



503073

WEB PROGRAMMING & APPLICATIONS

LESSON 5: JAVASCRIPT

Instructor: Mai Van Manh

Outline

- Basic Javascript
- Document Object Model
- Form Handling
- Callback
- Ajax
- Web API

Javascript Introduction

Introduction to Javascript

1. JavaScript is one of the **most popular** programming language.
2. JavaScript was initially created to “**make web pages alive**”.
3. When JavaScript was created, it initially had another name:
“**LiveScript**”.
4. Javascript can be used for both **front-end** and **back-end** web development.

Javascript in client side

- Add or remove HTML Elements.
- Add or remove/change HTML Element attributes.
- Change HTML Element content.
- Change HTML Style.
- Form validation.
- Load data from server.
- Post data to server.

Javascript in server side

- Create a web server (like apache)
- Access file system
- Access OS features
- Create a command line program
- And more

Three ways to use Javascript in HTML

1. Inline Javascript
2. Internal Javascript
3. External Javascript

Inline Javascript

Inline Javascript

- Javascript code can be written directly on an HTML element via the **on[event]** attributes such as: **onclick**, **oninput**, **onfocus**....

```
1 <button onclick="alert('Bạn đã nhấn vào button')">Click</button>
```

127.0.0.1:5500 says

Bạn đã nhấn vào button

OK

Internal Javascript

Inline Javascript

- Internal javascript is the js code written inside the `<script>` tag, these tags are usually placed inside the `<head>` tag or before the `</body>` tag.

```
1 <body>
2     <h4>Xin chào</h4>
3     <script>
4         let h4 = document.querySelector('h4');
5         h4.innerHTML = 'Xin chào mọi người';
6     </script>
7 </body>
```

External Javascript

External Javascript

- External javascript is js code written in a separate file with the extension ***.js**. This file is then referenced from HTML via the `<script>` tag.



main.js

```
1 let h4 = document.querySelector('h4');  
2 h4.innerHTML = 'Xin chào mọi người';
```



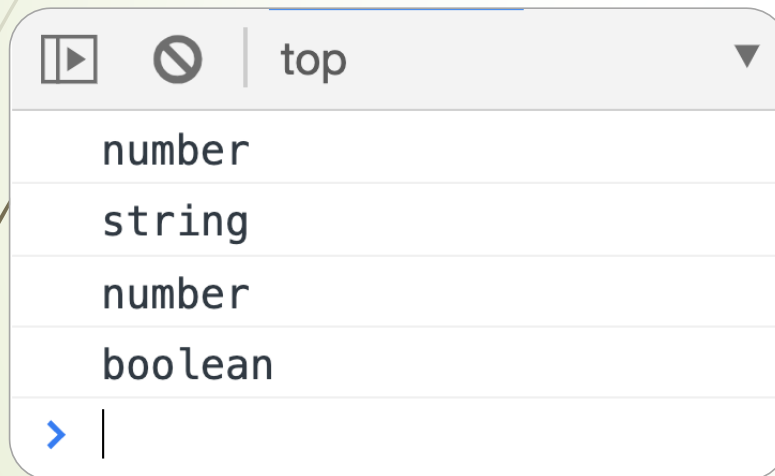
index.html

```
1 <body>  
2     <h4>Xin chào</h4>  
3     <script src="main.js"></script>  
4 </body>
```

Basic Syntax Examples

Data types

- JavaScript variables can hold many data types: numbers, strings, objects and more.



```
main.js

1 let x = 20;
2 console.log(typeof x);
3
4 x = 'Hello';
5 console.log(typeof x);
6
7 x = 2.5;
8 console.log(typeof x);
9
10 x = true;
11 console.log(typeof x);
```

== VS ===



```
1 console.log(3 == 3)           // true
2 console.log(3 == '3')        // true
3 console.log('' == 0)         // true
4 console.log('' == false)     // true
5 console.log(false == 0)      // true
```


