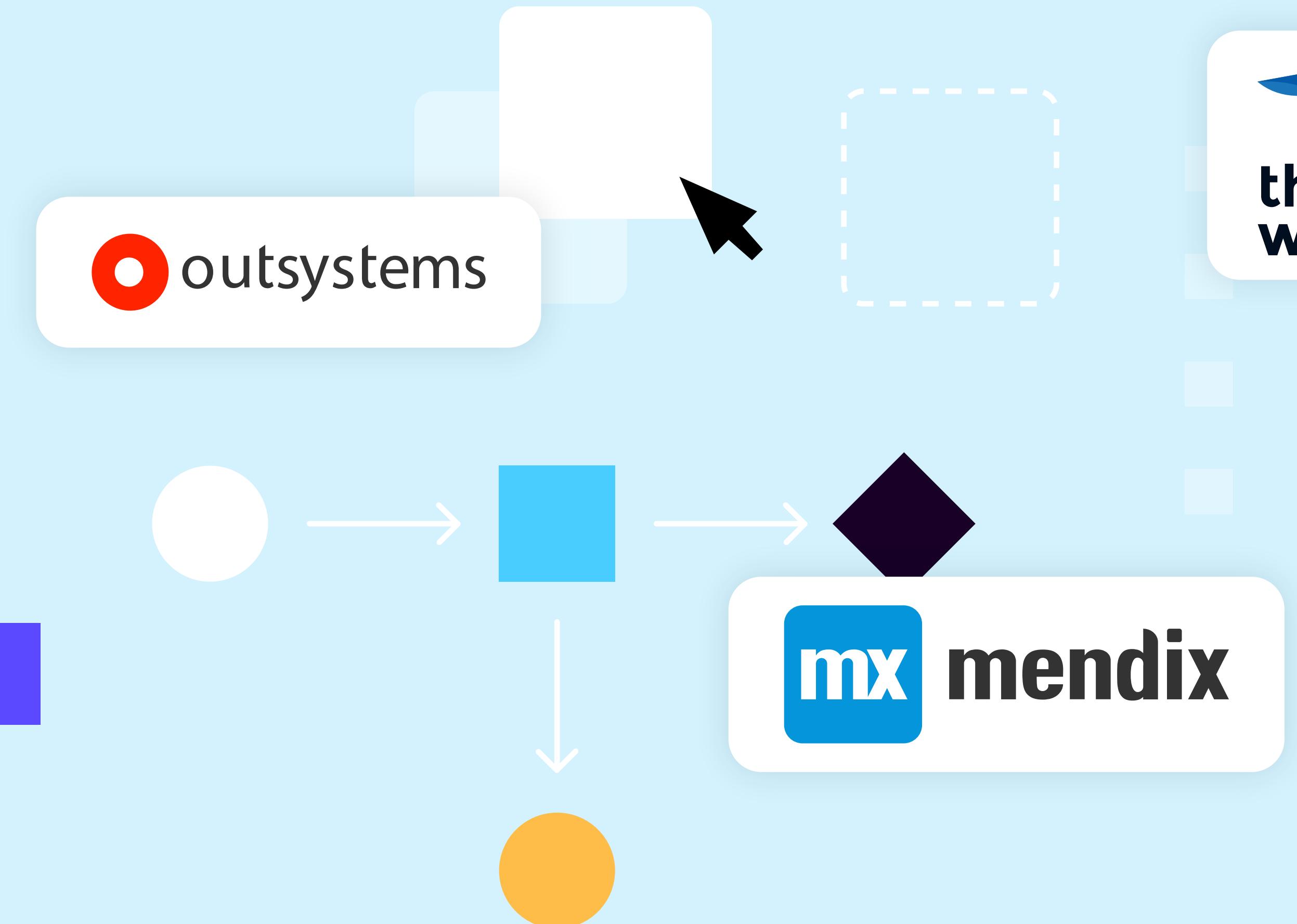




# Low code traineeship

trainingsdag 1: HTML, CSS, JS



# Inhoud les 1

## Intro

Inloop en voorstellen  
Wat gaan we doen?  
Visual Studio Code  
Wat is HTML, CSS en JavaScript?

## HTML

HTML syntax  
HTML tags  
HTML basis structuur website  
HTML a tag (maken linkjes)  
HTML lists  
HTML formulieren

## CSS

CSS syntax en voorbeelden  
CSS stylesheets & inline  
CSS cascading  
CSS Values en Units  
CSS Boxmodel  
CSS display  
CSS Flexbox  
CSS Position  
CSS fonts  
CSS pseudo classes  
CSS styling (box shadow, transform, transition)

## JavaScript

JS console.log, alert & prompt  
JS variabelen  
JS Strings  
JS Numbers & Math  
JS Conditional Statements  
JS Functions  
JS Arrays  
JS Loops  
JS Local Storage

# Visual Studio Code & Replit

## Weapon of choice Visual Studio Code

Voor het maken van websites in HTML CSS en JS heb je een goede code editor nodig. In deze cursus gaan we gebruik maken van Visual Studio Code.

Natuurlijk zijn er ook goede andere opties zoals: Atom, Notepad++, Adobe Dreamweaver en meer.

Je kan zelfs in je gewone notepad op je laptop coderen.



Visual Studio Code

## Replit

Een handig programma om code online te delen en om samen te werken tijdens het studeren.

### Linkjes

**Downloaden Visual Studio Code**  
<https://code.visualstudio.com/>

**Online code editor Replit**  
<https://replit.com/>

# Front End Webdevelopment

## Presentatielaag

In deze cursus gaan wij ons bezig houden met de presentatielaag van een applicatie of website. Dit is het gedeelte dat de bezoeker kan zien,

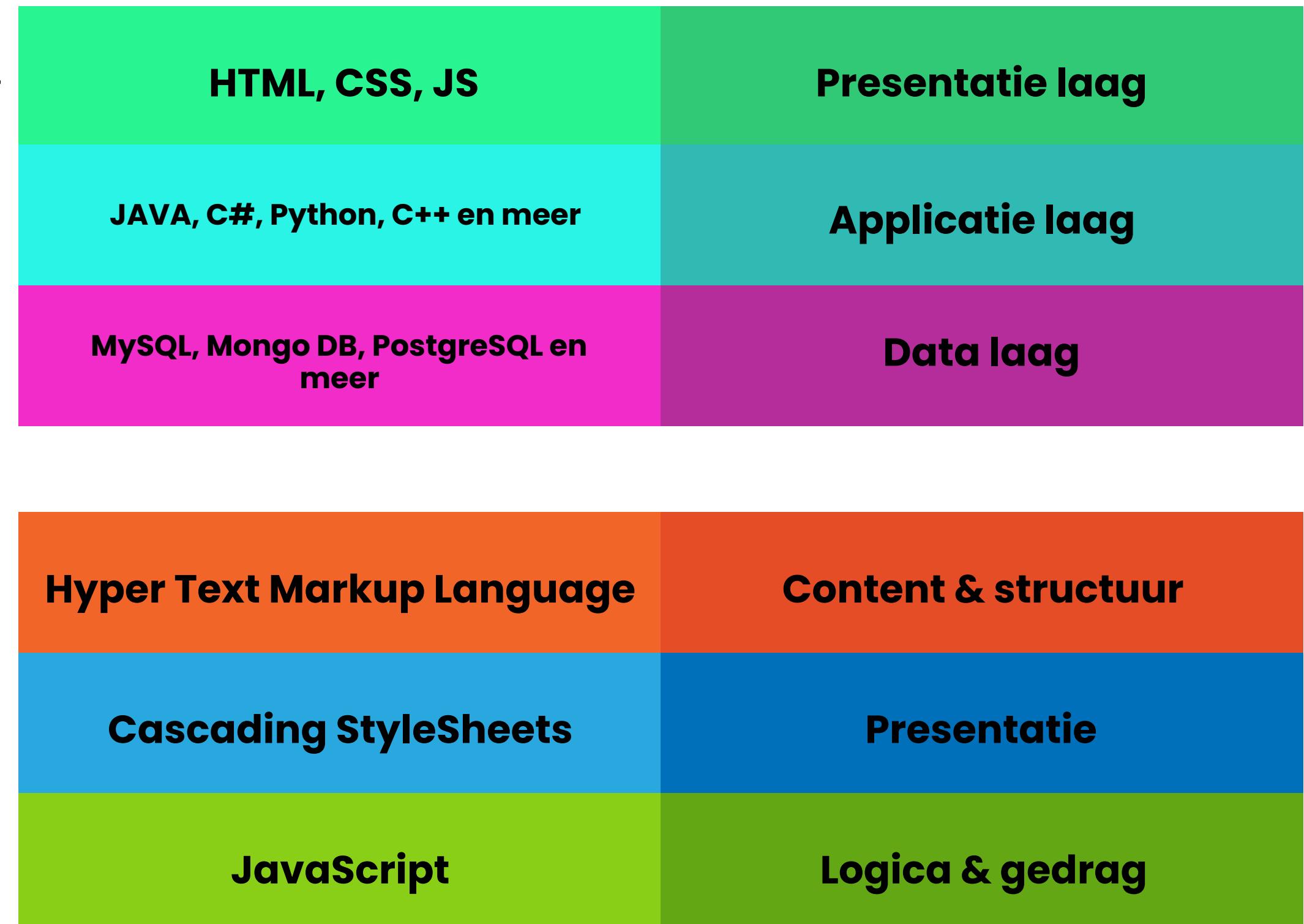
## De drie lagen van webdevelopment

### De presentatielaag bestaat uit:

HTML, hierin komt de content en structuur van de website. Dit zijn bij voorbeeld de headers, de paragrafen, de plaatjes en meer. Waar en in welke volgorde komt dit op de website? etc.

