WEEK 5: Javascript

```
Need for dynamics
```

HTML+CSS is just static, you can add some dynamic behavior to web pages with Javascript.

JavaScript is one of those few popular languages in the programming world. JavaScript has been ranked among the top 10 programming languages consistently for several years. Some of the most renowned tech companies are using JavaScript, like Facebook, Google, Microsoft.

What can Javascript do?

Javascript is a general purpose programming language, originally designed to run in the browser. JavaScript let web developers create web pages with client-side interactivity, or even to handle the HTML document as a dynamic object (containing all the information which is needed to render and describe functionality of its content)

Node: In 2009, a guy called Ryan Dahl took an open-source JavaScript engine (Chrome V8) and embedded it inside a standalone C++ program to make Javascript code run outside the browser. It is called Node.js, it is an open-source and cross-platform backend JavaScript runtime environment that lets developers use JavaScript to write command line tools and for server-side scripting. Running scripts server-side aims to produce dynamic web page content before the page is sent to the user's web browser.

ECMAScript: a specification of how Javascript should work. Javascript is a programming language that confirms this specification.

In the browser, it can:

- change HTML content
- HTML attribute values
    - Change, add, remove values
    - Show or hide elements
- change HTML style or CSS

For simple tasks, any text editor can be used.

For quick and better user experience, I suggest to use some online javascript sandbox, eg.

https://playcode.io/new/

If you want to install a code editor, I suggest Visual Studio Code with Live Server extension installed.

In medias res: Javascript Hello World

Changes the content of an HTML element:

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Hello World</h1>

<button type="button"
onclick="document.getElementById('demo').innerHTML = 'He'+'llo World' ">
Say hello</button>

<p id="demo">You could be mine</p>

</body>
</html>
```

Quoting: double or single, can embed easily into HTML quotes

Change properties or styles of HTML elements:

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Hello World</h1>

<button type="button"
onclick="document.getElementById('demo').innerHTML = 'Hell' + 'o World' ">
Say hello</button>

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

<!-- change style -->
<button type="button"
onclick="document.getElementById('demo').style.fontSize='30px'">Super Size
```

```
Me</button>

<!-- hide Hello -->
<button type="button" onclick="document.getElementById('demo').style.display
= 'none';">Hide Hello</button>

<!-- unhide Hello -->
<button type="button" onclick="document.getElementById('demo').style.display
= 'block';">Display Hello</button>

</body>
</html>
```

Default value of display property: ‘block’

Document Object Model - DOM

DOM treats HTML documents as a tree structure, where each element is an object representing a part of the document. Each element may contain other elements. Elements can have properties with their special values.

DOM has been standardized by the World Wide Web Consortium (W3C) from 1998 (as a result of browser wars). Modern browsers recognize and implement and render DOM.

The basic task of a browser is to display HTML. Most browsers use a DOM; they parse the HTML, create a DOM structure from it (which can also be used in JavaScript) and render the page based on that DOM.

- Document tree:
    - <html>
        - <head>
            - <title>
            -
        - <body>
            - <h1>
            -
                - innerHTML=
                - <i>
                -
                - <a>
            - …
            - <form>
                - <input>
                    - type=
                - <button>
                    - type=
                    - onclick=
                    - innerHTML=

Note: trees cannot have circles.

DOM elements can be accessed by

- name

```
age = document.getElementsByName("age")
```

- id

```
age = document.getElementById("age")
```

- class name

```
age = document.getElementsByClassName("age")
```

- tag name

```
age = document.getElementsByTagName("li")
```

- sequential index

```
age = document.getElementsByName("age")[0]
```

Where to place javascript code:

- HTML properties controlling events (onclick)
- HTML script tag
- external file/URL
- anywhere… (not a good practice…)

```
<!DOCTYPE html>
<html>
<head>
<script src='http://some.where/script.js'></script>
<script>
function first_func()
{
console.log("Find me");
}
</script>
</head>
<body>

<h1>JavaScript Hello World</h1>

<button type="button" onclick="alert('Hit an OK');">Push me</button>
<button type="button" onclick="first_func()">Push me again</button>
<button type="button" onclick="second_func()">Do not push</button>

</body>
</html>
```

Content of script.js:

```
<script>
function second_func()
{
console.log("You can see me on the console.");
}
</script>
```

There are no strict rules where to place javascript code.

