

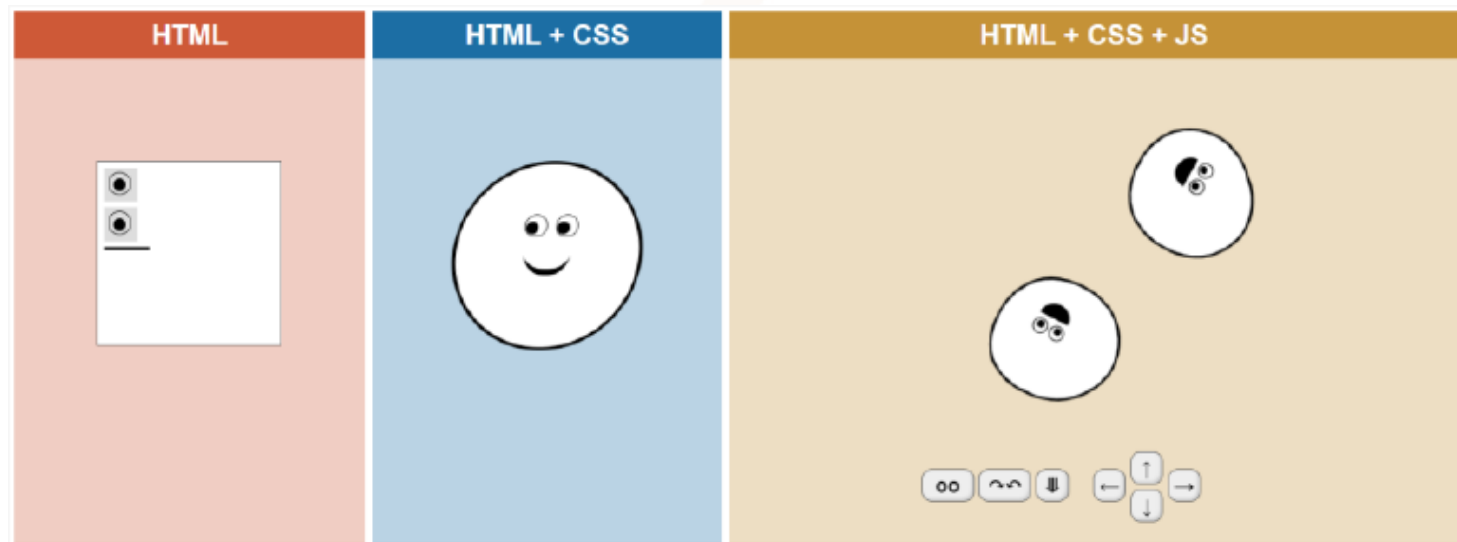
# Javascript Intro

# Inhoud

- **Basisbegrippen:** variabelen, primitieve datatypes.
- Eerste code schrijven
- Instructies en operatoren.
- Functies –Hoisting.
- Arrays.

# Client-Server Architectuur: Client

- **De client** (browser) begrijpt drie talen:
  1. **HTML–Hypertext Mark-up Language**: geeft aan je pagina **structuur** en **betekenis** door middel van headings, tabellen, afbeeldingen, lijsten, ...
  2. **CSS–Cascading Stylesheet**: geeft aan je pagina **opmaak** en **layout** door middel van logische indeling en eigenschappen aan de inhoud zoals lettertype –kleur –...
  3. **ECMAScript(= JavaScript)**: geeft interactie aan je pagina zoals het klikken op knoppen, met je cursor bewegen, vraagt data op en geeft deze in de browser weer, kan voor animatie zorgen (games) ...



# Basisbegrippen

## De bouwstenen

- JavaScript gebruikt variabelen om data bij te houden.
- JavaScript catalogeert de data in een aantal basistypes
  - Tekst
  - Getallen
  - Boolean
- JavaScript is een **loosely typed language**: geen type declaraties !! (in tegenstelling tot Java)

# Basisbegrippen

## De bouwstenen : variabelen

Een variabele is een geheugenlocatie met een unieke naam (identifieer).

- De namen van variabelen in JavaScript zijn:

### Hoofdlettergevoelig

- 1 woord, minimaal 1 karakter lang
  - beginnen met een letter, \$ of een underscore
  - Spaties en andere speciale karakters zijn NIET toegelaten.
  - Het is een goede gewoonte om CamelCase toe te passen (elke nieuw woord in de naam start met een hoofdletter).
- 3 manieren om variabelen te declareren: const, let, var

TIP: Kies altijd **betekenisvollen** namen voor je variabelen!

# Basisbegrippen

## De bouwstenen : variabelen

Een variabele is een geheugenlocatie met een unieke naam (identifieer).

- De namen van variabelen in JavaScript zijn:

### Hoofdlettergevoelig

- 1 woord, minimaal 1 karakter lang
  - beginnen met een letter, \$ of een underscore
  - Spaties en andere speciale karakters zijn NIET toegelaten.
  - Het is een goede gewoonte om CamelCase toe te passen (elke nieuw woord in de naam start met een hoofdletter).
- 3 manieren om variabelen te declareren: const, let, var

TIP: Kies altijd **betekenisvollen** namen voor je variabelen!

## De bouwstenen : variabelen

Declaratie van een variabele waarvan de waarde kan veranderen.

```
let variabeleNaam=initiële waarde;
```

- Naamgeving: zie vorige slide
- Type (numeriek, tekst,...) is afhankelijk van de waarde. (Javascript is loosely typed)
- Waarde kan veranderen!!

# Basisbegrippen - variabelen

ES6

JavaScript is **loosely typed**!

