LAB 12 (WEEK 45)

JavaScript

Table of Contents

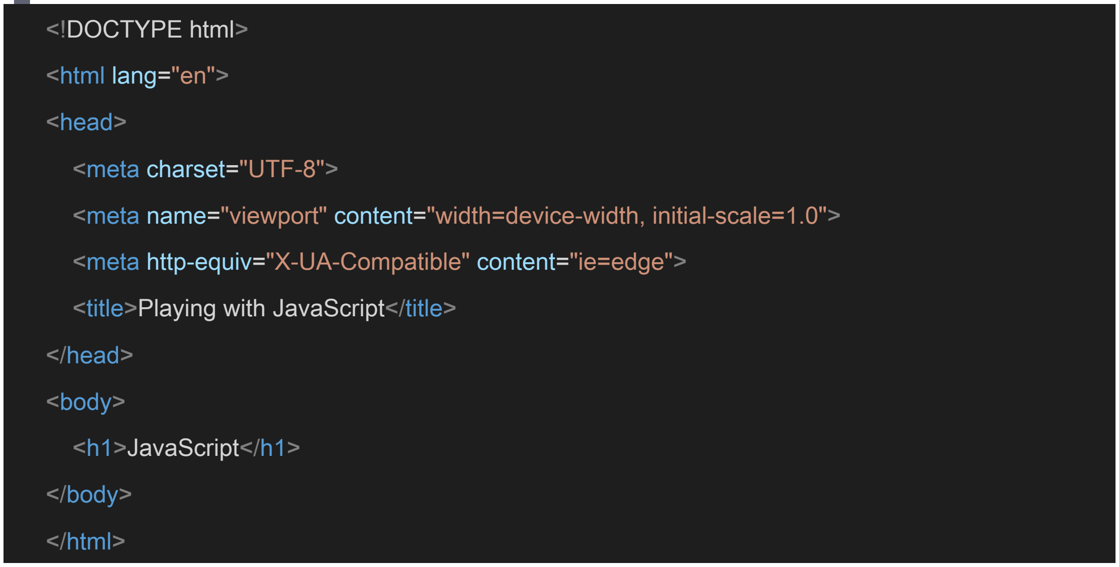
Introduction

In this lab, we are going to work with JavaScript and display the output using the console. Remember that JavaScript is a programming language[1] used to handle the dynamic nature of contents as well as the interaction between the browser and the user.

Generally speaking, each programming language has a specific syntax and some unique particularities, but most of programming languages have some structures in common: variables, program structures that include expressions and statements such as conditional statements or loops, functions or procedures, data structures, arrays, etc.

First of all, we are going to learn how to store JavaScript files and execute them in our browsers (i.e. inside our html files).

> ***Create an HTML5 file (index.html).***



```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title>Playing with JavaScript</title>
</head>
<body>
   <h1>JavaScript</h1>
</body>
</html>
```

Two ways to include JavaScript in our html files:

- JavaScript code can be inserted between <script> and </script> html tags.

[1] A programming language is a formal language that specifies a set of instructions that can be used to produce various kinds of output. Programming languages generally consist of instructions for a computer. Programming languages can be used to create programs that implement specific algorithms.

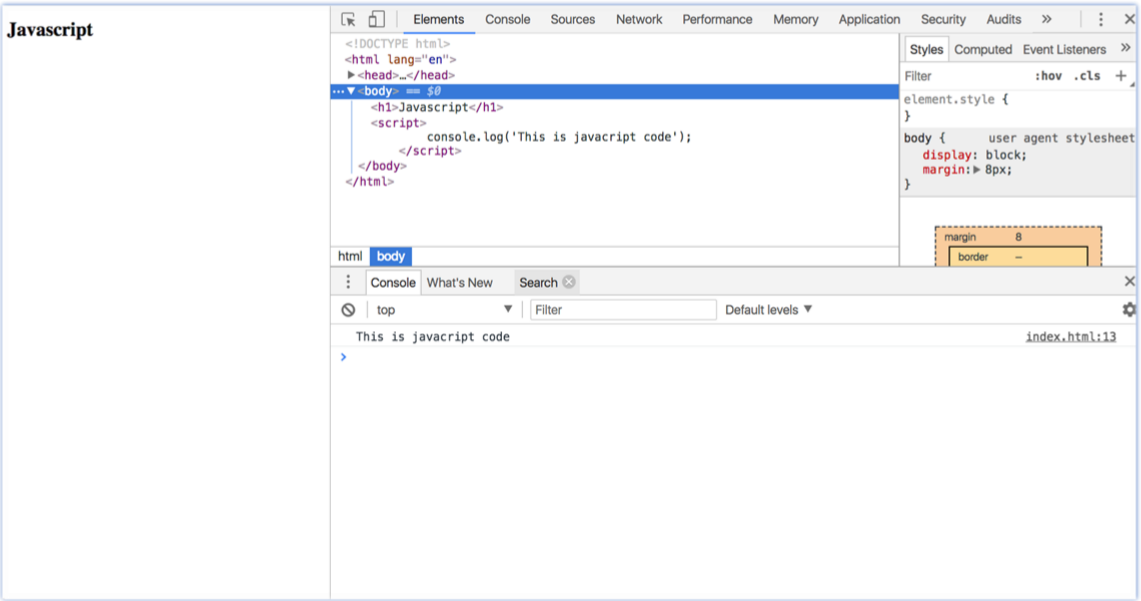
- **Scripts can also be placed in external files.**