Give me beer or not

```
<!DOCTYPE html>
<html>
<head>
<script>
function drink()
{
 result=document.getElementById("answer");
 age = document.getElementsByName("age")[0].value;

 if (age>=18)
    {
    answer = "Here you are your " + document.getElementById("drink").value;
    }
 else
    {
    answer = "You are only " + age + " thus here you are your milk";
    }

 result.innerHTML=answer;
}
</script>
</head>
<body>

<h2>Ask for more</h2>

What are you drinking?
<input type="text" id="drink">
<br>How old are you?
<input type="range" name="age" min="5" max="100"
onchange="document.getElementById('age_disp').innerHTML=document.getElementsB
yName('age')[0].value">
<span id="age_disp"></span>
<br><input type="button" value="Push me" onclick="drink()">
<div id="answer">Push the button to get answer</div>

</body>
</html>
```

Javascript conditional statements

if - else

```
<script>
if (condition1) {
  //  block of code to be executed if condition1 is true
} else if (condition2) {
  //  code to be executed if condition1 is false and condition2 is true
} else if (condition3) {
  //  code to be executed if condition1 and condition2 are false
  //  but condition3 is true
} else {
  //  code to be executed if all the three the conditions above are false
}
</script>
```

```
<!DOCTYPE html>
<html>
<body>

<button onclick="myFunction()">Get a time-based greeting</button>

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

<script>
function myFunction() {
  var greeting;
  var time = new Date().getHours();
  if (time < 10) {
    greeting = "Good morning";
  } else if (time < 20) {
    greeting = "Have a nice day";
  } else {
    greeting = "Good evening";
  }
  document.getElementById("demo").innerHTML = greeting;
}
</script>
```

```
</body>
</html>
```

Simple ?: operator also can be used

```
<script>
category = age>=18 ? “adult” : “child”;
</script>
```

switch

When using many conditions, switch statements can be easier to read (and write)

```
<!DOCTYPE html>
<html>
<head>
<script>
var day;
switch (new Date().getDay()) {
 case 0:
   day = "Sunday";
   break;
 case 1:
   day = "Monday";
   break;
 case 2:
   day = "Tuesday";
   break;
 case 3:
   day = "Wednesday";
   break;
 case 4:
   day = "Thursday";
   break;
 case 5:
   day = "Friday";
   break;
 case  6:
   day = "Saturday";
}
```

```
document.getElementById("result").innerHTML = "Today is " + day;
</script>
</head>
<body>
<p id="result">Today is ...</p>
</body>
</html>
```

If not using body tag, it will also work. Or not. Depending on the placement of the result paragraph. It is important to know DOM is created when you run code. (onload event, see next example)

You can do it simpler using arrays:

```
<!DOCTYPE html>
<html>
<head>
<script>
var dow =
["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"];
document.getElementById("result").innerHTML = "Today is " + dow[new
Date().getDay()];
</script>
</head>
<body>
<p id="result">Today is ...</p>

</body>
</html>
```

To be sure that your script starts when DOM is created, use body onload:

```
<!DOCTYPE html>
<html>
<head>
<script>
 function BigBang()
 {
var dow =
["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"];
document.getElementById("result").innerHTML = "Today is " + dow[new
```

```
Date().getDay()];
 }
</script>
</head>
<body onload="BigBang()">
<p id="result">Today is ...</p>

</body>
</html>
```

Javascript Arrays

Arrays store multiple values in a single variable. Items are numbered (indexed).
Indexes start from 0.

```
<script>
var dow =
["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"];
<!-- the same, OO-like -->
var dow = new Array
("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday");
best_day = dow[4];
</script>
```

Javascript variables and operators

```
<script>
var a=1;
var b=1;
var c= (a**2 + b**2)**0.5;
</script>
```

Arithmetic operators

+ - * / (four basic operations: addition, substraction, multiplication, division)
** (exponentiation)
% (modulus = division remainder)
++ -- (inplace increment-decrement)

Assignment operators also work: += -=, *= /= **= %=

```
<script>
var i=0;
i+=1; // same as i=i+1
</script>
```

Strings

- + and += operators can also be used on strings

```
<script>
var hi="High";
var five=5;
var concat=hi+five // mixing string with numbers results in string
</script>
```

Binary operators

>> shift right
& bitwise AND
| bitwise OR
^ bitwise XOR

Logical operators
&& logical AND
|| logical OR

Javascript programs tend to be more and more unreadable…
Many javascript expressions are used repeatedly in the same code

```
<script>
function id(id)
{
      return document.getElementById(id); // save 22 chars...
}


id('my_id').innerHTML = "Simple is beautiful"

<span id='my_id'></span>
```

```
</script>
```

Instead of id, JQuery started to use $ as a function name...

A complex example. Guess my number game

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Guess My number</title>
</head>

<body>
<h1>Guess My Number</h1>
<p>Try to find out my guessed number between 1 and 100.</p>

<input type="text" id="yournumber">
<input type="submit" id="submitguess" value="Check my guess">

<script type = "text/javascript">
   var mynumber = Math.floor(Math.random() * 100 + 1);

   var tipnum = 1;

   document.getElementById("submitguess").onclick = function()
   {
    var yournumber = document.getElementById("yournumber").value;

    if (yournumber > mynumber)
    {
       alert("Your guess #" + tipnum + " is greater than my secret number.");
       tipnum++;
    }
    else if (yournumber < mynumber)
    {
       alert("Your guess #" + tipnum + " is smaller than my secret number.");
       tipnum++;
    }
    else
    {
       alert("Congratulations, you guessed my number " + tipnum + " guess.");
```

```
    }
   }
</script>
</body>
</html>
```