== VS ===



```
1 console.log(3 === 3)           // true
2 console.log(3 === '3')         // false
3 console.log('' === 0)          // false
4 console.log('' === false)      // false
5 console.log(false === 0)       // false
```

String

main.js

```
1 let myName = "nguyễn văn a";  
2  
3 console.log(myName); // nguyễn văn a  
4 console.log(myName.length); // 12  
5 console.log(myName.toUpperCase()); // NGUYỄN VĂN A  
6 console.log(myName.indexOf('văn')); // 7  
7 console.log("I am " + myName); // I am nguyễn văn A
```

Variable and if-else statement



main.js

```
1 let age = 10;  
2 age = age * 2;  
3 age ++;  
4  
5 if (age > 25) {  
6     console.log( 'Age > 25' )  
7 }else {  
8     console.log( 'Age <= 25' );  
9 }
```

Constant and for/while loop



main.js

```
1 const numbers = [1,5,8,93,15];
2
3 for (let i = 0; i < numbers.length; i++) {
4     console.log(numbers[i]);
5 }
6
7 let index = 0;
8 while (index < numbers.length) {
9     console.log(numbers[index]);
10    index ++;
11 }
```

Function

```
main.js
1 function createArray(num) {
2     let arr = [];
3     for (let i = 1; i <= num; i++) {
4         arr.push(i * i);
5     }
6     return arr;
7 }
8
9 function printArray(arr) {
10     for (let i = 0; i < arr.length; i++) {
11         console.log(arr[i]);
12     }
13 }
14
15 let myArray = createArray(5);
16 printArray(myArray);
```



main.js

```
1 function createArray(num = 10) {  
2     let arr = [];  
3     for (let i = 1; i <= num; i++) {  
4         arr.push(i * i);  
5     }  
6     return arr;  
7 }  
8  
9 function printArray(arr) {  
10     arr.forEach(function(item) {  
11         console.log(item);  
12     });  
13 }  
14  
15 let myArray = createArray();  
16 printArray(myArray);
```

