

University of Science, HCMC



Lab: JavaScript

Phát triển ứng dụng Web –JavaScript

JavaScript

Page: 1

Jul 04th, 2020



1 Mục tiêu

Hướng dẫn sinh viên làm quen với ngôn ngữ JavaScript, ứng dụng xử lý một số tương tác cho trang web.

Nguồn tham khảo: <http://htmlcheatsheet.com/js/>

2 Quy định bài thực hành và định dạng nộp bài

- Thời gian thực hiện: **4 tiết**
- Sinh viên sẽ tạo một thư mục webroot có định dạng đặt tên: **<MSSV>_JS**
- Mỗi bài tập sẽ được lưu theo định dạng sau: **<MSSV>_<Tên của file>.js**
- Ví dụ: MSSV = **1234567**, file **Ex01** định dạng nộp bài làm là **1234567_Ex01.js**
- Công cụ sử dụng để lập trình là **Visual Studio Code**
- Link download:
 - o Visual Studio Code: <https://code.visualstudio.com/download>
- Yêu cầu:
 - o Phần resource (tài nguyên): được cung cấp trong thư mục images
 - o **Các trường hợp làm bài không đúng quy định sẽ không được chấp nhận.**

3 Hướng dẫn một số cú pháp cơ bản của JavaScript

3.1 Tạo và thực thi file JavaScript trên môi trường Node.js

- **Tạo file JavaScript làm việc trong Visual Studio Code**

Mở Visual Studio Code > Menu File > Open ... > Chọn thư mục *<MSSV>_JS* > Open
Menu File > New Document > Đặt tên file *<MSSV>_<Tên của file>.js*

- **Thực thi mã lệnh JavaScript trên môi trường Node.js**

Menu View > Terminal (hoặc nhấn tổ hợp phím Ctrl + `)
Gõ lệnh: `$ node <MSSV>_<Tên của file>` + Enter

3.2 Cú pháp cơ bản

- **Khai báo biến và kiểu dữ liệu**

```
var age = 18;           // number
var name = "Jane";      // string
var name = {first:"Jane", last:"Doe"}; // object
var truth = false;      // boolean
var sheets = ["HTML", "CSS", "JS"]; // array
var a; typeof a;         // undefined
var a = null;           // value null
```

- **Cấu trúc điều khiển:**

- **If - Else**

```
if ((age >= 14) && (age < 19)) {
    status = "Eligible.";
} else {
    status = "Not eligible.";
}
```

- **Switch**

```
switch (new Date().getDay()) {  
    case 6:  
        text = "Saturday";  
        break;  
    case 0:  
        text = "Sunday";  
        break;  
    default:  
        text = "Whatever";  
}
```

○ Vòng lặp:

▪ For

```
var sum = 0;  
for (var i = 0; i < a.length; i++) {  
    sum += a[i];  
} // parsing an array
```

▪ While

```
var i = 1;  
while (i < 100) {  
    i *= 2;  
    document.write(i + ", ");  
}
```

▪ Do While

```
var i = 1;  
do {  
    i *= 2;  
    document.write(i + ", ");  
} while (i < 100)
```

○ Khai báo hàm:

```
function addNumbers(a, b) {  
    return a + b; ;  
}  
x = addNumbers(1, 2);
```

- **Arrow Function:**

```
var addNumber = (a, b) => {  
    |   return a + b;  
    }  
x = addNumber(1, 2);
```

- **Đối tượng:**

```
var student = {                                // object name  
    firstName: "Jane",                        // list of properties  
    lastName: "Doe",  
    age: 18,  
    height: 170,  
    fullName : function() {                  // object function  
        return this.firstName + " " + this.lastName;  
    }  
};  
student.age = 19;                             // setting value  
student[age]++;                               // incrementing  
name = student.fullName();                   // call object function
```

