



Javascript

# Table of contents

- Inleiding
- Core Javascript
- Client Side Javascript
- Ajax

# Inleiding

# Javascript

= SCRIPTTAAL. Een implementatie van ECMAScript ( momenteel versie 6).

- OOP
- Platformonafhankelijk
- Cliënt-server toepassingen

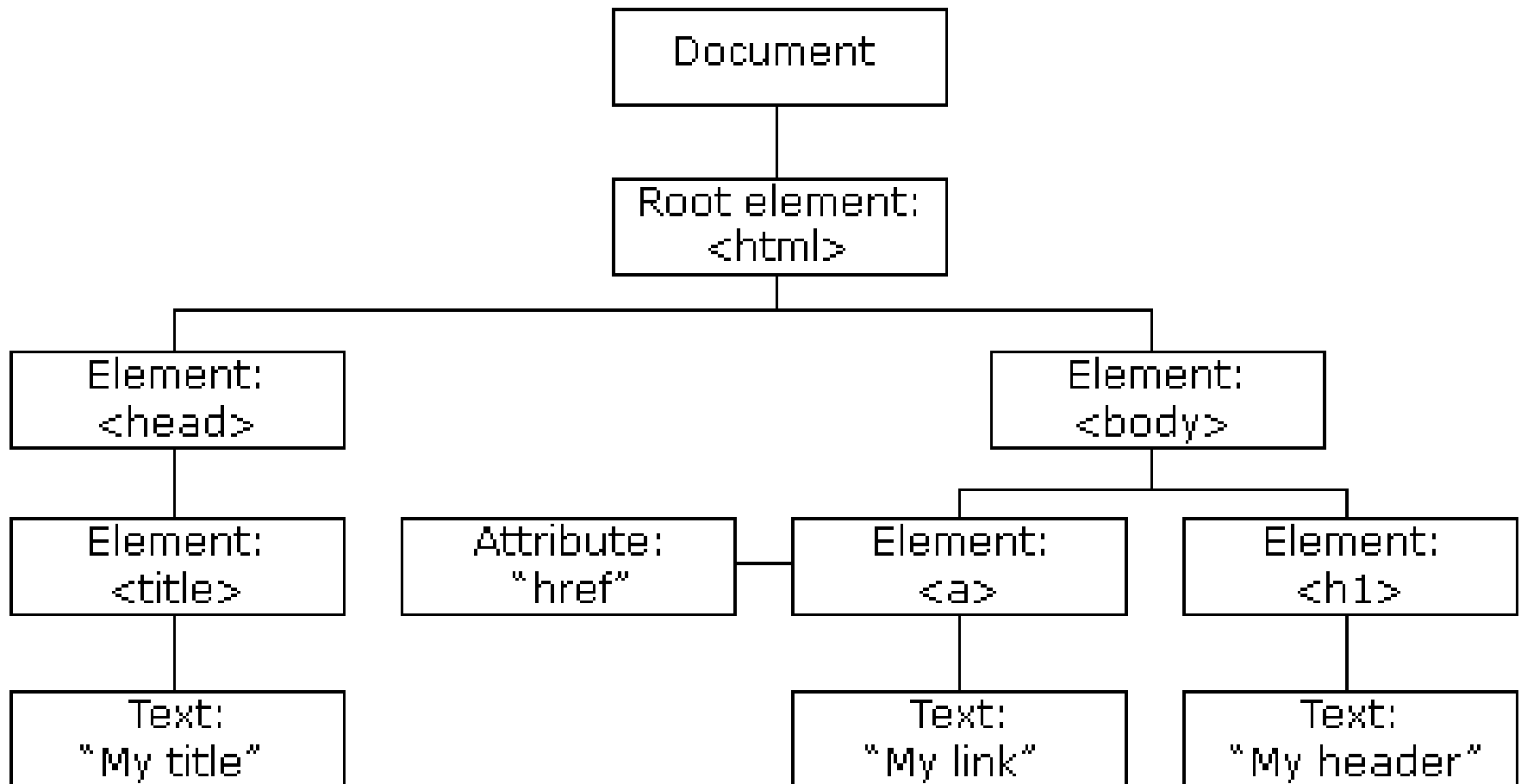
# Javascript

**Core Javascript:** Syntax van de programmeertaal

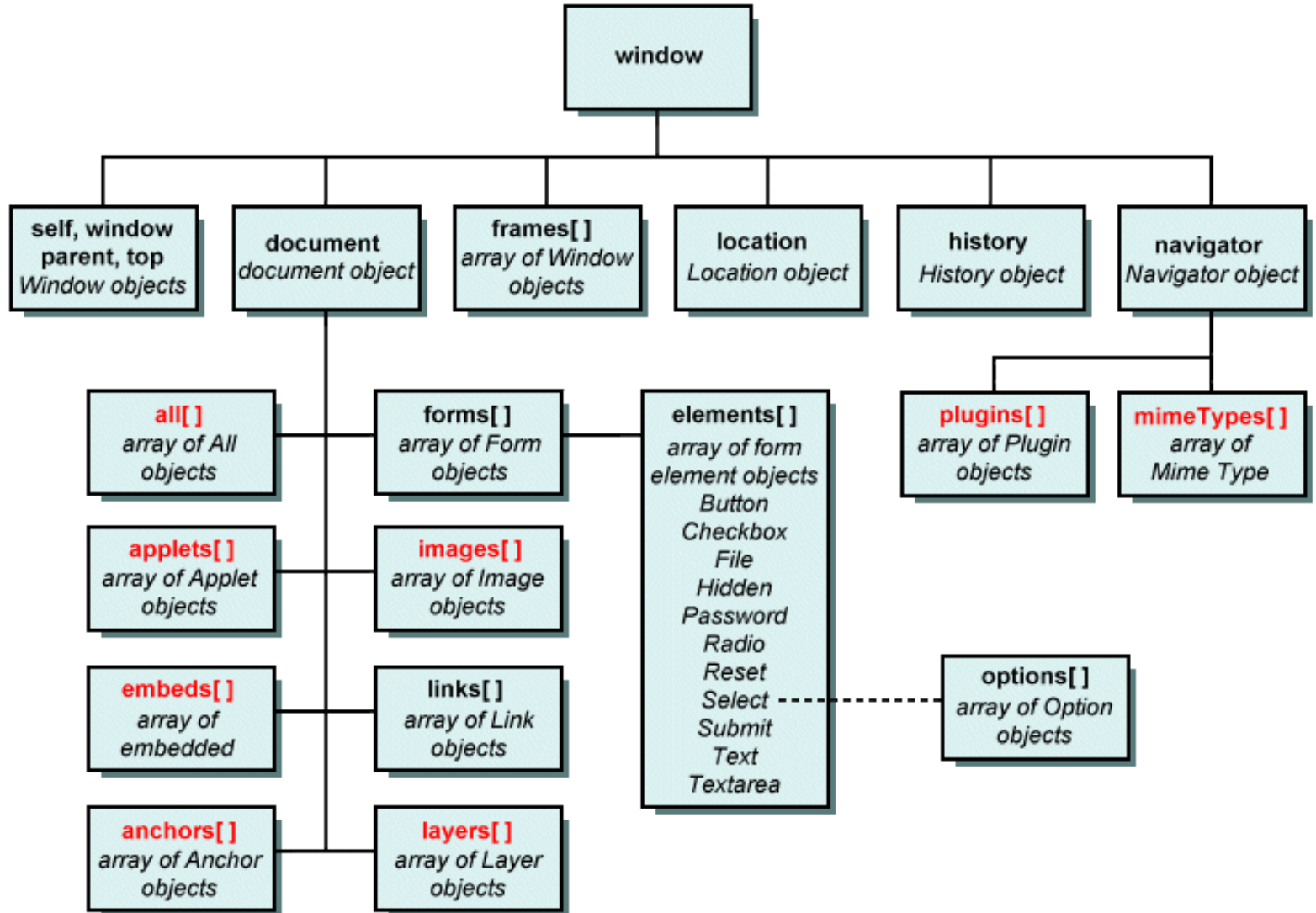
**Client side Javascript:** Gebaseerd op Core Javascript. Samenwerking met HTML/CSS

**Server Side Javascript:** Eerste geïnterpreteerd door webserver.

# Document Object Model



# Browser Object Model



# Javascript VS Java

Java	Javascript
Gecompileerd (Java Compiler) en Geïnterpreteerd (JVM)	Geïnterpreteerd (Browser)
Op klassen gebaseerd	Op prototypes gebaseerd
Strongly typed	Weakly/ dynamically typed



# Core Javascript

Werken met javascript

# Benodigdheden

Editor: Visual Studio Code

Browser: Chrome/ Firefox/ Safari/ Edge ...

