



A. MỤC TIÊU:

- Sử dụng thành thạo các cú pháp của ngôn ngữ lập trình Javascript
- Xây dựng được các tương tác linh động với người dùng thông qua các sự kiện của đối tượng DOM.

B. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

5.1. Tổng quan về JavaScript

JavaScript (JS) là một ngôn ngữ kịch bản (không cần biên dịch) được sử dụng để tạo các trang web tương tác. Ngôn ngữ này rất linh động hỗ trợ lập trình hướng đối tượng, hàm, sự kiện. JS có thể chạy trên rất nhiều nền tảng như trình duyệt web, máy chủ (Node.js), ứng dụng di động, và nhiều nền tảng khác. JS chạy trên trình duyệt của người dùng, giúp cải thiện tốc độ tải trang.

JavaScript trong trình duyệt có thể thực hiện mọi thứ liên quan đến thao tác trang web, tương tác với người dùng và máy chủ web, cụ thể:

- Thêm HTML mới vào trang, thay đổi nội dung hiện có.
- Tương tác với hành động của người dùng, chạy khi nhấp chuột, di chuyển con trỏ, nhấn phím.
- Gửi yêu cầu qua mạng đến các máy chủ từ xa, tải xuống và tải lên các tệp (được gọi là công nghệ AJAX và COMET).
- Nhận và đặt cookie, đặt câu hỏi cho khách truy cập, hiển thị tin nhắn.
- Ghi nhớ dữ liệu ở phía máy khách (lưu trữ cục bộ).

5.1.1.. Tập lệnh Script

Các chương trình JavaScript có thể được chèn vào bất kỳ phần nào của tài liệu HTML với sự trợ giúp của thẻ <script>.

Ví dụ:

```
<!DOCTYPE HTML>
<html>
<body>
  <p>Before the script...</p>
  <script type="text/javascript">
    alert( 'Hello, world!' );
  </script>
  <p>...After the script.</p>
</body>
</html>
```

Ngoài ra, code JavaScript có thể được đặt vào một tệp riêng và chỉ cần khai báo thẻ script đính kèm vào HTML với thuộc tính src có giá trị là tên đường dẫn tệp javascript

```
<script src="/path/to/script.js"></script>
```

5.1.2. Biến và câu lệnh

Một biến là tên đại diện cho vùng nhớ lưu trữ cho dữ liệu. Chúng ta có thể sử dụng các biến để lưu trữ thông tin, khách truy cập và dữ liệu khác. Để tạo một biến trong JavaScript, sử dụng từ khóa let hoặc var.

Câu lệnh khai báo một biến có tên là message và gán dữ liệu cần lưu trữ “Hello”.

```
let message;  
message = 'Hello!';  
alert(message); // shows the variable content  
Chúng ta cũng có thể khai báo nhiều biến trong một dòng:
```

```
let user = 'John', age = 25, message = 'Hello';
```

Một từ khóa khác để khai báo biến là var

```
var message = 'Hello';
```

Để khai báo hằng số không đổi, sử dụng const

```
const myBirthday = '18.04.1982';
```

Kết thúc câu lệnh JavaScript có thể có dấu chấm phẩy (;) hoặc không và quy ước chú thích cặp /* */

5.2. Các thành phần của JavaScript

5.2.1. Lệnh nhập xuất

✚ Nhận 1 giá trị chuỗi từ phím

```
let a = prompt("Nhập họ tên của bạn:");
```

✚ Xuất chuỗi giá trị lên trình duyệt

```
document.write("Tổng là: " + sum);
```

✚ Xuất hộp chuỗi giá trị

```
alert("Chào bạn");
```

5.3. Cấu trúc điều khiển

✚ Câu lệnh if

```
if (conditional1){  
  // khối lệnh 1 được thực thi nếu conditional là true  
} else if (conditional2){  
  // Khối lệnh 2;  
} else {  
  // Khối lệnh 3  
}
```

+ Câu lệnh switch... case

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

+ Vòng lặp for

```
for ([initialization]; [condition]; [final-expression])
  statement
```

+ Vòng lặp for ... of

```
for (variable of iterable) {
  statement
}
```

+ Vòng lặp for ... in

```
for (variable in object) {
  statement
}
```

+ Vòng lặp while

```
while (condition)
  statement
```

Mảng

```
let fruits = ['Apple', 'Banana'];
let first = fruits[0]; //Apple
let last = fruits[fruits.length - 1]; //Banana
```

5.4. Lập trình hướng đối tượng trong Javascript

Lập trình hướng đối tượng - OOP (Object-Oriented Programming) là một mô hình lập trình dựa trên việc sử dụng các đối tượng để tổ chức và quản lý thông tin. Trong JavaScript, OOP có thể được thực hiện theo nhiều cách khác nhau, vì JavaScript là một ngôn ngữ linh hoạt và hỗ trợ cả lập trình hướng đối tượng nguyên mẫu (Prototype-based OOP) và lập trình hướng đối tượng dựa trên lớp (Class-based OOP).

