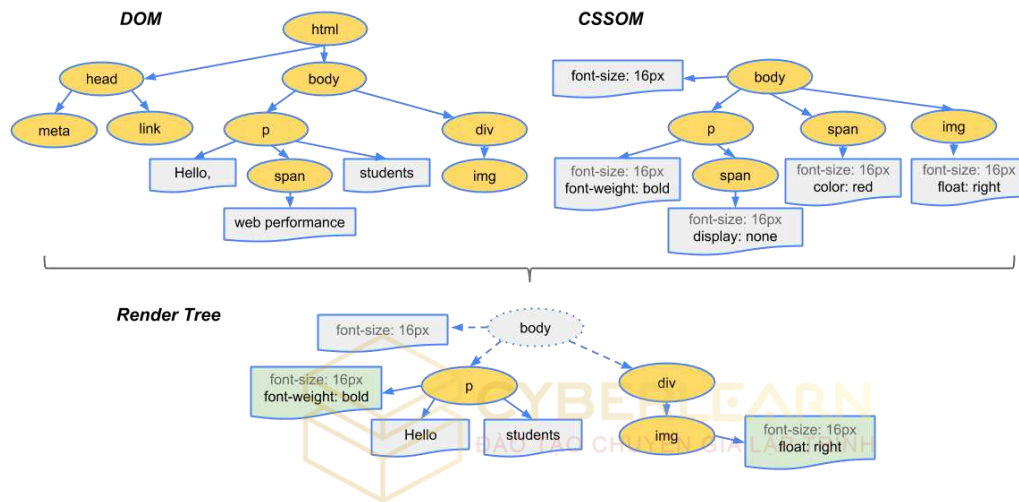


JavaScript



CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Quá trình thực thi hiển thị website



OOO....Ôi, Gì thế này?

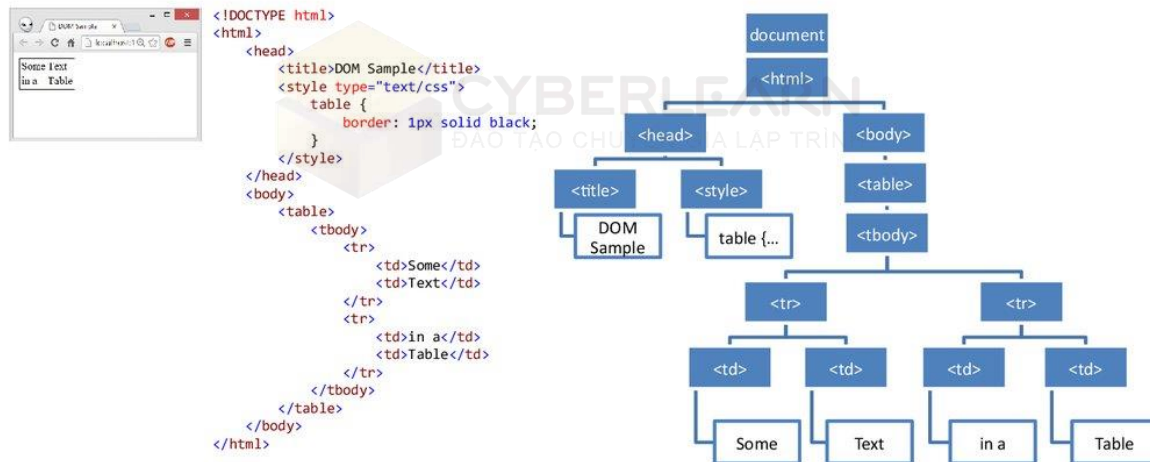
1. Qui tắc phân tích trang HTML
2. DOM (Document Object Model)
3. CSSOM (CSS Object Model)
4. Có liên quan gì JS không ? →

