

Módulo 1b - Javascript

T02 #TDW #MCTW

27/10/2021

Carlos Santos



HTML CSS and JavaScript for Beginners - A Web Design Course

<https://learning.oreilly.com/videos/html-css-and/9781838551278/>



Chapter 4 : Learn JavaScript Section





Welcome to JavaScript

- Browser
- DevTools
- Consola
- Editor / IDE

Write JavaScript

- <script>...</script>
 - + head
 - + body
- <button type="button" onclick="...">Click</button>
- O HTML já foi carregado antes de executar o JS?
- <script src="script.js"></script>
- “ e ”
 - + onclick="alert('Hi!')"



Update HTML element

- `console.log(...); console.dir(...);`
 - + Para desenvolvimento
- `document`
- `document.querySelector('h1').innerHTML = “...”;`



JavaScript Code (syntax)

- Comments
 - + //
 - + /*
 - */
- Espaços brancos no JS e no HTML
- Line breaks no JS -> OK
- Line breaks numa string em JS -> NOK
- Palavras reservadas do JS



JavaScript variables

- Números e Strings em JS
- var a = "hello";
- var b = 10;
- var c = b + ' ' + a;



Variables in action

- Tipos de dados são dinâmicos (podem mudar)
- Nomes das variáveis: letras, algarismos, _ e \$
 - + não pode começar com algarismo
 - + não pode usar palavras reservadas do JS
 - + nomes são case sensitive
- var para a declação de uma variável
- = é o operador de atribuição/assign
- \ sequência de escape
- typeof
- Tipo de dados booleano



Operators and Assignments

- Operadores: + - * /
- Incremento: a++ ++a
- Decremento: a-- --a
- % resto da divisão inteira (module)
- a = a + b;
- a += b;
- == Operador de comparação
- < >



JavaScript Functions

- Declaração
 - + `function message(){
}`
- Parâmetros/Argumentos
 - + `function message(a){
...
}
+ message('Olá');`
- Variáveis globais



Close look at functions

- Variáveis locais
 - + Local scope
- return



JavaScript Objects

- `var person = {};`
- `person.property = value;`
- variáveis com atribuição de uma função
- `this`
- `var person1 = {
 first:'Mike',
 last:'Svekis'
}`
- `person1.last`
- `person1["last"]`



JavaScript Arrays

- var myArray1 = [“Laurence”, “Svekis”, 30, false];
- Índice numérico (começa em 0)
 - + myArray1[1];
- myArray.length



Array Methods

- `myArray1.push(...); // adicionar ao final`
- `myArray1.pop(); // remover do final`
- `myArray1.unshift(...); // adicionar na frente`
- `myArray1.shift(); // remover da frente`
- `myArray1.indexOf('Svekis');`
- `myArray1.sort();`
- `myArray1.reverse();`
- `myArray1.splice(1,1); // retirar valores do array`
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array



Condition Statements JavaScript

- `if (condition) {
 statements;
}`
- `&& (AND)`
- `|| (OR)`
- `if (condition) {
 statements;
} else {
 statements;
}`
- `if (condition) {
 statements;
} else if (condition) {
 statements;
} else {
 statements;
}`



Combine with Functions

- Nada de novo 😊



JavaScript Switch

- `switch (variable) {
 case value1:
 statements;
 break;
 case value2:
 statements;
 break;
 default:
 statements;
}`



JavaScript Loops

- ```
for (var x = 0; x < 10; x++) {
 statements;
}
```
- ```
var myArray = [1, 3, 5, 7, 8];  
for (var x = 0; x < myArray.length; x++) {  
    console.log(myArray[x]);  
}
```
- ```
var myObj = {first:"Laurence", last:"Svekis"};
for (var x in myObj) {
 console.log(myObj[x] + ' ' + x);
}
```



# **JavaScript Loops**

- ```
var x = 0;
while (x < 10) {
    console.log(x);
    x++;
}
```
- ```
var i = 0;
do {
 console.log(i);
 i++;
}
while (i < 10);
```



# **JavaScript String Methods**

- `myStr1.length;`
- `myStr1.indexOf('...');`
- `myStr1.lastIndexOf('...');`
- `myStr1.indexOf('...', iniPosition);`
- `myStr1.search('...');`
- E muito mais...



