

JavaScript



CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Giới thiệu JAVASCRIPT

- LÀM SAO TƯƠNG TÁC NÚT ĐỂ GỞI DỮ LIỆU ?
- LÀM SAO THAY ĐỔI DỮ LIỆU MÀN HÌNH, NỘI DUNG,... ĐỘNG ?
- LÀM SAO THAY ĐỔI MÀU SẮC CÁC THÀNH PHẦN KHI NHẤN NÚT HOẶC TƯƠNG ỨNG XỬ LÝ TỪ BACKEND VỀ ?

JAVASCRIPT GIÚP TRẢ LỜI CÁC CÂU HỎI TRÊN

➡ TƯƠNG TÁC VỚI THÀNH PHẦN GIAO DIỆN

Giới thiệu JAVASCRIPT

- ❖ Javascript là ngôn ngữ lập trình **thông dịch phía client** (client-side) có khả năng hướng đối tượng.
- ❖ Javascript xuất hiện đầu tiên trên Netscape 2.0 năm 1995 với tên là LiveScript.
- ❖ Javascript có thể:
 - Thay đổi nội dung trang web.
 - Thay đổi thuộc tính các thành phần HTML.
 - Thay đổi css của các thành phần HTML.
 - Kiểm tra (validate) tính hợp lệ dữ liệu vào của người



JAVASCRIPT làm được những gì?

FRONT END

BACK END

FULLSTACK

MOBILE

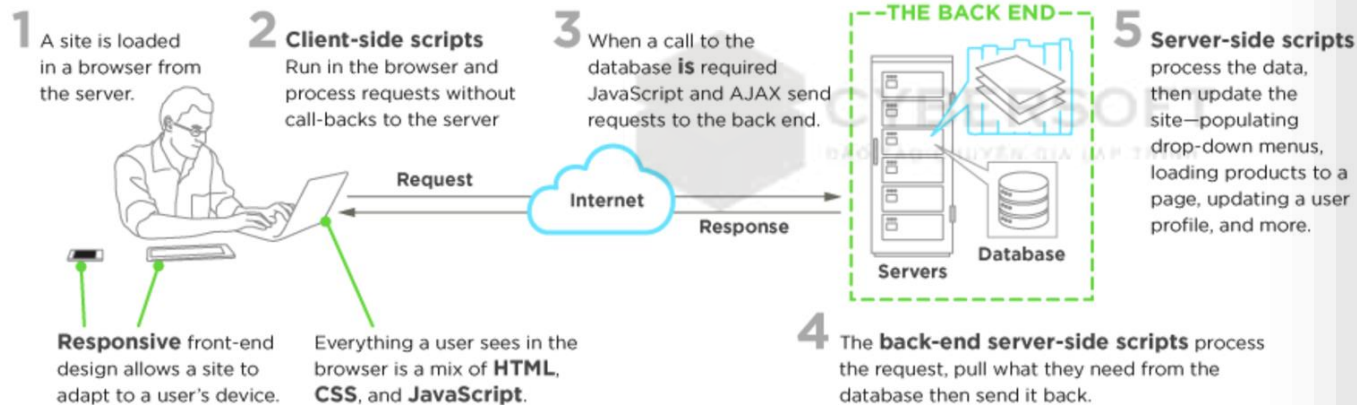
NHIỀU PHẾT NHỈ :) → HỌC NGAY THÔI



CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



JAVASCRIPT làm được những gì?



FRONT END



Kiểm tra dữ liệu người dùng nhập vào



Lấy dữ liệu người dùng nhập vào và gửi xuống BackEnd



Lấy thông tin từ BackEnd, phân tích và hiển thị lên các thành phần giao diện.