XEM SLIDE TIẾP

DOM - Document Object Model

Xem tương ứng giữa Code và cấu trúc cây DOM

DOM Tree



DOM - Document Object Model

- Mở thêm trình duyệt và sử dụng Inspect Element để thấy rõ hơn
- Vậy DOM để làm gì ? SUY NGHĨ THỬ NHÉ ?
 - Làm sao để lấy dữ liệu người dùng nhập ?
 - Làm sao để kiểm tra dữ liệu nhập đúng sai ?
 - Làm sao để đổ dữ liệu từ BACKEND lên ?
- → OK, hiểu rồi ? DOM ĐỂ GIÚP TA LẤY ĐƯỢC CÁC THẺ (OBJECT) ĐỂ TƯƠNG TÁC DỮ LIỆU

DOM - Document Object Model

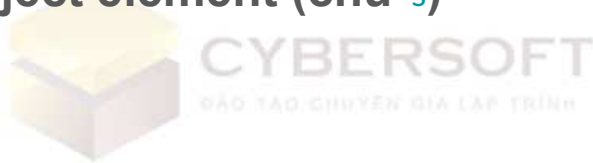
- Làm sao lấy đây ?
 - 1) LẤY QUA ID CỦA THẺ
 - 2) LẤY QUA TÊN THẺ (TAG NAME)
 - 3) LẤY QUA CLASS CSS CỦA THẺ
 - 4) Sử dụng phương thức `querySelector` & `querySelectorAll`

DOM - Cách 1 - Lấy qua id của thẻ

- LẤY QUA ID CỦA THẺ
- Cú pháp: `document.getElementById(elementID)`
 - Thẻ phải được gán ID
 - Ví dụ:
 - `var test = document.getElementById(<id thẻ>)`
 - Lưu ý: sau khi trả về cho biến test, biến test sẽ làm một OBJECT Element đó nhé! Nếu không trả về NULL

DOM - Cách 2 - Lấy qua tag name (tên thẻ)

- Cú pháp: `element.getElementsByTagName(tagname)`
 - **Element**: có thể là document, có thể là một element bất kì để lấy các element bên trong nó.
 - **Tagname**: tên thẻ cần lấy (ví dụ : thẻ <p>, thẻ <a>, thẻ)
 - **Trả về** danh sách các object element (chữ s)
 - → **DEMO slide tiếp**



```
<!DOCTYPE html>
<html>
<body>

<ul>
  <li>Front end</li>
  <li>Back end</li>
  <li>Front end</li>
  <li>CyberSoft</li>
</ul>

<p>Click nút để thay đổi item đầu tiên trong danh sách li
(index 0).</p>

<button onclick="myFunction()">NHẤN THỬ</button>

<script>
function myFunction() {
  var list = document.getElementsByTagName("UL")[0];
  list.getElementsByTagName("li")[0].innerHTML = "BUỔI";
}
</script>

</body>
</html>
```

- Front end
- Back end
- Front end
- CyberSoft

Click nút để thay đổi item đầu tiên trong danh sách li (index 0).

NHẤN THỬ

Sao là 0 chỗ này vậy hè ?

1. Mở Editor lên và code xem sao nhé



DOM - Cách 2 - Lấy qua tag name (tên thẻ)

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin: 5px;
}
</style>
</head>
<body>

<div id="myDIV">
  <p>Thẻ p đầu tiên trong div (index 0).</p>
  <p>Thẻ p thứ 2 trong div (index 1).</p>
  <p>Thẻ p thứ 3 div (index 2).</p>
</div>

<p>Nhấn nút và thay đổi background color của thẻ p thứ 2 sang
màu red.</p>

<button onclick="myFunction()">Thử ngay</button>

<script>
function myFunction() {
  var x = document.getElementById("myDIV");
  x.getElementsByTagName("P")[1].style.backgroundColor = "red";
}
</script>

</body>
</html>
```

Thẻ p đầu tiên trong div (index 0).

Thẻ p thứ 2 trong div (index 1).

Thẻ p thứ 3 div (index 2).

Nhấn nút và thay đổi background color của thẻ p thứ 2 sang màu red.

Thử ngay

1. Mở Editor lên và code xem sao nhé



DOM - Cách 2 - Lấy qua tag name (tên thẻ)

```
<!DOCTYPE html>
<html>
<head>
<style>
#myDIV {
  border: 1px solid black;
  margin: 5px;
}
</style>
</head>
<body>

