What is JavaScript?

JavaScript is

- A scripting language added to make webpages more interactive
  - http://www.histography.io/
What is JavaScript?

JavaScript is

- A scripting language added to make webpages more **interactive**
  - http://www.histography.io/
- An interpreted language
What is JavaScript?

JavaScript is

- A scripting language added to make webpages more **interactive**
  - [http://www.histography.io/](http://www.histography.io/)
- An interpreted language
- Object-based
- Case sensitive
What is JavaScript?

JavaScript is

- A scripting language added to make webpages more **interactive**
  - [http://www.histography.io/](http://www.histography.io/)
- An interpreted language
- Object-based
- Case sensitive
- Widely used and supported
- Accessible to the beginner
Comments

Single line comment:

    // my comment goes here

Multi-line comment:

    /* my comment goes here
       my comment goes here
       my comment goes here
    */
Variables

```javascript
var my_variable6 = -14.5;
console.log(typeof my_variable6);    // → number
```
Variables

```javascript
var my_variable6 = -14.5;
console.log(typeof my_variable6);     // → number

var a = "Hello world!";
console.log(typeof a);                // → string

var a = true;
console.log(typeof a);                // → boolean
```
Variables

```javascript
var my_variable6 = -14.5;
console.log(typeof my_variable6);  // → number

var a = "Hello world!";
console.log(typeof a);  // → string

var a = true;
console.log(typeof a);  // → boolean

var a = ["Hello", "COS", 426];
console.log(typeof a);  // → object
// can also be null or undefined
```
Objects
Objects

function professor(name, class) {
    this.name = name;
    this.class = class;
}

function professor(name, class_num) {
    this.name = name;
    this.class_num = class_num;
}

var prof1 = new professor("Szymon", 426);
var prof2 = new professor("Tom", 526);
function professor(name, class_num) {
    this.name = name;
    this.class_num = class_num;
}

var prof1 = new professor("Szymon", 426);
var prof2 = new professor("Tom", 526);

console.log(prof1.name);         // → Szymon
console.log(prof2.class_num);    // → 526
function professor(name, class_num) {
    this.name = name;
    this.class_num = class_num;
}

var prof1 = new professor("Szymon", 426);
var prof2 = new professor("Tom", 526);

console.log(prof1.name); // → Szymon
console.log(prof2.class_num); // → 526

prof1 = {name:"Szymon", class_num:426};
Objects

```javascript
prof1 = {name:“Szymon”, class_num:426};

prof1.num_students = 71;

console.log(prof1.num_students);        // → 71

prof1.status = function () {
    return this.name + “ is awesome”;
};

console.log(prof1.status());        // → Szymon is awesome
```
Objects

Can be an array of object:

```javascript
var journal = [
    {events: ["work", "ice cream", "pizza", "running", "television"],
     squirrel: false},
    {events: ["weekend", "cycling", "break", "peanuts", "beer"],
     squirrel: true}
];
```
Objects

Can be an array of object:

```javascript
var journal = [
    {events: ["work", "ice cream", "pizza", "running", "television"],
        squirrel: false},
    {events: ["weekend", "cycling", "break", "peanuts", "beer"],
        squirrel: true}
];
console.log(journal[0].events[1]); // → ice cream
```
Objects

Can be an array of object:

```javascript
var journal = [
    {events: ["work", "ice cream", "pizza", "running", "television"],
     squirrel: false},
    {events: ["weekend", "cycling", "break", "peanuts", "beer"],
     squirrel: true}
];

console.log(journal[0].events[1]); // → ice cream
for ( var prop in journal[0] ) {
    console.log(prop);
    console.log(journal[0][prop])
}
// → events
// → ["work", "ice cream", "pizza", "running", "television"]
// → squirrel
// → false
Variable Scope

In JavaScript, instead of braces, functions are the only things that create a new scope.

```javascript
var a = 1;
{
    var a = 2;
}
console.log(a); // → 2

-----------------------------

var a = "outside";
var f = function() {
    var a = "inside f";
};
f();
console.log(a); // → outside
```
Function expression

**Function declaration:**

```javascript
function sqr( x ) {return x * x; }
```
- not part of regular top-to-bottom flow of control
- can be used by all the code

**Function expression** act as names for a specific piece of the program:

```javascript
var Sqr = function( x ) { return x * x; };
```
Special functions

- `alert()` to display a message box
- `confirm()` to display a confirmation box
- `prompt()` to display a prompt box
- `open()` to open a new window
- `close()` to close a window
- `write()` write a string to the Web page
- `console.log()` outputs a message to the Web Console
Debugging

- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
More JavaScript Help

http://www.w3schools.com/js/
Simple HTTP server

Open up a terminal and type:

- $ cd /home/yourdir
- $ python -m SimpleHTTPServer

That's it! Now your http server will start in port 8000. You will get the message:

Serving HTTP on 0.0.0.0 port 8000

You can access it via

http://127.0.0.1:8000/yourhtml.html
Dat.Gui

A lightweight graphic user interface for changing variables in JavaScript

Link for tutorial (no need to learn how to use it)

https://workshop.chromeexperiments.com/examples/gui/
Questions?