

# Learning Javascript

## Ressources

W3Schools tutorials

<https://www.w3schools.com/js/default.asp>

Javascript (ES6) documentations

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<http://es6-features.org>

## Basics for this exercises

**Opening the console in your chrome browser:**

OS X: ALT + CMD + J

Windows: CTRL + Shift + J

**Printing on the console:**

```
console.log("*");
```

will write a \* character

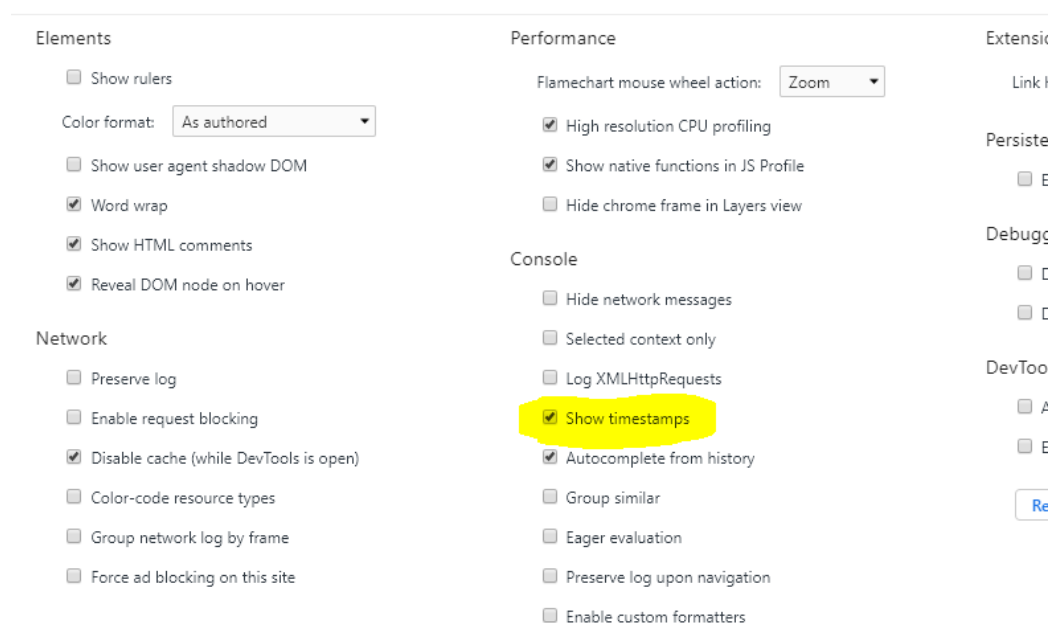
```
let text = "print me!";
```

```
console.log(text);
```

will write contents of variable text

**Showing timestamps of console logs, to avoid collapsing of similar consecutive messages (useful for these assignments)**

Go to console settings in chrome  and enable show timestamps



# Handin:

Upload results from exercise 1,2 and 3 to Canvas:

1. lab02-exercise01-triangles.html
2. lab02-exercise02-rectangles.html
3. lab02-exercise03-simpleRPG.html

## Guidelines

A ZIP file containing the following folders and files, **exactly structured like this:**

- lab02-groupXX.zip (Where **XX** is the group number on Canvas)
  - lab02-groupXX
    - lab02-exercise01-triangles.html
    - lab02-exercise02-rectangles.html
    - lab02-exercise03-simpleRPG.html

**MAC:** To compress to Zip, select a folder in the Finder, right-click and choose “compress”. You have to **Zip a folder containing the folder labXX-groupXX** for this to be structured properly:

- [AnyFolder] ←- CHOOSE COMPRESS HERE, then name .zip appropriately
  - labXX-groupXX
    - lab folders and files

**Windows:** Select the respective folder labXX-groupXX in the Explorer, right-click and choose Send To -> Compressed(zipped) Folder

**Any submission not following these guidelines automatically receives 0 points and has to be resubmitted properly structured.**

If submissions are **late by more than a week** after the assignment deadline, a **grade reduction of 10% per week** late will be introduced. **No Bonus Points** can be earned on late assignments.