Documentatie:

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- <https://www.w3schools.com/js/default.asp>

# Commentaar

Block commentaar:

```
/* Hallo  
   Javascript */
```

Programmaregel commentaar:

```
// Hallo Javascript
```

# Literals

<b>Gehele getallen</b>	
1234	Decimaal getal 1234.
035	Octaal getal 35 (= 29 decimaal). Octale getallen hebben als grondtal 8. De notatie begint met een 0.
0x22	Hexadecimaal getal 22 (= 54 decimaal). Hexadecimale getallen hebben als grondtal 16. De notatie begint met 0x.
<b>Reële getallen</b>	
34.235	Getal met komma notatie. Met gebruikt hiervoor steeds een punt.
34235E-3	Getal met wetenschappelijke notatie. 34235 is de <i>mantisse</i> en -3 de <i>exponent</i> . $34235 \times 10^{-3} = 34.235$ .
<b>Strings of tekenreeksen</b>	
"Hello World" 'Hello World'	Reeks van karakters tussen enkelvoudige of dubbele aanhalingstekens.
<b>Voorgedefinieerde waarden</b>	
true	Booleaanse waarde 'waar'.
false	Booleaanse waarde 'vals'.
NaN	<u>N</u> ot a <u>N</u> umber : geeft aan dat een waarde geen getal is.
undefined	Ongedefinieerd: onbepaalde waarde.
null / NULL	Niets: referentie naar niets (geen enkel object).

# Escape Characters

Code	Outputs
\'	single quote
\"	double quote
\\	backslash

Five other escape characters are valid in JavaScript:

Code	Outputs
\b	Backspace
\r	Carriage Return
\f	Form Feed
\t	Horizontal Tabulator
\v	Vertical Tabulator

# Variabelen

```
number = 5;
```

```
var myNumber = 10;
```

```
var myString = "Hello World";
```

```
var myUndefined;
```

```
var myBoolean = true;
```

```
var myDecimal = 10.10;
```

???

```
var myBoolean = true;  
var sum = 10 + myBoolean;  
console.log("Sum= " + sum);
```

```
var myString = "1";  
var sum1 = 10 + myString;  
var division = 10 - myString;  
console.log("Sum1= " + sum1);  
console.log("Division= " + division);
```

**Oplossing:**

Sum= 11

Sum1= 101

Division= 9

# Opdracht 1

Vraag aan de gebruiker zijn naam en leeftijd en druk deze af in de console.

Maak gebruik van de prompt() functie:

<https://developer.mozilla.org/en-US/docs/Web/API/Window/prompt>



# Operatoren

- Rekenkundige
- Relationale
- Logische
- Shift & bit
- Toekennings
- String
- Conditionele

# Operatoren

```
var getal1 = 10;  
var getal2 = "10";  
  
console.log(getal1 == getal2);  
console.log(getal1 === getal2); // Zelfde type  
console.log(getal1 != getal2);  
console.log(getal1 !== getal2); // Verschillend of ander type
```

# Opdracht 2

## **Opdracht 2: Operatoren gebruiken**

In deze opdracht maken we een script waarbij we aan de gebruiker zijn leeftijd vragen. Op basis van deze waarde maken we allerlei berekeningen.

- Maak een nieuwe HTML-pagina met een script.
- Definieer een variabele `age`.
- Vraag de leeftijd aan de gebruiker via `prompt()` en ken deze waarde toe aan de variabele `age`.
- Voeg een variabele `days` toe die het aantal dagen bevat. Bereken dit aantal op basis van de leeftijd. Druk het aantal dagen af.
- Voeg een variabele `weeks` toe die het aantal weken bevat. Bereken dit aantal op basis van het aantal dagen. Druk het aantal weken af.
- Ga na of de persoon meerderjarig is door te controleren of zijn leeftijd ligt tussen 0 en 18 jaar. Voeg hiervoor een variabele met de naam `adult` toe. Druk de waarde van deze variabele af.
- Gebruik de conditionele operator samen met de variabele `adult` om af te drukken of de persoon meerderjarig is:

```
console.log(adult?"Adult":"Minor");
```

- Ga na wat er gebeurt als de gebruiker op *Annuleren* klikt. Ken in dit geval een *default* waarde toe met behulp van de *default* operator.