Callback Function

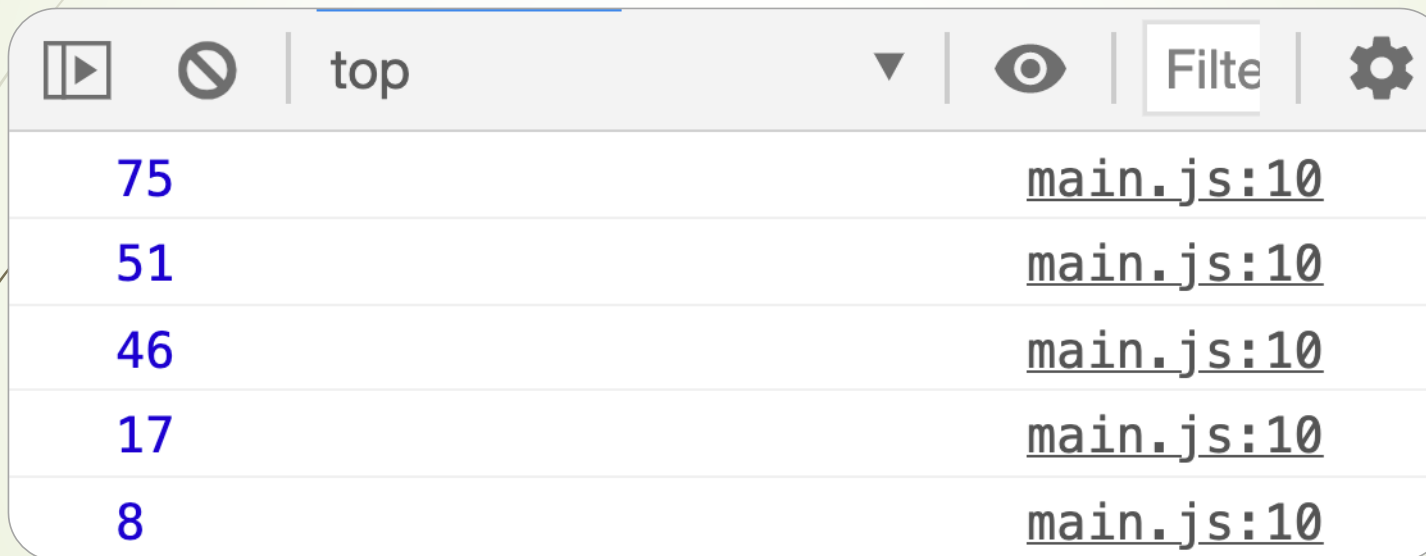
Callback function

- A callback is a function passed as an argument to another function.

```
main.js

1 function generateAndProcessArray(size, processor) {
2     let arr = [];
3     for (let i = 1; i <= size; i++) {
4         arr.push(Math.round(Math.random() * 100))
5     }
6     processor(arr);
7 }
8 function printArray(arr) {
9     for (let i = 0; i < arr.length; i++) {
10         console.log(arr[i]);
11     }
12 }
13 generateAndProcessArray(5, printArray);
```


Callback function



75	<u>main.js:10</u>
51	<u>main.js:10</u>
46	<u>main.js:10</u>
17	<u>main.js:10</u>
8	<u>main.js:10</u>

Callback function



main.js

```
1 function alertSumOfArray(arr) {  
2     let sum = 0;  
3     for (let i = 0; i < arr.length; i++) {  
4         sum += arr[i];  
5     }  
6     alert("Sum = " + sum);  
7 }  
8 generateAndProcessArray(5, alertSumOfArray);
```

127.0.0.1:5500 says

Sum = 279

OK

Arrow function

Arrow function

- Arrow functions allow us to write shorter function syntax.



```
1 function square(n) {  
2     return n * n  
3 }
```



```
1 let square = (n) => {  
2     return n * n  
3 }
```

Arrow function

- If the function has only one statement, and the statement returns a value, you can remove the brackets and the return keyword.



```
1 function square(n) {  
2     return n * n  
3 }
```



```
1 let square = (n) => n * n
```



```
1 let square = n => n * n
```

Arrow function as a callback

```
1 let arr = [1, 4, 9, 3, 5]
2 // normal callback function
3 arr.forEach(function(item) {
4     console.log(item)
5 })
```

```
1 let arr = [1, 4, 9, 3, 5]
2 // arrow callback function
3 arr.forEach((item) => {
4     console.log(item)
5 })
```

Reduced Arrow function



```
1 let arr = [1, 4, 9, 3, 5]
2 // normal callback function
3 arr.forEach(function(item) {
4     console.log(item)
5 })
```



```
1 let arr = [1, 4, 9, 3, 5]
2 // reduced arrow function
3 arr.forEach(item => console.log(item))
```

setTimeout

setTimeout

- Executes a function, after waiting a specified number of milliseconds.

setTimeout(*function*, *milliseconds*);



main.js

```
1 function task() {  
2     console.log('Câu lệnh này chạy cuối cùng, sau 1s');  
3 }  
4  
5 console.log('Câu lệnh này chạy đầu tiên');  
6  
7 setTimeout(task, 1000);  
8  
9 console.log('Câu lệnh này chạy thứ hai');
```

setTimeout v2



main.js

```
1 console.log('Câu lệnh này chạy đầu tiên');  
2  
3 setTimeout(function() {  
4     console.log('Câu lệnh này chạy cuối cùng, sau 1s');  
5 }, 1000);  
6  
7 console.log('Câu lệnh này chạy thứ hai');
```

setTimeout v3

main.js

```
1 console.log('Câu lệnh này chạy đầu tiên');  
2  
3 setTimeout(() => {  
4     console.log('Câu lệnh này chạy cuối cùng, sau 1s')  
5 }, 1000);  
6  
7 console.log('Câu lệnh này chạy thứ hai');
```

setTimeout v4



main.js

```
1 console.log('Câu lệnh này chạy đầu tiên');  
2  
3 setTimeout(() => console.log('Câu lệnh này chạy cuối cùng, sau 1s')  
4           , 1000);  
5  
6 console.log('Câu lệnh này chạy thứ hai');
```

setInterval

setInterval

- The **setInterval()** method calls a function or evaluates an expression at specified intervals (in milliseconds).