Days submitted late	Grade reduction
0 - 7 days	0%
8 -14 days	10%, no bonus points
14 -21 days	20%, no bonus points
etc..	30%+ , no bonus points

*Note from teacher:*

*I am sorry that I have to be so uptight about this, but it is a lot of work to check and correct so many exercises every week and I can save a lot of time when I can automate the fetching and extraction of the files from Canvas. When everything is properly set up and structured it is done in a few clicks instead of a huge hassle to manually sort and name everything myself. Since this is the first exercise, please ask if you are in doubt about how to submit. All future exercises will follow these exact same guidelines.*

# Setup

Create a new folder for the lab and create a copy of your template html (or index.html) file from the class inside. Name it: "lab02-exercise01-triangles.html". For now write your Javascript inside a `<script type="text/javascript" ></script>` html element at the end of the `<body></body>`.

**Make sure to start every script and Javascript file with the line:**

```
"use strict";
```

to use Javascript strict mode for better errors!

## 1: Simple Programming: Triangles (30%)

Create one or multiple function(s) to "draw" different sizes of Triangles out of \* characters on the console. The size determines the length of the longest line of \*. You can use for example one or multiple nested `for` loop(s). Text can be written on the console as shown on page 1.

**Tipp:** `console.log(message)` can only print the `message` as one line. You might want to assemble the `message` string first, before printing it out.

Here are examples of the output for triangles of size 2 and 5:

```
*
* *
*

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

**Make sure to start every script and Javascript file with the line:**

```
"use strict";
```

to use Javascript strict mode for better errors!

## 2: Object Oriented Programming: Rectangles (40%)

Create a new copy of the base html file. Name it: "lab02-exercise02-rectangles.html". Create a class that defines a rectangle and has a method to draw it on the console using + and - characters. Then create multiple rectangle objects and draw them.

Examples shown are 2x2, 5x5 and 10x4 rectangles.

```
++  
++
```

```
+++++  
+---+  
+---+  
+---+  
+++++
```

```
++++++  
+-----+  
+-----+  
++++++
```

**For extra practice using classes and object oriented programming:**

Extend the logic to use any given characters for the Rectangle. You can tell your Rectangles how they should look like in the constructor using parameters. When you are in doubt on how to use constructors in Javascript, check the lecture slides or

<https://www.w3schools.com/js/default.asp>

### 3: Event based Programming: A small RPG (20%)

Create a new copy of the base html file. Name it: "lab02-exercise03-simpleRPG.html".

Create a small RPG where the player can move around a 2D world made out of letters on the console. Maybe use the " " character as background and the "8" character as the player.

**Make sure to start every script and Javascript file with the line:**

```
"use strict";
```

to use Javascript strict mode for better errors!

A 10x10 world with the player at location (6,9):

```
//////////
//////////
//////////
//////////
//////////
//////////
//////////
//////////
//////////
//////////
      8
//////////
```

To move the player around, create an event listener for the "keydown" event like this:

```
function keyDownFunction(event){
    if(event.key === "ArrowDown"){ // arrow down(↓) key is
pressed

    }
}
// add an event listener that calls keyDownFunction
// whenever any key is pressed.
window.addEventListener("keydown", keyDownFunction);
```

The different keys to listen to, to move the player are "ArrowDown"(↓), "ArrowUp"(↑), "ArrowLeft"(←) and "ArrowRight"(→).

You only need one event listener, since it will listen to any key press. Then inside `keyDownFunction` you can determine with `event.key` what key was pressed and execute the respective logic.

See: [https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key/Key\\_Values](https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key/Key_Values)

For more values of `event.key` for other keyboard keys.

When drawing the world, you will first need to clear the console, this can be done by:

```
console.clear();
```

## 4: Extend the RPG (10%)

- Extend the RPG to include a goal position, when reaching it the player wins.
- Implement traps, when touched the player loses.
- Add obstacles that can not be traversed.
- .... Whatever else you can think of.

Further bonus for additional and outstanding work, up to 110%