# Statements

## **Uitdrukkingstatement:**

```
value1 = value2 + value3;  
function();  
value2++;
```

## **Declaratiestatement:**

```
var value = 8;
```

## **Programmaverloop statement:**

```
if {} else {}, while(), for()...
```

# Programmaverloop-statements

## Herhaling:

### **Opdracht 3: Het *while* statement gebruiken**

- Maak een script dat de getallen van 100 tot 120 afdrukt in omgekeerde volgorde.
- Maak een script dat de leeftijd aan de gebruiker vraagt. Indien de leeftijd niet tussen 0 en 150 ligt, dient met de vraag te herhalen totdat die wel ligt tussen de grenzen. Gebruik hiervoor een `do while`-lus.

### **Opdracht 4: Het *for* statement gebruiken**

- Maak een script dat de getallen van 100 tot 120 afdrukt in omgekeerde volgorde.

# Programmaverloop-statements

## Herhaling:

### Opdracht 5: Het *if else* statement gebruiken

- Maak een script dat de lengte en gewicht van de gebruiker opvraagt en hiervan de *body mass index* berekent ( $\text{gewicht}/(\text{lengte}^2)$ ). Druk de mate van overgewicht af naargelang het resultaat:

< 18: ondergewicht

18 - 25: normaal

25 - 27: licht overgewicht

27 - 30: matig overgewicht

30 - 40: ernstig overgewicht

> 40: obesitas

# Programmaverloop-statements

## Herhaling:

### ***Opdracht 6: Het `switch` statement gebruiken***

- Maak een script dat de gebruiker vraagt naar zijn taal ('nl', 'fr', 'en'). Druk de tekst "Dag wereld" af in de gekozen taal. Druk een foutmelding af indien een ongekende taal wordt opgegeven.
- Laat de `break` eens weg en ga na wat het resultaat is.

# Functions

```
function myFunctionName(params){  
    statements;  
}
```

Anonieme functie of function literal:

```
var functionName = function(params){  
    statements;  
}
```



# Lokale vs Globale variabelen

## **Globale variabelen:**

scope = hele document.

→ Variabelen buiten functies of variabelen zonder **VAR** declaratie.

## **Lokale variabelen:**

scope = binnen codeblok

# Lokale vs Globale variabelen

```
hello = "Hello World!";  
function myFunction(){  
    var hello = "Hello Javascript!";  
    console.log(hello);  
    console.log(this.hello);  
}  
myFunction();
```

```
Hello Javascript!  
Hello World!
```

# Let (since ECMAScript 6) vs var

```
function myFunction() {  
  //i is *not* visible out here  
  for( let i = 0; i < 5; i++ ) {  
    //i is only visible in here (and in the for() parentheses)  
    //and there is a separate i variable for each iteration of the loop  
  }  
  //i is *not* visible out here  
}
```

```
function myFunction() {  
  //i *is* visible out here  
  for( var i = 0; i < 5; i++ ) {  
    //i is visible to the whole function  
  }  
  //i *is* visible out here  
}
```

# Let vs var

```
let me = 'foo';  
let me = 'bar'; // SyntaxError: Identifier 'me' has already been declared
```

```
var me = 'foo';  
var me = 'bar'; // No problem, `me` is replaced.
```

# Functions

```
function sayHello(name) {  
  console.log("Hello " + name + "!");  
}
```

```
sayHello("Kenneth");  
sayHello();
```

```
Hello Kenneth!  
Hello undefined!
```

# Functions

```
function sayHello(name) {  
  name = name || "John Doe";  
  console.log("Hello " + name + "!");  
}
```

```
sayHello("Kenneth");  
sayHello();
```

```
Hello Kenneth!  
Hello John Doe!
```

# Functies met return type

```
function getSum(number1, number2){  
    return number1 + number2;  
}
```

```
var sum = getSum(10, 7);  
console.log("The sum= " + sum);
```

The sum= 17

# Opdracht

## **Opdracht 7: Functies definiëren en gebruiken**

- Maak voor het berekenen van de BMI een functie `calculateBmi()` en gebruik deze in het script.