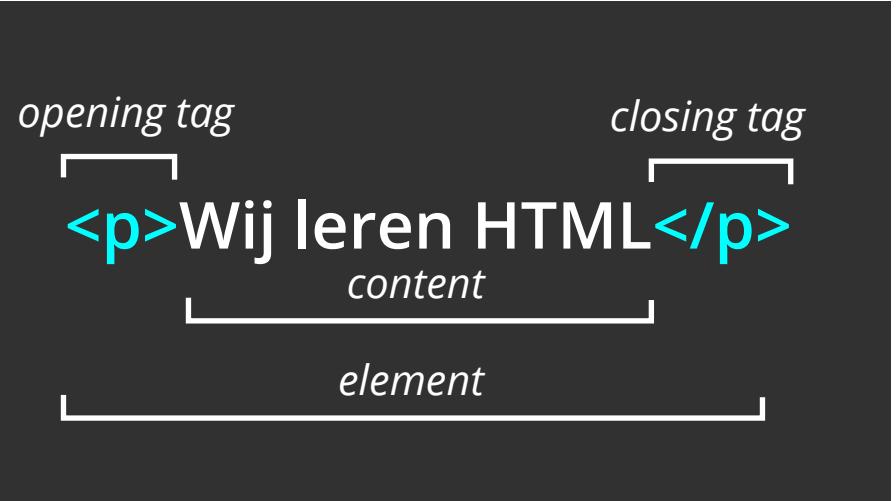
CSS, hierin wordt de presentatie van de content bepaald. Welke kleur heeft de tekst? Hoe breed wordt een blok? Heeft de achtergrond een kleur of een plaatje? etc.

JS, hierin wordt de logica en het gedrag bepaald. Wat gebeurd er als er op een button wordt geklikt? Je kan vanuit JS ook communiceren met backend talen (applicatielaag) of rechtstreeks met een database (data laag)



# HTML basics

## Syntax



HTML bestaat uit elementen. Een tag vertelt de browser wat voor soort element het is. Een element heeft een openings tag en vaak een closing tag. Hiertussen zit de content, dit kan tekst zijn, maar ook andere elementen. Een element kan attributen bevatten zoals een class of een id naam.



## Tags

Met een tag refereer je naar een bepaald type element. Bijvoorbeeld de `<header>` tag geeft aan dat de inhoud daarvan onderdeel van de kop van de website is. Een `<h1>` geeft aan dat dit de belangrijkste titel van de pagina is. Hier naast een lijstje van de belangrijkste tags:

<html></html>	<ul></ul>
<body></body>	<ol></ol>
<main></main>	<li></li>
<header></header>	<table></table>
<footer></footer>	<a></a>
<article></article>	<tr></tr>
<nav></nav>	<th></th>
<h1></h1> (tm h6)	<td></td>
<p></p>	<img></img>
<div></div>	<button></button>
<b></b>	

## Linkjes

### HTML tags

<https://www.w3schools.com/tags/default.asp>  
<https://developer.mozilla.org/nl/docs/Web/HTML/Element>

# HTML basics

## Nesting & indenting

Elementen kunnen andere elementen bevatten. Zo kan een 'parent' meerdere 'child' elementen bevatten.  
Om de code leesbaar te houden is indenting erg handig.

Gebruik tab en shift tab om een kolom of meerder kolommen naar voren of naar achter te schuiven

shortcut: selecteren en ctrl k daarna ctrl f

## Structuur

Een HTML bestand heeft een vaste structuur.  
Het bestand zal een **extensie .html** moeten hebben zodat de browser weet wat voor bestand het is: index.html

**!DOCTYPE html:** browser laten weten dat het een html bestand is.  
**html:** opening en closing tags, laat weten dat hiertussen html wordt geplaatst.  
**head:** bevat informatie over de pagina en links naar css en js  
**body:** hierin komt de content

shortcut: selecteren en type '!' en druk op 'tab'

```
<div class="element-name">
    <h1>Ik zit in de div</h1>
</div>
```

```
<!DOCTYPE html>
<html lang="nl">
    <head>
        <title>Leren Front End</title>
    </head>
    <body>
    </body>
</html>
```

# HTML basics

## Body (content van de website)

Hierin komt de content van de website. De header, navigatie, de hoofdcontent en de footer van de website zitten allemaal in de body.

### Header:

introduction  
logo  
navigation

```
<header>
<nav></nav>
<h1>Wij leren HTML</h1>
<p>Ik ga een website bouwen </p>
</header>
```

### Main:

sections  
page content (articles, blogs, reports, info).

```
<main>
<section>
<article>
<h2>Koptekst</h2>
<p>paragraaf</p>
</article>
</section>
</main>
```

### Footer:

contact information  
extra navigation

```
<footer>
<p>© 2021 Maurice</p>
</footer>
```

# HTML de 'a' tag: linkjes

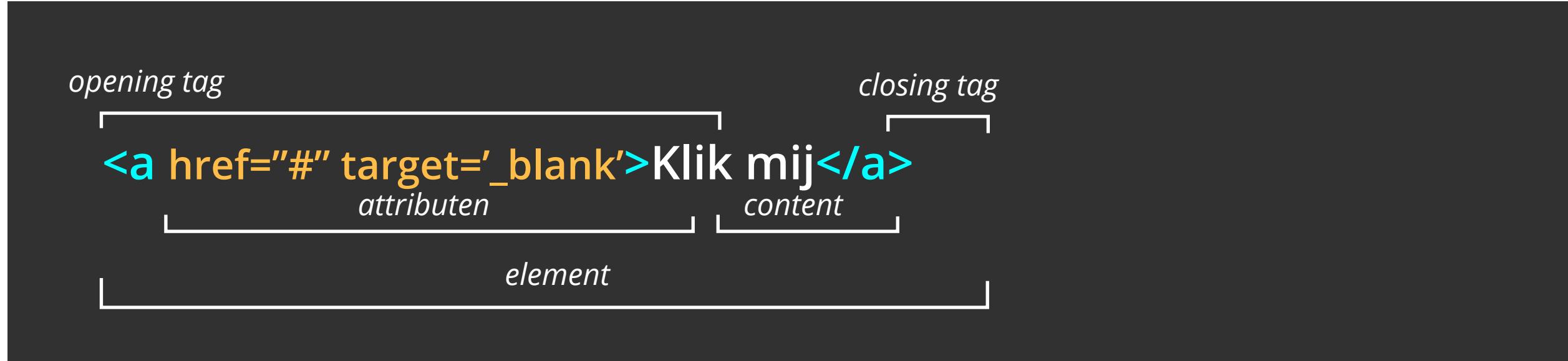
## Linken naar interne & externe pagina's en naar id's

**Interne links:** je linkt vanaf een pagina van je website naar een andere pagina binnen dezelfde website. bv: vanaf www.jouwwebsite.nl naar www.jouwwebsite.nl/over-ons

**externe links:** je linkt vanaf een pagina van je website naar een pagina van een andere website. bv: vanaf www.jouwwebsite.nl naar www.google.nl

**linken naar id:** je linkt vanaf een pagina van je website naar een sectie op dezelfde pagina. bv: vanaf www.jouwwebsite.nl/over-ons naar www.jouwwebsite.nl/over-ons#het-team

### Element link: de 'a' tag



### Uitzoeken

- linken naar een absolute URL en een relatieve URL
- link openen in zelfde tabblad en nieuw tabblad
- linken naar een id op dezelfde pagina

# HTML lijstjes

## Ordered & Unordered lists

Met html lists kan je gerelateerde items in een lijstje zetten. Dit kan een lijst zijn met getallen (ordered lists) of met bullets (unordered lists)

```
<ul>
  <li>Item</li>
  <li>Item</li>
  <li>Item</li>
  <li>Item</li>
</ul>
```

```
<ol>
  <li>John</li>
  <li>Jack</li>
  <li>Kate</li>
  <li>Hurley</li>
</ol>
```

Unordered HTML list:

- Item
- Item
- Item
- Item

Ordered HTML list:

1. John
2. Jack
3. Kate
4. Hurley

## Linkjes

### HTML lists

[https://www.w3schools.com/html/html\\_lists.asp](https://www.w3schools.com/html/html_lists.asp)

# HTML images

## Images

Afbeeldingen kunnen het design van een website verbeteren, maar ook ondersteuning bieden in het gebruikersgemak van de website.

### img tag en attributen

```

```

Een afbeelding maak je met een img tag.

De tag wil weten waar de afbeelding gevonden kan worden. De link geef je aan met het attribuut **src**. Dit kan een relatieve of absolute link zijn naar de afbeelding.

Een alt attribuut wordt gebruikt als alternatieve tekst voor de afbeelding, indien de gebruiker de afbeelding niet kan zien door een trage verbinding, een error in de src of als de gebruiker een screen reader gebruikt.

### images & CSS object fit

Met object fit bepaal je hoe een element als een afbeelding wordt geschaald ten opzichte van zijn container. Hiermee kan je bevoordeeld angeven dat een afbeelding de container vult. Of dat de afbeelding zijn maximale grootte aanneemt etc. Zie link voor voorbeelden.



### Linkjes

#### HTML img

[https://www.w3schools.com/tags/tag\\_img.asp](https://www.w3schools.com/tags/tag_img.asp)

#### CSS object fit

<https://developer.mozilla.org/en-US/docs/Web/CSS/object-fit>

# CSS basics

## Syntax

Een css element bestaat uit een **selector**, **property + value**

```
selector {  
    property: value;  
}  
  
.classname {  
    max-width: 720px;  
}
```

## Inline, internal stylesheet, external stylesheet

- **inline**. in de html in de opening tags
- **internal stylesheet**. in de head van de pagina in <style></style>
- **external stylesheet**. in een css bestand.

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

## CSS selectors

- **by html tag**. selecteer elementen aan de hand van de tag naam bv h1, header, main
- **by classname**. selecteer elementen met dezelfde klassennaam. plaats een punt voor de naam bv: .classname {}
- **by id**. selecteer een element met een bepaalde id naam. plaats een # voor de naam.  
bv: #idname

# CSS basics

## Cascade

Een element krijgt zijn styling van meerdere stylesheets en kan ook inline gestyled worden. Een h1 kan in een stylesheet zijn kleur krijgen via zijn tag, classname en id krijgen. En kan ook nog inline gestyled worden. Daarnaast heeft je computer een stylesheet en je browser ook.