<div id="myDIV">
  <h3>1. Thẻ H3 nè buri.</h3>
  <p>2. Thẻ P trong div.</p>
  <span>3. Thẻ Span trong div.</span>
  <h2>4. Thẻ H2 trong div.</h2>
  <div>5. Thẻ div trong div</div>
  <p>6. Một thẻ P khác trong div.</p>
</div>

<p>Nếu click vào NÚT bên dưới thì thẻ nào sẽ thay đổi đây ?.</p>

<button onclick="myFunction()">CODE MỚI BIẾT</button>

<script>
function myFunction() {
  var x = document.getElementById("myDIV");
  x.getElementsByTagName("*")[3].style.backgroundColor = "red";
}
</script>

</body>
</html>
```

Chỉ rì hè ? * chỗ này là sao?

1. Thẻ H3 nè buri.

2. Thẻ P trong div.

3. Thẻ Span trong div.

4. Thẻ H2 trong div.

5. Thẻ div trong div

6. Một thẻ P khác trong div.

Nếu click vào NÚT bên dưới thì thẻ nào sẽ thay đổi đây ?.

CODE MỚI BIẾT

1. Trả lời trước khi code ?
2. Mở Editor lên và code xem sao nhé



DOM - Cách 3 - Lấy qua classname

- Cú pháp: `element.getElementsByClassName(classname)`
 - Element: có thể là document, có thể là một element bất kì để lấy các element bên trong nó.
 - classname: tên class cần lấy
 - Trả về danh sách các object element (*chữ s số nhiều*)
 - → DEMO slide tiếp



CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

DOM - Cách 3 - Lấy qua classname (demo)

```
<!DOCTYPE html>
<html>
<body>

<div class="example">Thẻ div thứ nhất có class = "example".
</div>

<div class="example">Thẻ div thứ 2 có class="example".</div>

<p>Thay đổi Nội dung của thẻ Div đầu tiên class="example"
(index 0).</p>

<button onclick="myFunction()">THAY ĐỔI</button>

<p><strong>Chú ý:</strong> Phương thức getElementsByName()
không hỗ trợ Internet Explorer 8 trở về trước.</p>

<script>
function myFunction() {
    var x = document.getElementsByClassName("example");
    x[0].innerHTML = "Xin chào CyberSoft!";
    x[0].style.color = "red";
}
</script>

