SWA1 Exercises 1

# Exercise 1.1 – Installation

1. Install node.js from <https://nodejs.org/en/>
2. Run the program below. What does it do?

const chars = {

'1': 'e',

'8': 'r',

'11': '!',

'4': 'o',

'0': 'H',

'10': 'd',

'6': 'W',

'9': 'l',

'2': 'l',

'7': 'o',

'3': 'l',

length: 12

}

let msg = ''

for(let i = 0; i < chars.length; i++) {

if (chars[i])

msg = msg + chars[i]

else

msg = msg + ' '

}

console.log(msg)

# Exercise 1.2 – Implement composition

1. The class diagram below is a fine Java class diagram, but it is not how we program object-oriented in JavaScript. Implement the model in JavaScript using factory functions.



1. Create an array of circles. Use the array map() method to create an array with the radius of each circle.
2. We want to add an overloaded constructor to Circle:   
    Circle(x: double, y: double, radius: double)   
   JavaScript doesn't support overloading like Java. How do you implement this? (Hint: use [object destructuring](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment).)

# Exercise 1.3 – Extending objects

Consider the class diagram below



Realize the class diagram using factory functions. (Hint: Use the [spread syntax](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_syntax).)

# Exercise 1.4 – JavaScript Recap

## Exercise 1.4.1 – truthy

Which of the following are truthy?

1. 2 + 2 === 4
2. 2 + 2 === '4'
3. 2 + 2 == '4'
4. Number('4')
5. Number('0')
6. NaN
7. NaN === NaN
8. NaN !== NaN
9. Infinity === Infinity
10. 1/0 == 2/0
11. 2 \* null
12. 2 + null
13. 7
14. null || 7
15. '4'
16. ''

## Exercise 1.4.2 – loops

1. Make a loop that prints (using console.log) the numbers from 1 to 10
2. Make a loop that adds the numbers from 1 to 10
3. Make a loop that computes 10! (factorial)

## Exercise 1.4.3 – arrays

var a = [1, 2, 3, 5, 8] creates an array.

a.length is the length of the array (5)

a[0], …, a[4] are the elements of the array.

1. What’s a[5]?
2. Make a loop that prints the elements of a
3. Make a loop that adds the elements of a
4. Make a function that takes an array and returns the sum of its elements
5. Add an element to a like this: a[8] = 55
6. What’s a[8]?
7. What’s the length of a?
8. What happens if you print a to the console?
9. What happens with your loop from (c)?

## Exercise 1.4.4 – basic functions

1. Make a function, factorial, that takes a value n and returns n!
2. Make a function, power, that takes values m and n and returns mn.

## Exercise 1.4.5 – advanced functions

Make a function that takes two arguments, m and n. If n is undefined, the function should return m!, otherwise the function should return mn.  
What happens if you call the function with only one argument?