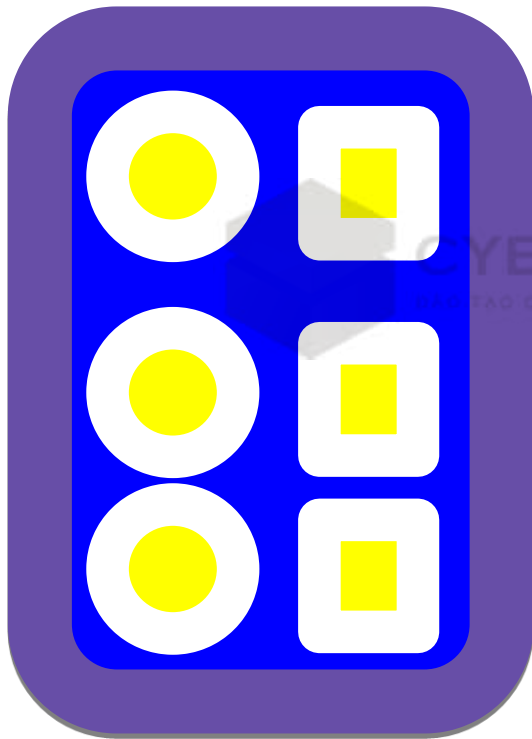
ReactJS, Angular, VueJS,...



Chia tải xử lý cho Server

JAVASCRIPT làm được những gì?

