JavaScript mini challlenges (with answers)

Exercise 1: hoisting

- 'use strict'

```
myString = 'something'
console.log(variableWithHosting)
```

```
▶ Uncaught ReferenceError: myString is not defined
    at script.js:16:10
```

- comment out 'use strict'

```
variableWithHosting = 'something'
console.log(variableWithHosting)
```

```
something
```

- var and let

```
16  console.log(myString)
17  console.log(myNumber)
18  var myString
19  let myNumber
```

```
undefined                                                              script.js:16
▶ Uncaught ReferenceError: Cannot access 'myNumber' before initialization   script.js:17
    at script.js:17:13
```

```
16  myString = 'something'
17  console.log(myString)
18  myNumber = 23          You,
19  console.log(myNumber)
20  var myString
21  let myNumber
```

```
<========== YOUR CODE ==========>                                        script.js:17
something                                                              script.js:17
▶ Uncaught ReferenceError: Cannot access 'myNumber' before initialization   script.js:18
    at script.js:18:10
```

Exrcise2: Variable mutation and comparison

```
16  const num = 7        You, 7 minutes ago • Uncommi
17  const str = '3'
18  console.log('variable mutation')
19  console.log(typeof num)
20  console.log(typeof str)
21  console.log(str + num, typeof (str + num))
22  console.log(str * num, typeof (str + num))
23  console.log(str * 1, typeof (str * 1))
24  console.log(+str, typeof (+str))
25  console.log('' + num, typeof ('' + num))
26  console.log('explicit type coersion')
27  console.log(String(num), typeof String(num))
28  console.log(Boolean(num), typeof Boolean(num))
29  console.log('falsy values')
30  console.log(Boolean(0))
31  console.log(Boolean(''))
32  console.log(Boolean(null))
33  console.log(Boolean(undefined))
34  console.log(Boolean(NaN))
35  console.log('truthy values')
36  console.log(Boolean(1))
37  console.log(Boolean(' '))
38  console.log(Boolean([]))
39  console.log(Boolean({}))
40  console.log(Boolean(Infinity))
```

```
variable mutation                script.js:18
number                           script.js:19
string                           script.js:20
37 string                        script.js:21
21 'string'                      script.js:22
3 'number'                       script.js:23
3 'number'                       script.js:24
7 string                         script.js:25
explicit type coersion           script.js:26
7 string                         script.js:27
true 'boolean'                   script.js:28
falsy values                     script.js:29
false                            script.js:30
false                            script.js:31
false                            script.js:32
false                            script.js:33
false                            script.js:34
truthy values                    script.js:35
true                             script.js:36
true                             script.js:37
true                             script.js:38
true                             script.js:39
true                             script.js:40
```

```
console.log('comparison')
// https://developer.mozilla.org/en-US/docs/Web/JavaScript/Equality_comparisons_and_sameness
console.log('0 == false: ', 0 == false)
console.log('1 == true: ', 1 == true)
console.log('1 === true: ', 1 === true)
console.log('1 == "1": ', 1 == '1')
console.log('1 === "1": ', 1 === '1')
console.log('null == undefined: ', null == undefined)
// when we have false in the comparison it will be coersed to number
console.log('null == false: ', null == false)
//NaN is not a valid number, and thus cannot be compared to any other value, even to itself
console.log('NaN === NaN: ', NaN === NaN)
console.log('!null: ', !null)
console.log('!!null: ', !!null)
console.log('!NaN: ', !NaN)
console.log('!!NaN: ', !!NaN)
```

```
comparison                          script.js:16
0 == false:  true                   script.js:17
1 == true:  true                    script.js:18
1 === true:  false                  script.js:19
1 == "1":  true                     script.js:20
1 === "1":  false                   script.js:21
null == undefined:  true            script.js:22
null == false:  false               script.js:24
NaN === NaN:  false                 script.js:26
!null:  true                        script.js:27
!!null:  false                      script.js:28
!NaN:  true                         script.js:29
!!NaN:  false                       script.js:30
```

Exercise 3: Objects and this keyword

1. Create a person object with attributes firstName, lastName, birthYear and a printInfo function (leave blank for now)
2. log persons firstName
3. log persons lastName but using something like person['last'+'Name']

4. get the date today and store to a variable the current year (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/getFullYear)
5. in the print info function try to return a string like that 'FullName: yourFullName, Age: yourAge'
6. log the result of the printInfo function
7. Use template literal to return the info (step 5)
8. copy the person object code and create a person2 object with only difference instead of a named function use an arrow function
9. log the function results, what do you observe?

```
45  const dateToday = new Date()
46  const currentYear = dateToday.getFullYear()

47  const person = {
48    firstName: 'Maria',
49    lastName: 'Mousiou',
50    birthYear: 1986,
51    printInfo () { return `FullName: ${this.firstName}
    ${this.lastName}, Age: ${currentYear - this.birthYear}` }
52  }

53  console.log(person.printInfo())

54  const person2 = {
55    firstName: 'Maria',
56    lastName: 'Mousiou',
57    birthYear: 1986,
58    printInfo: (rec) => { return `FullName: ${rec.firstName}
    ${rec.lastName}, Age: ${currentYear - rec.birthYear}` }
59  }

60  try {
61    // if we don't use try catch the execution will stop to the
    error. With try..catch the next line will also be printed
62    console.log(person2.printInfo())
63  } catch (err) {
64    console.error(err)
65  }

66  console.log(person2.printInfo(person2))
```

```
FullName: Maria Mousiou, Age: 37                                    script.js:53
▶ TypeError: Cannot read properties of undefined (reading 'firstName')  script.js:64
    at Object.printInfo (script.js:58:49)
    at script.js:62:23
FullName: Maria Mousiou, Age: 37                                    script.js:66
```

Exercise 4: Arrays

Given the array [4,5,-1,6,0,10,3]

1. calculate the min of the array using for loop or Array.prototype.forEach
2. calculate the max of the array using for loop or Array.prototype.forEach
3. calculate the sum of the elements using for loop, Array.prototype.forEach or Array.prototype.reduce

```
16  const myArray = [4,5,-1,6,0,10,3]
17  let min = myArray[0]
18  let max = myArray[0]
19  myArray.forEach(el => {
20    min = el < min ? el : min
21    max = el > max ? el : max
22  })
23  console.log(` min: ${min}, max: ${max}`)
24  const sum = myArray.reduce((acc, item) => acc + item, 0)
25  console.log('Sum of array items: ', sum)
```

```
min: -1, max: 10                                    script.js:23
Sum of array items:  27                             script.js:26
```