

BÀI GIẢNG

THIẾT KẾ WEB

Chương 4. JavaScript (P1)

Nguyễn Thành Thủy

BM. Tin học quản lý, Khoa Thống kê Tin học.

Trường Đại học Kinh tế, Đại học Đà Nẵng

e-Mail: thuynt@due.edu.vn

NỘI DUNG

1. Tổng quan về JavaScript
2. Dữ liệu kiểu mảng
3. Cấu trúc điều kiện
4. Cấu trúc lặp
5. Hàm trong Javascript
6. Nhúng JavaScript vào HTML
7. HTML DOM
8. HTML DOM CSS



TỔNG QUAN VỀ JAVASCRIPT

JavaScript - Tổng quan về Javascript

❑ Tổng quan:

- ❑ Là ngôn ngữ được xử lý trên Client (Client-Side);
- ❑ Được sử dụng để tương tác với người dùng trên trang web;
- ❑ Code Javascript có thể được lồng vào các thẻ HTML;
- ❑ Javascript đòi hỏi sự tương thích với từng phiên bản web browser khác nhau.

JavaScript - Tổng quan về Javascript

❑ Cú pháp lệnh trong JavaScript:

- ❑ Mỗi câu lệnh kết thúc bằng dấu chấm phẩy ; (*không bắt buộc*);
- ❑ Khối lệnh được đặt trong cặp dấu { } ;
- ❑ Tên định danh và từ khóa có phân biệt chữ hoa, chữ thường;
- ❑ Sử dụng cặp dấu // hoặc /* */ để tạo ghi chú (Comment).

JavaScript - Tổng quan về Javascript

❑ Xuất kết quả trong JavaScript

❑ InternalHTML: viết vào một phần tử HTML

```
<p id="demo"></p>
```

```
<script>
```

```
    document.getElementById("demo").innerHTML = 5 + 6;
```

```
</script>
```

❑ document.write(): viết trực tiếp lên màn hình

```
<button type="button" onclick="document.write(5 + 6)">
```

```
    Try it
```

```
</button>
```

JavaScript - Tổng quan về Javascript

❑ Xuất kết quả trong JavaScript

- ❑ **window.alert()**: viết vào hộp cảnh báo

```
<script>  
    window.alert(5 + 6);  
</script>
```

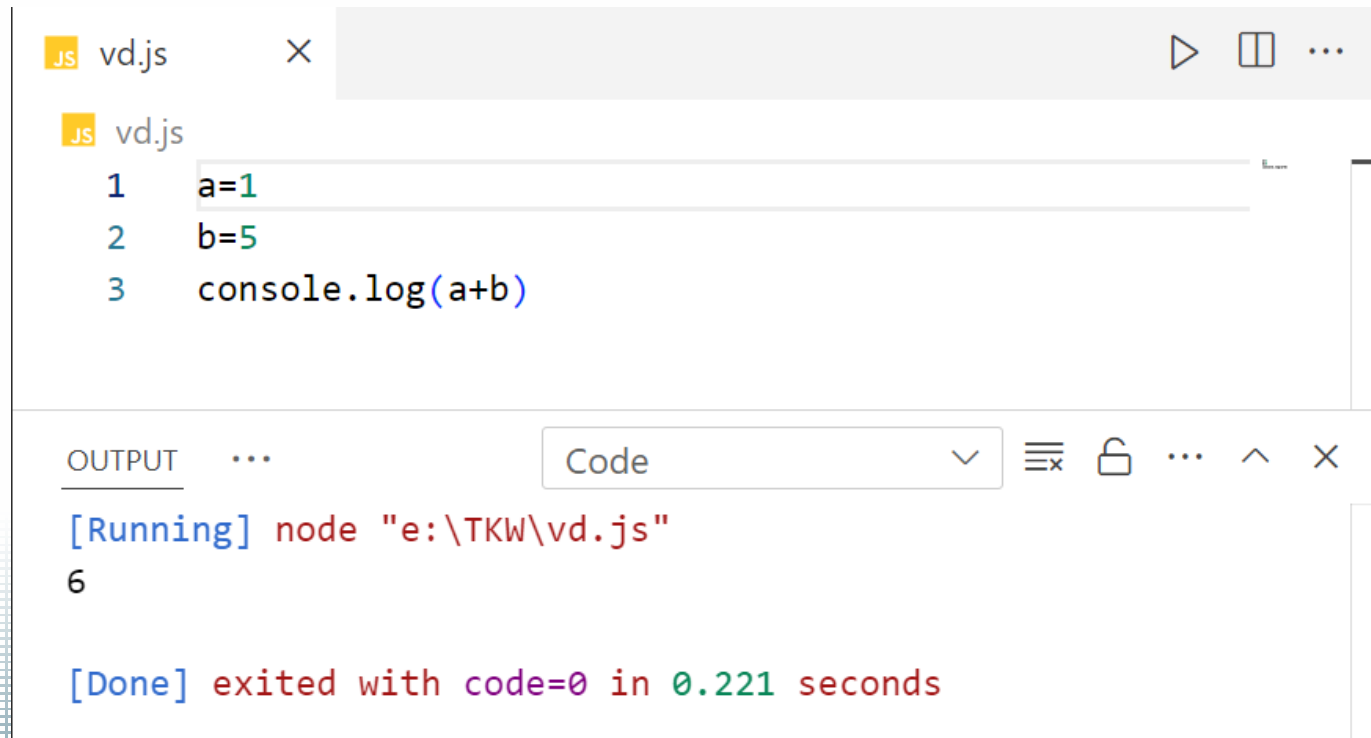
- ❑ **console.log()**: viết vào bảng điều khiển trình duyệt

```
<script>  
    console.log(5 + 6);  
</script>
```