Let’s add some JavaScript code in our html file. We will use *console.log* function to print text/output in our console.

> *Add a <script> tag at the end of your body and print the following text in the console “This is JavaScript code”.*

```
<body>
  <h1>JavaScript</h1>


  <script>
    console.log('This is Javacript code');
  </script>
</body>
```

> *Open the index.html file in your browser and display the console. (Chrome: Mac -> “Command+Option+J” Windows-> “Control+Shift+J”.*



Javascript

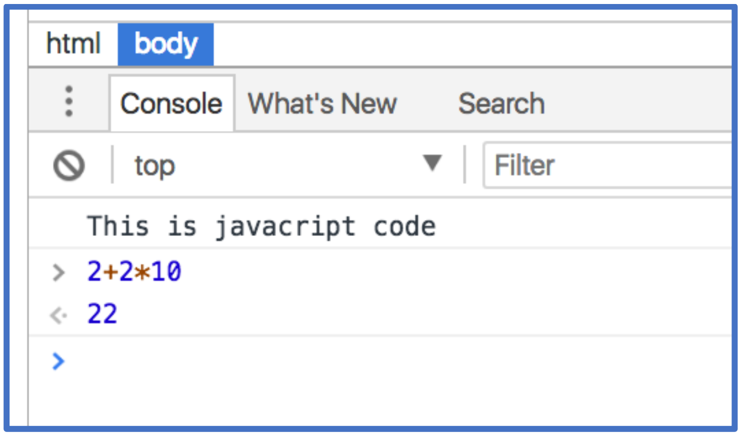
```
<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
…▼<body> == $0
    <h1>Javascript</h1>
    <script>
            console.log('This is javacript code');
        </script>
  </body>
</html>
```

```
element.style {
}
body {          user agent stylesheet
   display: block;
   margin:▶ 8px;
}
```

```
This is javacript code                                   index.html:13
>
```

Notice that the left-hand side of the image above is the web page and the right-hand side the chrome JavaScript console. In the console, after the > symbol, you can write JavaScript code that will be executed by the browser. The result is preceded by <. symbol.

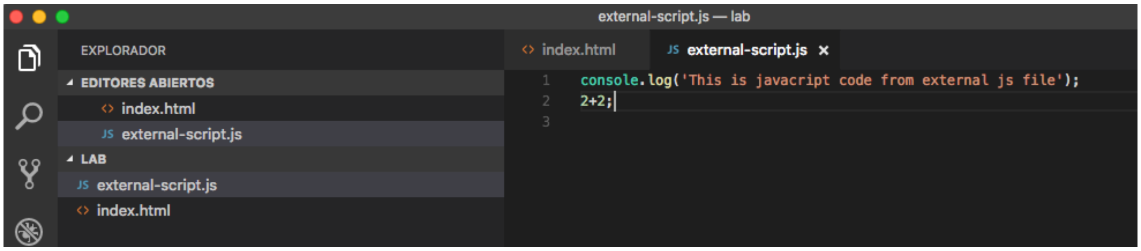
> *Try to add an arithmetic expression to the console (2+2*10)*



```
  This is javacript code
> 2+2*10
<· 22
>
```

Let's see how we can use external JavaScript files to do the same. JavaScript files use the ".js" extension.

