



Crash Course CSS / BOOTSTRAP

Trainingssessie Thomas More

Jorne Marx – Pieter Pauwels

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

INHOUDSTAFEL

Inhoud

INHOUDSTAFEL	3
1 INLEIDING/VOORBEREIDING	5
1.1 Wie, Wat, Waarom?	5
1.2 Het doel van de cursus	5
1.3 Benodigdheden	5
2 CSS	6
2.1 Inleiding	6
2.2 Het gebruiken van CSS	7
2.3 Het linken van stylesheets	8
2.4 Syntaxis van CSS	8
2.5 CSS Selectoren	9
2.5.1 Element selector	9
2.5.2 ID selector	9
2.5.3 Klasse selector	9
2.5.4 Universele selector	10
2.5.5 Komma combinatie	10
2.5.6 Descendant selector	11
2.5.7 Child selector	11
2.5.8 Adjacent sibling selector	11
2.5.9 General Sibling Selector	11
2.5.10 Pseudo-klasses selectoren	12
2.5.11 Pseudo elementen	13
2.6 Commentaar	14
2.7 Kleuren	15
2.7.1 Color	16
2.7.2 Background-color	16
2.8 Overriding	17
2.9 Lettertype wijzigen	18
2.9.1 Font-family	18
2.9.2 Font-size	19
2.9.3 Font-weight	19
2.9.4 Tekst-decoration	20
2.10 Waarde van lengtes	21
2.10.1 Absolute lengte-eenheden	21
2.10.2 Relatieve lengte-eenheden	21
2.10.3 Percentage	21
2.11 Border	22
2.12 Border-radius	23
2.13 Positionering	24
2.14 Margin en Padding	25
2.15 Icoontje	31
3 BOOTSTRAP	32
3.1 Inleiding	32
3.2 Installatie / integratie van Bootstrap	33
3.2.1 Installeren via pre-compiled bestanden	33
3.2.2 Integreren via een CDN	33
3.3 Bootstrap Utilities	34
3.3.1 Margin en padding	34
3.3.2 Width	35
3.3.3 Achtergrondkleur	36

3.3.4	Tekstkleur.....	36
3.4	Bootstrap Grid.....	37
3.4.1	Mobile first	37
3.4.2	De werking van het grid systeem	37
3.4.3	Oefening op het grid systeem	38
3.4	Bootstrap componenten	38
3.4.4	Knoppen	39
3.4.5	Cards	40
3.4.6	Alerts	41
3.4.7	Oefening op componenten.....	42

1 INLEIDING/VOORBEREIDING

1.1 Wie, Wat, Waarom?

Wij zijn Pieter Pauwels en Jorne Marx. We studeren beide Application Development / Artificial Intelligence aan de Thomas More Hogeschool in Geel.

Wij geven deze les in het kader van het vak Professional Skills, waar we de opdracht kregen een groep mensen een IT-gerelateerde skill bij te brengen door middel van een trainingssessie.



1.2 Het doel van de cursus

In deze cursus zal je kennismaken met CSS en het framework Bootstrap. Het doel is om je een goede basiskennis van beide aan te leren. Op het einde van de cursus zal je in staat zijn om een HTML pagina een mooie basisopmaak te geven.

1.3 Benodigheden

Voor het volgen van deze cursus heb je een goede web-ontwikkelingstool nodig, alsook een goede basiskennis van HTML. Wij werken bij voorkeur met [Visual Studio Code](#). Indien een andere tool je voorkeur heeft kan je die gerust gebruiken.



2 CSS

2.1 Inleiding

Cascading Style Sheets (CSS) is een opmaak taal die we gebruiken om stijl toe te kennen op HTML documenten. Het is geen echte "programmeertaal", maar een reeks instructies voor een web browser om webpagina's te vormen.



CSS is geïntroduceerd omwille van een eenvoudigere en consistentere vormgeving van webpagina's. Het World Wide Web Consortium (W3C) heeft daarom de standaard vastgelegd. Het W3C is een organisatie die de webstandaarden voor het wereldwijde web ontwerpt. Denk hier bijvoorbeeld aan HTML, XML en CSS.