## Wie heeft er voorrang?

- 1 tag (minst)
- 2 class (combo class is specifieker dan de hoofclass)
- 3 id
- 4 inline

## Inheritance

Zitten er meerdere elementen in bv een div. En wil je alle elementen styling meegeven. Dan kan je de div bv een text kleur geven, waardoor alle tekst die kleur krijgt tenzij overschreven.

# CSS values & units

## Values & units

**values:** elke css property heeft een waarde (value) zodat de browser weet hoe iets gestyled moet worden. Om bijvoorbeeld tekst een kleur te geven of een header een achtergrondskleur te geven gebruik je een **color value**. Om aan te geven hoe breed een div moet zijn of hoe groot tekst moet zijn gebruik je een **length value**

**color units:** een value kan in verschillende units worden uitgedrukt. Zo kan color worden uitgedrukt met een **keyword** (blue, cyan of transparant), met een **rgb()** waarde (#000000) of met een **hsl()** & **hsla()** waarde (hsla(120,100%,50%,0.3))

**length units:** de breedte, hoogte etc van een value kan op verschillende manieren worden uitgedrukt. Populair zijn px, %, vw, vh, em en rem.

```
selector {  
    property: value;  
}  
  
.mydiv {  
    width: 60%;  
    min-width: 400px;  
    background-color: black;  
    color: #ffffff;  
}
```

In dit voorbeeld zijn width, min-width, background-color en color properties. De color is misschien verwarrend maar dit is geen value maar deze property heeft gewoon de naam color. Alles achter de dubbele punt zijn de values uitgedrukt in units.

- |                         |   |
|-------------------------|---|
| width: 60%              | - de 60% is een length value uitgedrukt in de unit % (percentage)         |
| min-width: 400px        | - de 400 px is een length value uitgedrukt in px (pixels)                 |
| background-color: black | - de black is een color value uitgedrukt met een keyword                  |
| color: #ffffff          | - de #ffffff is een color value uitgedrukt in een rgb hexadecimale waarde |

## Linkjes

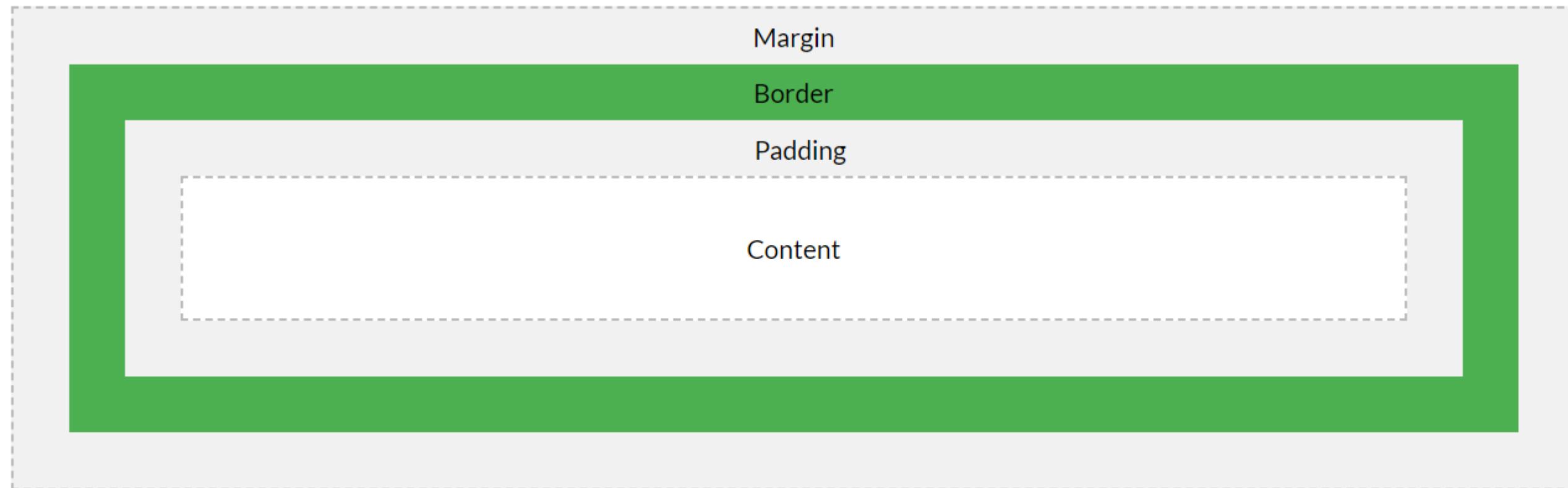
### Values & units

[https://developer.mozilla.org/en-US/docs/Learn/CSS/Buidling\\_blocks/Values\\_and\\_units](https://developer.mozilla.org/en-US/docs/Learn/CSS/Buidling_blocks/Values_and_units)

# CSS box model

## CSS box model

Een website bestaat eigenlijk uit allemaal blokken in blokken. Deze blokken 'boxes' bestaan vervolgens zelf uit content, padding, border en een margin



- **content:** de inhoud van de box, dit kan tekst zijn of een afbeelding, maar ook andere boxes.
- **padding:** de ruimte in de box om de content heen. Geef je het element een achtergrond kleur? dan veranderd het gedeelte van de padding ook.
- **border:** de rand om de content en de padding. Deze rand kan je een solide kleur geven, maar ook een dotted line. Ook kan je hiermee een element een afgerond erand geven.
- **margin:** de ruimte tussen dit element (box) en het element ernaast, boven of onder.

## Online voorbeelden en try it yourself tools

Op deze pagina staat een link naar w3schools, hier vind je over verschillende HTML, CSS, CSS en meer onderwerpen informatie met try it yourself tools. Hierin krijg je een stukje voorbeeld code en de daarbij horende uitkomst te zien. Het leuke aan deze tools is dat je de code kan aanpassen en de resultaten kan bekijken

### Linkjes

#### Box model

[https://www.w3schools.com/css/css\\_boxmodel.asp](https://www.w3schools.com/css/css_boxmodel.asp)

Margin

Border

Padding

Content

# CSS display

## CSS box model display

**block:** displayed een element als een blok element. Elk blok start op een nieuwe regel.

**inline-block:** displayed een element als een inline element, maar je kan hoogtes en breedte waarden toepassen

**inline:** displayed een element als een inline element, hoogte en breedte waarden hebben geen effect.

**flex:** displayed een element als een block element. Het element wordt een flex container en de elementen in dit blok worden de flex children.

**grid:** displayed een element als een block element. Het element wordt een grid container en de elementen in dit block worden grid children.

**none:** Het element wordt verwijderd en is daardoor niet zichtbaar.

voor een volledig overzicht van alle display values bekijk de link van w3schools.

### Linkjes

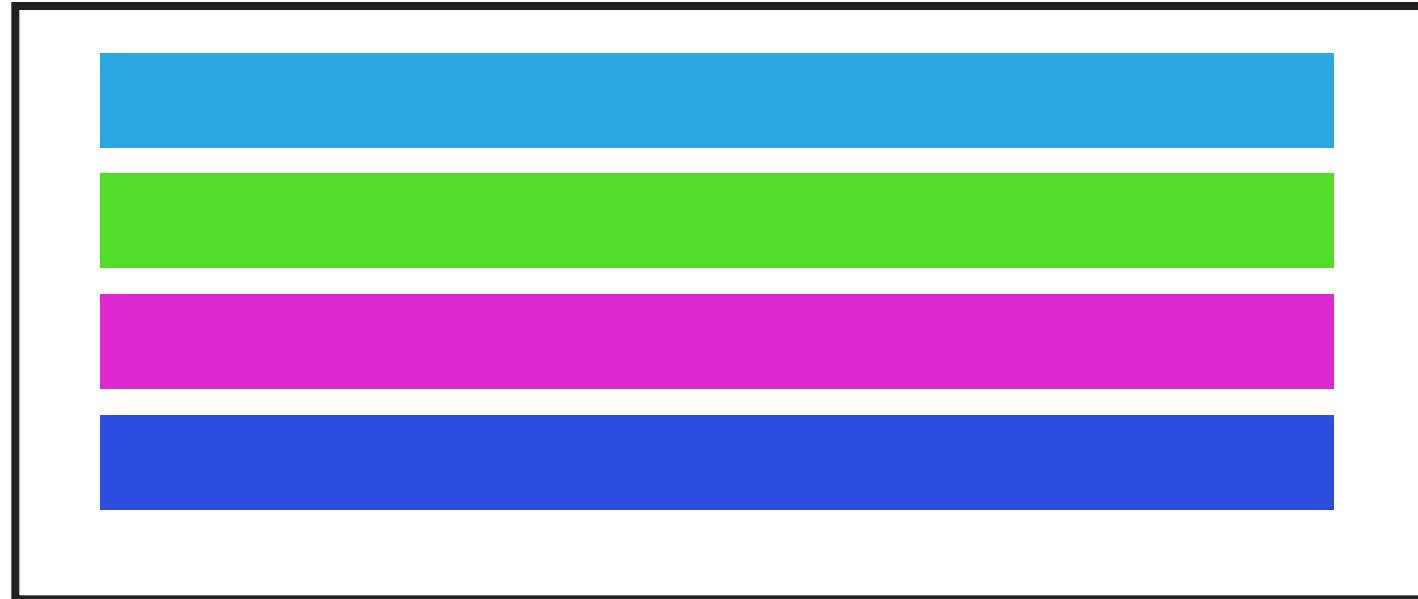
#### Box model display values

[https://www.w3schools.com/cssref/pr\\_class\\_display.asp](https://www.w3schools.com/cssref/pr_class_display.asp)