JavaScript - Tổng quan về Javascript

❑ Hướng dẫn chạy file .js trên VsCode

- ❑ Cài đặt **Note.js**: <https://nodejs.org/en>
- ❑ Trên VsCode cài Extension **Code Runner** của **Jun Han**



```
JS vd.js × [Run] [Debug] [More]
```

```
JS vd.js
1 a=1
2 b=5
3 console.log(a+b)
```

OUTPUT ... Code [Menu] [Lock] [More] [Up] [Close]

```
[Running] node "e:\TKW\vd.js"
6

[Done] exited with code=0 in 0.221 seconds
```


JavaScript - Tổng quan về Javascript

❑ Khai báo biến

- ❑ Chế độ tự động nhận diện (*không khuyến khích*)

```
x = 5;  
y = 6;  
z = x + y;
```

- ❑ Khai báo bằng từ khóa **var** (*không tương thích với mọi trình duyệt*)

```
var x = 5;  
var y = 6;  
var z = x + y;
```

- ❑ Khai báo bằng từ khóa **let** (*hỗ trợ tốt với các trình duyệt web hiện hành*)

```
let x = 5;  
let y = 6;  
let z = x + y;
```

- ❑ Khai báo bằng từ khóa **const** (*sử dụng khi khai báo hằng*)

```
const x = 5;  
const y = 6;  
const z = x + y;
```

JavaScript - Tổng quan về Javascript

❑ Khai báo biến

❑ Khai báo đồng thời nhiều biến

```
let person = "John Doe", carName = "VinFast", price = 200;
```

```
let person = "John Doe",  
    carName = "VinFast",  
    price = 200;
```

❑ Có thể sử dụng dấu \$ để đặt tên cho biến

```
let $ = "Hello World";  
let $$$ = 2;  
let $myMoney = 5;
```

JavaScript - Tổng quan về Javascript

- ❑ **Ví dụ 1**, Viết chương trình JavaScript để tính diện tích của hình chữ nhật.

```
let width = 5;  
let height = 10;  
let area = width * height;  
console.log("Diện tích hình chữ nhật là: " + area);
```

JavaScript - Tổng quan về Javascript

- ❑ **Ví dụ 2**, Viết chương trình JavaScript để tính chu vi và diện tích của hình tròn với bán kính r đã biết.

```
let bankinh = 7;  
let chuvi = 2 * Math.PI * bankinh;  
let dientich = Math.PI * bankinh * bankinh;  
  
console.log("Chu vi hình tròn là: " + chuvi);  
console.log("Diện tích hình tròn là: " + dientich);
```