# Functions als object

```
function sumOfTwoNumbers(a, b){  
    return a + b;  
}
```

```
function doSomething(a, b, sum){  
    return sum(a, b) > 0 ? "Positive number" :  
        "Negative number";  
}
```

```
var result = doSomething(10, -9, sumOfTwoNumbers);  
console.log(result);
```

# parseFloat()

```
var value = "10.13";  
var myFloat = parseFloat(value);  
console.log(myFloat);
```

```
var value1 = "10.13Hello";  
var myFloat1 = parseFloat(value1);  
console.log(myFloat1);
```

```
var value2 = "Hello10.13";  
var myFloat2 = parseFloat(value2);  
console.log(myFloat2);
```

10.13

10.13

NaN

# Opdracht

## **Opdracht 9: De globale functies gebruiken**

- Zoek in de documentatie de beschrijving van de `parseInt()` en de `isNaN()` functies.
- Maak een script dat de gebruiker twee getallen vraagt. Druk de som van de twee getallen af op het scherm. Geef een foutmelding indien een van de waarden geen getal is.

# OOP in javascript

Object heeft eigenschappen en methoden.

objectname.property (opvragen)

objectname.property = "Hello" (waarde toekennen)

objectname.method();

# Klassen versus prototypen

Klasse = blauwdruk van object → hele reeks objecten maken van de klasse = instanties.

Prototype = objecten baseren op reeds bestaande objecten.  
(alleen concrete objecten)

# Klassen versus prototypen

1. Object definiëren en onmiddellijk maken.
2. Nieuw object creëren door constructor.

# Person object

```
function Person(name, age, length, weight) {  
  this.name = name;  
  this.age = age;  
  this.length = length;  
  this.weight = weight;  
}  
var jos = new Person("Jos", 67, 188, 78);  
var eddy = new Person("Eddy", 43, 178, 70);  
console.log(jos.name);  
console.log(eddy);
```

Jos

Person { name: 'Eddy', age: 43, length: 178, weight: 70 }

# Person object

```
function Person(name, age, length, weight) {  
    this.name = name;  
    this.age = age;  
    this.length = length;  
    this.weight = weight;  
}  
  
var empty = new Person();  
var private = new Person("Ludo");  
console.log(empty);  
console.log(private);  
console.log(private.weight);
```

```
Person {  
  name: undefined,  
  age: undefined,  
  length: undefined,  
  weight: undefined }  
  
Person {  
  name: 'Ludo',  
  age: undefined,  
  length: undefined,  
  weight: undefined }  
undefined
```



# Person object

```
function Person(name, age, length, weight) {  
  this.name = name;  
  this.age = age;  
  this.length = length;  
  this.weight = weight;  
}
```

```
Person.prototype.toString = function(){  
  return this.name;  
};
```

```
var empty = new Person();  
var private = new Person("Ludo");  
console.log(empty.toString());  
console.log(private.toString());  
console.log("" + private);
```

undefined

Ludo

Ludo

# Person object

```
var eddy = new Person("Eddy", 43, 178, 70);  
for (var prop in eddy) {  
    console.log(prop);  
}
```

name  
age  
length  
weight  
toString

```
var eddy = new Person("Eddy", 43, 178, 70);  
for (var p in eddy) {  
    if (eddy.hasOwnProperty(p)) {  
        var element = eddy[p];  
        console.log(element);  
    }  
}
```

Eddy  
43  
178  
70

# Opdracht

- Maak een auto object aan via constructor. De auto heeft volgende properties:
  - color
  - year
  - cc
  - fuel

Maak nadien een aantal auto objecten aan en print de waardes ervan uit.

# Voorgedefinieerde objecten

Documentatie: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects)

Strings

Arrays

Date

Boolean

Number

...