</body>
</html>
```

Thẻ div thứ nhất có class = "example".
Thẻ div thứ 2 có class="example".

Thay đổi Nội dung của thẻ Div đầu tiên class="example" (index 0).

THAY ĐỔI

Chú ý: Phương thức getElementsByName() không hỗ trợ Internet Explorer 8 trở về trước.

DOM - Cách 4 - Lấy qua querySelector (1)

- Cú pháp: `document.querySelector(<selector>)`
 - Trả về **element đầu tiên** nếu khớp với **selector**
 - **selector**: có thể là thẻ, có thể là className
 - Ví dụ:
 - `document.querySelector("p");` → Chọn thẻ p đầu tiên
 - `document.querySelector(".example");` → Chọn thẻ có class = "example" đầu tiên
 - `document.querySelector("#demo").innerHTML = "CyberSoft!";` → Thay đổi nội dung HTML bên trong thẻ có id = "demo"
 - `document.querySelector("div > p");` → Chọn thẻ <p> đầu tiên nằm trong thẻ <div>

DOM - Cách 4 - Lấy qua querySelectorAll (2)

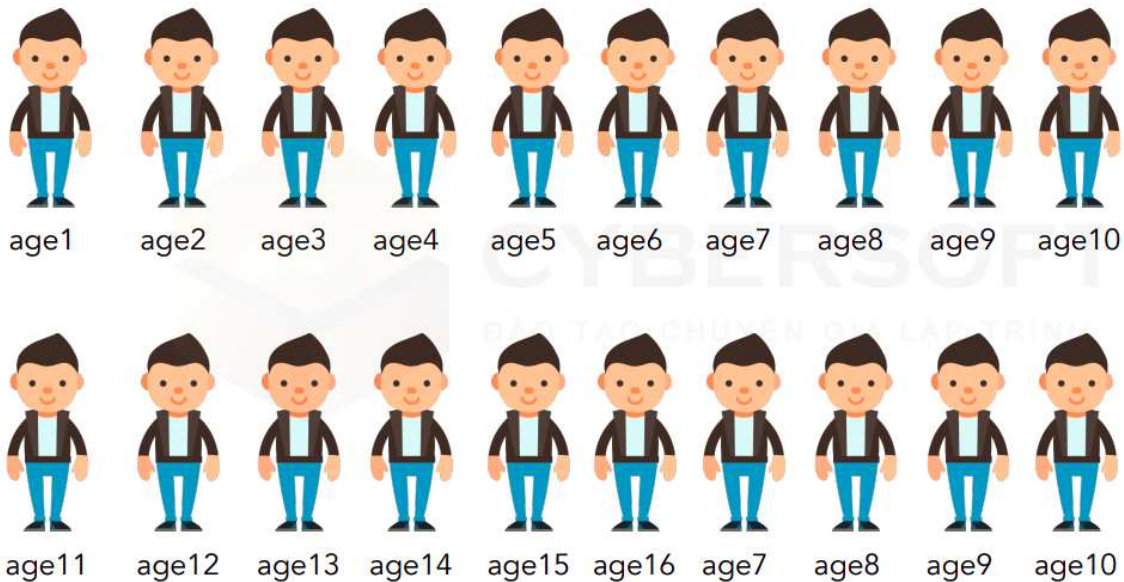
- Cú pháp: `document.querySelectorAll(<selector>)`
 - Trả về **tất cả element** nếu khớp với **selector**
 - **selector**: có thể là thẻ, có thể là className
 - Ví dụ:
 - `document.querySelectorAll("p");` → Tất cả thẻ `<p>`
 - `document.querySelectorAll(".example");` → Tất cả thẻ có class = "example"
 - `document.querySelectorAll("#demo").innerHTML = "CyberSoft!";` → Thay đổi nội dung HTML bên trong thẻ có id = 'demo'
 - `document.querySelectorAll("div > p");` → Tất cả thẻ `<p>` nằm bên trong thẻ `<div>`

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)



GIỚI THIỆU MẢNG (ARRAY)

1		Drew Vosburg	9,735
2		Jesse Nelson	8,364
3		Joel Bullis	7,621
4		Carmen SanDiego	4,582
5		Charles Barkley	2,651

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]

					14															
--	--	--	--	--	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

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

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
```

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

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

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

DEMO MẢNG (ARRAY)

```
var a = [9, 3, 5, 8];  
  
// Chiều dài mảng  
console.log("Chiều dài mảng: " + a.length); // 4  
  
// Truy cập vào các phần tử mảng  
for (var i = 0; i < a.length; i++) {  
    console.log("a[" + i + "] = " + a[i]);  
}
```

Chiều dài mảng: 4

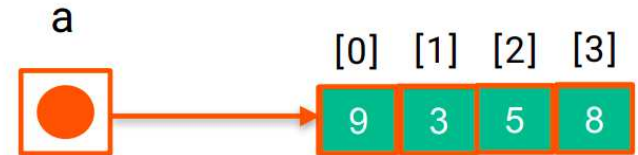
a[0] = 9

a[1] = 3

a[2] = 5

a[3] = 8

RSO
TRUNG GIÁ LẬP TRÌNH



DEMO MẢNG (ARRAY)

```
var animals = ["dog", "cat", "tiger"];

// Chiều dài mảng
console.log("Chiều dài mảng: " + animals.length); // 3

// Truy cập vào các phần tử mảng
for (var i = 0; i < animals.length; i++) {
  console.log("animals[" + i + "] = " + animals[i]);
}
```

Chiều dài mảng: 3

animals[0] = dog

animals[1] = cat

animals[2] = tiger

RSOFT
KÊNH GIẢI LẬP TRÌNH

DUYỆT MẢNG BẰNG FOREACH

```
/* Mỗi lần duyệt mảng
- item đại diện cho phần tử trong mảng,
- index đại diện cho số chỉ mục trong mảng
*/
array.forEach(function(item, index){
    // Các câu lệnh
})
```

