Women Techmakers

HTML, CSS & JS Workshop

WTM Hamburg

JS Documentation

During our Workshops we're going to develop parts of our WTM Hamburg Website together.

At this point, we'll continue with JS (JavaScript) for the site.

This document will give you a general definition of what JS is, and show you some of the rules that this language follows.

You can refer to this documentation at all Workshops, during the development of your tasks.

What is JS?

JS stands for JavaScript.

JavaScript is a cross-platform, object-oriented scripting language originally created by Netscape.

It is a small and lightweight language. Inside a web browser, JavaScript can be connected to the objects of its environment to provide programmatic control over them.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements.

Read further:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction#What_is_JavaScript

What is ECMAScript?

ECMAScript (ES) is a standard for a scripting language, and the JavaScript language is based on the ECMAScript standard.

JavaScript was originally created at Netscape, and they wanted to standardize the language. So, they submitted the language to the European Computer Manufacturer's Association (ECMA) for standardization. But, there were trademark issues with the name Javascript, and the standard became called ECMAScript, which is the name it holds today as well.

Read further:
http://www.programmerinterview.com/index.php/javascript/javascript-what-is-ecmascript/

JS Comments

JS Comments are pretty self-explanatory, they let you add notes and other hints that help you understand what's going on in your own code.

They also help other developers looking into your code. To have an easier understanding of why you wrote certain JS functions, variables, objects, etc.

You'll find in our scripts (*.js* files) many JS comments.

E.g.

```
JS

// I am a single JavaScript comment


/**
 * I'm also a JavaScript comment,
 * but a bit longer
 * than a single comment.
 * You can call me multi-line comment.
**/


/*
I'm also a JavaScript comment.
Another version of the multi-line comment.
*/
```

Read further:
http://www.w3schools.com/js/js_comments.asp

JS Variables

You use variables as symbolic names for values in your application. The names of variables, called identifiers, conform to certain rules.

A JavaScript identifier must start with a letter, underscore _, or dollar sign $. Subsequent characters can also be digits 0-9.

Because JavaScript is case sensitive, letters include the characters "A" through "Z" (uppercase) and the characters "a" through "z" (lowercase).

Declaring variables

You can declare a variable in three ways:

- With the keyword *var*.
  This syntax can be used to declare both local and global variables.

```
JS

var x = 42;

var y = ‘Moin, Welt!’
```

- By simply assigning it a *value*.
  This always declares a global variable. It generates a strict JavaScript warning. **You shouldn't use this variant**.

```
JS

x = 42;

y = ‘Moin, Welt!’
```

- With the keyword *let*.
  This syntax can be used to declare a block scope local variable.

  Variable scope in JS:
  https://msdn.microsoft.com/library/bzt2dkta(v=vs.94).aspx

```
JS

let x = 42;

let y = ‘Moin, Welt!’
```

- With the keyword *const*.
  This syntax can be used to declare a constant: a variable that can't be changed or overwritten.

```
JS

const x = 'foo';

const y = 'Moin, Welt!'
```

Read further:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Grammar_and_types#Basics

JS Functions

A function is a block of code that is defined to perform a certain task. The function needs to be defined and then called / executed to run the code inside it.

A function is defined the following way:

```
JS
 //1      2    3
function foo(arg1, arg2...){
  //code...
}
//1: function keyword
//2: function name
//3: function arguments = values passed to the
function that can be used inside the code
```

- Defining and calling functions example:

```
JS

//first function
function foo(){
  return 3;
}

//second function with argument
function bar(baz){
  let a = baz;
  return a + foo();
}

//call function bar with an argument
bar(4); // 7 is printed in the console
```

Manipulating the DOM with JavaScript

We can change the DOM with JavaScript after all our HTML and CSS is already loaded and put in place. DOM elements can be changed in style and HTML content (among other things).

```
JS

/**
 * Imagine we have a <div id="container"></div> in our HTML.
 * The variable container will contain a reference to that HTML element once the
 * following code is executed. The code calls the function "getElementById" on
 * the globally available variable "document" (which contains the whole HTML
 * document). We pass the ID of the div we want to select to the function as an
 * argument.
 **/
 var container = document.getElementById('container');

 // Now we will put some text content inside the container
 container.innerHTML = 'Moin, World!';

 // Change the background color of the container
 container.style.backgroundColor = 'blue';
```

Learn JavaScript

- https://www.javascript.com/
- https://developer.mozilla.org/en-US/docs/Web/JavaScript

Happy Coding!
Women Techmakers HH Team