MOBILE



<https://cybersoft.edu.vn>

MOBILE

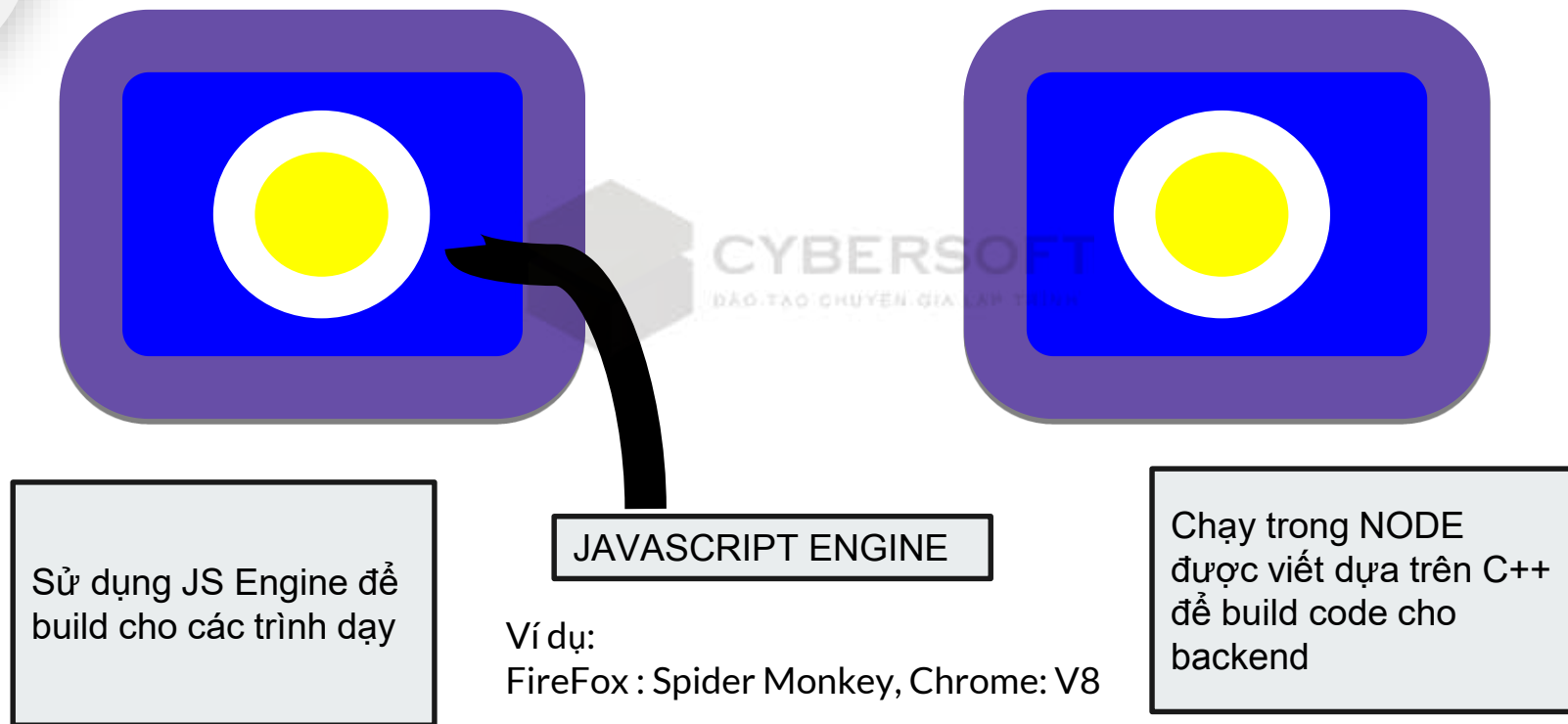


React Native, Flutter



Xây dựng và hiển thị
các component cho app

JAVASCRIPT chạy ở đâu, như thế nào?



Javascript & ECMAScript ? Phiên bản JS

ECMAScript

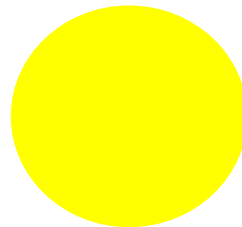
Bảng đặc tả - chuẩn
(Specification)

JavaScript

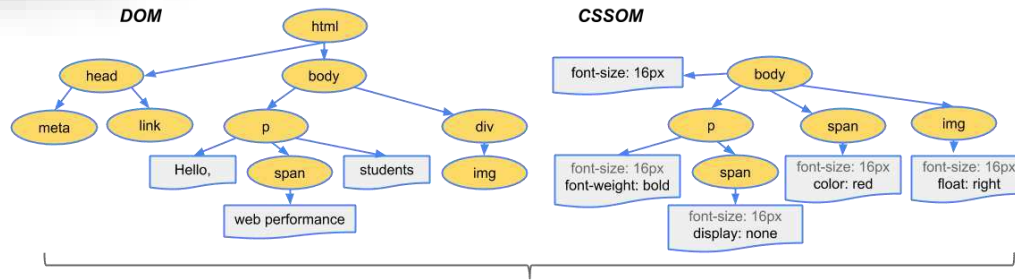
Ngôn ngữ lập trình
(Programming Language)



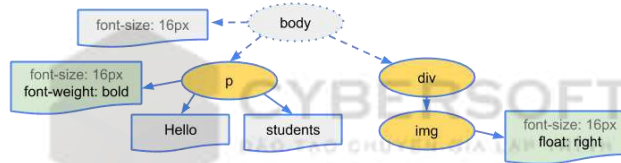
CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



Quá trình thực thi hiển thị website



Render Tree



Chèn Javascript vào tài liệu HTML

- ★ Quá trình đọc và hiển thị trang HTML như slide trước → Sẽ phân tích HTML → Phân tích CSS → Nếu gặp Javascript → Dừng để đọc → **nếu script OK chạy tiếp (KHÔNG OK THÌ SAO?)** → Hiển thị lên Trình duyệt.

★ CHÈN JS VÀO TRANG

- Sử dụng TAB `<script></script>`
- Chèn thẳng vào trang
- Code từ file bên ngoài
- Chèn bất kì nơi đâu

```
<html>
  <head>
    <script type="text/javascript">
      some statements
    </script>
  </head>
  <body>
    <script type="text/javascript">
      some statements
    </script>
    <script src="Tên_file_script.js"> </script>
    <script type="text/javascript">
      // gọi thực hiện các phương thức được định nghĩa
      // trong "Tên_file_script.js"
    </script>
  </body>
</html>
```