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

Ví dụ:

```
var colors = ["red", "green", "blue"];  
document.writeln(colors.pop()); // blue  
colors.push("yellow", "green"); // 4  
document.writeln(colors); // red, green, yellow, green  
document.writeln(colors.shift()); // red  
colors.unshift("gray"); // 4  
document.writeln(colors); // gray, green, yellow, green
```

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

CÁC PHƯƠNG THỨC THÔNG DỤNG

Ví dụ:

```
var colors = ["red", "green", "blue", "green"];  
document.writeln(colors.indexOf("green")); // 1  
document.writeln(colors.indexOf("green", 2)); // 3  
document.writeln(colors.lastIndexOf("green")); // 3  
document.writeln(colors.lastIndexOf("green", 2)); // 1
```

CÁC PHƯƠNG THỨC THÔNG DỤNG

- ❑ **reverse()**: đảo ngược mảng.
- ❑ **sort()**: sắp xếp mảng.

```
array.sort ( [<hàm-so-sánh>] ) ;
```

```
var numbers = [1, 4, 7, 3];  
numbers.reverse();  
document.writeln(numbers); // 3, 7, 4, 1  
numbers.sort(); // sắp tăng  
document.writeln(numbers); // 1, 3, 4, 7  
  
numbers = [1, 6, 7];  
numbers.sort(function(a, b) { // sắp giảm  
    return b - a;  
});  
document.writeln(numbers);
```

CÁC PHƯƠNG THỨC THÔNG DỤNG

- ❑ **join()**: dùng kết hợp các phần tử của mảng thành một chuỗi.

```
array.join(separator);
```

Ví dụ:

```
var colors = ["red", "green", "blue"];  
document.write(colors.join(", ")); // red, green, blue  
document.write(colors.join("; ")); // red; green; blue  
document.write(colors.join("|")); // red|green|blue
```

CÁC PHƯƠNG THỨC THÔNG DỤNG

- ❑ **slice()**: cắt một đoạn trong mảng và trả về mảng mới.

```
array.slice(beginIndex [, endIndex]);
```

- beginIndex và endIndex: lấy đoạn con từ vị trí beginIndex đến trước endIndex, endIndex mặc định là chiều dài mảng.
- Nếu các chỉ số là số âm sẽ được tính từ cuối mảng.

Ví dụ chỉ số âm:

0	1	2	3	4
red	green	blue	yellow	gray
-5	-4	-3	-2	-1

CÁC PHƯƠNG THỨC THÔNG DỤNG

Ví dụ:

```
var colors = ["red", "green", "blue", "yellow", "gray"];  
document.writeln(colors.slice(2)); // blue, yellow, gray  
document.writeln(colors.slice(1, 3)); // green, blue  
document.writeln(colors.slice(-2)); // yellow, gray  
document.writeln(colors.slice(-3, -2)); // blue
```

CÁC PHƯƠNG THỨC THÔNG DỤNG

- ❑ **splice()**: dùng thay đổi nội dung của mảng, thêm hoặc xóa các phần tử trong mảng và trả về các phần tử đã bị xóa.

```
array.splice(index, num, [e1, e2, ..., en]);
```

- ❑ **index**: chỉ định vị trí bắt đầu thay đổi mảng.
- ❑ **num**: chỉ định số lượng phần tử sẽ bị xóa từ index, mặc định là 0.
- ❑ **e1, e2, ..., en** chỉ định các phần tử sẽ được thêm vào từ vị trí index.

CÁC PHƯƠNG THỨC THÔNG DỤNG

Ví dụ:

```
var colors = ["red", "green", "yellow"];  
colors.splice(1, 0, "orange");  
document.writeln(colors); // red, orange, green, yellow  
colors.splice(1, 2, "gold");  
document.writeln(colors); // red, gold, yellow  
colors.splice(1, 1);  
document.writeln(colors); // red, yellow
```

CÁC PHƯƠNG THỨC THÔNG DỤNG

- ❑ **concat()**: trả về mảng mới bao gồm các phần tử nối mảng đang xét với các mảng hoặc phần tử chỉ định.