–Je geeft **geen** type op bij declaratie van een variabele!  
Dit betekent niet dat de variabele geen type heeft. Het type is undefined

```
let getal; //type is undefined
```

–Toekennen van een waarde aan een variabe

```
getal = 10; //type is number
```

–Je kan ook een variabele declareren en instantiëren in 1 keer

```
let getal = 10; //type is number
```

–Het type van een variabele kan veranderen en is afhankelijk van de huidige waarde => **impliciete type conversie**

```
let getal; //type is undefined
```

```
getal = 10; //type is number
```

```
getal = 'nu ben ik plots van type string'; //type is string (tekst)
```

(Gebruik ' of " voor strings)



# Basisbegrippen - variabelen

ES6

Enkele voorbeelden:

```
let getal; //declaratie
getal; //>>undefined
getal = 200; //waarde toekennen, getal bevat waarde 200
getal; //>>200 -retourneert de waarde van de variabele getal
getal = 'nu plots een stukje tekst'; //>>"nu plots een stukje tekst"
let naam = 'John'; //declaratie en toekennen van waarde
naam; //>>"John"
let tekst = 'va'+ 'kan'+ 'tie'+ '!!!'; //concatenatie
tekst; //>>"vakantie!!!"
//meerdere declaraties in 1 lijn
let jaar= 2019, maand= 1, dag = 14;
let getal=10; //geeft syntax error Identifier 'getal' has already been declared
```

# Basisbegrippen - variabelen

ES6

## Samenvatting

- JavaScript is **loosely typed**. Er wordt geen type toegekend aan variabelen.
- Maak gebruik van het keyword **let** om een variabele te declareren en kennen een waarde toe aan de variabele en hierdoor wordt het type toegekend.
- Je kan een variabele declareren en een waarde geven, of eerst declareren en dan pas een waarde geven.
- Bij decalaratie wordt het type Undefined toegekend.
- Indien we daarna de waarde 200 toekennen is het type van de variabele Number.
- Daarna kennen we aan deze variabele een stuk tekst toe ("Hello World"), dan is het type van de variabele String geworden
- **Dit betekent niet dat JavaScript geen rekening houdt met het datatype.**
- Het behandelt getallen anders dan strings, anders dan booleans,...

# Basisbegrippen - variabelen

ES6

## De bouwstenen : variabelen

Men kan ook het **var** keyword (ES5) gebruiken voor de declaratie van een variabele.

Nadelen:

- var** kan opnieuw gedeclareerd worden binnen dezelfde scope
  - Er is **function** scope (zie later). Dwz dat de variabele binnen de functie beschikbaar is (ook voor alle blokken in de functie).
  - Als een var niet binnen een functie gedefinieerd, is dan is krijgt deze globale scope (in een web applicatie is dit het window object)
- GEBRUIK NOOIT het "var" KEYWORD (indien mogelijk)

## De bouwstenen : constanten

Een constante krijgt bij de creatie onmiddellijk een VASTE waarde die niet meer kan worden gewijzigd

- Naamgeving: idem als bij variabelen.
- Je geeft **geen** type op bij declaratie van een constante. Het type is afhankelijk van de toegekende waarde. Er moet onmiddellijk een waarde toegekend worden
- Het toekennen van een nieuwe waarde is niet toegestaan!!! Muteren is wel mogelijk(object) –zie later.
- Er is block scope (zie later), zoals bij let.

```
const naam =constante waarde;
```

**Voorbeelden:**

```
const intrest = 0.15;
```

```
const saldo = 2000;
```

```
const intrestPerJaar= saldo * intrest;
```

```
console.log(intrestPerJaar) //>> 300
```

```
intrestPerJaar= 400; //>>throwt een exception: TypeError
```

```
const total; //>>missing initializer in const declaration
```

## De bouwstenen : variabelen

### Best practice

- Gebruik altijd **const** indien je een vaste waarde aan de variabele wil toekennen.
- Gebruik **let** als de variabele gewijzigd mag worden
- Nooit **var** gebruiken, tenzij voor compatibility issues met IE11 (zie <http://kangax.github.io/compat-table/es6/>)

### Oefening

- Gegeven: de prijs van een product is 200 euro, de korting is 10%.
- Print de prijs na korting.

# JavaScript debuggen (chrome browser)

## JavaScript Console : Developer Tools in Browser

- Ga naar de sources tab

The screenshot shows the Chrome Developer Tools interface with the 'Sources' tab selected. The left pane displays the 'File navigator' with a tree view of files including 'eventsEnBoxes.js' and 'functies.js'. The main pane shows the code for 'functies.js', which includes functions for logging, alerting, confirming, and prompting. The right pane shows the 'Console' tab with a search bar and a list of messages. Annotations with arrows point to various parts of the interface: 'File navigator' points to the left pane; 'Elk script andere tab' points to the file list; 'Pause, resume, step through code' points to the toolbar; 'Display console' points to the console tab; 'Docking options' points to the window control buttons; and 'Settings' points to the gear icon.

File navigator

Elk script andere tab

Pause, resume, step through code

Display console

Docking options

Settings

```
1 function hello() {  
2   console.log("hello");  
3 }  
4  
5 var show_alert = function () {  
6   alert("I am an alert box!");  
7 }  
8  
9 var show_confirm = function () {  
10   var antwoord = confirm("Press a button");  
11   if (antwoord == true)  
12     alert("You pressed OK!");  
13   else  
14     alert("You pressed Cancel!");  
15 }  
16  
17 var show_prompt = function () {  
18   var name = prompt("Please enter your name", "Harry");  
19 }
```

Line 25, Column 1

# Referenties

<https://www.learn-js.org/>

[https://www.w3schools.com/js/js\\_let.asp](https://www.w3schools.com/js/js_let.asp)

<https://www.javascript.com/try>