# String

```
var text = new String("Hello World");  
var sameText = new String("Hello World");  
var sameTextAgain = "Hello World";  
console.log(text == sameText);  
console.log(text === sameText);  
console.log(text == sameTextAgain);  
console.log(sameText === sameTextAgain);  
console.log(text.length);  
console.log(text.toUpperCase());
```

false

false

true

false

11

HELLO WORLD

# Opdracht

**Originele tekst:** Hallo ik ben javascript aan het leren.

**Nieuwe tekst:** H4llo ik ben J4V4 44n het leren.

→ Maak gebruik van de String methodes.

# Array

= reeks van elementen met een index.

1. Via constructor
2. Via array literal

```
var array = new Array(2);  
array[0] = "Hello";  
array[1] = "World";  
var array1 = new Array("Hello", "World");  
var arrayLiteral = ["Hello", "World"];
```

```
Hello10
5
20
1010
undefined
c:\Users\vangike\workspace
html\VDAB_GENK\js\tempCodeRunnerFile.js:11
console.log(arrayLiteral[3].substring(0,2));
^
```

TypeError: arrayLiteral[3].substring is not a function

```
var arrayLiteral = ["Hello", "World", 10, true, 10.10];
console.log(arrayLiteral[0] + 10);
console.log(arrayLiteral[1].length);
console.log(arrayLiteral[2] + 10);
console.log(arrayLiteral[2] + "10");
console.log(arrayLiteral[4].length);
console.log(arrayLiteral[3].substring(0,2));
```



# Itereren over Array

for loop:

```
for (var i = 0; i < array.length; i++) {  
  var element = array[i];  
  console.log(element);  
}
```

for-of loop:

```
for (element of array) {  
  console.log(element);  
}
```

forEach function:

```
array.forEach((element)=>{  
  console.log(element);  
});
```

# Opdracht

Maak een array aan met volgende elementen: “Javascript”, “Html”, “Css”, “Ajax”, “Angular” , “jQuery”.

Maak gebruik van functies Array:

1. Voeg 2 elementen toe en print af.
2. Print de array af in omgekeerde volgorde.

# Opdracht

## **Opdracht 12: Reeksen maken en gebruiken**

- Maak een script dat een reeks bevat met de namen van verschillende kleuren.

0	Zwart
1	Wit
2	Rood
3	Groen
4	Blauw

- Sorteer deze reeks en druk de kleuren op het scherm af.

# Tweedimensionale Array

= reeks van reeksen

1. Via constructor
2. Via array literal

```
var array2d = new Array([1, 3, 4], [2, 5, 6], [9, 10]);
```

```
var array2d = [[1, 3, 4], [2, 5, 6], [9, 10]];
```

# TweeDArray afdrukken

```
var array2D = [[1, 2, 3], [5, 6], [9, 10, 11, 12]];

for (var i = 0; i < array2D.length; i++) {
  for (var j = 0; j < array2D[i].length; j++) {
    process.stdout.write(array2D[i][j] + "\t");
  }
  console.log();
}

array2D.forEach(function (array) {
  array.forEach((el) => {
    process.stdout.write(el + "\t");
  })
  console.log();
});
```

# Opdracht

## **Opdracht 13: Tweedimensionale reeksen gebruiken**

- Maak van de reeks van kleuren een tweedimensionale reeks. In de eerste kolom komt de naam van de kleur en in de tweede kolom de gestandaardiseerde kleurnaam of hexadecimale RGB-waarde:

0	Zwart	Black
1	Wit	White
2	Rood	Red
3	Groen	Green
4	Blauw	Blue
5	Speciaal	#AE56BC

- Druk de inhoud van de tabel af.

# Reeks van eigenschappen

In Javascript kan een object beschouwd worden als een reeks van eigenschappen.

```
var jos = new Person("Jos", 18, 1.88, 68);  
console.log(jos["name"]);  
jos["length"] = 1.55;  
console.log(jos.length);
```

# Argumentenlijst als reeks

= Argumenten die doorgegeven worden aan een functie worden opgeslagen als een reeks (array)

```
function test(){  
    for (var index in arguments){  
        console.log(arguments[index]);  
    }  
}  
test(1, 2 , "Kenneth", "Javascript");
```



# Argumentenlijst als reeks

= Argumenten die doorgegeven worden aan een functie worden opgeslagen als een reeks (array)

```
function test(){  
  for (var index in arguments){  
    console.log(arguments[index]);  
  }  
}  
test(1, 2 , "Kenneth", "Javascript");
```

# Opdracht

## **Opdracht 14: De argumentenlijst gebruiken**

- Maak een functie `average()` die het gemiddelde berekent van alle getallen die als argument worden meegegeven.

