        chemistry: 5
    };
    student.calAverage = function() {
        // TO DO ...
        return ...;
    }
```

## Thử thách cơ bản số 4

Mark và John muốn so sánh chỉ số BMI với nhau (Body Mass Index - Chỉ số khối cơ thể). Hãy giúp Mark và John tính toán chỉ số này bằng công thức bên dưới:

$$BMI = mass/height^2 = KhoiLuong/ChieuCao^2$$

## Thử thách cơ bản số 4

Mark và John muốn so sánh chỉ số BMI với nhau (Body Mass Index - Chỉ số khối cơ thể). Hãy giúp Mark và John tính toán chỉ số này bằng công thức bên dưới:

$$BMI = mass/height^2 = KhoiLuong/ChieuCao^2$$

1. Ứng dụng kiến thức object để lưu thông tin John và Mark.
2. Mỗi Object đều cần phải tính chỉ số BMI -> Tạo một phương thức tính toán và trả về chỉ số BMI, lưu giá trị này cho mỗi Object
3. Hiển thị kết quả ra màn hình xem ai có chỉ số cao hơn.

## **Solution - Thử thách cơ bản 04**

## Cấu trúc lặp (For, While Loop)

---

# Vòng lặp (Loop)

- Vòng lặp dùng để thực thi một đoạn code nhiều lần liên tiếp.

# Vòng lặp (Loop)

- Vòng lặp dùng để thực thi một đoạn code nhiều lần liên tiếp.
- Lặp 50 lần tính toán điểm trung bình của 50 học sinh.
- Lặp 15 lần hiển thị 15 bài viết lên một trang tin tức.
- ...

## Các dạng vòng lặp trong Javascript

- for
- for / in
- while
- do / while



# Vòng lặp for

## For loop

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```

- **Statement 1:** Thực thi một lần **trước khi** chạy đoạn code bên trong (code block).
- **Statement 2:** Định nghĩa điều kiện để có thể chạy vòng lặp tiếp theo.
- **Statement 3:** Được thực thi **sau khi** toàn bộ đoạn code bên trong hoàn tất.

# Các biến thể của vòng for

## Biến thể dựa vào ý nghĩa của mỗi Statement

```
for(var i = 1; i <= 10; i++) {  
    console.log("i = ", i);  
}
```

```
var i = 1;  
for( ; i <= 10; i++) {  
    console.log("i = ", i);  
}
```

```
var i = 1;  
for( ; i <= 10 ; ) {  
    console.log("i = ", i);  
    i++; // i = i + 1  
}
```

# Vòng lặp for/in

## For/in loop

```
// Object Student
var student = {
  fullName: 'Nguyen Van A',
  year: 1992,
  maths: 9.0,
  physics: 7.5,
  chemistry: 5.4
}

for(key in student) {
  console.log("key = ", key, " | value = ", student[key]);
}
```

# Vòng lặp While, Do/While

## Cú pháp (syntax)

```
while (condition) {  
    // code block to be executed  
}  
/////////  
do {  
    // code block to be executed  
}  
while (condition);
```

- While chỉ thực thi khi điều kiện **condition** trả về **true** => Kiểm tra trước làm sau.
- Do/While luôn luôn thực thi đoạn code lần đầu tiên, sau đó mới kiểm tra điều kiện của những vòng lặp tiếp theo => Làm trước kiểm tra sau.

## Thử thách cơ bản số 5

John và gia đình của anh ta đi nghỉ dưỡng tại Phan Thiết và đã đến thưởng thức tại 3 nhà hàng khác nhau. Hoá đơn cho mỗi nhà hàng lần lượt là \$124, \$48 và \$268.

Vì đây là nhà hàng 5 sao nên để giúp John trả tiền tip hợp lí cho nhân viên theo quy luật như sau. Tip là **20%** giá trị hoá đơn khi hoá đơn nhỏ hơn \$50. Là **15%** khi hoá đơn nằm trong khoảng \$50 đến \$200, và bằng **10%** nếu giá trị hoá đơn lớn hơn \$200.

## Thử thách cơ bản số 5

John và gia đình của anh ta đi nghỉ dưỡng tại Phan Thiết và đã đến thưởng thức tại 3 nhà hàng khác nhau. Hoá đơn cho mỗi nhà hàng lần lượt là \$124, \$48 và \$268.

Vì đây là nhà hàng 5 sao nên để giúp John trả tiền tip hợp lí cho nhân viên theo quy luật như sau. Tip là **20%** giá trị hoá đơn khi hoá đơn nhỏ hơn \$50. Là **15%** khi hoá đơn nằm trong khoảng \$50 đến \$200, và bằng **10%** nếu giá trị hoá đơn lớn hơn \$200.

1. Hiện thực lại thử thách số 3, ứng dụng tổng hợp Array, Object và Loop.
2. Tăng số lượng hoá đơn lên **n** phần tử.
3. Tạo một hàm random dữ liệu hoá đơn, truyền vào tham số **n**. Trả về một mảng random.

## **Solution - Thử thách cơ bản 05**



W3schools.com

**Nền tảng Website giáo dục cho việc học các công nghệ Web**

Website: <https://www.w3schools.com/js/default.asp>



Jonas Schmedtmann

**The Complete JavaScript Course 2019: Build Real Projects!**

Website: <https://www.udemy.com/the-complete-javascript-course/>