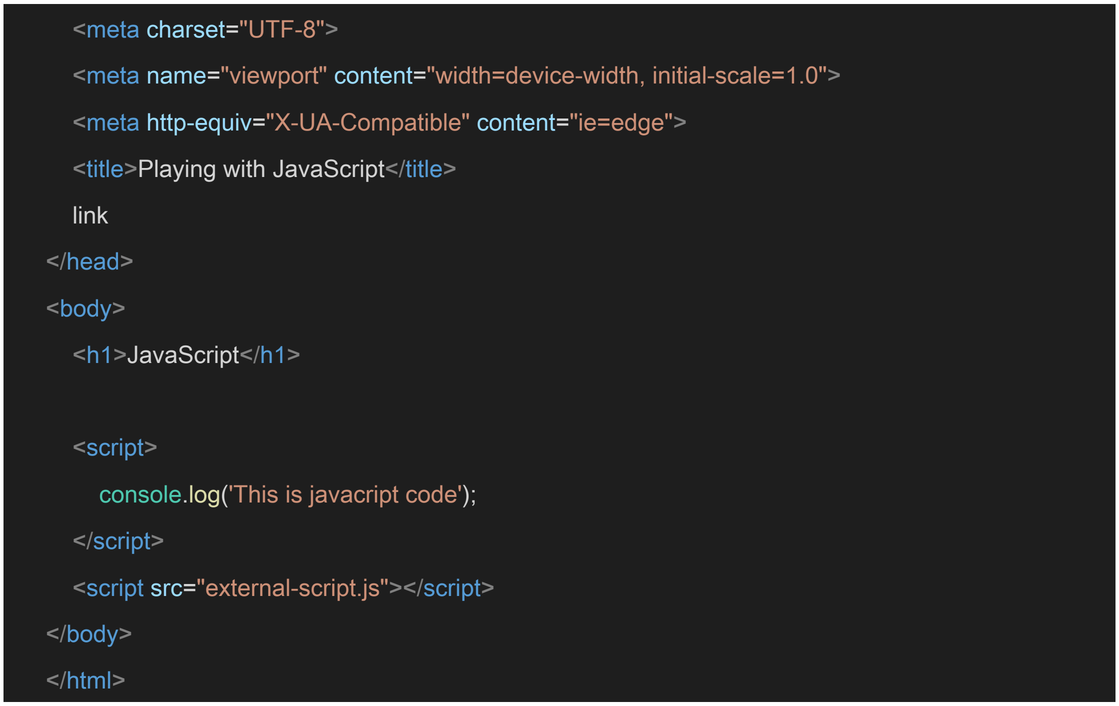
> *Create a JavaScript file in the same folder of your index.html. Name the file "external-script.js" and include the JavaScript code from the previous example.*



```
console.log('This is javacript code from external js file');
2+2;
```

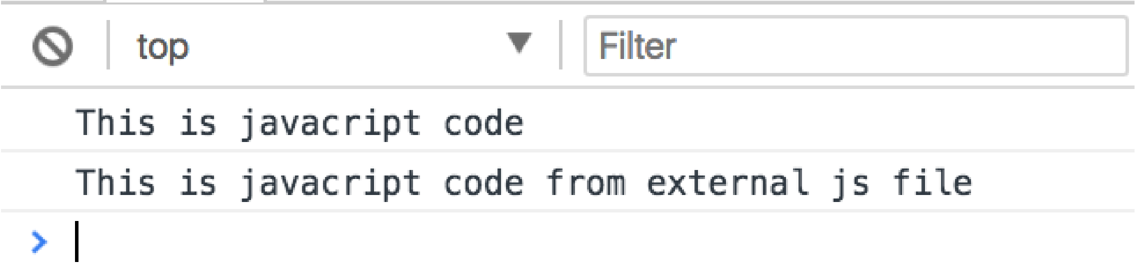
> *Load the script file in your index.html. Use the <script src=""> syntax.*

```
<!DOCTYPE html>
<html lang="en">
<head>
```



```
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
        <title>Playing with JavaScript</title>
        link
    </head>
    <body>
        <h1>JavaScript</h1>

        <script>
            console.log('This is javacript code');
        </script>
        <script src="external-script.js"></script>
    </body>
    </html>
```

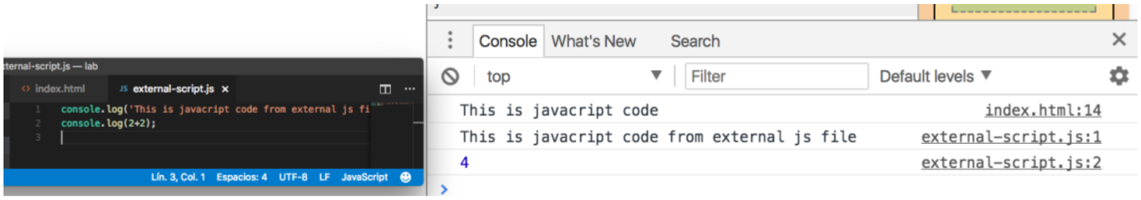
> *Refresh the page and see the result*



```
This is javacript code
This is javacript code from external js file
```

Notice that the result of the expression 2+2 is missing. If we want to output the result of our code, we should use the console.log function.

