JavaScript syntax Bootcamp

What will be discussed

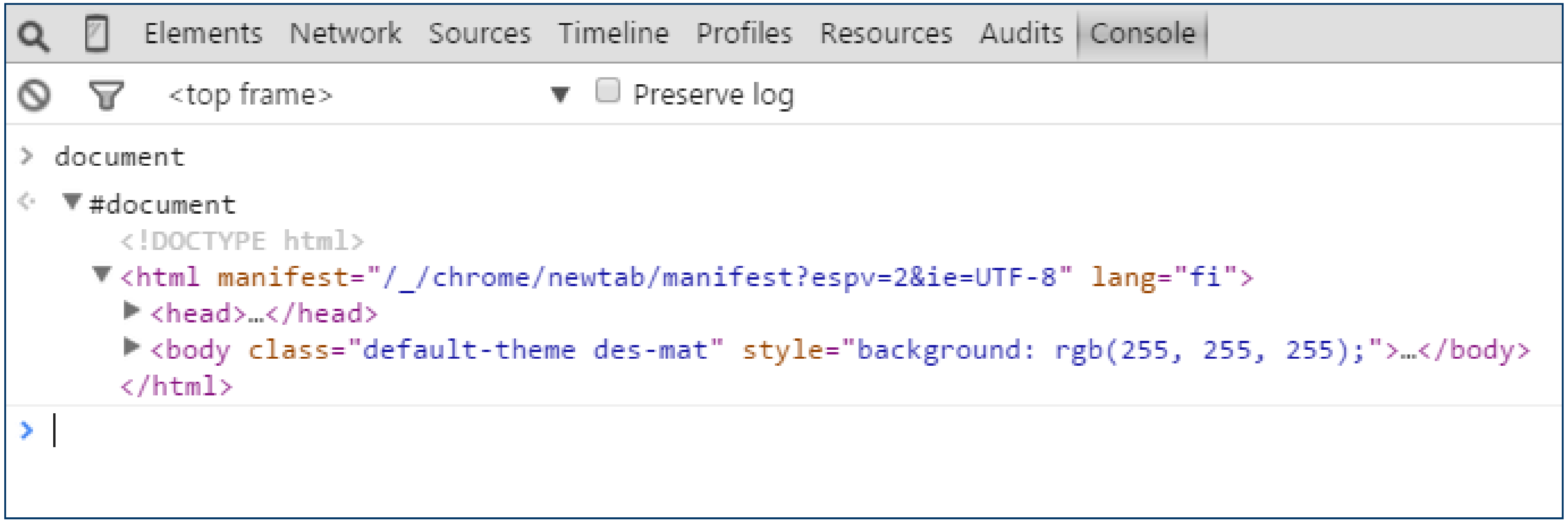
1. Using JavaScript console in the browser
2. Output
3. Variables and datatypes
4. Conditions
5. Loops
6. Functions and anonymous functions
7. Objects
8. Prototypes

How to try out JavaScript

- Playing with JavaScript console is the easiest way to start learning JavaScript. It is also used for testing and debugging.
- Each browser nowadays has one (press F12)
- In console, you can give JS commands and see the browser responding to them
- Google Chrome has great material if you want to read more and there's a handy cheat sheet availeble.
- After testing a piece of code in console, you need to be place the final code in an HTML file or external .js file

JavaScript console

- Typing ”document” in the console will return us a presentation of document and all its elements
- We can also browse the DOM tree with mouse
- Try also typing alert(”Hello world”);



```
> document
<· ▼#document
      <!DOCTYPE html>
    ▼<html manifest="/_/chrome/newtab/manifest?espv=2&ie=UTF-8" lang="fi">
      ▶<head>…</head>
      ▶<body class="default-theme des-mat" style="background: rgb(255, 255, 255);">…</body>
      </html>
> |
```

1. Output in JS

- JS does not have any built-in print functions
- One can output data in the following ways:

1. Writing into an alert box, using **alert()**.
2. Writing into the HTML output using **document.write()**.
3. Writing into an HTML element, using **innerHTML**.
4. Writing into the browser console, using **console.log()**.

1. Output in JS

- One can use quoted ” or single quotes ’ when printing

```
document.write(’Hello world’);
document.write(”Hello world”);
```

- Just dont mix the two when printing HTML

```
document.write(’<p class=”red”> This is HTML</p>’);
```

HINT. You can print multiple lines using ”backticks” (= ASCII 96, ALT+96):

```
var html = ` <h1> <b>Some HTML
here</b> </h1>
<p>More text here </p>
`;
```

2. Variables

- Variables are pretty much used like in any other language

```
var price1 = 5;
price2 = 6; // this is ok too
var total = price1 + price2;

var fname = “John”;
const lname = ‘Doe’;
var fullname = fname +” “+lname;
let result = "Sum of "+a+" and "+b+ " is "+ (a+b);
```

In ES6:

```
let a = 10;
let b = 20;
let result = `Sum of ${a} and ${b} is ${a+b}.`;
```

2. Data Types

```
var length = 16;  // Number
var lastName = "Johnson"; // String
var cars = ["Saab", "Volvo", "BMW"]; //
Array
var x = {firstName:"John", lastName:"Doe"};
// Object
```

3. Conditions

- Familiar from other programming languages

```
var time = 9;
if (time < 10) {
    greeting = "Good morning";
}else {
    greeting = "Good evening";
}
console.log(greeting)
```

4. Loops

- JS has regular for and while loops

```
 for (var i=0; i < 10; i++) {
   console.log(i); // logs ”0, 1,2,3,4,5,6,7,8,9"
}
```

- In addition, for..of and for..in
  - *for...in iterates over property names*
  - *for...of iterates over property values*

```
 var arr = [3, 5, 7];
 for (var i in arr) {
   console.log(i); // logs "0", "1", "2"}

 for (var i of arr) {
   console.log(i); // logs "3", "5", "7"
}
```

5. Functions

- A function is a block of code designed to perform a particular task.

```
function doSomething() {
    document.write(“Time to code!”);
}
// The function returns the product of p1 and p2
      function myFunction(p1, p2) {
          return p1 * p2;
      }
```

- A JavaScript function is executed when "something" invokes it (calls it).

```
      doSomething();
      var x = myFunction(4, 3);
```

5. Anonymous functions

- JS allows us to use self executing functions without names, see the following to get an idea

```
var helloWold = function() {
  alert('Hello World');
}
```

- Or we could even write:

```
(function() {
  alert('Hello World');
})();
```

6. Objects

- Object is a single variable with multiple properties and actions (functions)

```
var person = {
    firstName:"John",
    lastName:"Doe",
    age:50,
    eyeColor:"blue"
};
console.log(person.age);
```

6. Objects and methods

- Object is a single variable with multiple properties and actions (functions)

```
var person = {
    firstName:"John",
    lastName:"Doe",
    age:50,
    eyeColor:"blue“
    getFullName: function() {
   return this.firstName + " " + this.lastName;
       }
};
console.log(person.getFullName() );
```

7. Prototypes

- Java-like objects can be created using prototypes:

```
function person(first, last, age, eyecolor) {
    this.firstName = first;
    this.lastName = last;
    this.age = age;
    this.eyeColor = eyecolor;
}

var myFather = new person("John", "Doe", 50, "blue");
var myMother = new person("Sally", "Rally", 48, "green");
```

Questions or comments?