CẤU TRÚC ĐIỀU KIỆN

JavaScript – Cấu trúc điều kiện

□ Dạng 1

```
if (hour < 18) {  
    greeting = "Good day";  
}
```

□ Dạng 2

```
if (hour < 18) {  
    greeting = "Good day";  
}  
else {  
    greeting = "Good evening";  
}
```

JavaScript – Cấu trúc điều kiện

□ Dạng 3

```
if (time < 10) {  
    greeting = "Good morning";  
}  
else if (time < 20) {  
    greeting = "Good day";  
}  
else {  
    greeting = "Good evening";  
}
```

JavaScript – Cấu trúc điều kiện

- ❑ **Ví dụ 1**, Viết chương trình JavaScript để kiểm tra xem một số nguyên n là số chẵn hay số lẻ.

```
let n = 7;
```

```
if (n % 2 === 0) {  
    console.log("Số " + n + " là số chẵn");  
} else {  
    console.log("Số " + n + " là số lẻ");  
}
```


JavaScript – Cấu trúc điều kiện

- ❑ **Ví dụ 2**, Viết chương trình JavaScript để xếp loại học lực của học sinh dựa trên điểm trung bình average.

```
let $dtb = 7.2;
```

```
if ($dtb >= 8) {  
    console.log("Học lực Giỏi");  
} else if ($dtb >= 6.5) {  
    console.log("Học lực Khá");  
} else if ($dtb >= 5) {  
    console.log("Học lực Trung bình");  
} else {  
    console.log("Học lực Yếu");  
}
```

BÀI TẬP THỰC HÀNH 1

CẤU TRÚC LẬP

JavaScript – Cấu trúc lặp for ... loop

❑ Cú pháp - Dạng 1

```
for (expression 1; expression 2; expression 3) {  
    // code block to be executed  
}
```

❑ Ví dụ 1: in lên màn hình dãy thông báo theo mẫu

```
for (let i = 0; i < 5; i++) {  
    text = "The number is " + i;  
    console.log(text);  
}
```

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4

❑ Ví dụ 2: in lên màn hình tổng các số nguyên chẵn trong dãy số từ 1 đến 10

```
sum=0  
for (let x = 1; x <= 10; x++)  
    if (x%2==0)  
        sum+=x  
console.log(sum)
```

30

JavaScript – Cấu trúc lặp for ... in

❑ Cú pháp - Dạng 2

```
for (key in object) {  
    // code block to be executed  
} // key nhận về giá trị index của object
```

- ❑ Ví dụ: in lên màn hình tổng giá trị các phần tử trong một mảng

```
const numbers = [45, 4, 9, 16, 25];  
let n = 0;  
for (let x in numbers) {  
    n += numbers[x];  
}  
console.log(n)
```

99

JavaScript – Cấu trúc lặp while

❑ Cú pháp

```
while (condition) {  
    // code block to be executed  
}
```

❑ Ví dụ: in lên màn hình dãy thông báo theo mẫu

```
let i=1;  
while (i < 10) {  
    console.log("The number is " + i);  
    i++;  
}
```

The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9

JavaScript – Cấu trúc lặp do ... while

❑ Cú pháp

```
do {  
    // code block to be executed  
}  
while (condition);
```

❑ Ví dụ: in lên màn hình dãy thông báo theo mẫu

```
let i=0;  
do {  
    text = "The number is " + i;  
    console.log(text);  
    i++;  
}  
while (i < 10);
```

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9

BÀI TẬP THỰC HÀNH 2

DỮ LIỆU KIỂU MẢNG

JavaScript – Dữ liệu kiểu mảng (Array)

❑ Khai báo mảng

```
const array_name = [item1, item2, ...];
```

❑ Ví dụ 1:

```
const cars = ["Mazda", "VinFast", "BMW"];
```

❑ Ví dụ 2:

```
const cars = [];  
cars[0] = "Mazda";  
cars[1] = "VinFast";  
cars[2] = "BMW";
```