Best practice khi sử dụng Javascript

!! Nhớ : Code JS ở ngoài và include vào khi cần → Tái sử dụng & SEO tốt hơn

!! Đặt các include ở cuối thẻ Body → **Tại sao?** Xem lại quá trình Render HTML → Nếu đặt đầu → **JS lỗi** → **Trang lỗi luôn** → **Trắng béc** → **TIÊU**

Nhóm kiến thức cần nắm

1. Biến số
2. Lệnh gán
3. Hằng số
4. Kiểu dữ liệu

5. Lệnh điều kiện
6. Vòng lặp
7. Mảng
8. Hàm

Ví dụ đầu tiên - Vị trí chèn JS & Console.log



```
index.html • JS buoi1.js •
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8" />
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <title>CyberSoft test buoi 1</title>
8   <meta name="viewport" content="width=device-width, initial-scale=1">
9   <link rel="stylesheet" type="text/css" media="screen" href="main.css" />
10 </head>
11
12 <body>
13   <!--Code trên này -->
14
15   <!--Chèn JS cuối body nhé-->
16   <script src="buoi1.js"></script>
17
18 </body>
19
20 </html>
```

**NHỚ CHÈN
CUỐI THẺ <BODY>
NHÉ NGƯỜI ƠI**

Mở lên làm ngay nào các buổi ơ :

- 1) Tạo Folder JS-Buoi1
- 2) Tạo trang index.html
- 3) Tạo buoi1.js
- 4) Nhúng vào trang index.html



Console.log Thần thánh

Code nào....

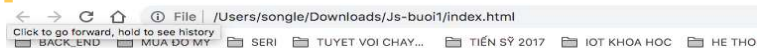
- 1) Mở file buoi1.js và gõ
 - `Console.log('HelloWorld');`
 - `Console.log('Bưởi ơi');`
- 2) Chạy trang index.html
- 3) Mở tab Console → Xem sao ?
- 4) Gõ thêm `Console.log('chuối nè');` vào Console



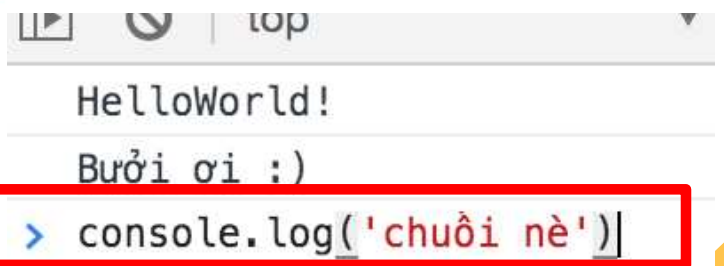
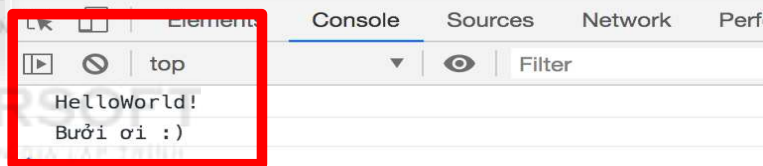
- Hiện thị trong Tab Console của trình duyệt
- Giúp gỡ rối (Debug), in kết quả của một biến, biểu thức mà không xuất hiện ra cho người dùng
- Có thể code trên file JS hoặc gõ trực tiếp vào



NHỚ XÓA KHI DEBUG XONG NHÉ → ĐỂ NHIỀU MẮT VIỆC ĐÂY



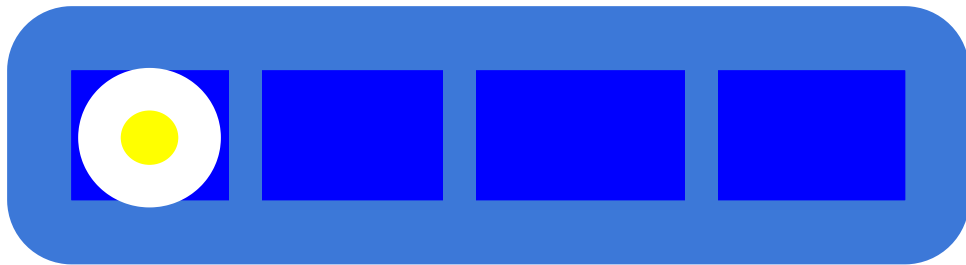
Console.log(...) Không hiển thị lên đây gì nhé



