

ES6

1: let, const

2: Template String

3: Arrow Function

4: Object, Array destructuring assignment

5: Spread, Rest syntax

6: promise

7: async/await

.....



let, const

- ❖ Các từ khóa let và const giúp ta tạo ra các biến block-scope, tức là các biến chỉ tồn tại và được phép sử dụng trong phạm vi mà nó được tạo ra, các phạm vi này được giới hạn bởi các khối lệnh nằm trong cặp dấu {}, ví dụ:
 - Trong một hàm
 - Trong một vòng for
 - Trong một khối lệnh if
- ❖ Một file js có thể được hình dung như đang được bọc lại bởi một cặp dấu ngoặc {}, nó là phạm vi lớn nhất

Arrow Function

- ❖ Tuy nhiên tùy theo việc function của ta nhận vào bao nhiêu tham số, làm những việc gì, arrow function có thể tiêu giảm một số thành phần và vẫn không ảnh hưởng đến việc hoạt động của hàm
 - Khi hàm của ta chỉ nhận một tham số truyền vào
 - Khi hàm của ta chỉ sử dụng một dòng lệnh duy nhất là return
 - Khi hàm của ta chỉ sử dụng một dòng lệnh return, nhưng kết quả trả về là một object
 -

```
const myFunction = function (param1, param2, param3) {  
  console.log(param1);  
  console.log(param2);  
  console.log(param3);  
  return "Done";  
};
```

```
const myFunctionArrow = (param1, param2, param3) => {  
  console.log(param1);  
  console.log(param2);  
  console.log(param3);  
  return "Done";  
};
```

```
const myFunctionArrow1 = (param1) => {  
  console.log(param1);  
  return "Done";  
};
```

```
const tinhTong = (so1, so2) => {  
  // const tong = so1 + so2;  
  return so1 + so2;  
};  
  
const tinhTong2 = (so1, so2) => so1 + so2;
```

Arrow Function

```
const arr = [1, 2, 3, 4, 5];  
const result = arr.map((item, index, arr) => {  
  return item * item;  
});  
console.log(result);
```

```
const arr = [1, 2, 3, 4, 5];  
const result = arr.map((item) => item * item);  
console.log(result);
```

TẠO THUỘC TÍNH CHO OBJECT

- ❖ Khi ta muốn tạo một thuộc tính cho một object với thông tin đang được lưu trong một biến nào đó, ta chỉ việc đặt biến đó vào vị trí value của key thuộc tính mà ta muốn đặt cho object đó,
- ❖ Es6 cho ta một cú pháp ngắn gọn hơn trong trường hợp key của ta trùng với tên của biến chứa dữ liệu

```
const address = "Hà Nội"
const obj = {
  name:"nghiem",
  age:18,
  address:address
}
// const obj = {
//   name:"nghiem",
//   age:18,
//   address
// }
console.log(obj); { name: 'nghiem', age: 18, address: 'Hà Nội' }
```

```
const myData = "some infomation";
const name = "Nghiem";

const myObject = {
  id: 1,
  info: myData,
  name
}
console.log(myObject) { id: 1, info: 'some infomation', name: 'Nghiem' }
```

```
const myF = (input) => {
  return {
    name: input,
  };
};
console.log(myF("nghiem"));
```



```
const myF = (name) => {
  return {
    name,
  };
};
console.log(myF("nghiem"));
```

destructuring assignment

- ❖ Destructuring assignment là cú pháp ngắn gọn của việc ta tạo ra một biến và muốn gán nó có giá trị bằng dữ liệu từ một nguồn là object hay array khác;
- ❖ Nguồn dữ liệu có thể là bên phải của dấu "=", hoặc từ dữ liệu truyền vào của hàm
- ❖ Với Object: Ta có thể tạo luôn một biến có tên trùng với key mà ta muốn lấy dữ liệu, hoặc cũng thể đặt một tên khác nếu muốn, nếu ta cố tình tạo một biến lấy một thông tin không tồn tại, biến đó sẽ có giá trị undefined

```
const address = "Hà Nội"
const obj = {
  name: "nghiem",
  age: 18,
  address: address
}

const hisName = obj.name;
const hisAge = obj.age;
const hisAddress = obj.address;

console.log(hisName);  nghiem
console.log(hisAge);   18
console.log(hisAge);   18
```



```
const {name, age, address: x} = obj;
console.log(name)  nghiem
console.log(age);  18
console.log(x);    Hà Nội
```