JavaScript – Dữ liệu kiểu mảng (Array)

❑ Cập nhật giá trị của mảng

```
const cars = ["Mazda", "Toyota", "BMW"];  
cars[0]="VinFast";  
console.log(cars)
```

```
[ 'VinFast', 'Toyota', 'BMW' ]
```

❑ Thêm phần tử cho mảng

```
const fruits = ["Banana", "Orange", "Apple"];  
fruits.push("Lemon");  
console.log(fruits)
```

```
[ 'Banana', 'Orange', 'Apple', 'Lemon' ]
```

❑ Đếm số phần tử trong mảng

```
const fruits = ["Banana", "Orange", "Apple"];  
console.log(fruits.length)
```

```
3
```

JavaScript – Dữ liệu kiểu mảng (Array)

- ❑ **Ví dụ 1**, Viết chương trình JavaScript sử dụng mảng để lưu các số và tính tổng của các phần tử trong mảng.

```
let numbers = [1, 2, 3, 4, 5];  
let sum = 0;
```

```
for (let i = 0; i < numbers.length; i++) {  
    sum += numbers[i];  
}
```

```
console.log("Tổng các phần tử trong mảng là: " + sum);
```

Tổng các phần tử trong mảng là: 15

JavaScript – Dữ liệu kiểu mảng (Array)

❑ **Ví dụ 2**, viết chương trình trên javascript thực hiện các yêu cầu sau:

- Khởi tạo 1 mảng gồm 3 số nguyên;
- Cập nhật giá trị của phần tử thứ hai trong mảng có giá trị bằng 5;
- Thêm một phần tử mới vào cuối mảng, có giá trị bằng 10;
- Đếm và in ra số lượng phần tử hiện có trong mảng;
- Tính tổng giá trị của các phần tử trong mảng, in kết quả lên màn hình.

```
const m=[1,2,3]
```

```
m[1]=5
```

```
m.push(10)
```

```
console.log("Số phần tử: " + m.length)
```

```
let s=0
```

```
for (let i=0; i<m.length; i++)
```

```
    s+=m[i]
```

```
console.log("Tổng = " + s)
```

Số phần tử: 4

Tổng = 19

BÀI TẬP THỰC HÀNH 3

HÀM TRONG JAVASCRIPT

JavaScript – Hàm trong Javascript

❑ Khai báo và gọi hàm

```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

❑ Ví dụ:

```
function myFunction(a, b) {  
    // Function returns the product of a and b  
    return a * b;  
}
```

```
// Function is called, the return value will end up in x  
let x = myFunction(4, 3);
```


JavaScript – Hàm trong Javascript

❑ Tham số mặc định của hàm

❑ Ví dụ:

```
function sum(x, y = 5) {  
    console.log(x + y);  
}
```

```
sum(5); // 10
```

```
sum(5, 15); // 20
```

JavaScript – Hàm trong Javascript

❑ Khai báo và gọi hàm trong ES6 (Arrow Function)

❑ Ví dụ 1, hàm không có tham số:

```
hello = () => {  
    return "Hello World!";  
}  
console.log(hello());
```

Hàm truyền thống

```
function hello(){  
    return "Hello World!";  
}  
console.log(hello());
```

❑ Ví dụ 2, hàm có tham số:

```
x = (a, b) => {  
    return a*b;  
}  
console.log(x(5,10))
```

```
function x(a,b){  
    return a*b  
}  
console.log(x(5,10))
```

* **ES6: ECMAScript 6**, được Hiệp hội sản xuất máy tính Châu Âu đề xuất để làm tiêu chuẩn cho ngôn ngữ lập trình JavaScript.

JavaScript – Hàm trong Javascript

❑ Ví dụ:

```
const m=[1,-2,0,3,-3,5,8,-6]
```

```
demSoDuong=(m)=>
```

```
{
```

```
  s=0
```

```
  for (let i in m)
```

```
    if (m[i]>0)
```

```
      s++
```

```
  return s
```

```
}
```

```
console.log("So phan tu duong: " + demSoDuong(m))
```

So phan tu duong: 4

BÀI TẬP THỰC HÀNH 4