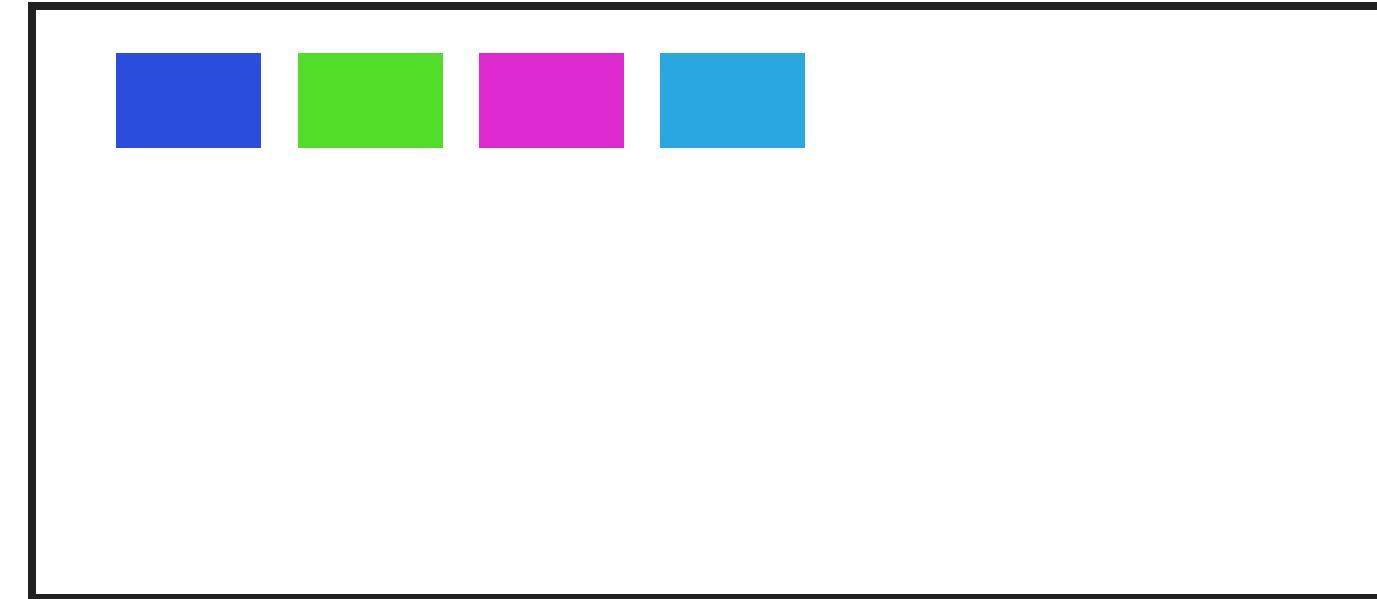
# CSS flex

## Wat is flexbox?

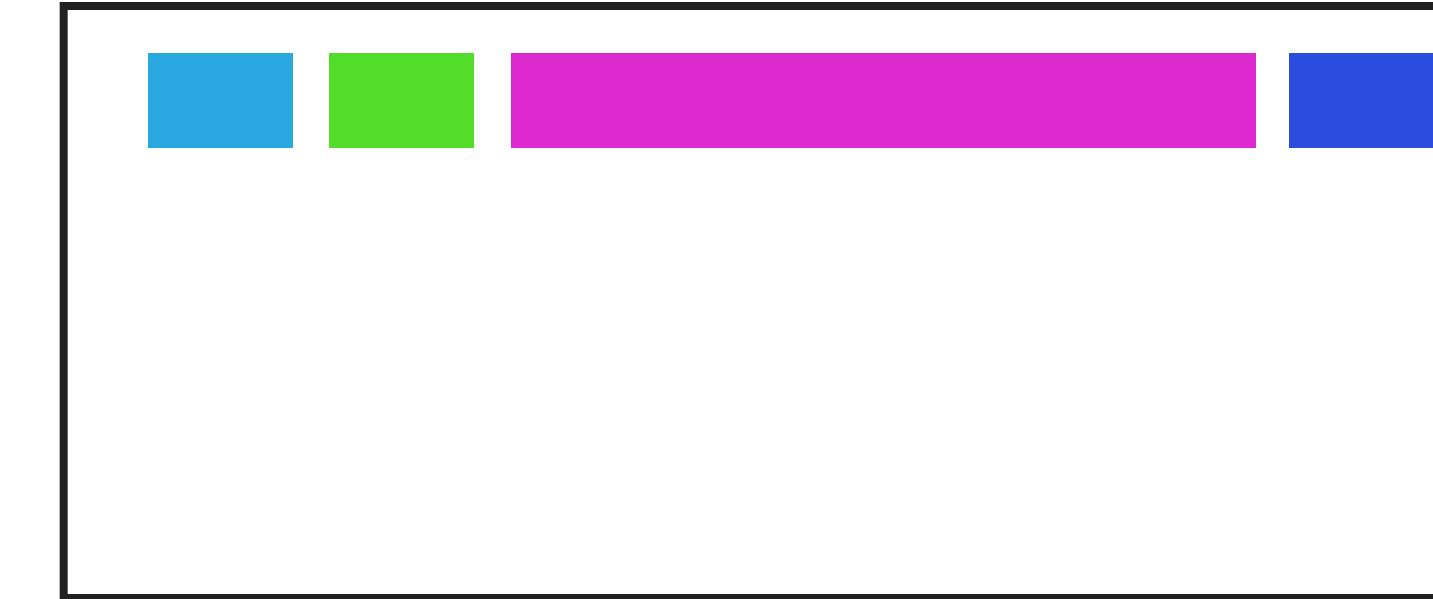
Flexbox is een geweldige layout display setting, waar je makkelijk responsive websites mee maakt.