> *Modify the code so that the result of the arithmetic function is displayed in the console.*



```
console.log('This is javacript code from external js fi
console.log(2+2);
```

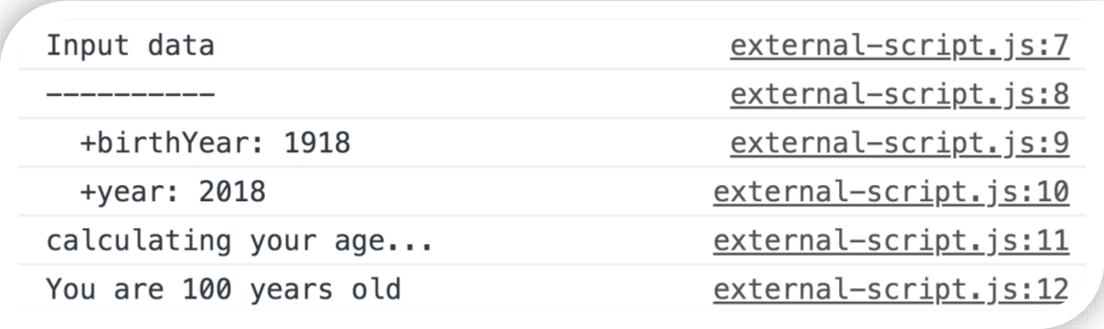
```
This is javacript code                              index.html:14
This is javacript code from external js file   external-script.js:1
4                                              external-script.js:2
```

Now it’s your turn. Do the following exercises.

Exercise 1

Exercise 1.1

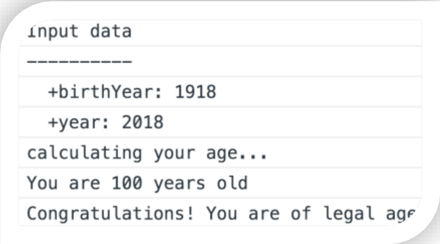
Create a html file and name it ex1.html. Create a JS file and name it ex1.js. Create two variables (birthYear and year). Initialise the birthYear using the year you were born. The year variable must be set with the 2018 value. Implement a small Javascript program to calculate your age. The expected result is the following.



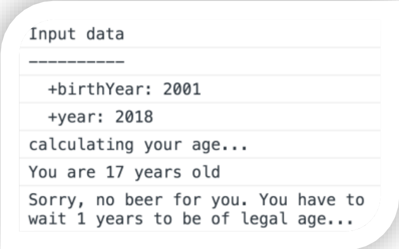
```
Input data                                external-script.js:7
----------                                external-script.js:8
  +birthYear: 1918                        external-script.js:9
  +year: 2018                            external-script.js:10
calculating your age...                  external-script.js:11
You are 100 years old                    external-script.js:12
```

Exercise 1.2

Use a condition statement using the previous code to calculate if you are of legal age. If legal age, show the message "Congratulations! You are of legal age". Otherwise, show the message "Sorry, no beer for you. You have to wait XXX years to be of legal age…". See the expected result below (test different input values).



```
Input data
----------
  +birthYear: 1918
  +year: 2018
calculating your age...
You are 100 years old
Congratulations! You are of legal age
```

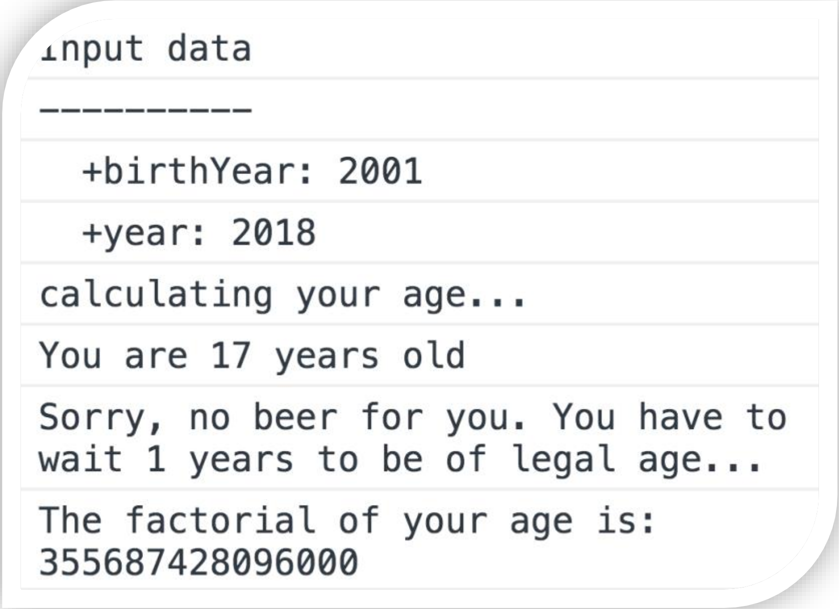


```
Input data
----------
  +birthYear: 2001
  +year: 2018
calculating your age...
You are 17 years old
Sorry, no beer for you. You have to
wait 1 years to be of legal age...
```

