JavaScript Basics

Agenda

JS Location

JS Input / Output

Data types

Conditions and flow control

Functions and events

Software development tools

JavaScript

Programming language

Object-oriented, imperative, functional programming

Untyped

Prototype-based (class-based vs. prototype-based)

Interpreted (just-in-time compilation)

WWW core technology

Tutorials: MDN w3schools

JS Location

Internal

HTML files (head / body)

```
<script> … JavaScript code … </script>
```

External

JavaScript files

```
<script src="/js/myScript.js"></script>
```

Input / Output

Dialog / popup windows: alert, confirm, prompt

Console: Chrome DevTools

HTML DOM

```
var x = document.getElementById("elementID").textContent;

document.getElementById("elementID").innerHTML = "text";
```

Data types and variables

Boolean

Null

Undefined

Number

String

Object

Dynamic types

Variable declaration

```
var abc = value;
```

Variable scope

global, local

[Grammar and types]

Strings and numbers

Strings

```
var name = "John May";
```

Numbers

```
var price = 120.5;
```

String functions JavaScript String Reference

Number functions JavaScript Number Reference

Numbers and dates

Arrays

Store multiple values in a single variable

Array declaration

```
var array-name = [item1, item2, ...];
```

Array methods to operate on array

Flow control

Block statement	`{ statements; }`
Conditional statements	`if…else, switch`
Loops and iteration	`for, do…while, while`

Functions

```
function name(parameter1, parameter2, …) {
    code to be executed
    return value;  // optional
}
```

Defining and calling functions

JavaScript Functions

Events

HTML code

```
<HTML-element some-event="some JavaScript">
```

JS code

```
element.addEventListener(event, function);
```

```
eg.  var el = document.getElementById("h1");
     el.addEventListener("click", myFunction);
```

Event reference

Software development tools

Static code analysis: **JSLint**

http://www.jslint.com/

Notepad++ plug-in

Authoring and debugging: **Chrome DevTools**

https://developers.google.com/web/tools/chrome-devtools/

To do

Display message

1. **The variables `name` and `surname` contain your personal data.**
2. **In the message.html , write a JavaScript program to display your data both in a popup window and in the console.**
3. **Try to display your name and surname in separate lines.**
4. **Check the results in the Chrome DevTools (ctrl+shift+j or F12).**

Enter data

1. **The variables** `name, surname` **and** `age` **contain your personal data.**
2. **Enter the data and save them in the variables.**
3. **In the enter.html document, write a JavaScript program to display your data both in a popup window and in the console.**
4. **Try to display your personal data in a single line.**
5. **Check the results in the Chrome DevTools.**

Execute a condition statement

1. **The variables** `name, surname` **and** `age` **contain your personal data.**
2. **Enter the data and save them in the variables.**
3. **In the condition.html document, write a JavaScript program to display your data in the console provided you pressed Ok button in the „confirm” popup window.**
4. **Check the results in the Chrome DevTools.**

Try to guess

1. **Write a JavaScript program where the program takes a random integer between 1 to 5, the user is then prompted to input a guess number. If the user input matches with guess number, the program will display a message "Good Work" otherwise display a message "Sorry".**
2. **Complete guessing.html with a JavaScript code.**
3. **To enter a user's number and to display a message, use popup windows.**

Fix that code

1. **Fix a JavaScript code in the code.html.**
2. **To improve the code quality, use JSLint (either notepad++ plug-in or www.jslint.com).**

Construct a pattern

1. **Write a JavaScript program to construct the following pattern:**

```
*
* *
* * *
* * * *
* * * * *
```

2. **Use a for loop and a repeat string method.**
3. **Complete pattern.html with a JavaScript code.**
4. **Display results in the console.**

Check an array

1. An array of 5 elements includes random integers in the range of 1 to 5. To create a random number, use Math.random() function.
2. Check whether the array is sorted, i.e. each element of the array is not smaller than its predecessor.
3. Complete array.html with a JavaScript code.
4. Display in the console the array elements and information whether the array is sorted.

Make a good speech

1. Write a JavaScript program to write/generate a speech.
2. Use speech.html with speech text excerpts.
3. To create a sentence, take any text from the first array, any text from the second one, any text from the third one, and finally, any text from the fourth array.
4. Complete the n() function to create a random number for selecting text from arrays. Use Math.random()
5. Display your speech, containing 5 sentences, in the console.

Create an external script

1. In the external.html, write a JavaScript program to display a current date in a popup window.
2. Place a JavaScript code in the external file date.js
3. In the current folder, create a folder ‘js’. Then put the script file in that folder.

Events

1. **In the events.html, write a JavaScript program to display a message in the console ‘You clicked the button X times’.**
2. **Put a code in an external script.**
3. **Create a function and a button event.**
4. **Display a message in the console each time the user clicks on the button.**
5. **Replace X with the number of clicks.**
6. **Check the results in the Chrome DevTools.**