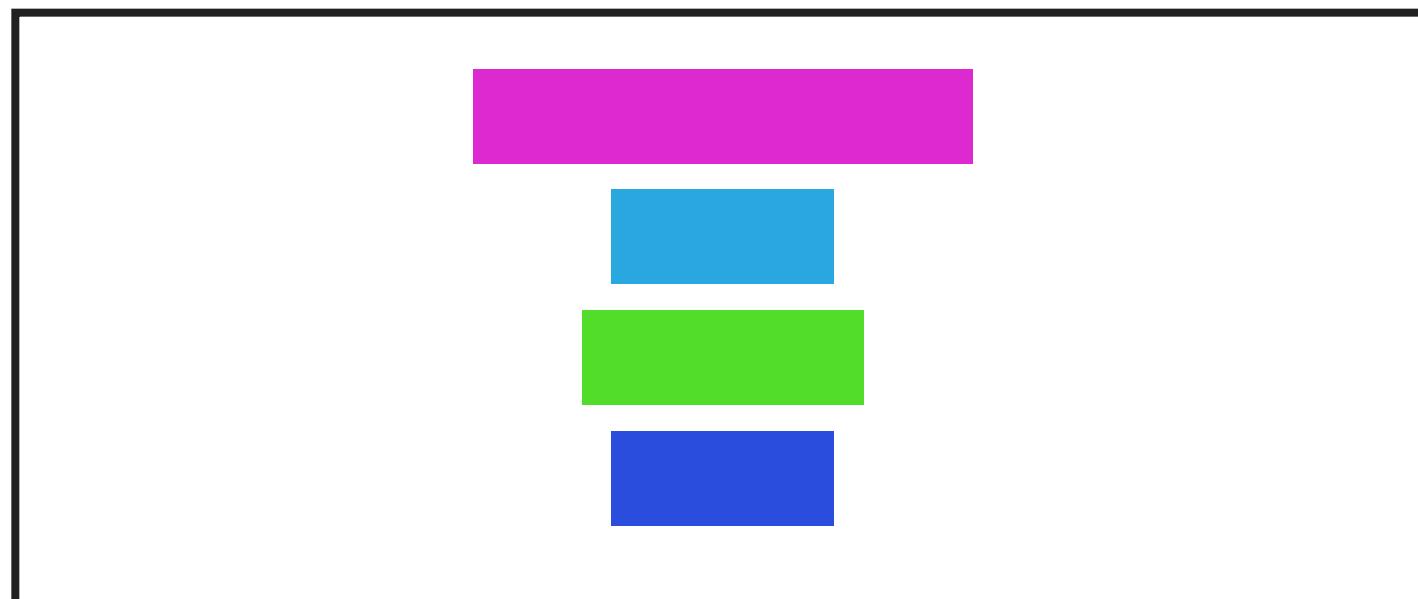
standaard display block



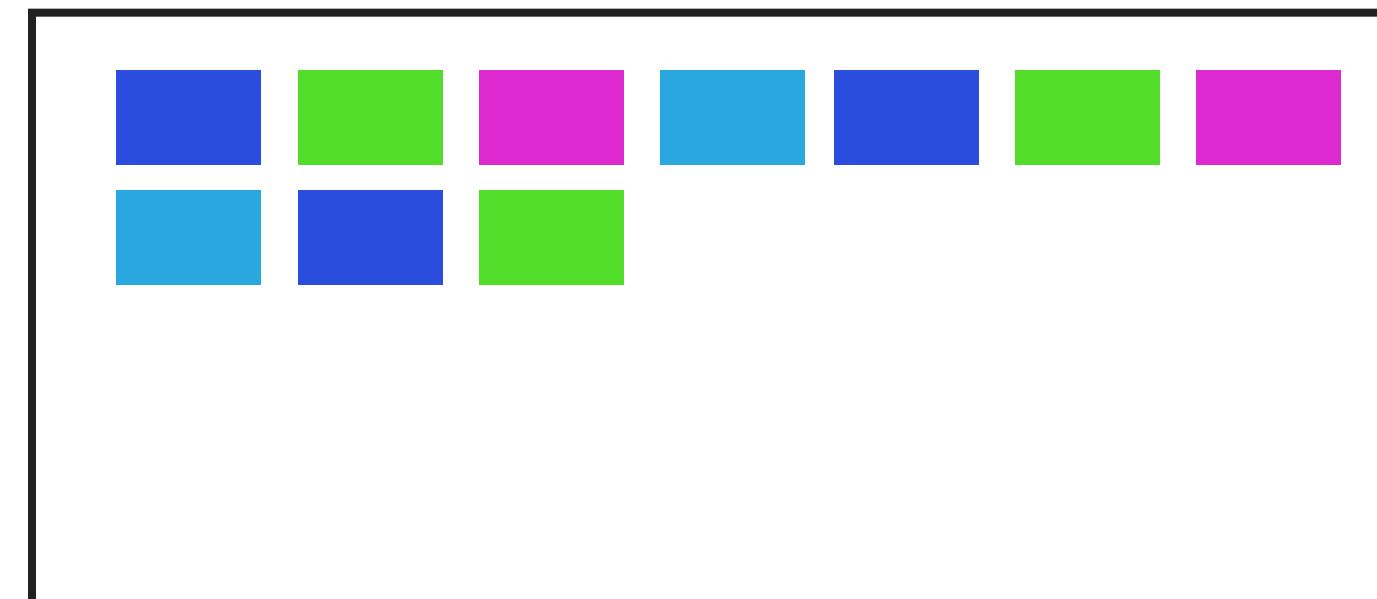
display flex



display flex variant



display flex variant



display flex variant

# CSS flex

## Wat is flexbox?

flex container ↗

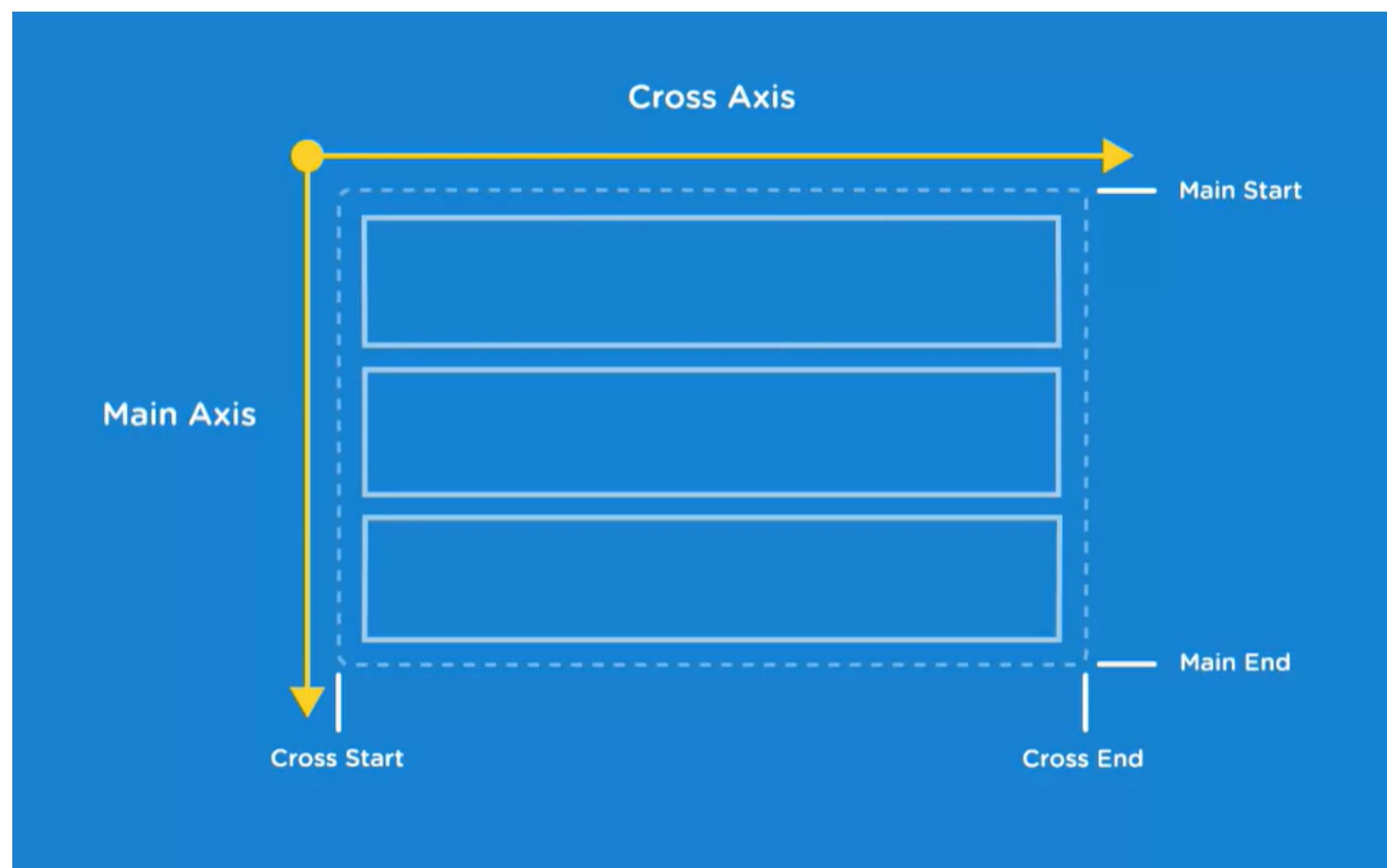


flex items ↗

### Linkjes

#### Lezen informatie flexbox

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>  
[https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)



# CSS position

## Wat is position?

CSS position is een basis element van CSS. Het bepaalt hoe en waar elementen getoond worden op het scherm. Ook kan de position van een element gevolgen hebben voor het positioneren van zijn child elements. Position mogelijkheden: static, relative, absolute, fixed, sticky

position: relative;

### Static

Alle elementen hebben by default de waarde position: static; Een static element wordt gerenderd in de volgorde waarin ze voorkomen in de HTML. Ook kan je de positie niet verschuiven met bijvoorbeeld left: 200px;

### Relative

Elementen met position: relative; worden gepositioneerd ten opzichte van zijn static positie. Dit doe je door top, left, right, bottom toe te kennen aan een element in de css.

position: relative;  
left: -100px;

### Linkjes

#### Lezen informatie position

[https://www.w3schools.com/css/css\\_positioning.asp](https://www.w3schools.com/css/css_positioning.asp)

# CSS position

## Absolute

Absoluut positioneren werkt anders dan relatief positioneren. Het is namelijk tenopzichte van zijn parent element. Maar hierbij moet je rekening houden dat het tenopzichte van het eerste parent element met een andere position dan static. Als je een element in de rechteronderhoek van zijn directe parent wil positioneren en je opeens ziet dat het element helemaal rechtsonder de pagina komt, dan zal een logische verklaring zijn dat zijn parent position static heeft. Zodra je dit veranderd naar position relative is het probleem opgelost.

```
position: absolute;  
bottom: 0;  
right: 0;
```

## Fixed

Een fixed element wordt geïnitialiseerd relatief aan de viewport. Dit element is dus altijd op dezelfde plek als je scrolled.

```
position: fixed;  
top: 0;  
left: 0;  
right: 0;
```

## Sticky

Een sticky is een beetje anders dan de andere mogelijkheden. Een sticky element switches tussen relative en fixed, afhankelijk van de scroll positie. Het element is relatief totdat hij voldoet aan een waarde en plakt dan vast op een positie.

```
position: sticky;  
top: 10px;
```

# CSS fonts

## Font sizes & weights

Een stukje tekst kan op veel manier gestyled worden. Je hebt de grootte van de font (font-size) de afstand tussen regels (line-height) de dikte van de tekst (font-weight) en nog veel meer!

### b tag i tag en span tag

Wil je de nadruk op een stuk leggen door het dikker te maken of juist schuin. Geef dit dan een b of i tag **< b > woord < /b >**

Wil je een stukje tekst anders dan de rest van een header of paragraaf, dan is de span tag super handig. Wrap een stukje tekst tussen de span tag, geef het een class of id en style dit stukje tekst hoe jij dat wil!

Er zijn nog meer tags te vinden op w3schools

### Uitzoeken

- verander de grootte van een stuk tekst en de regelafstand
- maak 1 woord dikker en een andere kleur met span
- verander het lettertype
- leg de nadruk op 1 woord met een b tag
- maak een link van 1 woord en style deze

# CSS Pseudo Classes

## Pseudo classes

Een pseudo class definieert de staat van een element. Dit kan bijvoorbeeld een hover zijn wanneer de gebruiker met zijn muis over een element gaat. Het kan een visited zijn als de link al eerder is bekeken door de gebruiker. etc

[This is a link](#)

[This is a link](#)

[This is a link](#)

[This is a link](#)

## Linkjes

**Lezen informatie pseudo classes**

[https://www.w3schools.com/css/css\\_pseudo\\_classes.asp](https://www.w3schools.com/css/css_pseudo_classes.asp)

```
/* unvisited link */  
a:link {  
    color: #FF0000;  
}  
  
/* visited link */  
a:visited {  
    color: #00FF00;  
}  
  
/* mouse over link */  
a:hover {  
    color: #FF00FF;  
}  
  
/* selected link */  
a:active {  
    color: #0000FF;  
}
```

# CSS Making it pretty

## Boxshadow

Een boxshadow kan ervoor zorgen dat je website er minder vlak uit ziet. Ook wordt het veel gebruikt in de populaire 'material design' van google

```
#example1 {  
    border: 1px solid;  
    padding: 10px;  
    box-shadow: 5px 10px 8px 10px #888888;  
}  
  
        ↑      ↑      ↑      ↑  
horizontale offset  blurr  spread   kleur  
        ↑  
verticale offset
```

## Transition

css transition heeft de mogelijkheid om een verandering van een property value smooth over te laten gaan.