+ Lập trình hướng đối tượng nguyên mẫu (Prototype-based OOP)

Đối tượng là tập hợp các thuộc tính (properties) và phương thức (methods).

Ví dụ:

```
let person = {
  name: "John",
  age: 30,
  greet: function () {
    document.write("Hello, my name is "+this.name);
  }
};

person.greet();
```



Lập trình hướng đối tượng dựa trên lớp (Class-based OOP)

Class là mẫu để tạo ra các đối tượng. Object là tập hợp thuộc tính và phương thức để tổ chức và quản lý thông tin, hành vi của đối tượng.

Các đặc điểm, tính chất của OOP trong JS tương tự như ngôn ngữ lập trình khác:

- ☐ Encapsulation: Bảo vệ dữ liệu bằng cách sử dụng thuộc tính private.
- ☐ Inheritance: Kế thừa thuộc tính và phương thức từ lớp cha.
- ☐ Polymorphism: Ghi đè phương thức.
- ☐ Abstraction: Trừu tượng hóa phương thức.

```
class Person {
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }
  greet() {
    document.write('Hello, my name is '+this.name);
  }
}
//khai báo đối tượng
const john = new Person("John", 30);
john.greet();
```

5.5. Thao tác với DOM

DOM (Document Object Model) cho phép JavaScript giao tiếp với HTML và CSS. Với DOM, mọi phần tử HTML là một đối tượng JavaScript được lưu trữ trong đối tượng document và có thể truy cập với mã lệnh JavaScript.

Một số phương thức trong DOM

- Chọn phần tử theo ID
`var element = document.getElementById(id);`
- Chọn tất cả các phần tử theo Tên thẻ HTML
`var elements = document.getElementsByTagName(name);`
- Chọn tất cả các phần tử theo Tên class
`var elements = document.getElementsByClassName(name);`
- Chọn phần tử đầu tiên thỏa theo các thức chọn CSS Selector

```
var element = document.querySelector(selectors);
```

- Chọn tất cả phần tử thỏa theo các thức chọn CSS Selector

```
var elementList = parentNode.querySelectorAll(selectors);
```

Ví dụ: chọn phần tử thẻ <a> đầu tiên và thay đổi đường link trong tài liệu HTML

```
let link = document.querySelector('a');  
link.href = 'http://thongthai.work';
```

- Tạo, xóa node

```
var newDiv = document.createElement('div');  
newDiv.innerHTML = 'Hello there!';  
document.body.appendChild(newDiv);  
document.body.removeChild(div1);
```

- Quản lý style

```
para.style.color = 'white';  
para.style.backgroundColor = 'black';  
para.style.padding = '10px';  
para.style.width = '250px';  
para.style.textAlign = 'center';
```

- Quản lý sự kiện

```
document.getElementById('div1').  
  addEventListener('mouseover', function(event) {  
    event.target.style.backgroundColor = '0000ff';  
  });
```

```
document.addEventListener('DOMContentLoaded', function(event){  
  alert('Page has been loaded');  
});
```

```
var button1 = document.getElementById('button1');  
button1.addEventListener('click', function(event){  
  target = event.target;  
  alert(target.innerHTML);  
});
```

Giới thiệu bài tập mẫu

Bài 1: Viết chương trình nhập hai số từ người dùng và in ra tổng của chúng.

```
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="utf-8" />  
  <title></title>  
</head>  
<body>  
  <p>  
    <script>  
      let a = parseFloat(prompt("Nhập số thứ nhất:"));  
      let b = parseFloat(prompt("Nhập số thứ hai:"));
```

```
        let sum = a + b;
        document.write("Tổng là: " + sum);
    </script>
</p>
</body>
```

Bài 2: Viết chương trình tìm phần tử lớn nhất trong một mảng.

```
<script>
    let arr = [10, 20, 5, 30, 15];
    let max = arr[0];
    for (let i = 1; i < arr.length; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }
    document.write("Phần tử lớn nhất là: " + max);
</script>
```

Bài 3: Viết chương trình thay đổi nội dung của thẻ <p> khi nhấn vào nút <button>.