Biến - Variable

- Javascript sử dụng từ khóa var để khai báo biến.
- Sử dụng ký hiệu “=” để gán giá trị cho biến.
- Khi khai báo biến không cần khai báo kiểu của biến.
- Biến trong javascript có thể giữ giá trị của bất cứ kiểu dữ liệu nào (untyped language).

Memory - Bộ nhớ



Tên biến -
Variable name

Biến như những chiếc hộp,
giống sao ta ????

- Nhãn (label)
- Chứa đồ (value)



Biến - Variable



index.html

JS buoi1.js



```
1
2 //Khai báo biến nhé cả nhà
3
4 var name = 'CyberSoft'; //Hợp lệ
5 var address = '376 Võ Văn Tần';
6 var numberStudents = 1800; // Camel case – Đặt tên kiểu lạc đà
7
8 //Các khai báo sao không hợp lệ
9 //var if ; --> B0000M
10 //var else; --> B0000M
11 /*-->>> Tên không được đặt trùng
12 |      với từ khóa của ngôn ngữ lập trình */
```


Biến - Một số lưu ý

- Tên biến không được trùng với các từ khóa
- Tên phải có ý nghĩa : **firstName**, **lastName**,....
- Không bắt đầu bằng số (Ví dụ: **2name**)
- Không chứa khoảng trắng hoặc gạch giữa (-)
- Không nên khai báo nhiều biến cùng một dòng
- Tên biến phân biệt chữ hoa, thường.

Hằng số - Constant

1. Mở Editor lên và gõ như bên
2. Sửa **var** → **const**
3. Chạy lại và xem console ?



HẰNG SỐ ĐỂ LÀM GÌ?

- Định nghĩa 1 lần và xài nhiều lần
- Khi sửa chỉ sửa 1 chỗ duy nhất
- Chỉ được gán 1 lần duy nhất

<https://cybersoft.edu.vn>

<https://cybersoft.edu.vn>

The screenshot shows a web browser with a code editor open. The editor has two tabs: 'index.html' and 'boui1.js'. The 'boui1.js' tab is active, showing the following code:

```
1 //CyberSoft.edu.vn Demo
2
3 var PI = 3.14;
4 PI = 3.1414;
5 console.log(PI);
```

Below the code editor, the browser's developer tools are open, showing the 'Console' tab. The console displays the output '3.1414' from the log statement in the code. The source of the log is identified as 'boui1.js:5'.

Kiểu dữ liệu

Kiểu giá trị (Primitives/ Value Type)

Kiểu tham chiếu (Reference)

String

Number

Boolean

Undefined

```
index.html JS buoi1.js
2
3 var name = "CybersSoft"; //String
4 var age = "8"; // Number
5 var isLogin = false; //boolean
6 var firstName; // Chưa được gán giá trị nên undefined
```

Object

Array

Function

Sẽ demo sau nhé :)

Câu lệnh điều kiện & Toán tử điều kiện (conditions)

Các bài toán hay yêu cầu nghiệp vụ có những thứ sau:

- Xét điều kiện
- Các trường hợp
- Nếu này nọ nếu này kia

Biểu thức điều kiện - A , B : true | false

- $A \ \&\& \ B \rightarrow$ trả về **true** khi cả 2 đều **true**, còn lại **false**
- $A \ || \ B \rightarrow$ trả về **false** khi cả 2 đều **false**, còn lại **true**
- $!A \rightarrow$ ngược lại của A (phủ định của A)
- $A == B$ (So sánh không cần kiểu dữ liệu, ví dụ "0" == 0 \rightarrow true)
- $A === B$ (So sánh cả kiểu dữ liệu và giá trị, ví dụ "0" === 0 \rightarrow false \rightarrow **KHUYẾN DÙNG NHÉ**)

- ```
if (<biểu thức điều kiện>) {
 // các dòng lệnh
}
```
- ```
if (<biểu thức điều kiện>) {  
    // các dòng lệnh  
} else {  
    // các dòng lệnh  
}
```
- ```
switch (<lựa chọn/trường hợp>) {
 case TH1:
 // các dòng lệnh
 break;
 case TH2:
 // các dòng lệnh
 break;

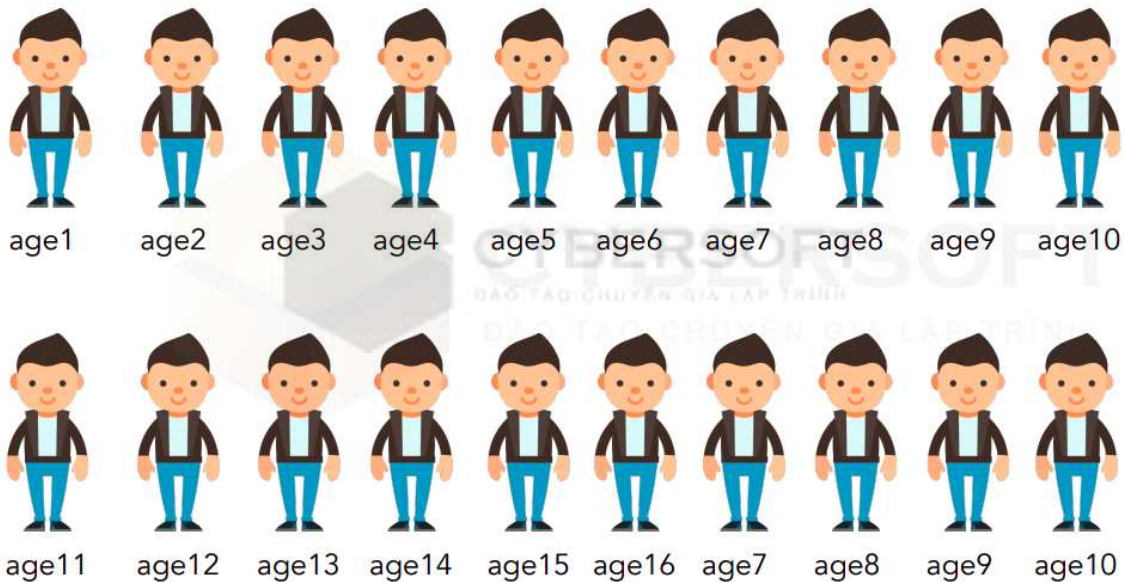
 default:
 // các dòng lệnh
}
```

# MẢNG (ARRAY)

- Mảng là một biến lưu trữ nhiều giá trị cùng lúc.
- Trong javascript, các phần tử trong mảng có thể có **kiểu dữ liệu khác nhau**.
- Truy cập một phần tử mảng dùng chỉ số mảng.



# GIỚI THIỆU MẢNG (ARRAY)



# CÁC PHẦN LIÊN QUAN ĐẾN MẢNG (ARRAY)

```
var age = [...]
```

age[0] age[1] age[2] age[3] age[4] age[5] age[6] age[7] age[8] age[9] age[10] age[11] age[12] age[13] age[14] age[15] age[16] age[17] age[18] age[19] age[20]



3. Chỉ số đầu tiên / index

1. Chỉ số của mảng

2. Độ dài của mảng = 20

4. Chỉ số cuối cùng = độ dài - 1

5. Phần tử thứ 6 có giá trị = 14

# KHAI BÁO MẢNG (ARRAY)

Từ khóa khai báo (var)

tên mảng

=

[...] hoặc new Array(...);