Exercise 1.3

Calculate the factorial of your age. Use a loop structure to get the number.

Here you can see how to calculate the factorial of a non-negative number: https://en.wikipedia.org/wiki/Factorial.



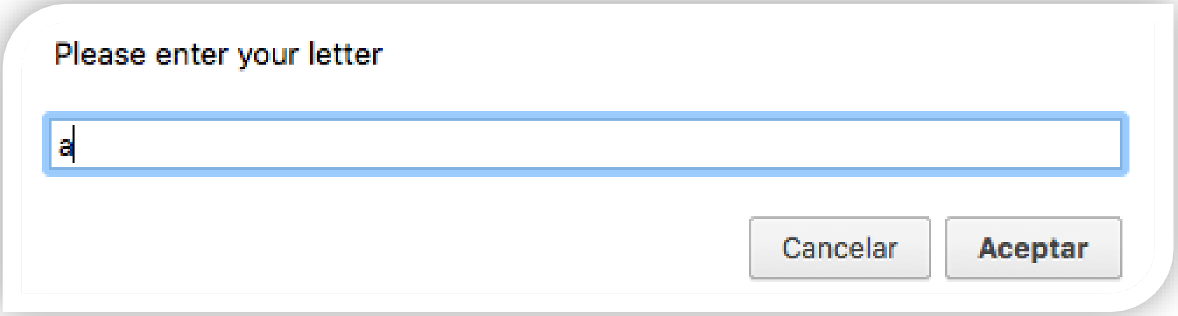
```
input data
----------
  +birthYear: 2001
  +year: 2018
calculating your age...
You are 17 years old
Sorry, no beer for you. You have to
wait 1 years to be of legal age...
The factorial of your age is:
355687428096000
```

Exercise 1.4

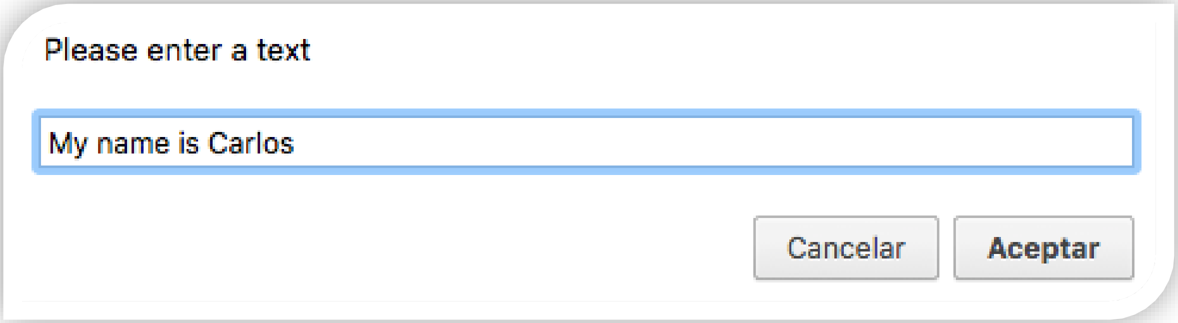
Create a small program to count the number of times that a given letter appears within a given text. The user will introduce the text and the letter using the prompt() function. Use the alert() function to display the result. Use loops and conditional statements to count the number of times that the letter appears in the text.