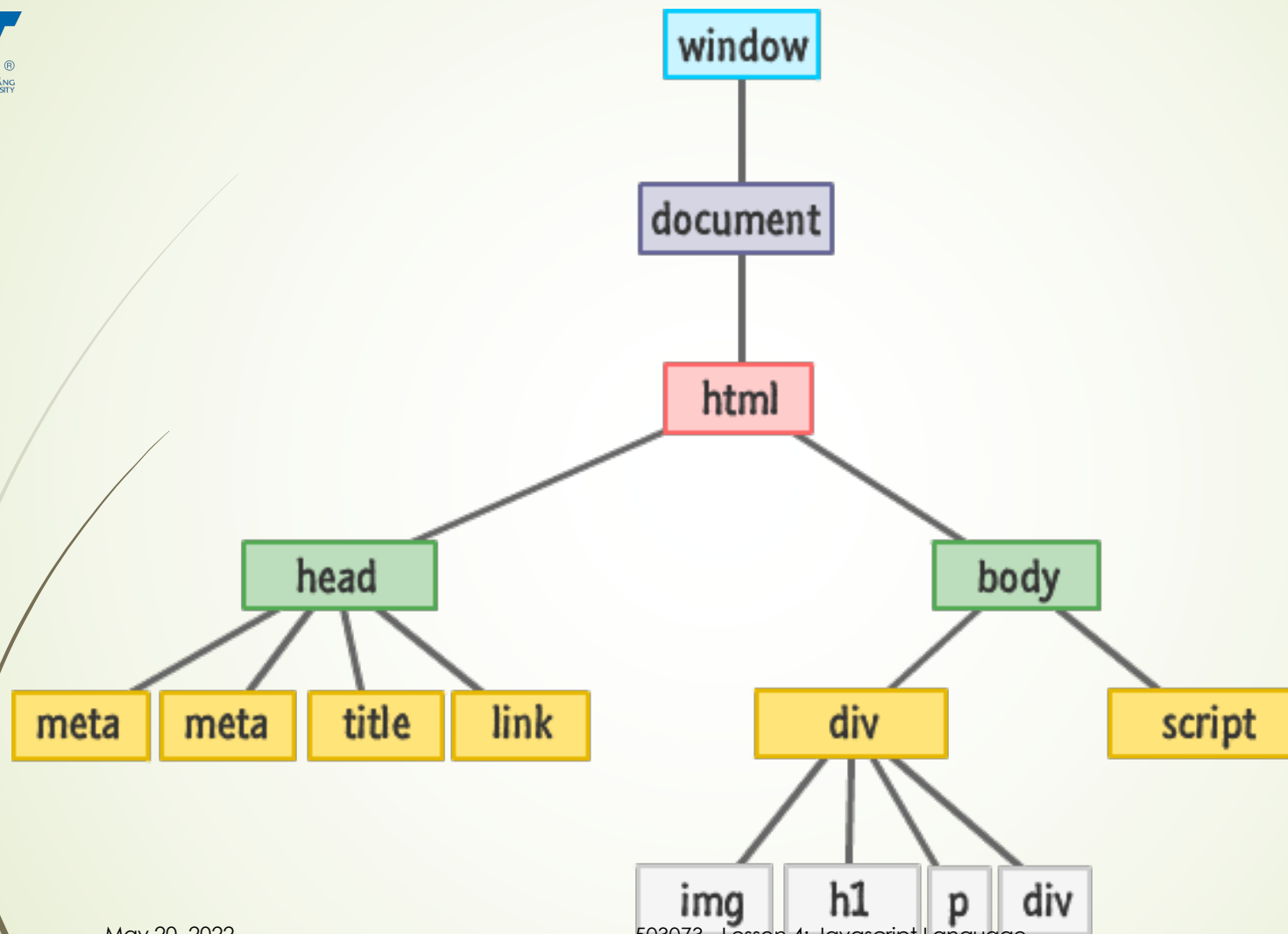
setInterval(*function*, *milliseconds*);

```
1 let wait = 10;
2 let id = setInterval(() => {
3     console.log(`Bạn phải chờ sau ${wait}s nữa`)
4     wait = wait - 1
5
6     if (wait === 0) {
7         clearInterval(id)
8     }
9 }, 1000);
```