Ví dụ:

```
var numbers = [1, 2];  
var nNumbers = numbers.concat([3, 4], [5], 8, 9);  
document.writeln(nNumbers); // 1, 2, 3, 4, 5, 8, 9
```

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

CÁC PHƯƠNG THỨC THÔNG DỤNG

- ❑ **map()**: trả về mảng mới (cùng kích thước mảng đang xét) từ kết quả xử lý mỗi phần tử mảng đang xét trong một hàm cho trước.

Ví dụ:

```
var colors = ["red", "green", "yellow"];  
var upperColors = colors.map(function(c) {  
    return c.toUpperCase();  
});  
  
// RED, GREEN, YELLOW  
document.writeln(upperColors);
```

CÁC PHƯƠNG THỨC THÔNG DỤNG

- ❑ **filter()**: trả về mảng mới là một số phần tử trong mảng đang xét thỏa bộ lọc định nghĩa trong một hàm cho trước.

Ví dụ:

```
var numbers = [12, 5, 3, 8, 9, 26];  
var evenNum = numbers.filter(function(n) {  
    return n % 2 == 0;  
});  
document.writeln(evenNum); // 12, 8, 26
```

BÀI TẬP LUYỆN MẢNG

1. Cho người dùng nhập vào số nguyên và lưu trữ vào mảng.

1. Tính tổng các số chẵn
2. Đếm có bao nhiêu số âm
3. Tính tổng các số âm

Nhập vào số nguyên

Thêm số
























CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

BÀI TẬP LUYỆN MẢNG

2. Dựa vào layout từ đề bài:

- Tìm sinh viên có điểm cao nhất
- Tìm sinh viên có điểm thấp nhất
- Đếm số sinh viên giỏi
- Danh sách sinh viên có điểm trung bình lớn hơn 5
- Sắp xếp điểm tăng dần

[Link layout bài luyện](#)

QUẢN LÝ SINH VIÊN				THÊM MỚI
<input type="checkbox"/>	HÌNH ẢNH	TÊN	ĐIỂM TRUNG BÌNH	THAO TÁC
<input type="checkbox"/>		Nguyễn Văn A	9.4	 
<input type="checkbox"/>		Đỗ Văn Nhơn	8.2	 
<input type="checkbox"/>		Vin Thị Liên	3.4	 
<input type="checkbox"/>		Đen Thị Giầu	6.4	 
<input type="checkbox"/>		Ngô Văn Mạnh	2.4	 
<input type="checkbox"/>		Văn Sỹ Lâm	1.4	 
<input type="checkbox"/>		Bùi Đình Sen	9.8	 
Hiện thị 10 trên 25 sinh viên				Trước 1 2 3 4 5 Sau
Tìm SV Có Điểm Cao Nhất				Tìm SV Có Điểm Thấp Nhất
Đếm Số SV Giỏi				Danh Sách SV Điểm TB Lớn Hơn 5
Sắp xếp tăng dần				

BÀI TẬP LUYỆN MẢNG

Cho người dùng nhập vào số nguyên và lưu trữ trong mảng.

Thiết kế layout cho người dùng chọn thực hiện các chức năng bên dưới:

1. Tổng các số dương trong mảng.
2. Đếm có bao nhiêu số dương trong mảng.
3. Tìm số nhỏ nhất trong mảng.
4. Tìm số dương nhỏ nhất trong mảng.
5. Tìm số chẵn cuối cùng trong mảng. Nếu mảng không có giá trị chẵn thì trả về -1.
6. Đổi chỗ 2 giá trị trong mảng theo vị trí (Cho nhập vào 2 vị trí muốn đổi chỗ giá trị).
7. Sắp xếp mảng theo thứ tự tăng dần.
8. Tìm số nguyên tố đầu tiên trong mảng. Nếu mảng không có số nguyên tố thì trả về -1.
9. Nhập thêm 1 mảng số thực, tìm xem trong mảng có bao nhiêu số nguyên?
10. So sánh số lượng số dương và số lượng số âm xem số nào nhiều hơn.