- Javascript prompt docs: https://www.w3schools.com/jsref/met_win_prompt.asp
- Prompt example: https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_prompt
- Javascript alert docs: https://www.w3schools.com/jsref/met_win_prompt.asp
- Alert example: https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_alert
- Javascript lenght property: https://www.w3schools.com/jsref/jsref_length_string.asp

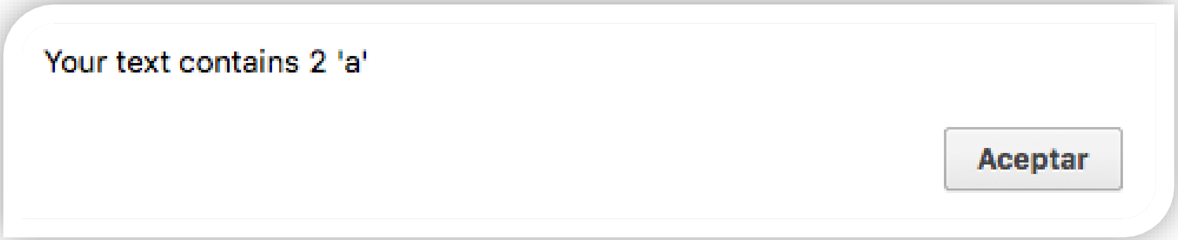
See images below.


Please enter your letter

a


Please enter a text

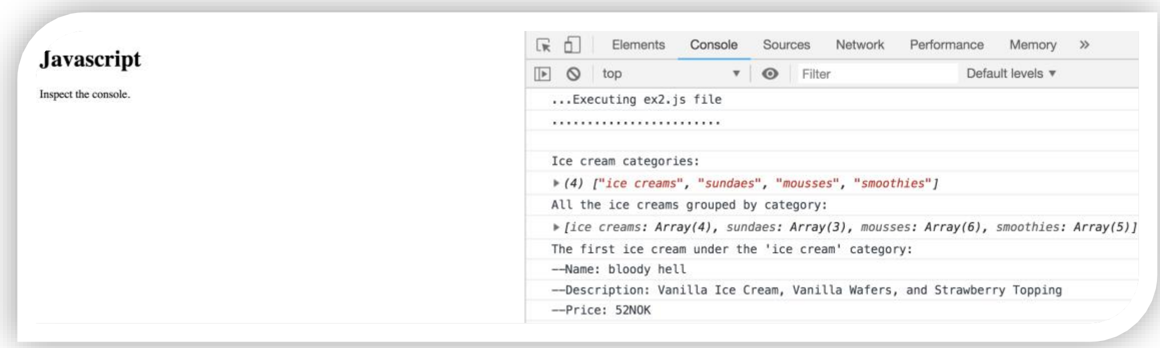
My name is Carlos


Your text contains 2 'a'

Exercise 2

Let’s try to print all the ice creams from coursework 3 using JavaScript.

We will start opening the “ex2.html” file (provided with the sources of this lab) in our browser and checking the output of the console.



