JAVASCRIPT COURSE

PART FOUR - 30.10.2017.

ICT HUB

IN THIS CLASS

- HTML, CSS and JavaScript calculator
- JavaScript in the browser — Intro to DOM

BEFORE WE BEGIN

Presentations and homework:

https://github.com/JSBelgrade/course-2017

PREVIOUS HOMEWORK

Create a calculator UI using HTML and CSS.

Recommended design:
https://dribbble.com/shots/3344091-Daily-Ui-004-Calculator


4 + 8 + 15 + 16 + 23 + 42
= 108
AC
+/-
%
x
7
8
9
/
4
5
6
-
1
2
3
+
0
,
=

LET'S DO IT TOGETHER


4 + 8 + 15 + 16 + 23 + 42
= 108
AC
+/-
%
x
7
8
9
/
4
5
6
-
1
2
3
+
0
,
=

Live coding on jsbin.com.

Hint:
Use Flexboxes, see
http://bit.ly/flexbox-tricks

Result:

http://jsbin.com/vukuzex

Code is submitted in the Github repository.

QUICK REMINDER
FROM PREVIOUS CLASS

ARROW FUNCTIONS

Shorter syntax than a function expression and does not bind its own this.

```
(arg1, arg2) => expression;
```

```
argument => expression;
```

```
argument => {
   expression1;
   return expression2;
}
```

```
function multiplier(factor) {
  return number => number * factor;
}

const twice = multiplier(2);
console.log(twice(5)); // 10
```

CLASSES & PROTOTYPES

```
function Animal(type) {
  this.type = type;
}

Animal.prototype.getType =
  function() {
    return this.type;
  };

const cow = new Animal('cow');
cow.getType(); // cow
```

```
class Animal {
  constructor(type) {
    this.type = type;
  }

  getType() {
    return this.type;
  }
}

const cow = new Animal('cow');
```

RUN JAVASCRIPT IN THE BROWSER

```
<!doctype html>
<html>
   <head>
      <title>Hello</title>
   </head>
   <body>
   </body>
<html>
```

```
<!doctype html>
<html>
  <head>
    <title>Hello</title>
    <script src="file.js"></script>
  </head>
  <body>
  </body>
<html>
```

```
<!doctype html>
<html>
  <head>
    <title>Hello</title>
  </head>
  <body>
    <script src="file.js"></script>
  </body>
<html>
```

CONTINUE

EXERCISE 1

Define a calculator functionalities.

1. Which functionalities does it need to have?
2. How should they work?
3. Edge cases?

List of MVP functionalities:

1. Add numbers.
2. Add operators (+, –, * and /).
3. Calculate result.
4. Reset calculation.
5. Anything else?

EXERCISE 2

Build calculator class.

Live coding…

(Code is submitted in Github repository)

```
class Calculator {
  constructor() {} // Setup
calculator
  addNumber(n) {} // Add number
or digit
  addOperator(o) {} // Add
operator (+, -, * or /)
  equals() {} // Calculate
current total
  reset() {} // Reset all fields
}
```

NEXT STEP:

CONNECT CALCULATOR CLASS WITH UI (HTML AND CSS)

HTML AND JAVASCRIPT

<script> tag

Usage:

```
<!doctype html>
<html>
  <head>
    <title>Hello</title>
  </head>
  <body>
    <script>alert('Hello!');</script>
  </body>
<html>
```

```
<!doctype html>
<html>
  <head>
    <title>Hello</title>
  </head>
  <body>
    <script src="script.js"></script>
  </body>
<html>
```

```
<!doctype html>
<html>
  <head>
    <title>Hello</title>
  </head>
  <body>
    <button onclick="alert('Hello!')">
      Press Me
    </button>
  </body>
<html>
```

JAVASCRIPT CODE
AND
BROWSER SANDBOX

THE DOCUMENT OBJECT MODEL (DOM)

The Document Object Model (DOM) is a programming interface for HTML and XML documents.

It represents the page so that programs can change the document structure, style and content.

