

# WEB DEVELOPMENT

## Practicum - Week 7.1

### Exercise guidelines

#### Folder structure

For this exercise, **create a new folder called week 7.1.** in your web development folder. Only put the files necessary to complete the exercise in here.

**DO NOT** add any source materials in this folder (screenshots, pdf files, ...)

#### Submission deadline

These exercises do not require submission.

### Plan of attack

If you've paid attention in the lectures, you should be able to complete every "basic" exercise. Some exercises are marked as optional. You are free to complete these as an additional practice tool, but are not strictly required.

You can create these exercises in an online tool or others, complete it in the console directly or link an HTML page to a JavaScript file. Choose the method of your preference.

#### FANCY NANCY

For those among you with some more programming experience, or those looking to go the extra mile, there are **FANCY NANCY** challenges.

Try these to enhance your JS development skills!

#### PAPER TRAIL

Every successful exercise starts with a paper trail, which means a plan of attack which you have written down - on paper - in your own words. For some exercises, these paper trails are mandatory. These are marked with the paper trail icon.

## Exercise 1: Hello world!

Write “Hello Howest!” in the browser console window

## Exercise 2: Template Literals

Modify the code sample below by using template literals.

Note that there is no need for \n!

```
let myName = 'Edward Scissorhands';
let myJob = 'Gardener';
let myAge = 29;

console.log('My name is' + myName + 'and I am a ' + myJob + '.\n I am ' +
myAge + ' years old.');
```

Console output:

```
My name is Edward Scissorhands and I am a gardener.
I am 29 years old.
```

## Exercise 3: Prompt

Request the user for a **number** by using the prompt command.

Save this number in a variable.

Create a second variable which will hold a text message, saying “the number you entered was “

Write the result of the number along with the text message in the console window.

Documentation for the prompt method <https://developer.mozilla.org/nl/docs/Web/API/Window/prompt>

### PAPER TRAIL

Provide a paper trail of your thought process before coding this exercise

## Exercise 4: Sum

This is an **expansion of the previous exercise**.

Convert the following instructions to JavaScript code:

1. The user supplies a number

2. The user supplies a second number
3. Both numbers are added together
4. The result of the sum is posted in the console in the following format:  
Given numbers 7 and 4: the console reads: "7 + 4 = 11".

### PAPER TRAIL

Provide a paper trail of your thought process before coding this exercise

## Exercise 5: Array

Carry out the following array operations:

1. Create an array named **styles** containing two elements, "Jazz" and "Blues".
2. Add "Rock-n-Roll" at the end of the array
3. Replace the middle value with "Classics".  
Careful: do not simply replace the second element. Your code should still function, regardless of the array size (as long as its number is odd).
4. Remove the array's first element and display it in the console
5. Add two elements, "Rap" and "Reggae" at the start of the array.

Console display after all commands have been executed:

```
Jazz, Blues
Jazz, Blues, Rock-n-Roll
Jazz, Classics, Rock-n-Roll
Jazz
Rap, Reggae, Classics, Rock-n-Roll
```

## Exercise 6: Seconds [optional]

Ask the user by using a prompt-window for a number of seconds. Display in the console how many hours, minutes and seconds that makes.

E.g. input **7263** displays "7263 seconds is **2h 1m 3s**" in the console window

## Exercise 7: Even or odd

Ask the user for a number by using a prompt-window.

Check whether that number is even or odd. Write the result in the console.

E.g. input **7**, console displays '**7 is odd**'

## Exercise 8: Graduation

Students graduating from an institution, gain a certain grade.

1. the grade **'pass'**, when scoring between 50% and 68%
2. the grade **'merit'** when scoring at least 68%
3. the grade **'distinction'** when scoring at least 77%
4. the grade **'cum laude'** when scoring at least 85%
5. the grade **'summa cum laude'** when scoring at least 90%

Only a single grade can be given to a student (i.e. you cannot be awarded both "merit" and "distinction", the higher grade trumps the lower)

### PAPER TRAIL

Provide a paper trail of your thought process before coding this exercise

## Exercise 9: fish, chips or fish & chips?

Purpose: Determine which numbers between x and y are divisible by 3 as well as 5?

1. Create a for loop which counts down from 1 to 100
2. Check for every number whether it's divisible by **3**. If so, display **'fish'**
3. Check if that same number is divisible by **5**, if so display **'chips'**
4. In case neither conditions are true, simply write the number itself to the console
5. Careful: a number can be divisible by both! In that case, display **'fish & chips'**

```
1
2
fish
4
chips
fish
7
8
fish
chips
11
fish
13
14
fish & chips
16
...
```

### PAPER TRAIL

Provide a paper trail of your thought process before coding this exercise

### FANCY NANCY

1. Allow the user to determine the start and end of the loop by asking for min and max values through a prompt window
2. Allow the user to choose which divisible values should be checked

## Exercise 10: Password protect [optional]

Use a **while-loop** for this exercise.

Create a hardcoded const value names **secretPassword** containing your password (e.g. 'stephenfry')

Display a **prompt upon page load**, where you ask for the password. As long as the user does not enter the correct password, the prompt window reappears.

Upon entering the correct value, the console reads: **'Welcome, he who guessed the secret password after 7 tries`**.

### PAPER TRAIL

Provide a paper trail of your thought process before coding this exercise

### FANCY NANCY

1. Show the secret password in the console, but randomise its letters (e.g. frsteyhpen) This enables the user to provide a more accurate guess.