Document Object Model

Document Object Model

- The HTML DOM is a standard object model and programming interface for HTML. It defines:
 1. The HTML elements as objects
 2. The properties of all HTML elements
 3. The methods to access all HTML elements
 4. The events for all HTML elements



The HTML **DOM** is a standard for
how to get, change, add, or
delete HTML elements.

► With the object model, JavaScript can:

1. change all the HTML elements
2. change all the HTML attributes
3. change all the CSS styles
4. remove HTML elements and attributes
5. add new HTML elements and attributes
6. react to all existing HTML events
7. create new HTML events

Finding HTML Elements

Finding HTML Elements

1. Finding HTML elements by **id**
2. Finding HTML elements by **tag** name
3. Finding HTML elements by **class** name
4. Finding HTML elements by **CSS selectors**
5. Finding HTML elements by HTML object collections

Find Element By Id

```
1 <body>
2   <p id="primary">Đây là tiêu đề chính</p>
3   <p id="secondary">Đây là tiêu đề phụ</p>
4   <script>
5       let p1 = document.getElementById('primary')
6       let p2 = document.getElementById('secondary')
7
8       p1.innerHTML = '<b>Tiêu đề 1</b>'
9       p2.innerHTML = '<b>Tiêu đề 1</b>'
10  </script>
11 </body>
```

Find Element By Class Name

```
1 <body>
2   <p class="title">Đây là tiêu đề chính</p>
3   <p class="title">Đây là tiêu đề phụ</p>
4   <script>
5       let p1 = document.getElementsByClassName('title')[0]
6       let p2 = document.getElementsByClassName('title')[1]
7
8       p1.innerHTML = '<b>Tiêu đề 1</b>'
9       p2.innerHTML = '<b>Tiêu đề 2</b>'
10   </script>
11 </body>
```

Find Element By Tag Name

```
1 <body>
2   <div>Đây là thẻ div</div>
3   <p>Đây là thẻ p</p>
4   <script>
5       let div = document.getElementsByTagName('div')[0]
6       let p = document.getElementsByTagName('p')[0]
7
8       div.innerHTML = '<b>Tiêu đề 1</b>'
9       p.innerHTML = '<b>Tiêu đề 2</b>'
10  </script>
11 </body>
```


Change Elements Attributes

Change Image Source



```
1 
2 
3
4 <script>
5     let img1 = document.
6         getElementsByTagName( 'img' )[0]
7
8     let img2 = document.
9         getElementsByTagName( 'img' )[1]
10
11     img1.src = 'image3.jpg'
12     img2.src = 'image4.jpg'
13 </script>
```

Change Link Source and Text

```
1 <body>
2     <a id="link"
3         href="https://tdtu.edu.vn">Web trường</a>
4     <script>
5         let link = document.getElementById('link')
6
7         link.innerHTML = 'Web Khoa'
8         link.href = 'https://it.tdtu.edu.vn'
9     </script>
10 </body>
```

Change Elements Style

Change Text Style



```
1 <p id="header">Xin chào</p>
2 <script>
3     let p = document.getElementById( 'header' )
4
5     p.style.fontWeight = 'bold'
6     p.style.fontSize = '150%'
7     p.style.padding = '10px'
8     p.style.textAlign = 'center'
9     p.style.textDecoration = 'underline'
10    p.style.backgroundColor = 'gold'
11 </script>
```

Change Text Style

Xin chào

Xin chào