JavaScript For Beginners Part 1

http://www.cileria.com/jsp.pdf

jan.schulz@devugees.org

Agenda

- 1. Introduction to JavaScript
- 2. Variables & Data-Types
- 3. If / Else Statements
- 4. Boolean Logic
- 5. Switch Statement
- 6. First Coding Challenge
- 7. Functions
- 8. Arrays
- 9. Objects
- 10. Loops
- 11. Second Coding Challenge

1. Introduction to JavaScript

- JavaScript is a lightweight, cross-platform, object-oriented computer programming language
- JavaScript is one of the tree core technologies of web development
- JavaScript is most commonly used as a part of webpages
- Today, JavaScript can be used in different places:
 - Client-side: JavaScript was traditionally only used in the browser
 - Server-side: Thanks to node.js, we can use JavaScript on the server as well
- JavaScript is what made modern web-development possible:
 - Dynamic effects and interactivity
 - Modern web applications that we can interact with

1. Introduction to JavaScript







NOUNS

means "paragraph"

ADJECTIVES

p {color: red;}
means "the paragraph
 text is red"

VERBS

p.hide();
means "hide the
paragraph"

Some hints ...

- 1. Please follow the course and do not get distracted by Facebook etc.
- 2. Please make notes with your text editor
- 3. Source Code will be published after the lecture
- 4. Do not worry if this course first seems too "easy" for you. The difficulty will go up over time.

2. Variables & Data-Types

 Variable: A container in which we can store a value in order to use it later again in our code

[CODING]

- 1) Create a folder
- 2) Create a index.html
- 3) Create a main.js

2. Variables & Data-Types

- Number: Floating point numbers, for decimals and integers.
- String: Sequence of characters, used for text.
- Boolean: Logical data type that can only be true or false.
- Undefined: Data type of a variable which does not have a value yet.
- Null: Also means 'non-existent'.

2. Variables & Data-Types

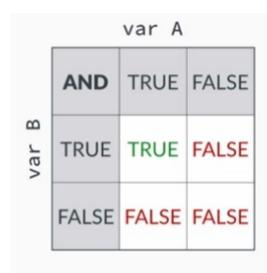
Operator Precedence

https://developer.mozilla.org/de/docs/Web/Java Script/Reference/Operators/Operator_Precede nce

3. If / Else Statements

```
if ( conditional statement ) {
    do something;
} else {
    do something else;
}
```

4. Boolean Logic



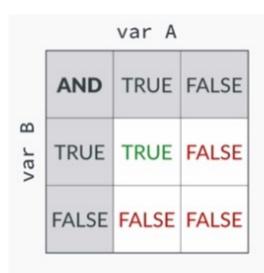
var A

OR	TRUE	FALSE
TRUE	TRUE	TRUE
FALSE	TRUE	FALSE

В

- AND (&&) => true if ALL are true
- OR(||) => true if ONE is true
- NOT(!) => inverts true/false value

4. Boolean Logic



var A

	OR	TRUE	FALSE
Var D	TRUE	TRUE	TRUE
	FALSE	TRUE	FALSE

```
    AND (&&) => true if ALL are true
    OR (||) => true if ONE is true
    NOT (!) => inverts true/false value
```

```
var age = 16;

age >= 20;    // => false
age < 30;    // => true
!(age < 30);    // => false

age >= 20 && age < 30;    // =>
age >= 20 || age < 30;    // =>
```

5. Switch Statements

```
switch( conditional statement ) {
  case expression:
     do something;
     break;
  case another expression:
     do something else;
     break;
  default:
     do something different else;
     break;
```

6. Coding Challenge

John and a friend invented a simple game where the player with the highest value of his height (in centimeters) plus five times his age wins (what a silly game :)

- 1. Create variables for the heights and ages of two friends and assign them some values
- 2. Calculate their scores
- 3. Decide who wins and print the winner to the console. Include the score in the string that you output to the console. Don't forget that there can be a draw (both players with the same score).
- 4. EXTRA: Add a third player and now decide who wins. Hint: you will need the && operator to take the decision. If you can't solve this one, just watch the solution, it's no problem:)

7. Functions

- Re-Usable piece of code
- Containers that hold lines of code
- Functions CAN return results
- DRY principle: Don't repeat yourself

```
function doubleMe( x ) {
    return 2 * x;
}
```

8. Arrays

A list of things, e.g. values
 var names = ['John', 'Jane', 'Mark'];

 https://www.w3schools.com/jsref/jsref_obj_arra y.asp

9. Objects

- A set of key-value pairs
- Do not have particular order like in Arrays

```
var obj = {
   name: 'John',
   lastName: 'Smith',
   age: 26
}
```

10. Loops

```
    Repeat tasks

• Using loops:
 while( statement ) {
    do something;
• Or
 for(expression; statement; expression) {
    do something;
```

11. Second Coding Challenge

- 1. Create an array with some years where persons were born
- 2. Create an empty array (just [])
- 3. Use a loop to fill the array with the ages of the persons
- 4. Use another loop to log into the console whether each person is of full age (18 or older), as well as their age
- 5. Finally, create a function called printFullAge which receives the array of years as an argument, executes the steps 2., 3. and 4. and returns an array of true/false boolean values: true if the person is of full age (>= 18 years) and false if not (< 18 years)
- 6. Call the function with two different arrays and store the results in two variables: full_1 and full_2

Example input: [1965, 2008, 1992]

Example output: [true, false, true]

Hint: you can use a loop not only to read from an array, like y[i], but also to set values in an array, like y[i] = ... You can also use the specific array methods.