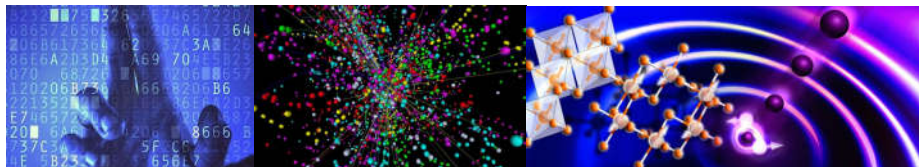


THIẾT KẾ WEB

BỘ MÔN KHOA HỌC MÁY TÍNH



Chương 4.

JAVASCRIPT

1. Giới thiệu JavaScript
2. Cách sử dụng JavaScript
3. Các thành phần của Javascript
4. Một số đối tượng trong Javascript
5. Xử lý sự kiện trong Javascript

1. Giới thiệu Javascript

- JavaScript là ngôn ngữ lập trình phổ biến nhất thế giới
- JavaScript là ngôn ngữ lập trình cho Web
- JavaScript dễ học
- JavaScript có thể thay đổi HTML Content

```
<body>
<h2>JavaScript có thể thay đổi HTML Content</h2>
<p id="demo">JavaScript có thể thay đổi HTML Content</p>
<button type="button" onclick="document.getElementById('demo').innerHTML='Hello
JavaScript!'"> Click Me to Change Content!</button>
</body>
```

file:///D:/02. NAM HOC 2020-

JavaScript có thể thay đổi HTML Content

Hello JavaScript!

Click Me to Change Content!

3

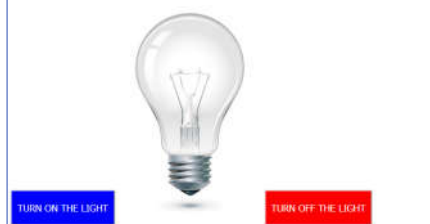
1. Giới thiệu Javascript

- JavaScript có thể thay đổi giá trị thuộc tính trong HTML

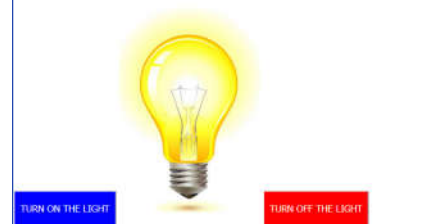
```
<body>
<h2>JavaScript có thể thay đổi giá trị thuộc tính trong HTML</h2>
<input type="button" value="TURN ON THE LIGHT" style="height:50px; background-color:#00F; color:white;" onclick=
"document.getElementById('myImage').src='Images/LightOn.JPG'" />

<input type="button" value="TURN OFF THE LIGHT" style="height:50px; background-color:#F00; color:#FFF;" onclick=
"document.getElementById('myImage').src='Images/LightOff.JPG'" />
</body>
```

JavaScript có thể thay đổi giá trị thuộc tính trong HTML



JavaScript có thể thay đổi giá trị thuộc tính trong HTML



4

2. Cách sử dụng

- Trong HTML, javascript được đặt trong cặp thẻ `<script>` và `</script>`.
- Các hàm trong Javascript là một khối mã lệnh mà có thể được thực thi khi nó được gọi. (một hàm có thể được gọi khi một sự kiện xảy ra ví dụ người dùng click vào button)
- Hàm JavaScript có thể được đặt trong `<body>` hoặc trong `<head>`

<pre> <!DOCTYPE html> <html> <body> <h1>A Web Page</h1> <p id="demo">A Paragraph</p> <button type="button" onclick="myFunction()">Try it</button> <script> function myFunction() { document.getElementById("demo").innerHTML = "Paragraph changed."; } </script> </body> </pre>	<pre> <!DOCTYPE html> <html> <head> <script> function myFunction() { document.getElementById("demo").innerHTML = "Paragraph changed."; } </script> </head> <body> <h1>A Web Page</h1> <p id="demo">A Paragraph</p> <button type="button" onclick="myFunction()">Try it</button> </body> </html> </pre>
--	--

5

2. Cách sử dụng

- Javascript có thể được tổ chức trong một file độc lập bên ngoài **myScript.js**

```
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```
- Sử dụng đoạn script bên ngoài