## Transform

Met css transform kan je een element onder andere: schalen, roteren, verplaatsen aan de x en y as,.

```
.changeme {  
    transform: rotate(20deg);  
}
```

## Linkjes

### Informatie boxshadow

[https://www.w3schools.com/css/css\\_pseudo\\_classes.asp](https://www.w3schools.com/css/css_pseudo_classes.asp)

### Informatie transform

<https://developer.mozilla.org/en-US/docs/Web/CSS/transform>

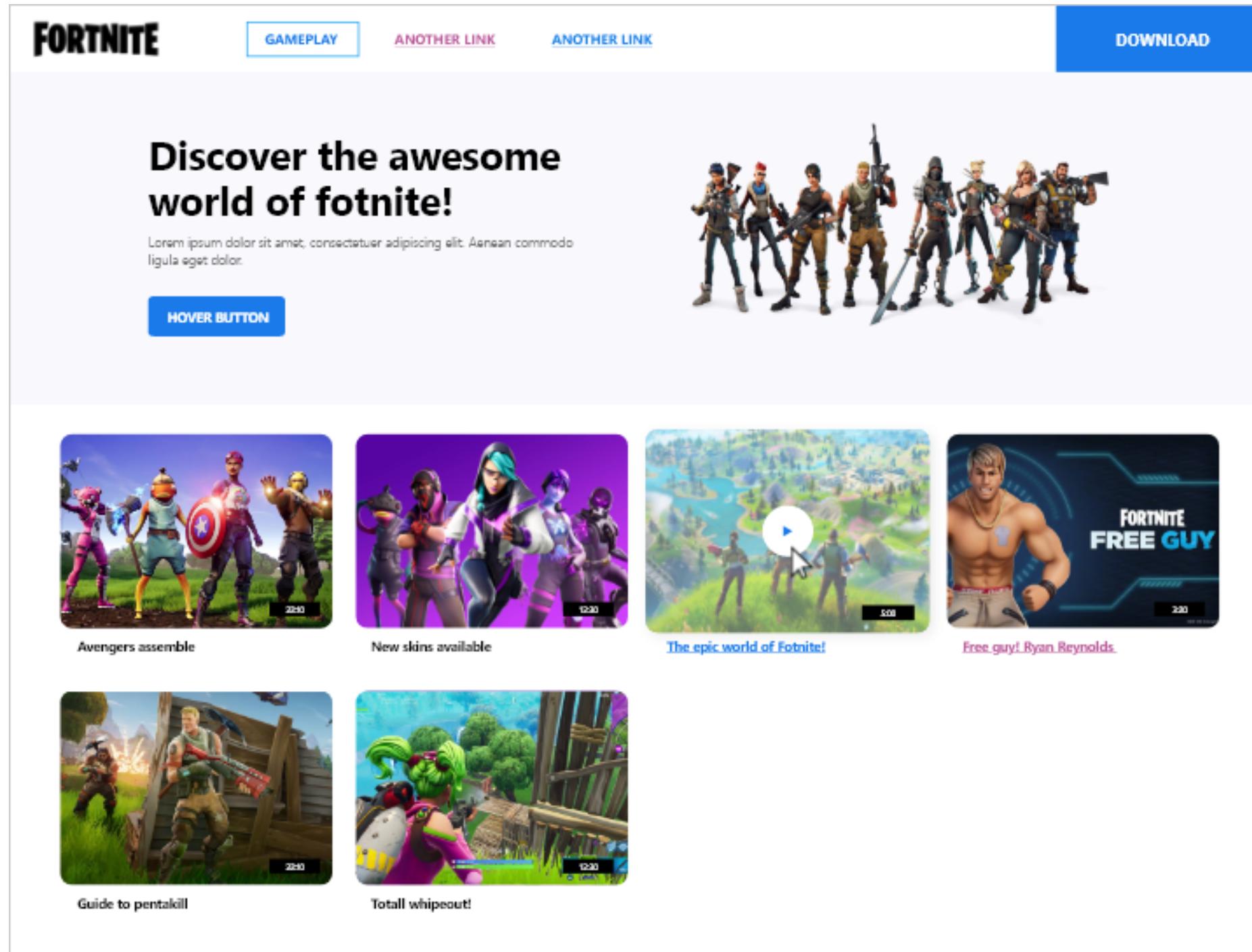
### Informatie transition

[https://www.w3schools.com/css/css3\\_transitions.asp](https://www.w3schools.com/css/css3_transitions.asp)

# HTML CSS opdracht

In deze opdracht combineer je de geleerde HTML en CSS technieken. Bouw het ontwerp na.

<https://xd.adobe.com/view/96f5fb9a-b106-4d2c-a7eb-ee33007def04-c3a6/>



## Kleuren

- #000000
- #707070
- #1A7AEA
- #20A6F5
- #BB5C9B
- #F9F8FD

## Afbeeldingen

<https://www.dropbox.com/s/c6n1etygdd2gdue/img-opdracht-1.zip?dl=1>

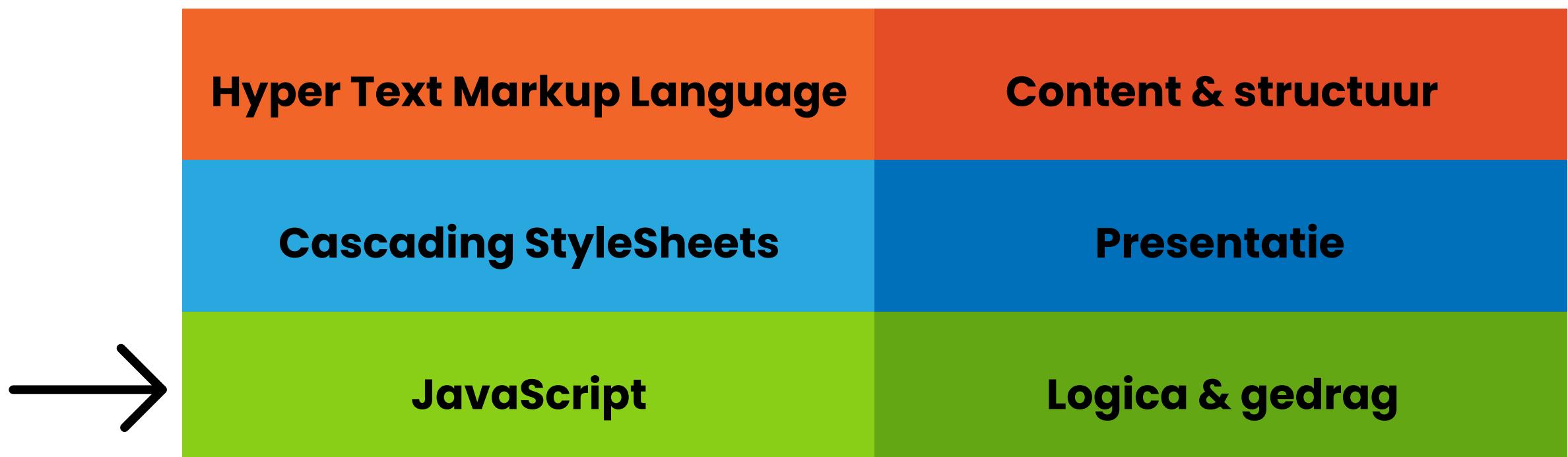
# JavaScript basics

## Wat is Javascript?

Veel gebruikte programmeertaal. Wij gaan het gebruiken om webpagina's interactiever te maken.

Javascript kan HTML en CSS updaten en veranderen.

In javascript kan je functies schrijven, variabelen maken, if statements schrijven, arrays maken en daaroverheen loopen.



## Client side vs server side

Javascript is net als HTML en CSS een client side script-taal. De code wordt gedownload naar de browser van de bezoeker en daar uitgevoerd.

Een server side script-taal zoals Java, C# en anderen draaien op een andere locatie dan de browser. De front end kan wel deze server aanroepen om gebruik te maken van deze fucnties met bijvoorbeeld endpoints. Maar dit valt buiten de scope van de cursus.

# JavaScript basics

## Chrome dev tools voor javascript

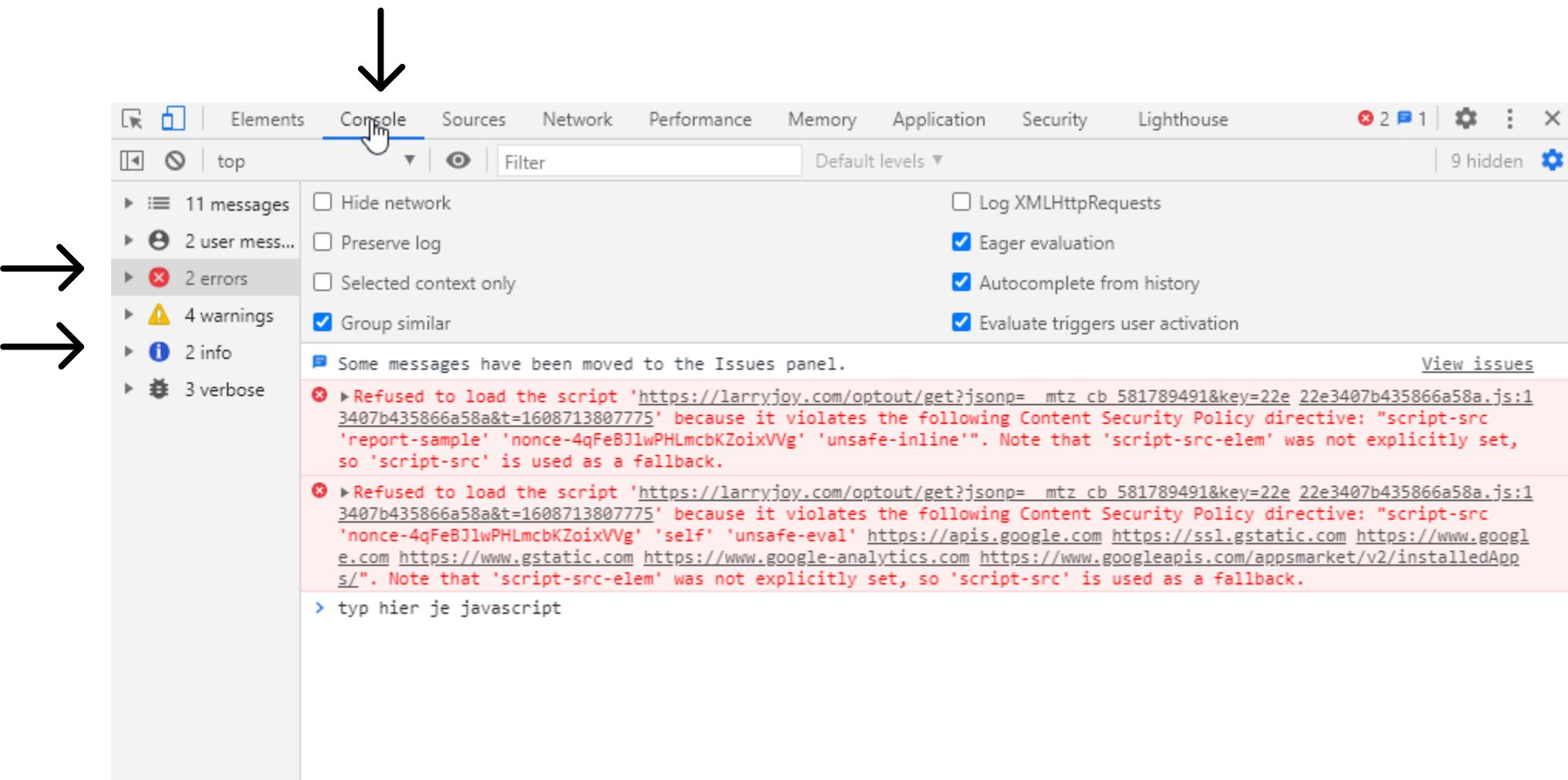
Voor debuggen en leuk om mee te experimenteren!