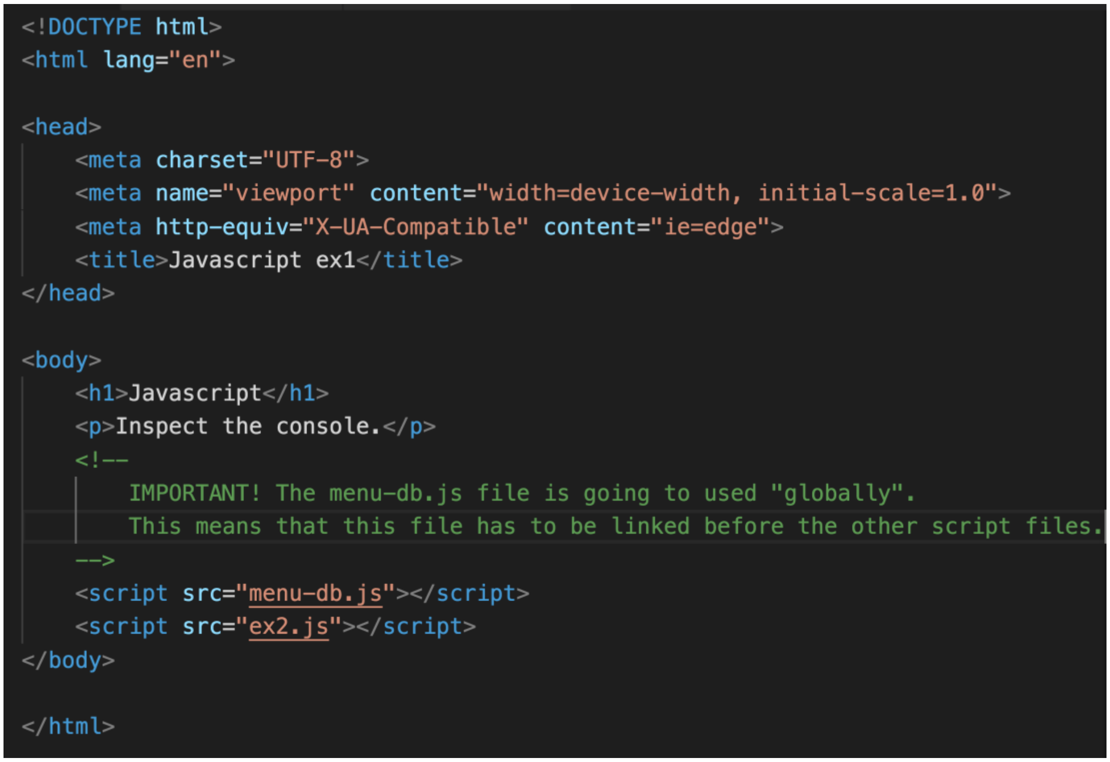
Javascript

Inspect the console.

```
...Executing ex2.js file
........................

Ice cream categories:
▸(4) ["ice creams", "sundaes", "mousses", "smoothies"]
All the ice creams grouped by category:
▸[ice creams: Array(4), sundaes: Array(3), mousses: Array(6), smoothies: Array(5)]
The first ice cream under the 'ice cream' category:
--Name: bloody hell
--Description: Vanilla Ice Cream, Vanilla Wafers, and Strawberry Topping
--Price: 52NOK
```

First thing you should notice is that all the text printed in the console is executed by the ex2.js file.

Also, inspecting the ex2.html code, you will see that the source of the html file is first loading the “menu-db.js” script and after that it also links the ex2.js file. The order is important. The menu-db.js file has to be linked first.

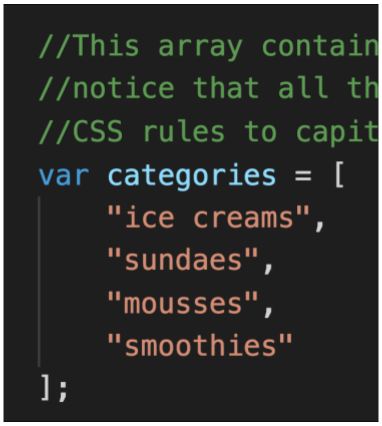


```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Javascript ex1</title>
</head>

<body>
    <h1>Javascript</h1>
    <p>Inspect the console.</p>
    <!--
        IMPORTANT! The menu-db.js file is going to used "globally".
        This means that this file has to be linked before the other script files.
    -->
    <script src="menu-db.js"></script>
    <script src="ex2.js"></script>
</body>

</html>
```

Exercise 2.1

Open the ex2.js file and read the comments. After that, remove all the code and implement your solution.

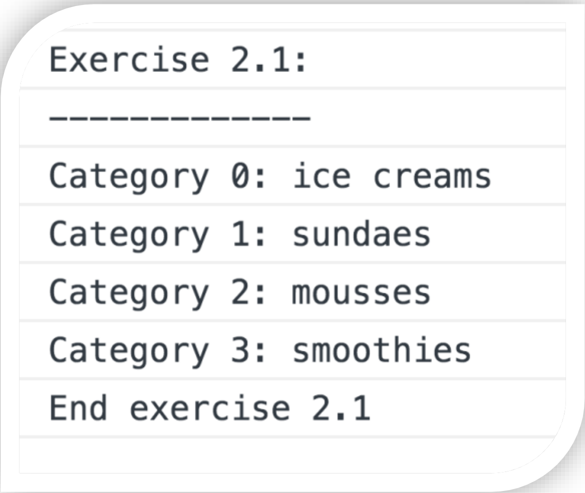
Your task consists in printing all the “categories” (menu-db.js file) in the console.



```
//This array contain
//notice that all th
//CSS rules to capit
var categories = [
    "ice creams",
    "sundaes",
    "mousses",
    "smoothies"
];
```

The previous code displays the names of all the categories: ice creams, sundaes, mousses and smoothies.

You have to use a loop to go through all of the elements of the “categories” variable (that contains a list of elements, this is, an array). The output of the console should look like the following image.