```
<script src="myScript.js"></script>
```
- Một số ích lợi của external Script
 - Tách biệt code HTML và JS
 - Dễ đọc hơn và dễ quản lý hơn
 - Các file js có thể được cache giúp page load nhanh hơn.

6

3. Các thành phần của JavaScript

- 3.1. JavaScript Output
- 3.2. Syntax
- 3.3. Conditional Expression
- 3.4. JS Loop (For; For in; While; break)

7

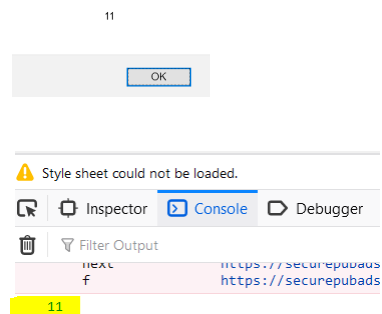
3.1. JavaScript Output

- Sử dụng `document.write()` :

```
<script>
document.write(5 + 6);
</script>
```
- Sử dụng `window.alert()` :

```
<script>
window.alert(5 + 6);
</script>
```
- Sử dụng `console.log()` :

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



8

3.2. Syntax

- Kết thúc câu lệnh JavaScript có thể có dấu chấm phẩy (;) hoặc không. Tuy nhiên, khuyến cáo luôn sử dụng.
- Tạo chú thích

```

</div>

<script>
// Comment một dòng lệnh
/*
Comment một khối lệnh
*/
function myFunction() {
    document.getElementById("panel").style.display = "block";
}

```

9

3.2. Syntax

- Các kiểu dữ liệu cơ bản

Kiểu dữ liệu	Ví dụ	Giải thích
Kiểu số	3 8.7 -17	Biểu diễn kiểu số nguyên và số thực
Kiểu chuỗi	"Hello World" '100'	Biểu diễn chuỗi, có thể sử dụng dấu nháy đơn hoặc kép để biểu thị chuỗi trong js
Boolean	true false	Biểu diễn dữ liệu đúng hoặc sai
Null	null	đại diện cho giá trị không có gì trong js
undefined	undefined	tương tự, nhưng biểu thị cho giá trị chưa được định nghĩa

10

3.2. Syntax

- Biến và hằng

Khai báo	Ví dụ	Giải thích
let	let test; let test = 5;	Khai báo biến test, và nó mang giá trị undefined. Khai báo biến test và gán cho nó giá trị 5. Biến khai báo bằng let có phạm vi trong một block { }
var	var test; var test = 5;	Tương tự như let. Tuy nhiên, biến khai báo bằng var có phạm vi trong một hàm.
const	const test = 5;	Khai báo hằng số

11

3.3. Câu lệnh rẽ nhánh

- So sánh. Giả sử: x = 5

Toán tử	Tên	Ví dụ	Kết quả
>	lớn hơn	x > 20	false
>=	lớn hơn hoặc bằng	x >= 5	true
<	nhỏ hơn	x < -100	false
<=	nhỏ hơn hoặc bằng	x <= 25	true
==	bằng (tự động thay đổi kiểu dữ liệu)	x == '5'	true
!=	không bằng	x != 'a'	true
===	bằng (kiểm tra cả giá trị và kiểu dữ liệu)	x === '5'	false
!==	khác giá trị hoặc kiểu dữ liệu	x !== '5'	true

12

3.3. Câu lệnh rẽ nhánh

- Toán tử Logic. Giả sử: $x = 5$ và $y = 10$

Toán tử	Tên	Ví dụ	Kết quả
&&	AND	$x < 20 \ \&\& \ x !== 5$	false
	OR	$y > 10 \ \ x === 5$	true
!	NOT	$!(x === y)$	true

13

3.3. Câu lệnh rẽ nhánh

- Câu lệnh if

```

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

```

- Ví dụ

```

let age = 18;
if (age < 18) {
  alert("Bạn chưa đủ tuổi");
} else {
  alert("Mời vào !!!");
}

```

14

3.3. Câu lệnh rẽ nhánh

- Switch được sử dụng để biểu diễn các hành động khác nhau dựa trên các điều kiện khác nhau

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

15

3.4. Vòng lặp

- Vòng lặp for

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

- Ví dụ

```
let str = '';

for (let i = 0; i < 9; i++) {
  str = str + i;
}

console.log(str);
// output: "012345678"
```

16

3.4. Vòng lặp

- Vòng lặp for ... of

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

- Ví dụ

```
const iterable = [10, 20, 30];  
  
for (let value of iterable) {  
  value += 1;  
  console.log(value);  
}  
// 11  
// 21  
// 31
```

17

3.4. Vòng lặp

- Vòng lặp for ... in

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

- Ví dụ