- Tạo mảng theo một trong 2 ví dụ sau:

```
var colors = new Array("red", "green", "blue");
var animals = ["dog", "cat", "tiger"];
```



# KHAI BÁO MẢNG (ARRAY)

- Trong đó cách thứ 2 được ưa dùng hơn. Việc sử dụng new Array sẽ dẫn đến một số nhập nhằng khi viết code, chẳng hạn:

```
var a = new Array(30, 20);
document.write(a[0]); // 30

var b = new Array(30);
document.write(b[0]); // undefined
```

→ Định nghĩa mảng có 2 phần tử là 30 và 20

→ Định nghĩa mảng có 30 phần tử đều có giá trị undefined.

# CÁC PHƯƠNG THỨC THÔNG DỤNG

- ❑ **push()**: thêm một hoặc nhiều phần tử ở cuối mảng và trả về chiều dài mới của mảng.
- ❑ **pop()**: lấy phần tử ở cuối ra khỏi mảng.
- ❑ **unshift()**: thêm một hoặc nhiều phần tử ở đầu mảng và trả về chiều dài mới của mảng
- ❑ **shift()**: lấy phần tử ở đầu ra khỏi mảng.

# CÁC PHƯƠNG THỨC THÔNG DỤNG

- ❑ **indexOf()**: trả về vị trí đầu tiên tìm thấy phần tử hoặc -1 nếu không tìm thấy.

```
array.indexOf(searchElement [, fromIndex]);
```

fromIndex là tham số tùy chọn, dùng chỉ định vị trí bắt đầu để tìm kiếm, mặc định là 0.

- ❑ **lastIndexOf()**: trả về vị trí cuối cùng tìm thấy phần tử hoặc -1 nếu không tìm thấy.

```
array.lastIndexOf(searchElement [, lastIndex]);
```

lastIndex chỉ định vị trí kết thúc tìm kiếm, mặc định là chiều dài mảng

# Vòng Lặp

- for - vòng lặp for theo index
- for/in - lặp trên từng thuộc tính của đối tượng (**Sẽ thảo luận trong phần JS OOP**)
- for/of - lặp thông qua Các Giá trị của một đối tượng có tính lặp (Mảng, chuỗi, Map, ...)
- while - lặp thông qua điều kiện đúng
- do/while - tương tự while nhưng thực hiện lệnh trước 1 lần