- **Định nghĩa thư viện module**

 - Random.js**

```
function random(min, max) {  
    |   return min + Math.random()*(max - min);  
    }  
...  
module.exports = random;
```

Controller.js

```
var controller = {};  
controller.add = (x, y) => x + y;  
controller.subtract = (x, y) => x - y;  
module.exports = controller;
```

Using.js

```
var random = require('./Random');  
console.log(random(0, 10));  
  
var controller = require('./Controller');  
console.log(controller.add(2, 3));  
console.log(controller.subtract(4, 6));
```

3.3 Xử lý bất đồng bộ

○ Sử dụng Callback

```
function add(x, y, callback) {  
    setTimeout(() => {  
        callback (x + y);  
    }, 1000);  
}  
add(2, 3, x => console.log(x));
```

○ Sử dụng Promise

```
function add(x, y) {  
    return new Promise((resolve, reject) => {  
        setTimeout(() => {  
            resolve(x + y);  
        }, 1000);  
    });  
}  
add(2, 3).then(x => console.log(x));
```

○ Sử dụng Async – Await

```
async function asyncThing(value) {
    return new Promise((resolve, reject) => {
        setTimeout(() => resolve(value), 1000);
    });
}

async function add(x, y) {
    return await asyncThing(x + y);
}

(async function () {
    let x = await add(2, 3);
    console.log(x);
})();
```

4 Hướng dẫn làm quen với Javascript

4.1 Yêu cầu

- Với mỗi bài tập bên dưới, tạo tập tin html với tên của bài tương ứng. Ví dụ nếu bài có tên là **Ex01-Main**, mã số sinh viên **1234567** thì tạo tập tin tên **1234567_Ex01-Main.js** và gõ lại nội dung như được cung cấp.

4.2 Hướng dẫn thực hiện

4.2.1 Ex01-Main

Định nghĩa một mảng đối tượng chứa danh sách đối tượng sinh viên, mỗi sinh viên lưu thông tin mã số sinh viên, họ tên và mảng danh sách các đối tượng môn học, trong đó mỗi môn học lưu thông tin tên môn học và điểm số. Mặc định điểm tất cả các môn học là 0 điểm. Sử dụng lệnh console.log và JSON.stringify để xuất thông tin danh sách sinh viên đã định nghĩa ra màn hình.

4.2.2 Ex02-Random

Định nghĩa module Ex02-Random trả về hàm random(min, max) phát sinh ngẫu nhiên một số thực trong khoảng [min, max].

4.2.3 Ex03-StudentController

Định nghĩa module Ex03-StudentController, trả về một object có chứa các hàm nhận tham số đầu vào là một mảng danh sách các sinh viên theo cấu trúc đã được định nghĩa trong module Ex01-Main, trong đó:

1. Hàm `generate(students)` xử lý phát sinh điểm ngẫu nhiên cho tất cả các SV trong danh sách sử dụng hàm phát sinh ngẫu nhiên đã định nghĩa trong module `Ex02-Random`.
2. Hàm `average(students)` xử lý bổ sung thuộc tính điểm trung bình cho mỗi SV.
3. Hàm `normalize(students)` xử lý định dạng điểm của SV theo dạng số thực có 2 chữ số thập phân.

Lần lượt gọi thực thi các hàm đã định nghĩa trong file `Ex01-Main` và hiển thị kết quả thông tin của tất cả sinh viên.

4.2.4 Ex04-SlowRandom

Định nghĩa module Ex04-SlowRandom xử lý giống Ex02-Random nhưng chỉ trả về giá trị sau thời gian 1 giây.

4.2.5 Ex05-SlowStudentController

Định nghĩa module Ex05-SlowStudentController xử lý giống Ex03-StudentController nhưng gọi hàm SlowRandom để phát sinh điểm ngẫu nhiên cho SV

Lần lượt gọi thực thi các hàm đã định nghĩa trong file Ex01-Main và hiển thị kết quả thông tin của tất cả sinh viên.