Hướng dẫn:

Bước 1: Tạo trang HTML (BT3_DOM_1.html)

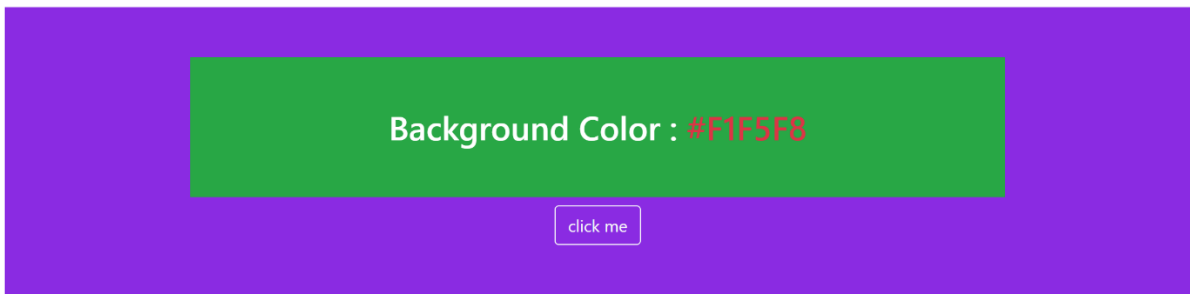
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title></title>
</head>

<body>
    <p id="demo">Xin chào!</p>
    <button onclick="changeText()">Nhấn vào đây</button>
    <script src="BT3_DOM_1.js"></script>
</body>
</html>
```

Bước 2: Viết hàm javascript để xử lý sự kiện Click (BT3_DOM_1.js)

```
function changeText() {
    document.getElementById("demo").innerHTML = "Ngày mới vui vẻ";
}
```

Bài 4: Viết chương trình thay đổi nội dung và định dạng của thẻ HTML khi người dùng nhấn vào nút button.



Yêu cầu:

- Khi người dùng nhấn vào nút Click me. Màu nền sẽ thay đổi ngẫu nhiên và hiển thị code màu lên màn hình (như hình).

Hướng dẫn:

Bước 1: Tạo trang HTML

```
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <link href="css/bootstrap.min.css" rel="stylesheet" />
</head>
<body class="text-center">
  <div class="container mt-5 p-5 text-center"
      style="background-color:blueviolet">
    <h2 class="p-5 w-75 bg-success text-white d-inline-block
        text-capitalize">
      background color :
      <span class="text-danger text-uppercase">#f1f5f8</span>
    </h2>
    <div class="w-100 text-center">
      <button class="btn btn-outline-light" id="btn">
        click me
      </button>
    </div>
  </div>
  <script src="BM4_JS.js"></script>
</body>
</html>
```

Bước 2: Viết javascript để xử lý sự kiện Click

```
const hex = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'A', 'B', 'C', 'D', 'E', 'F'];
let btn = document.getElementById("btn");
const color = document.querySelector("span");

btn.addEventListener("click", function () {
  let hexColor = "#";
  for (let i = 0; i < 6; i++) {
```

```

        hexColor += hex[getRandomNumber()];
    }
    color.textContent = hexColor;
    document.querySelector("div").style.backgroundColor = hexColor;
});

function getRandomNumber() {
    return Math.floor(Math.random() * hex.length);
}

```

Bài 5: Tạo trang hiển thị bộ đếm như hình



Yêu cầu:

- Khi load trang thì số đếm bằng 0.
- Click vào Tăng sẽ tăng số lên 1, click Giảm sẽ giảm số đi 1. Click Đặt lại sẽ quay về 0.
- Nếu số là âm thì hiển thị màu đỏ, số dương hiển thị màu xanh, bằng 0 hiển thị màu đen.

Hướng dẫn:

- Tạo trang HTML

```

<main>
  <div class="container">
    <h1>Bộ đếm</h1>
    <span id="value">0</span>
    <div class="button-container">
      <button class="btn decrease">Tăng</button>

```



```

        <button class="btn reset">Đặt lại</button>
        <button class="btn increase">Giảm</button>
    </div>
</div>
</main>

```

- Thiết kế giao diện hiển thị (tự viết css hoặc sử dụng thư viện bootstrap).
- Tạo eventListener cho các nút

