Name: Bob Borsboom
Studentnumber: 10802975
Course: Data Processing

Questions WK 2

Answer the following questions in your own words. This assignment will only be graded pass or fail.

1. Explain the difference between the == operator and the === operator.

==: Test whether a value has a real value instead of null or undefined
===: Tests whether a value is precisely equal to the other

2. Explain what a closure is. (Note that JavaScript programs use closures very often.)

The ability to treat functions as values, combined with the fact that local variables are “re-created” every time a function is called

Being able to reference a specific instance of local variables in an enclosing function—is called closure. A function that “closes over” some local variables is called a closure

being able to reference a specific instance of local variables in an enclosing function

3. Explain what higher order functions are.

higher order functions: Functions that operate on other functions, either by taking them as arguments or by returning them

Higher-order functions allow us to abstract over actions, not just values.
They come in several forms. For example, you can have functions that create new functions:

```
function greaterThan ( n ) {
return function ( m ) { return m > n ; };
}
var greaterThan10 = greaterThan (10) ;
console . log ( greaterThan10 (11) ) ;
// → true
```

4. Explain what a query selector is and give an example line of JavaScript that uses a query selector.

Query selector: takes a selector string and returns an arraylike object containing all the elements that it matches.

EXAMPLE:

```
<p > And if you go chasing
< span class =" animal " > rabbits </ span > </p >
<p > And you know you ' re going to fall </ p >
<p > Tell ' em a < span class =" character " > hookah smoking
< span class =" animal " > caterpillar </ span > </ span > </p >
<p > Has given you the call </ p >
< script >
function count ( selector ) {
return document . querySelectorAll ( selector ) . length ;
}
console . log ( count (" p ") ) ; // All <p > elements
// → 4
console . log ( count (". animal ") ) ; // Class animal
// → 2
console . log ( count (" p . animal ") ) ; // Animal inside of <p >
// → 2
console . log ( count (" p > . animal ") ) ; // Direct child of <p >
// → 1
</ script >
```

This one is useful if you want a specific, single element. It will return only the first matching element or null if no elements match.