```
var obj = {a: 1, b: 2, c: 3};  
  
for (const prop in obj) {  
  console.log(prop);  
}  
  
// Output:  
// a  
// b  
// c
```

18

3.4. Vòng lặp

- Vòng lặp while

```
while (condition)
    statement
```

- Ví dụ

```
var n = 0;
var x = 0;

while (n < 3) {
    n++;
    x += n;
}
```

19

3.4. Vòng lặp

- Vòng lặp do ... while

```
do
    statement
while (condition);
```

- Ví dụ

```
var result = '';
var i = 0;
do {
    i += 1;
    result += i + ' ';
}
while (i > 0 && i < 5);
// Mặc dù i == 0 vòng lặp vẫn thực hiện trước khi thực hiện kiểm tra
```

20

3.4. Vòng lặp

- break: thoát khỏi vòng lặp ngay lập tức
- Ví dụ

```
let i = 0;

while (i < 6) {
  if (i === 3) {
    break;
  }
  i = i + 1;
}

console.log(i);
// output: 3
```

21

3.4. Vòng lặp

- continue: dừng vòng lặp hiện tại và thực hiện vòng lặp mới.
- Ví dụ

```
let text = '';

for (let i = 0; i < 10; i++) {
  if (i === 3) {
    continue;
  }
  text = text + i;
}

console.log(text);
// output: "012456789"
```

22

3.5. Hàm

- Khai báo hàm

```
function fnName(param1 = defaultValue1, ..., paramN = defaultValueN) { /* ... */ }
```

- Ví dụ

```
function multiply(a, b = 1) {
  return a * b
}

multiply(5, 2)      // 10
multiply(5)         // 5
```

23

3.6. Mảng

- Tạo mảng

```
let fruits = ['Apple', 'Banana']
console.log(fruits.length)
// 2
```

- Truy xuất mảng sử dụng chỉ số

```
let first = fruits[0]
// Apple

let last = fruits[fruits.length - 1]
// Banana
```

24

3.6. Mảng

- Lặp qua các phần tử

```
fruits.forEach(function(item,
index, array) {
    console.log(item, index)
})
// Apple 0
// Banana 1
```

25

3.6. Mảng

- Một số phương thức

Phương thức	Diễn giải	Ví dụ
push	Thêm phần tử vào cuối mảng	fruits.push('Orange')
pop	Xóa phần tử cuối mảng	fruits.pop()
shift	Xóa phần tử đầu mảng	fruits.shift()
unshift	Thêm phần tử đầu mảng	fruits.unshift('Strawberry')
indexOf	Tìm chỉ số phần tử của mảng	fruits.indexOf('Banana')
splice	Xóa phần tử bằng chỉ số	fruits.splice(index, 1)
slice	Copy một mảng	fruits.slice()

26

4. Thao tác với DOM

- DOM (Document Object Model) là giao diện cho phép sử dụng JavaScript với HTML và CSS. Trình duyệt biến mọi phần tử HTML thành một đối tượng JavaScript được lưu trữ trong đối tượng tài liệu và có thể truy cập trong mã JavaScript.
- 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';
```

27

4. Thao tác với DOM

- Một số phương thức chọn phần tử DOM

Phương thức	Diễn giải
<code>var element = document.getElementById(id);</code>	Chọn phần tử theo ID
<code>var elements = document.getElementsByTagName(name);</code>	Chọn tất cả các phần tử theo Tên thẻ HTML
<code>var elements = document.getElementsByClassName(names);</code>	Chọn tất cả các phần tử theo Tên class
<code>var element = document.querySelector(selectors);</code>	Chọn phần tử đầu tiên thỏa theo các thức chọn CSS Selector
<code>var elementList = parentNode.querySelectorAll(selectors);</code>	Chọn tất cả phần tử thỏa theo các thức chọn CSS Selector

28

4. Thao tác với DOM

- Tạo/Xóa Node HTML

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

```
document.body.removeChild(div1);
```

29

4. Thao tác với DOM

- Quản lý style với DOM

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

30

4. Thao tác với DOM

- Xử lý Sự kiện

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

- Xem thêm các sự kiện tại:
https://www.w3schools.com/jsref/dom_obj_event.asp

31

4. Thao tác với DOM

- Page Load Event

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

- MouseOver Event

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

32

4. Thao tác với DOM

- Điều khiển form

```
<form id='form'>
  Name: <input type='text'>
  <input type='submit' value='Save'>
</form>
```

```
var form = document.getElementById('form');
form.addEventListener('submit', function(event){
  event.preventDefault()
  var textInput = form.elements[0]
  alert(textInput.value)
});
```