Bronnen – dag 3

Objects and Arrays

The Coding Train - What is an array? (video)

array index

= order of an element in the array (starts with 0)

The Coding Train - JavaScript Objects (video)

dot syntax

```
var circle {
      x: 0,
      y: 200,
      diameter: 30
}

ellipse(circle.x)
```

MAAR: vanaf ES5 function circle() bestaat, dus gebruik niet een variabele genaamd ‘circle’, want dit werkt niet meer (bron: comments)

Eloquent JavaScript - Data (book)

array index

Zero-based counting has a long tradition in technology and in certain ways makes a lot of sense, but it takes some getting used to. **Think of the index as the amount of items to skip**, counting from the start of the array.

So if you know that the property you are interested in is called *color*, you say `value.color`. If you want to extract the property named by the value held in the binding `i`, you say `value[i]`. **Property names are strings. They can be any string, but the dot notation works only with names that look like valid binding names.** So if you want to access a property named *2* or *John Doe*, you must use square brackets: `value[2]` or `value["John Doe"]`.

`array.length` is ook gewoon een property en zou dus geschreven kunnen worden als `array["length"]`

methods

```
let sequence = [1, 2, 3];
```

```
sequence.push(4);
sequence.push(5);
console.log(sequence);
// → [1, 2, 3, 4, 5]
console.log(sequence.pop());
// → 5
console.log(sequence);
// → [1, 2, 3, 4]
```

The `push` method adds values to the end of an array, and the `pop` method does the opposite, removing the last value in the array and returning it.

delete operator

```
let anObject = {left: 1, right: 2};
console.log(anObject.left);
// → 1
delete anObject.left;
console.log(anObject.left);
// → undefined
console.log("left" in anObject);
// → false
console.log("right" in anObject);
// → true
```

keys method

```
console.log(Object.keys({x: 0, y: 0, z: 2}));
// → ["x", "y", "z"]
```

assign method

```
let objectA = {a: 1, b: 2};
Object.assign(objectA, {b: 3, c: 4});
console.log(objectA);
// → {a: 1, b: 3, c: 4}
```

mutablility of const objects

```
const score = {visitors: 0, home: 0};
// This is okay
score.visitors = 1;
// This isn't allowed
score = {visitors: 1, home: 1};
```

Jacques the Weresquirrel

Het verhaal van de Jacques the Weresquirrel is aangrijpend, maar de code is op een gegeven moment niet te begrijpen. Ook is er een tweede einde, dat eigenlijk niets toevoegt aan het verhaal; puur voor shock value. 6/10

Idee voor dating app: SCU (Shrek Cinematic Universe)

indexOf & lastIndexOf

```
console.log([1, 2, 3, 2, 1].indexOf(2));
// → 1
console.log([1, 2, 3, 2, 1].lastIndexOf(2));
// → 3
```

concat & slice

The following example shows both `concat` and `slice` in action. It takes an array and an index, and it returns a new array that is a copy of the original array with the element at the given index removed.

```
function remove(array, index) {
  return array.slice(0, index)
    .concat(array.slice(index + 1));
}
console.log(remove(["a", "b", "c", "d", "e"], 2));
// → ["a", "b", "d", "e"]
```

strings aanpassen

```
console.log("coconuts".slice(4, 7));
// → nut

console.log("coconut".indexOf("u"));
// → 5

console.log("one two three".indexOf("ee"));
// → 11

console.log("  okay \n ".trim());
// → okay
```

(The `trim` method removes whitespace (spaces, newlines, tabs, and similar characters) from the start and end of a string.)

```
console.log(String(6).padStart(3, "0"));
// → 006
```

(`padStart` and takes the desired length and padding character as arguments)

```
let sentence = "Secretarybirds specialize in stomping";
let words = sentence.split(" ");
console.log(words);
// → ["Secretarybirds", "specialize", "in", "stomping"]
console.log(words.join(". "));
// → Secretarybirds. specialize. in. stomping

console.log("LA".repeat(3));
// → LALALA
```

rest parameters: three dot notation

```
function max(...numbers) {
  let result = -Infinity;
  for (let number of numbers) {
    if (number > result) result = number;
  }
  return result;
}
console.log(max(4, 1, 9, -2));
// → 9

let numbers = [5, 1, 7];
console.log(max(...numbers));
// → 7

let words = ["never", "fully"];
console.log(["will", ...words, "understand"]);
// → ["will", "never", "fully", "understand"]
```