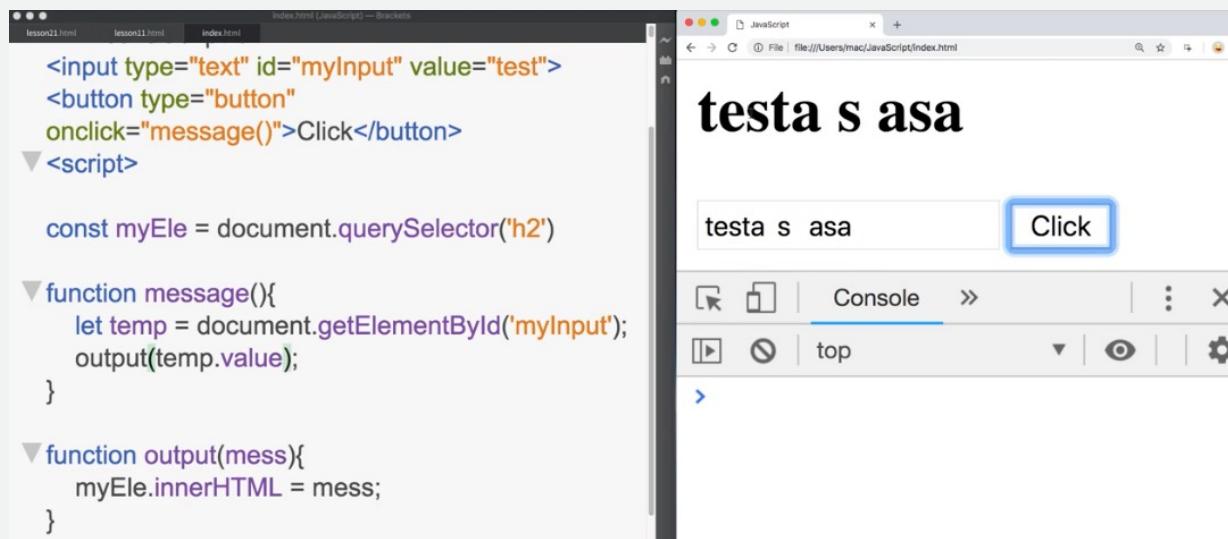
# **ES6 JavaScript**

- {  
    let a = ‘test’;  
    console.log(a);  
}
- for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
- {  
    const a = “test”;  
    a = “world”; // Error  
}



# **Document Object Model (DOM)**

- document
- `document.getElementById('myInput').value;`



The image shows a screenshot of the Brackets IDE interface. On the left, the code editor displays an HTML file with a text input field and a button. The button has an onclick event that calls a JavaScript function named 'message'. This function retrieves the value of the input field using document.getElementById and outputs it to the browser's console. On the right, a browser window shows the resulting page with the text 'testa s asa' and a blue-bordered 'Click' button.

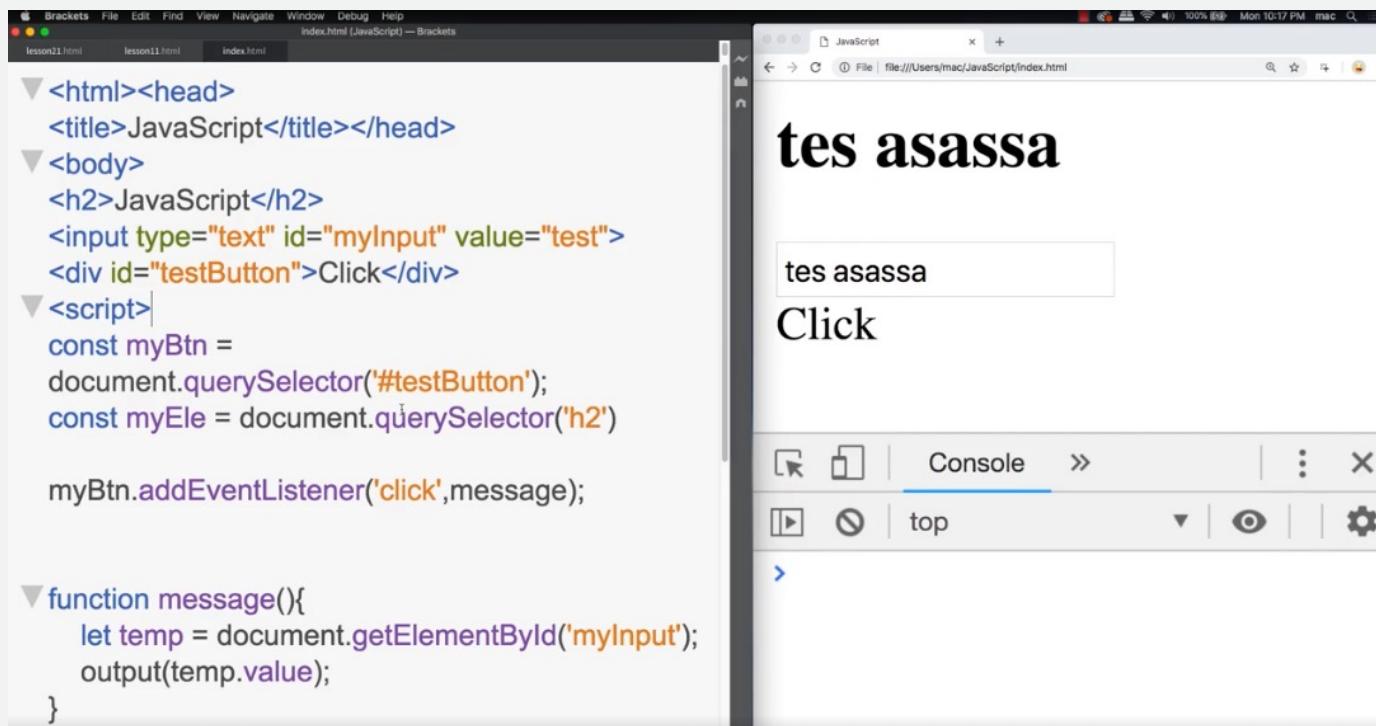
```
<input type="text" id="myInput" value="test">
<button type="button"
 onclick="message()">Click</button>
<script>

 const myEle = document.querySelector('h2')

 ▼function message(){
 let temp = document.getElementById('myInput');
 output(temp.value);
 }

 ▼function output(mess){
 myEle.innerHTML = mess;
 }

```



The screenshot shows the Brackets IDE interface. On the left, the code editor displays an HTML file named 'index.html' with the following content:

```
<html><head>
<title>JavaScript</title></head>
<body>
<h2>JavaScript</h2>
<input type="text" id="myInput" value="test">
<div id="testButton">Click</div>
<script>
const myBtn =
document.querySelector('#testButton');
const myEle = document.querySelector('h2')

myBtn.addEventListener('click',message);

function message(){
let temp = document.getElementById('myInput');
output(temp.value);
}


```

On the right, a browser window titled 'JavaScript' shows the rendered HTML. The page contains the text 'tes asassa' and a text input field with the value 'tes asassa'. Below the input is a button labeled 'Click'. The browser's developer tools are open, showing the 'Console' tab which is currently selected, and the 'top' frame.

# Eventlisteners JavaScript