destructuring assignment

```
const myObj = {  
  name:"nghiem",  
  age:18,  
  address:{  
    city:"Cầu Giấy",  
    conuntry:"Việt Nam"  
  }  
}  
// mong muốn lấy giá trị name , age , conuntry
```

destructuring assignment

```
const myFn = (input) => {  
  const herName = input.name;  
  const herAge = input.age  
  console.log(herAge) 18  
  console.log(herName) nghiem  
}  
myFn({name:"nghiem",age:18})
```



```
const myFn = ({name,age}) => {  
  console.log(name) nghiem  
  console.log(age) 18  
}  
myFn({name:"nghiem",age:18})
```


destructuring assignment

```
const students = [  
  {  
    name: "phong",  
  },  
  {  
    name: "nghiem",  
  },  
  {  
    name: "son",  
  },  
  {  
    name: "dai",  
  },  
];  
  
const student1 = students[0];  
const student3 = students[2];  
const student4 = students[3];  
// console.log(student1);  
// console.log(student3);  
// console.log(student4);  
💡  
const [st1, st2, st3, st4, st5] = students;  
console.log(st1);  
console.log(st2);  
console.log(st3);  
console.log(st4);  
console.log(st5);
```

Spread, Rest syntax

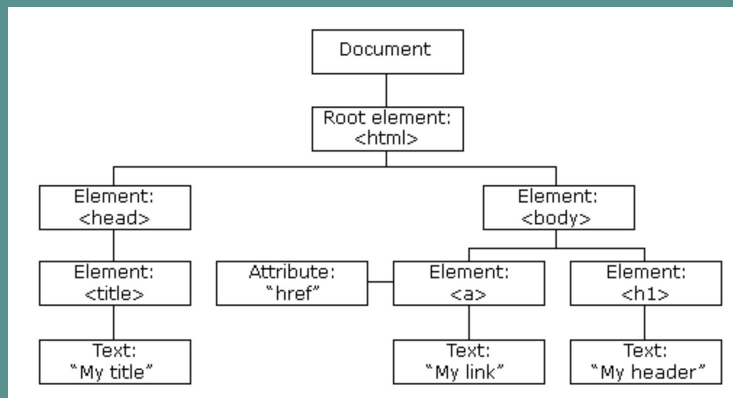
- ❖ Spread operator là cú pháp giúp ta “Trải” thông tin của một object hoặc một array để sử dụng thay vì phải trỏ đến từng phần tử trong mảng hay từng key trong array
- ❖ Lưu ý: các key đứng sau nếu trùng tên sẽ ghi đè dữ liệu của các key đứng trước
- ❖ Sử dụng dấu “...” để trải các thông tin của object ra, thường dùng để copy dữ liệu sang một object mới
- ❖ Với object, các thông tin được trải sẽ được liệt kê lần lượt giống như sau khi dòng code được chạy
 - Key1:value1
 - Key2:value2
 - Key3:value3
 - Key n:value n

HTML DOM

- ❖ DOM là gì ?
- ❖ Cách DOM kết nối với các thành phần HTML
- ❖ Cách thay đổi các thành phần HTML thông qua DOM
- ❖ DOM event.
- ❖ DOM event listener

HTML DOM

- ❖ DOM là viết tắt của cụm từ Document Object Model, cụ thể DOM giống như một object lớn chứa các thông tin của một trang web, cụ thể là cấu trúc tương ứng của HTML mà trang web đó có
- ❖ DOM lưu trữ, quản lý và giúp ta thực hiện các thao tác với các thành phần HTML của trang web
- ❖ Về cơ bản, DOM là một object javascript được tạo ra dựa trên cấu trúc của HTML để ta có thể thao tác với HTML trong javascript, DOM được tạo ra khi trình duyệt load cấu trúc HTML của trang web.
- ❖ Các thông tin mà DOM có bao gồm
 - Các thẻ HTML dưới dạng các object
 - Các thuộc tính của các thẻ HTML đó
 - Các Hàm để truy cập tới những thẻ HTML nhất định
 - Các sự kiện có thể xảy ra của các thẻ HTML
-
- ❖ Mọi thao tác xử lý HTML từ phía javascript đều phải thông qua đối tượng element