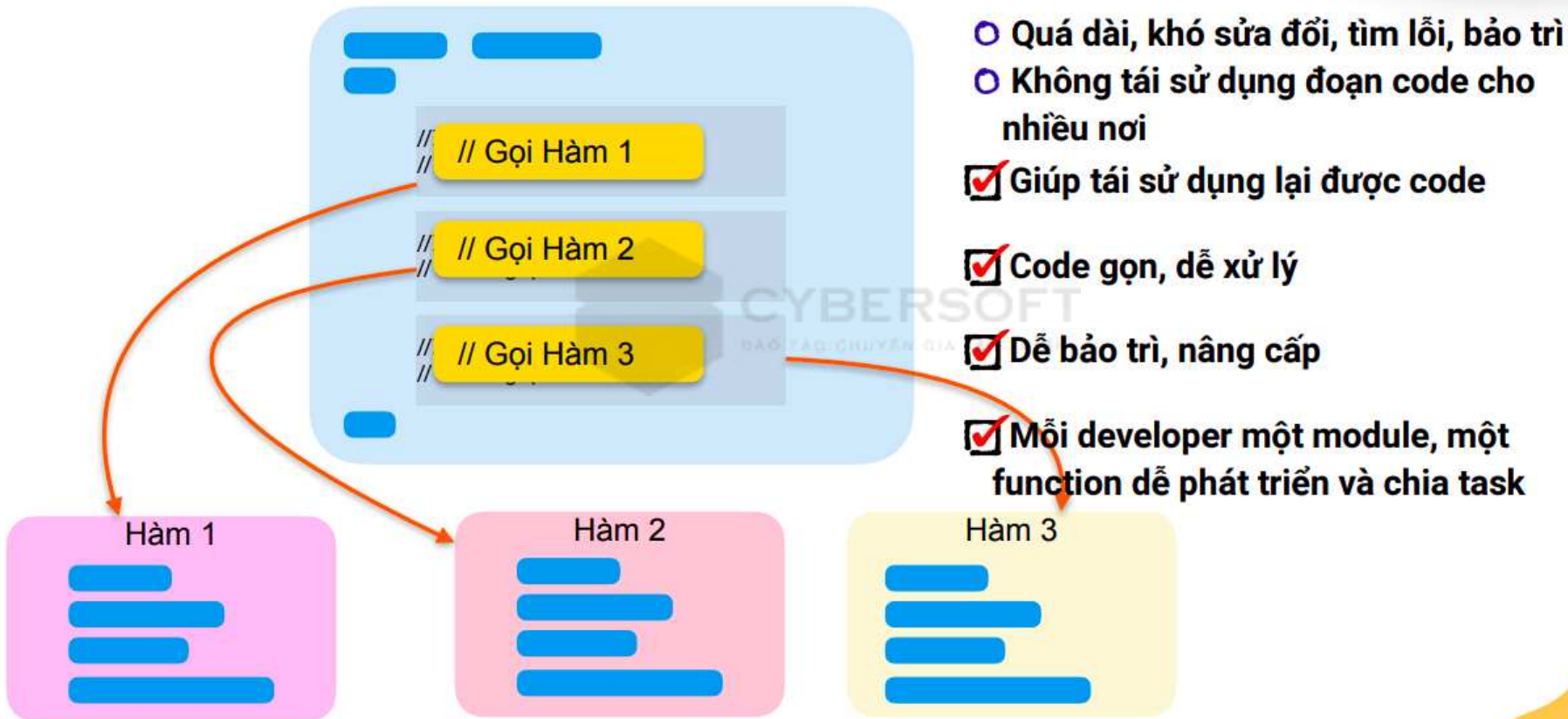
```
var people = ['Nam', 'Lien', 'Nguyen'];
```

```
for (var i = 0; i < people.length; i++) {
 console.log(people[i]);
}
```

```
for (var item of people){
 console.log(item);
}
```

```
var i = 0 ;
while (i < a.length){
 console.log(people[i]);
 i++;
}
```

# HÀM - LÝ DO TÁCH HÀM



# ĐỊNH NGHĨA HÀM

function

Tên hàm

(tham số truyền vào)

{

Thân hàm, các dòng lệnh xử lý

}

# MỘT SỐ VÍ DỤ HÀM

```
function xuatThongTin(){\n console.log("Hello world!");\n}
```

Hàm không có tham số,  
không có giá trị trả về

```
function tingTong(a, b){\n var total = a + b;\n console.log(total);\n}
```

Hàm có tham số, không  
có giá trị trả về

```
function tingLuong(soNgayLam, luongNgay){\n var luong = soNgayLam * luongNgay;\n return luong;\n}
```

Hàm có tham số, có giá  
trị trả về

# GỌI HÀM

Để gọi sử dụng hàm dùng **tên hàm** và **truyền danh sách đối số** tương ứng vào hàm.

```
function tinhTong(a, b){
 var total = a + b;
 console.log(total);
}
tinhTong(5,5);
```

- ❖ Khi gọi hàm nếu các tham số không được truyền sẽ nhận giá trị **undefined**.



# GỌI HÀM

- ❖ Phân biệt hai khái niệm:
  - **Tham số** (parameters) là danh sách các tên tham số trong định nghĩa hàm.
  - **Đối số** (arguments) là những giá trị thực sự được truyền vào hàm.
- ❖ Chú ý:
  - Không cần chỉ định **kiểu tham số**.
  - Javascript không kiểm tra **kiểu đối số** truyền vào.
  - Javascript không kiểm tra **số đối số** truyền vào.



CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Thank You