Controleer HTML CSS en Javascript en haal bugs uit je website.

Je kan errors in javascript zien. Je kan output in de console bekijken en je kan javascript runnen in de console!

### Openen DevTools:

- rechter muisklik -> inspecteren of functie f12
- klik op console hier kan je javascript typen
- klik op errors voor debugging
- klik op info om de console.log te zien



# JavaScript basics

## Chrome dev tools voor javascript

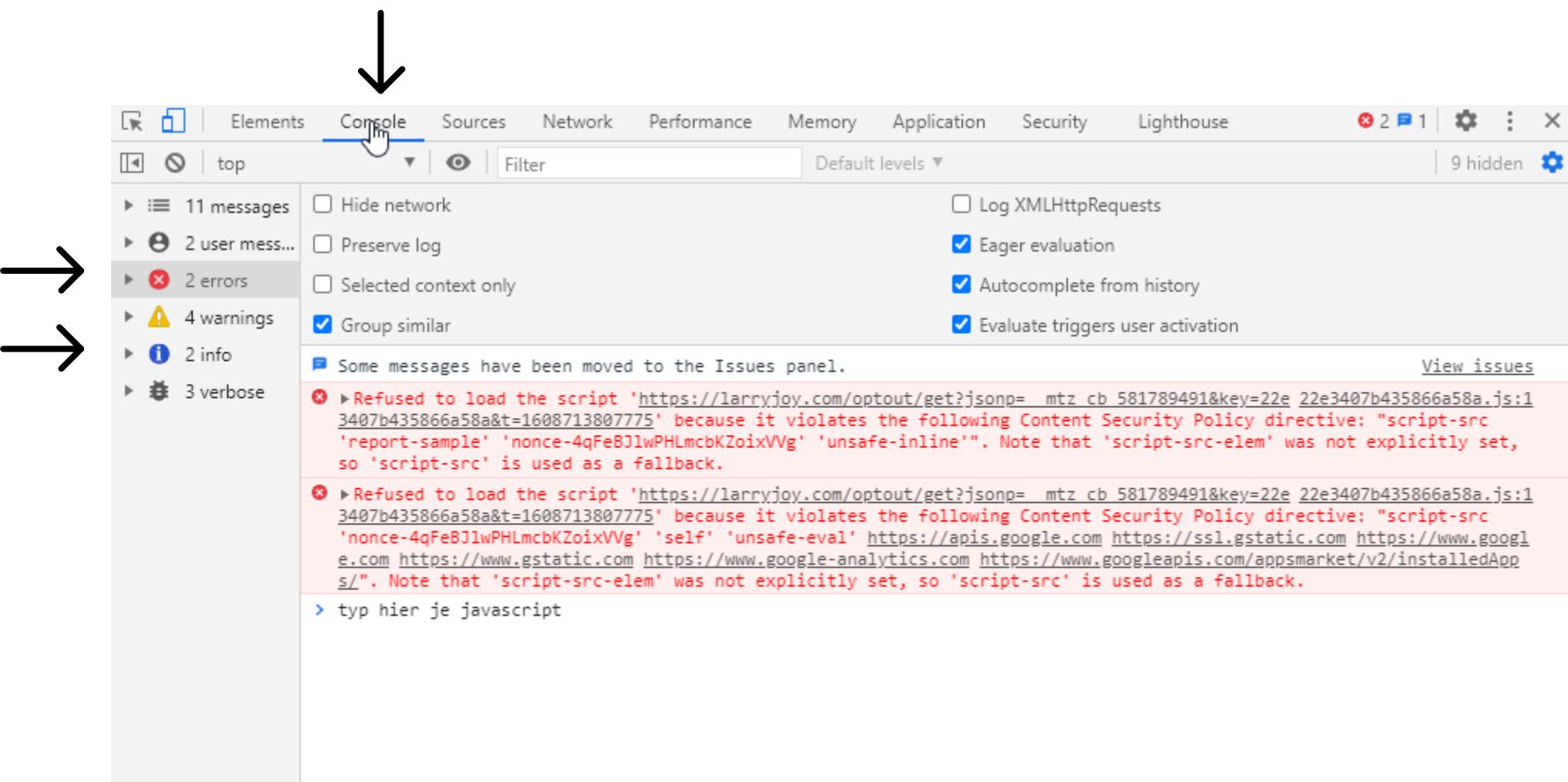
Voor debuggen en leuk om mee te experimenteren!

Controleer HTML CSS en Javascript en haal bugs uit je website.

Je kan errors in javascript zien. Je kan output in de console bekijken en je kan javascript runnen in de console!

### Openen DevTools:

- rechter muisklik -> inspecteren of functie f12
- klik op console hier kan je javascript typen
- klik op errors voor debugging
- klik op info om de console.log te zien



# JavaScript basics

## console.log alert en prompt

### console.log

Een geweldige tool om te testen! Hiermee kan je iets in de console te loggen. Verwacht je een bepaalde uitkomst van een functie, wil je controleren wat er in een variabele is opgeslagen? log het in de console!

```
console.log('dit is een log' + var1);
```

### alert

Deze invoer is onjuist! Het programma is vastgelopen! Uw sessie is verlopen! En vele andere belangrijke meldingen zie je vaak in een alert. In een alert kan je een variabele melden of een String of een combinatie van beiden.

```
alert('dit is een alert' + var1);
```

### prompt

Hiermee kan je om invoer van de gebruiker vragen. Deze invoer kan je direct opslaan als een variabele. Een prompt kan je een default waarde geven.

```
var naam1 = prompt("Please enter your name");
```

```
var naam2 = prompt("Please enter your name", "Harry Potter");
```

# JavaScript basics

## Variabelen: var, let en const

### Wat kan in een variabele?:

In een variabele kan je vanalles opslaan denk aan:

String (woorden en zinnen)

Number (getallen)

Objecten (buiten scope van cursus)

Array (meerdere elementen, bijvoorbeeld reeks getallen)

Boolean (true of false)

Functies (verwijs naar een functie)

Andere verwijzingen (bijvoorbeeld naar iets uit de HTML)

### var

- heeft geen scope, laatste waarde telt.
- kan opnieuw een waarde krijgen
- kan opnieuw gedeclareerd worden

### let

- leeft alleen binnen zijn eigen scope
- kan niet opnieuw gedeclareerd worden binnen dezelfde scope
- kan opnieuw een waarde krijgen

### const

- leeft alleen binnen zijn eigen scope
- kan niet opnieuw gedeclareerd worden binnen dezelfde scope
- kan niet opnieuw een waarde krijgen

### Linkjes

#### JS variabelen

[https://www.w3schools.com/js/js\\_variables.asp](https://www.w3schools.com/js/js_variables.asp)

# JavaScript Strings

## Strings

Strings zijn handig om text data op te slaan. Doordat Strings worden behandeld als objecten kan je een aantal functionaliteiten gebruiken zoals alle karakters omzetten naar lowercase of uppercase. Ook kan je de lengte van de String makkelijk berekenen.

## Creeëren String

Een String maken kan met dubbele of enkele quotes.

```
"dit is een string"
```

```
console.log("dit is String in een log");
```

```
const zinnetje = ("dit is een String in een variabele");
```

## Strings transformeren en manipuleren

```
let zinnetje = "Hello World";
console.log(zinnetje.length);
console.log(zinnetje.toUpperCase());
```

## Strings combineren

```
let name = prompt("Wat is je naam?");
```

```
let message = "Welkom " + name + " bij deze
javascript les!";
```

```
alert(message);
```

# JavaScript Strings

## Strings

Strings zijn handig om text data op te slaan. Doordat Strings worden behandeld als objecten kan je een aantal functionaliteiten gebruiken zoals alle karakters omzetten naar lowercase of uppercase. Ook kan je de lengte van de String makkelijk berekenen.