# Boolean

Wrapper object.

```
var myBoolean = new Boolean(false);  
if (myBoolean) {  
    console.log("True");  
} else {  
    console.log("False");  
}
```

```
var myBoolean = new Boolean(false);  
if (myBoolean.valueOf()) {  
    console.log("True");  
} else {  
    console.log("False");  
}
```

# Date

```
var today = new Date();  
console.log(today);  
console.log(today.toLocaleString());  
console.log(today.toLocaleDateString());  
console.log(today.toLocaleTimeString());  
console.log(today.getDate());  
console.log(today.getTime());
```

2017-10-09T10:57:37.073Z

2017-10-9 12:57:37

2017-10-9

12:57:37

9

1507546657073

# Opdracht

## **Opdracht 15: Het `Date`-object gebruiken**

- Zoek in de documentatie de beschrijving van het `Date`-object en ga na welke constructors, eigenschappen en methoden dit object heeft.
- Maak een script dat de datum van vandaag afdruckt op het scherm en de gebruiker begroet met "*Goede morgen*", "*Goede middag*" of "*Goede avond*" naargelang het tijdstip.
- Maak een script dat de gebruiker vraagt om zijn geboortedatum. Druk de dag van de week af. Hint: gebruik de constructor waarbij je een datum in de vorm van een string kan meegeven. Voor het weergeven van de dag kan je een `switch` of een reeks als *lookup-table* gebruiken.

## **Opdracht 16: Het `Math`-object gebruiken**

- Zoek de beschrijving van het object `Math` in de *JavaScript*-documentatie.
- Maak een script voor het invullen van je Lottoformulier. Geef telkens 6 verschillende getallen tussen 1 en 42.

# Regular Expressions

= patroon voor zoekacties binnen  
tekenreeks

RegEx Object:

<b>Methode</b>	<b>Omschrijving</b>
<code>exec()</code>	Voert een zoekactie uit op een string en geeft een reeks van strings terug die aan het patroon beantwoorden.
<code>test()</code>	Gaat na of een bepaald patroon aanwezig is in een string. Geeft <code>true</code> of <code>false</code> terug.

```
[ 'Hello Javascript', index: 0, input: 'Hello Javascript' ]  
true  
null  
false
```

```
var myRegex = new RegExp("^Hello [a-zA-Z]+");
```

```
// TEST 1:
```

```
var regexToTest = "Hello Javascript";
```

```
var result = myRegex.exec(regexToTest);
```

```
console.log(result);
```

```
console.log(myRegex.test(regexToTest));
```

```
// TEST 2:
```

```
var regexToTest2 = "Hello 6789";
```

```
var result2 = myRegex.exec(regexToTest2);
```

```
console.log(result2);
```

```
console.log(myRegex.test(regexToTest2));
```

# Regular Expressions

```
var otherRegex= /^D\d{3}/  
console.log(otherRegex.exec("H198"));  
console.log(otherRegex.exec("H1234"));  
console.log(otherRegex.exec("K1111123"));  
console.log(otherRegex.exec("m678"));
```



# Opdracht

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions)

## Opdracht 17: Regular Expressions

- Maak een script dat de gebruiker vraagt om een datum in de vorm DD/MM/YYYY of DD-MM-YYYY of DD.MM.YYYY. Ga met een *regular expression* na of de datum correct is.
- Optioneel: zet de datum om in een `Date`-object indien de syntax correct is. Haal de dag, maand en jaar uit de string door middel van de methode `split()` van een string in combinatie met een *regular expression*.
- Maak een script dat de gebruiker vraagt om zijn rekeningnummer. Ga na of het een geldig nummer is.
- Optioneel: ga ook na of het controlegetal correct is.
- Maak een script dat de gebruiker om zijn E-mail adres vraagt en ga na of dit een correct adres is.
- Maak een script dat de gebruiker vraagt zijn nummerplaat in te geven en controleer of het nummer geldig is.