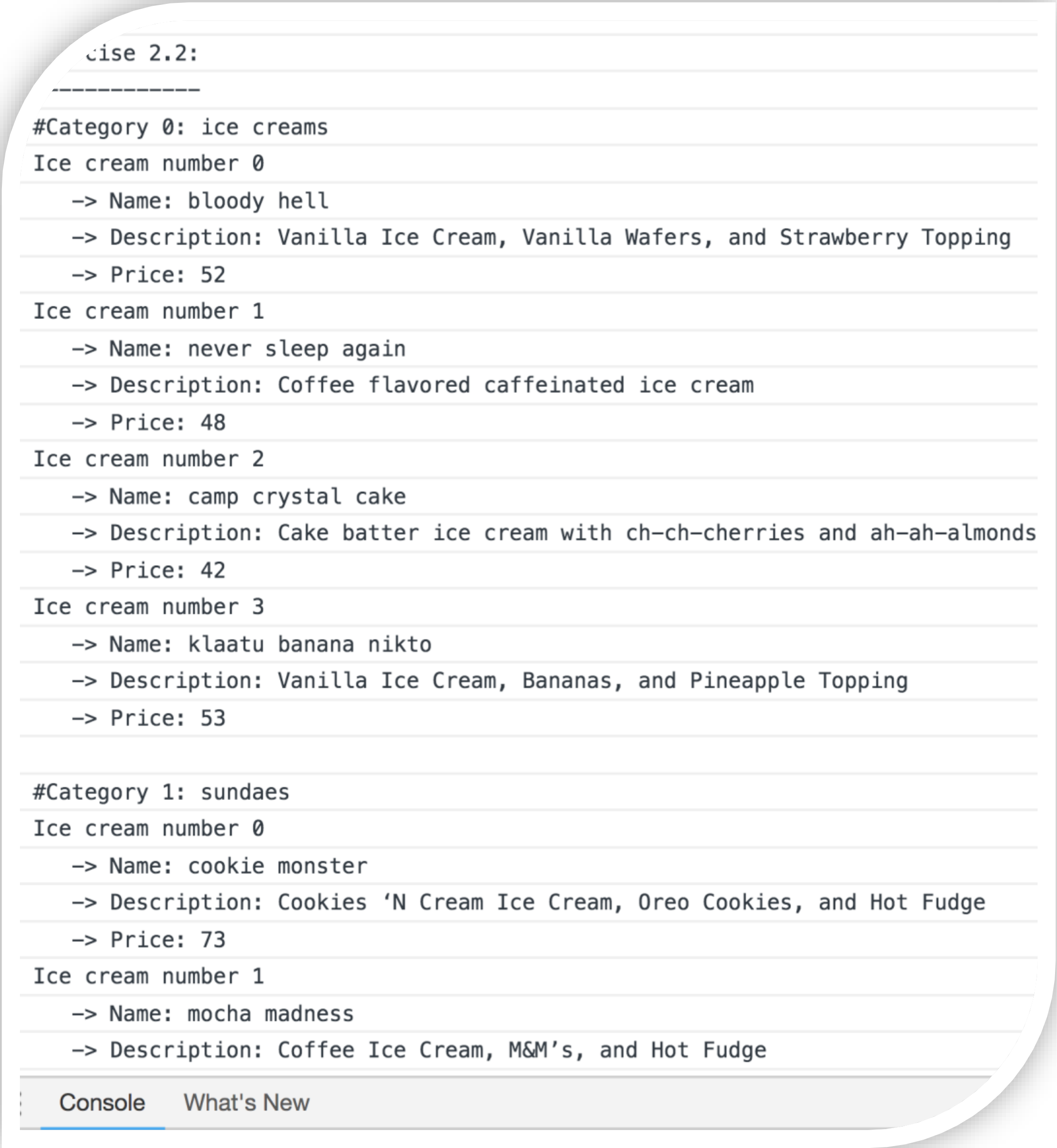


```
Exercise 2.1:
-------------
Category 0: ice creams
Category 1: sundaes
Category 2: mousses
Category 3: smoothies
End exercise 2.1
```

Exercise 2.2

The final exercise aims to print all the ice creams under each one of the categories in the console. This can be done using a nested loop. This is, the first loop will go through all the categories (categories variable) and inside that loop there will be another loop going through all of the ice creams of that category.

The final result will look like this:



```
 cise 2.2:
------------
#Category 0: ice creams
Ice cream number 0
   -> Name: bloody hell
   -> Description: Vanilla Ice Cream, Vanilla Wafers, and Strawberry Topping
   -> Price: 52
Ice cream number 1
   -> Name: never sleep again
   -> Description: Coffee flavored caffeinated ice cream
   -> Price: 48
Ice cream number 2
   -> Name: camp crystal cake
   -> Description: Cake batter ice cream with ch-ch-cherries and ah-ah-almonds
   -> Price: 42
Ice cream number 3
   -> Name: klaatu banana nikto
   -> Description: Vanilla Ice Cream, Bananas, and Pineapple Topping
   -> Price: 53

#Category 1: sundaes
Ice cream number 0
   -> Name: cookie monster
   -> Description: Cookies ‘N Cream Ice Cream, Oreo Cookies, and Hot Fudge
   -> Price: 73
Ice cream number 1
   -> Name: mocha madness
   -> Description: Coffee Ice Cream, M&M’s, and Hot Fudge
```