## Creeëren String

Een String maken kan met dubbele of enkele quotes.

```
"dit is een string"  
  
console.log('dit is String in een log');  
  
const zinnetje = ("dit is een String in een variabele");
```

## Strings transformeren en manipuleren

```
let zinnetje = "Hello World";  
  
console.log(zinnetje.length);  
console.log(zinnetje.toUpperCase());
```

## Strings combineren

```
let name = prompt("Wat is je naam?");  
  
let message = "Welkom " + name + " bij deze  
javascript les!";  
  
alert(message);
```

# JavaScript Strings

## Template literals

Template literals gebruiken back-ticks (`) ipv (`""`). Je kan binen de template literals gewoon enkele en dubbele quotes gebruiken als tekst. Ook kan je makkelijk meerdere regels gebruiken.

Ook is het mogelijk om variabelen in template literals te gebruiken

```
let name = "Maurice";
let woonplaats = "Groningen";
let number = 4;

let text = `Welkom ${name},
volgens onze gegevens woon je in ${woonplaats}.
Een random sommetje ${2*number};
Einde bericht.`;
```

## Linkjes

### JS Strings

[https://www.w3schools.com/js/js\\_strings.asp](https://www.w3schools.com/js/js_strings.asp)

### JS template literals

[https://www.w3schools.com/js/js\\_string\\_templates.asp](https://www.w3schools.com/js/js_string_templates.asp)

# JavaScript Numbers & Math

## Arithmetical Operators

Tijdens het programmeren zul je veel gebruik maken van calculaties. Optellen aftrekken vermenigvuldigen en delen.

```
3 + 9 // 12  
5 - 2 // 3  
8 / 4 // 2  
10 * 9 // 90
```

```
let score;  
let ronde1 = 100;  
score = score + ronde1;
```

## Numbers en Strings

Wat gebeurt er als je '10' + '5' doet? Javascript ziet dit als Strings en zal als uitkomst '105' hebben.

Een string kan je omzetten naar een number.

```
let getal1 = '12';  
let getal2 = 6;  
  
let som = parseInt(getal1) + getal2
```

# JavaScript Numbers & Math

## random nummer

Met het Math object van javascript kan je wiskundige taken uitvoeren zoals af- ronden naar beneden wortel berekenen en ook random getallen genereren.

```
Math.random(); // getal tussen 0 en 1  
5 * Math.random(); // getal tussen 0 en 5  
Math.floor(5 * Math.random()); // 0, 1, 2, 3 of 4  
Math.floor(5 * Math.random()) + 1; // 1, 2, 3, 4 of 5
```

## Linkjes

### JS Math

[https://www.w3schools.com/jsref/jsref\\_obj\\_math.asp](https://www.w3schools.com/jsref/jsref_obj_math.asp)

### JS Math.random()

[https://www.w3schools.com/jsref/jsref\\_random.asp](https://www.w3schools.com/jsref/jsref_random.asp)

# JavaScript Conditional Statements

## If, else en else if

Conditional statements bepalen welke acties worden uitgevoerde bij verschillende waarheden. In een if en else if wordt altijd gekeken of de vergelijking `true` of `false` is. (boolean)

`if`: een block code die wordt uitgevoerd als een conditie waar is.

`else`: een block code als de condities erboven niet waar zijn.

`else if`: om een nieuwe conditie te testen als de voorgaande conditie niet waar is.

```
let leeftijd = 20;

if (leeftijd < 18) {
    console.log('je bent te jong');
} else if (leeftijd > 60) {
    console.log('je bent te oud');
else {
    console.log('kom binnen');
}
```

## Comparison Operators

Om te kijken of waarden gelijk of juist niet gelijk zijn worden comparison operators gebruikt.

Bekijk de link.

## Linkjes

### JS if else if else

[https://www.w3schools.com/js/js\\_if\\_else.asp](https://www.w3schools.com/js/js_if_else.asp)

### JS Comparison Operators

[https://www.w3schools.com/js/js\\_comparisons.asp](https://www.w3schools.com/js/js_comparisons.asp)

# JavaScript Conditional Statements

## Logical Operators

Logical operators worden gebruikt om de logica tussen variabelen of waarden te bepalen.

Zie link comparison operators.

```
let leeftijd = 20;

if (leeftijd >= 18 && leeftijd < 60) {
    console.log('je mag naar binnen');
} else {
    console.log('te jong of te oud');
}
```

# JavaScript functions

## Functions

Een functie is een block met code die taken uitvoert zodra iets deze functie aanroept.

Een functie kan iets returnen en argumenten mee krijgen van de aanroeper

```
function myFunction() {  
    console.log('ik ben een standaard functie');  
}  
  
function myFunction2() {  
    console.log('ik ben een functie en return het getal 20');  
    return 20;  
}  
  
function myFunction3(p1,p2) {  
    console.log('ik ben een functie met parameters');  
    return p1 + p2;  
}
```

## Aanroepen functie

Een functie kan worden aangeroepen vanuit een andere plek in de JavaScript code. Ook kan je vanuit de html een functie triggeren met een event. Zoals met een click of een mouseover

## Linkjes

### Functions

[https://www.w3schools.com/js/js\\_functions.asp](https://www.w3schools.com/js/js_functions.asp)

### Events

[https://www.w3schools.com/tags/ref\\_eventattributes.asp](https://www.w3schools.com/tags/ref_eventattributes.asp)

# JavaScript functions

## innerHTML

Het is handig om de html te kunnen veranderen via JavaScript. Hiermee kan je bijvoorbeeld de inhoud van een div veranderen en er tekst in plaatsen of de bestaande tekst te vervangen met andere tekst. Ook kun je hele stukken HTML plaatsen in iets als een div.

## veranderen HTML Style

Ook is het handig om de style van HTML elementen te kunnen veranderen met JavaScript. Hiermee kan je bijvoorbeeld een stuk tekst tonen bij een click en weer verbergen bij een volgende click etc.

```
document.getElementById("demo").innerHTML = "Paragraph changed!";
document.getElementById("demo").style.display= "none";
```

## Linkjes

### InnerHTML

[https://www.w3schools.com/jsref/prop\\_html\\_innerhtml.asp](https://www.w3schools.com/jsref/prop_html_innerhtml.asp)

### Style

[https://www.w3schools.com/js/js\\_html\\_idom\\_css.asp](https://www.w3schools.com/js/js_html_idom_css.asp)

# JavaScript arrays

## Arrays

Arrays worden gebruikt om meerdere waarden in 1 variabele op te slaan. Zie het als soort van lijstjes. Het 1e item in een array begint bij 0.

```
let studenten= ["Kees", "Piet", "Simone"];
let cars=[];
cars[0]= "Saab";
cars[1]= "Volvo";
cars[2]= "BMW";

let arrayitems= ["Kees", 12, false];
```

```
const stad= {naam:"Amsterdam", inwoners:820000, provincie:"Noord-Holland"};
```

## numbers and names

Arrays gebruiken standaard nummers voor het ophalen van items, ook kun je de items een naam geven.

```
studenten[1];
stad.naam;
```

## Linkjes

### Arrays

[https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp)

# JavaScript arrays

## Multi dimensional Arrays

Het is soms ook handig om lijstjes in lijstjes te hebben. Denk aan een school met klassen, dit is een 1 dimensionaal array, echter wil je van al die klassen ook de studenten hebben. Dan wordt het een 2 dimoensionaal array

```
let numberitems = [
  [1, 6, 9],
  [12, 4 , 8],
  [4, 9]
];

console.log(numbersitems[2,1];
```

```
let school = [
  {klas:"groep1",leerlingen:["Piet","Kees"]},
  {klas:"groep2",leerlingen:["Clara","Simone"]}
];

console.log(school[1].leerlingen[1])
```

# JavaScript Loops

## Wat zijn loops?

Loops zijn erg handig als je dezelfde code opnieuw en opnieuw wil uitvoeren met elke keer een andere waarde.

Het is helemaal handig als je een array hebt met waarden waarmee je elke keer iets wil uitvoeren

### For Loop

```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}  
  
let studenten = ['Kees','Piet','Clara'];  
for (let x = 0; x < studenten.length; x++) {  
    console.log('welkom ' + studenten[x]);  
}
```

### While loop

De while loop runt zolang een conditie waar is.

```
let number = 0;  
  
while (number < 10) {  
    number++;  
}
```

## Linkjes

### For Loops

[https://www.w3schools.com/js/js\\_loop\\_for.asp](https://www.w3schools.com/js/js_loop_for.asp)

### While Loops

[https://www.w3schools.com/js/js\\_loop\\_while.asp](https://www.w3schools.com/js/js_loop_while.asp)

# HTML form elements

## input

Het meest gebruikte form element is de <input>. Een input kan een simpel tekst veld zijn maar ook een nummer, wachtwoord checkbox en veel meer. Het type input dat getoond wordt hangt af van het attribuut type.

### Input checkbox

```
<input type="checkbox" id="toppings1" name="toppings1" value="Ananas">
<label for="toppings1">Ananas</label>
<input type="checkbox" id="toppings2" name="toppings2" value="Bacon">
<label for="toppings2">Bacon</label>
<input type="checkbox" id="toppings3" name="toppings3" value="Tonijn">
<label for="toppings3">Tonijn</label>
```

### Input radio

```
<input type="radio" id="student" name="my_role" value="student">
<label for="student">Student</label><br>
<input type="radio" id="docent" name="my_role" value="docent">
<label for="docent">Docent</label><br>
```

### Input text

```
<input type="text" id="fname" name="fname">
```

## Linkjes

### Form Elements

[https://www.w3schools.com/html/html\\_form\\_elements.asp](https://www.w3schools.com/html/html_form_elements.asp)

### While input types

[https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)

# JavaScript LocalStorage

## Web Storage

Het is in veel situaties handig om informatie op te slaan in de browser van de gebruiker. Dit werd vroeger gedaan in cookies, maar kan tegenwoordig ook in localStorage en sessionStorage.

```
localStorage.setItem("naam", "Maurice");

localStorage.getItem("naam");
```

## Linkjes

### Webstorage

[https://www.w3schools.com/html/html5\\_webstorage.asp](https://www.w3schools.com/html/html5_webstorage.asp)