DOM KẾT NỐI CÁC THÀNH PHẦN HTML

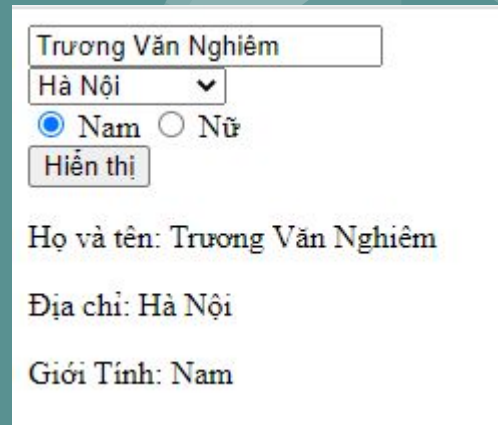
- ❖ Bản chất một trang web được cấu thành bởi một hệ thống cây với các node hay DOM node chính là các thẻ html và thuộc tính của chúng, trong đó đứng cao nhất và to nhất chính là đối tượng object document là một node lớn nhất để chứa các node con nhỏ hơn là thành phần tạo nên một cây
- ❖ Các DOM node có thể có các quan hệ với nhau cũng như các thẻ html: node cha, node con, node anh chị em
- ❖ Để trở tới được các element (node) trong cái cây đó , ta cần dùng tới các hàm của đối tượng document để tìm các element (node) con mà nó đang quản lý, các hàm có thể kể đến, mỗi node đều có thể thực hiện tương tự để tìm các con của chúng
 - document.getElementById(a): tìm MỘT element (node) có thuộc tính id là a
 - document.getElementsByClassName(a): tìm TẤT CẢ các element(node) có thuộc tính class chứa a
 - document.getElementsByTagName(a): tìm TẤT CẢ các element(node) có tên thẻ là a
 - document.querySelector(a): Tìm element ĐẦU TIÊN thỏa mãn selector a (a là selector giống css)
 - document.querySelectorAll(a): Tìm tất cả các element thỏa mãn selector a (a là selector giống css)
 -

DOM EVENT

- ❖ DOM event là một sự kiện có thể xảy đến với một node element, hay một thẻ html mà nó đại diện, đó có thể là một thao tác đến từ người dùng hoặc đến từ một sự kiện nào đó khác
- ❖ Một số sự kiện cơ bản có thể kể đến như
 - Key event: các sự kiện được tạo ra khi ta ấn bàn phím
 - keydown: khi ta ấn xuống 1 key
 - keyup: khi ta nhả key đó ra
 - Keypress: khi ta đã ấn và giữ key
 - ...
 - Mouse event:
 - Click: khi ta click chuột
 - Dbclick: khi ta click đúp
 - Mousedown: khi ta ấn chuột xuống
 - Mouseup: khi ta nhả chuột
 - Mousemove: khi ta ấn và kéo chuột
 - ...

Bài Tập

Câu 1: dựng giao diện gồm: nhập họ và tên , chọn địa chỉ , radio . khi nhập đầy đủ thông tin kích vào button hiển thị, thì sẽ hiện tất cả thông tin ở bên dưới



Trương Văn Nghiêm

Hà Nội ▼

☒ Nam ☐ Nữ

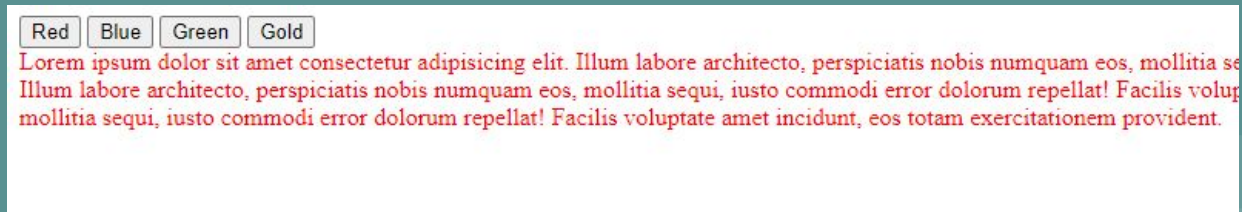
Hiển thị

Họ và tên: Trương Văn Nghiêm

Địa chỉ: Hà Nội

Giới Tính: Nam

Bài 2: tạo giao diện như hình: khi kích vào button red => màu chữ đổi sang đỏ , tương tự button khác ...