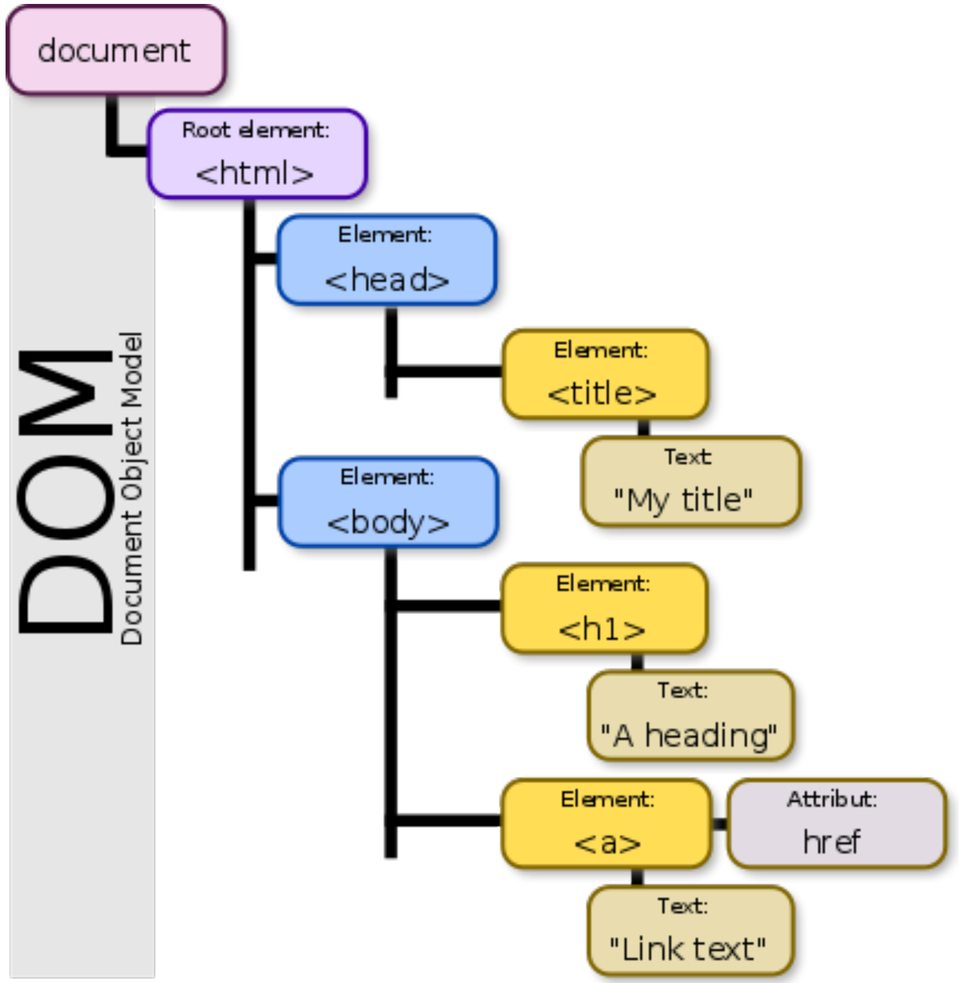
A Web page is a document. This document can be either displayed in the browser window, or as the HTML source. But it is the same document in both cases.

The DOM is an object-oriented representation of the web page, which can be modified with a scripting language such as JavaScript.

TREE

The DOM model represents a document with a logical tree. Each branch of the tree ends in a node, and each node contains objects.

```
<!doctype html>
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <h1>A heading</h1>
    <a href=example.com>Link text</a>
  </body>
<html>
```


DOM
Document Object Model
document
Root element:
<html>
Element:
<head>
Element:
<title>
Text
"My title"
Element:
<body>
Element:
<h1>
Text:
"A heading"
Element:
<a>
Attribut:
href
Text:
"Link text"

```
document.documentElement
```

document.getElementsByTagName

```
document.getElementsByTagName('h1')
```

DOM METHODS

DOM methods allow programmatic access to the tree; with them you can change the document's structure, style or content.

FINDING ELEMENTS

```
document.getElementById(id)

document.getElementsByClassName(class)

document.getElementsByName(name)

document.getElementsByTagName(name)
```

```
document.getElementById(id)

document.getElementsByClassName(class)

document.getElementsByName(name)

document.getElementsByTagName(name)
```

CHANGING THE DOCUMENT

```
element.removeChild

element.appendChild

element.insertBefore

element.replaceChild
```

EVENT LISTENERS

Nodes can have event handlers attached to them. Once an event is triggered, the event handlers get executed.

element.addEventListener

element.removeEventListener

```
el.addEventListener('click', e => {
  // Do something
  console.log(e.target)
})
```

LET'S CONNECT CALCULATOR

Live coding…

HOMEWORK

PART 1:

Add +/-, decimal and % functionalities to the calculator class.

PART 2:

Connect +/−, decimal and % functionalities to the calculator UI.

PART 3:

Find edge cases.

Bonus: Fix the most important ones.

READ (AND LEARN) MORE

Free Code Camp

Learn to code for free.

https://www.freecodecamp.org

Eloquent JavaScript

Marijn Haverbeke

https://eloquentjavascript.net

You Don’t know JavaScript

Kyle Simpson

https://github.com/getify/You-Dont-Know-JS

JavaScript: The Definitive Guide

David Flanagan

http://shop.oreilly.com/product/9780596805531.do

JavaScript: The Good Parts

Douglas Crockford

http://shop.oreilly.com/product/9780596517748.do

THE END

OF PART TWO