```

let count = 0;
const value = document.querySelector("#value");
const btns = document.querySelectorAll(".btn");

btns.forEach(function (btn) {
    btn.addEventListener("click", function (e) {
        const styles = e.currentTarget.classList;
        if (styles.contains("decrease")) {
            count--;
        } else if (styles.contains("increase")) {
            count++;
        } else {
            count = 0;
        }
        if (count > 0) {
            value.style.color = "green";
        }
        if (count < 0) {
            value.style.color = "red";
        }
        if (count === 0) {
            value.style.color = "#222";
        }
        value.textContent = count;
    });
});

```

6. Bài tập tại lớp

Bài 6: Tạo một ứng dụng To-Do List đơn giản cho phép người dùng thêm và xóa công việc.

- Đi siêu thị X
- Đi học anh văn X
- Đi thư viện X
- Đi đá banh X
- Đi cafe cùng bạn X

Yêu cầu:

- Danh sách công việc được hiển thị trong thẻ với tên công việc là nội dung hiển thị của 1 thẻ
- Nút “Thêm”: thêm nội dung được nhập trong thẻ <input> hiển thị vào danh sách công việc.
- “X”: Xóa công việc tương ứng trong thành phần

Bài 7: Thiết kế trang theo hình như sau:

Danh Sách Công Việc

Yêu cầu:

- Hiển thị danh sách công việc với thông tin liệu kiểu Mảng được lưu trữ trong localStorage (khi mở lại trang web danh sách vẫn không mất đi. Xem thêm tại: <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>).

Danh Sách Công Việc

Đọc Sách

☐
☐

Làm Bài Tập

☐
☐

Xóa Hết

- Khi gõ Công việc và bấm nút Thêm sẽ Thêm vào Mảng trong localStorage.
- Mỗi công việc trong danh sách bên dưới có nút Sửa và Xóa để người dùng có thể thực hiện.

- Có một nút Xóa hết khi nhấn vào sẽ Xóa hết danh sách.

Bài 8: Tạo trang theo mẫu như sau:

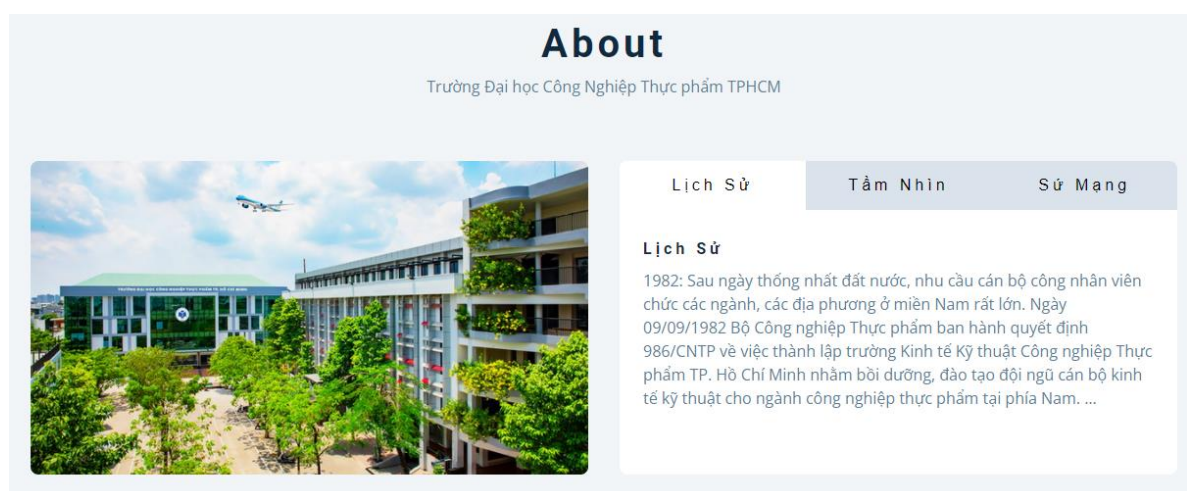


Yêu cầu:

- Phần trả lời ban đầu không hiển thị, khi click vào nút bên phải thì phần trả lời mới hiển thị. Click phần trả lời cho câu hỏi nào thì các phần trả lời cho các câu hỏi khác tự động đóng lại (sử dụng hàm **toggle** kết hợp **remove** để đóng hoặc mở các câu trả lời tương ứng).

7. Bài tập về nhà

Bài 9: Tạo trang như sau:



Yêu cầu:

- Trình bày như hình.

- Click từng Tabs sẽ hiển thị nội dung khác nhau.

Bài 10: Thiết kế trang `Sale_off.html` theo hình như sau:



Yêu cầu:

- Định dạng như hình
- Cho trước 1 ngày giờ kết thúc. Hiển thị đồng hồ đếm ngược (cập nhật mỗi giây) đến ngày kết thúc đó.
- Nếu đã vượt qua ngày giờ kết thúc, hiển thị thông báo kết thúc