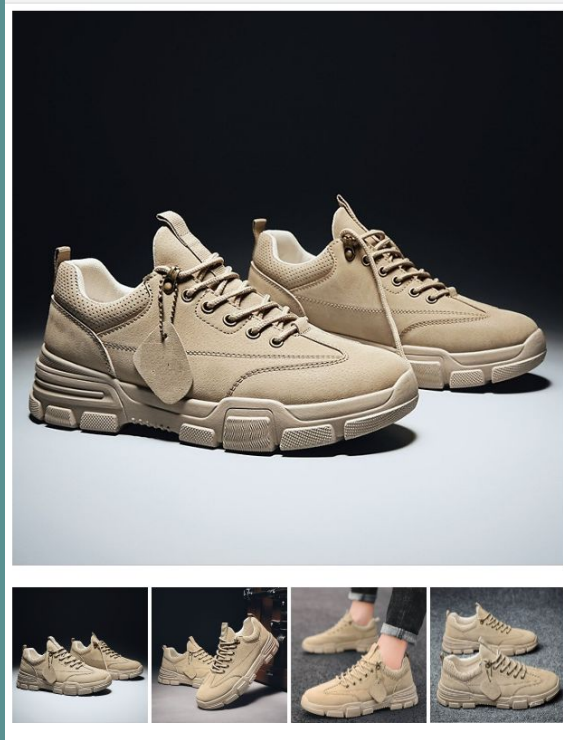


Red Blue Green Gold

Lorem ipsum dolor sit amet consectetur adipisicing elit. Illum labore architecto, perspiciatis nobis numquam eos, mollitia sequi, iusto commodi error dolorum repellat! Facilis voluptate amet incidunt, eos totam exercitationem provident.

Bài Tập

Câu 3: dựng giao diện như hình ảnh , khi kích vào hình ảnh nhỏ bên dưới , ảnh to bên trên cũng thay đổi theo



Bài Tập

Câu 4: dựng giao diện như hình ảnh , làm chức năng thêm sửa xóa , validate mã học sinh + họ và tên

- + khi không có data ở bảng hiển thị text : no data
- + khi kích vào thêm , xóa hay edit hiển thị modal để thông báo xác nhận

Thông tin sinh viên

Mã Sinh Viên

không bỏ trống

Họ Và tên

không bỏ trống

Năm Sinh:

Quê Quán:

Điểm Môn Toán:

Điểm Môn Lý:

Điểm Môn Hóa:

Thêm

reset

!

Bạn có muốn xóa không?

Trương Văn Nghiễm sẽ bị xóa

Cancel

OK

Mã SV	Tên	Năm Sinh	Quê Quán	Môn Toán	Môn Lý	Môn Hóa	ĐTB	Action
1	Trương Văn Nghiễm	2000	Hà Nội	9	6	3	6	<div>Xóa</div> <div>Edit</div>

Bài Tập lớn: Hình ảnh trang tiếp theo

1. tạo một form để lấy thông tin đăng ký thẻ tín dụng của người dùng , các thông tin gồm
name,

email,

password,

ảnh đại diện.

ngày sinh,

địa chỉ,

thu nhập trung bình,

mô tả bản thân.

giới tính,

tình trạng hôn nhân

thời hạn thẻ (1,2,3...10 năm)

2. Tạo một nút bấm để lưu dữ liệu đã lấy của người dùng vào một bảng bên dưới form, đồng thời reset các thông tin của form về rỗng, mỗi lần thêm một dòng vào bảng , mỗi dòng hiện các thông tin tương ứng của người đó

3. Thêm chức năng xóa thông tin bằng một nút bấm trong bảng

4. Thêm chức năng sửa thông tin bằng một nút bấm trong bảng , các thông tin của người dùng được đưa lại form để sửa , ấn một nút để lưu lại.

5. Thêm chức năng tìm kiếm thông qua 3 trường là (name hoặc email)và tình trạng hôn nhân,

6. thêm chức năng sắp xếp thu nhập trung bình (theo 1 trong 2 chiều)

7:Thêm chức năng khi load lại trang ko mất data (tham khảo localstote)

Bài Tập

Id:

Name:

Avatar: 313417483_...46834_n.jpg

Date Of Birth:

Address

Average Salary:

dsdsd

ds

ds

Description


Gender:

Male: ☒ Female: ☐ Other: ☐

Married: ☒

Duration:

Married: All ☒ Married ☐ Single ☐

ID	Name	Avatar	Dob	Address	Salary	Description	Gender	Married	Duration	Action
1670299781283	Trương Văn Nguyễn		2022-12-14	Hà Nội	56565656	ngheiem dsmskdasd dsad	female	Đã kết hôn	5 Năm	<input type="button" value="Xoa"/> <input type="button" value="Sua"/>