We maken bij voorkeur externe stijlbestanden, hierdoor houden we de content en de presentatie van een webpagina onafhankelijk van elkaar. Deze CSS-bestanden kunnen we vervolgens linken aan HTML-documenten om de beschreven stijlgeving toe te passen op deze HTML-documenten. Door deze onafhankelijkheid kunnen we de CSS-file(s) aanpassen die vervolgens de wijzigingen toepast op alle gekoppelde HTML-files.

2.2 Het gebruiken van CSS


Er zijn 3 manieren om CSS te gebruiken:

- **Inline**
Hierbij schrijven we stijlen die van toepassing zijn voor HTML-elementen binnen het element zelf. De stijl is enkel van toepassing op het HTML-element waarin deze stijl is beschreven.

A code editor window titled 'index.html' with a dark background. It contains a single line of HTML code: `<h1 style="font-size: large;">Crash Course CSS</h1>`. The text 'Crash Course CSS' is displayed in a large font size.

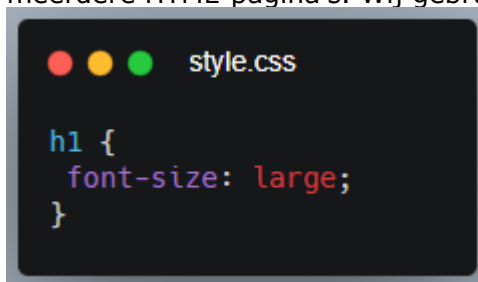
```
<h1 style="font-size: large;">Crash Course CSS</h1>
```

- **Internal**
Hierbij schrijven we de stijlgeving die van toepassing is in je HTML-file binnen een style-element. De stijlen die we hierin definiëren zijn enkel van toepassing binnen deze HTML-file.

A code editor window titled 'index.html' with a dark background. It contains HTML code for a heading and an internal style block. The heading text 'Crash Course CSS' is displayed in a large font size.

```
<h1>Crash Course CSS</h1>
<style type="text/css">
  h1 {
    font-size: large;
  }
</style>
```

- **External**
Hierbij schrijven we de stijlgeving die van toepassing is in een aparte .css-file. Zo kunnen we vervolgens deze stijlgeving toepassen op meerdere HTML-pagina's. Wij gebruiken deze manier.

A code editor window titled 'style.css' with a dark background. It contains a CSS rule for the h1 element, which is shown as 'h1' in the preview area.

```
h1 {
  font-size: large;
}
```

2.3 Het linken van stylesheets

Wanneer we een HTML-pagina hebben en we willen hier een stijlgeving in CSS aan toekennen kunnen we dit doen met behulp van de link-tag. We doen dit binnen de head-tag.



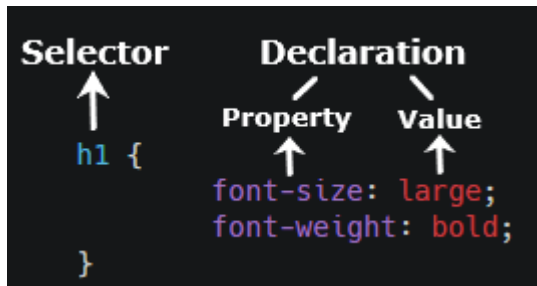
```
index.html

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="style.css">

  <title>Crash Course CSS</title>
</head>
```

2.4 Syntaxis van CSS



De selector verwijst naar het HTML-element dat je wilt stijlen.

Het declaratie-blok { ... } bevat 1 of meerdere declaraties, gescheiden door een puntkomma.

Elke declaratie bevat een CSS-eigenschapsnaam en een waarden, gescheiden door een dubbelpunt.

Belangrijk om te weten is dat niet alle CSS-eigenschappen ondersteunt worden door alle browsers. De meest voorkomende worden wel ondersteunt, maar voor andere moet je soms een omweg gebruiken om hetzelfde resultaat te bekomen.

De website <https://caniuse.com/> kan je gebruiken om te controleren of je een bepaalde eigenschap wel of niet kan gebruiken.

2.5 CSS Selectoren

Op de website <https://flukeout.github.io/> kan je de CSS selectoren leren en oefenen.

2.5.1 Element selector

Selecteert een HTML-element op basis van de naam van het element.



2.5.2 ID selector

Maakt gebruik van het attribuut "id" van een HTML-element. Een id is uniek binnen een pagina, dus deze selector wordt gebruikt om 1 uniek element binnen de pagina te selecteren.

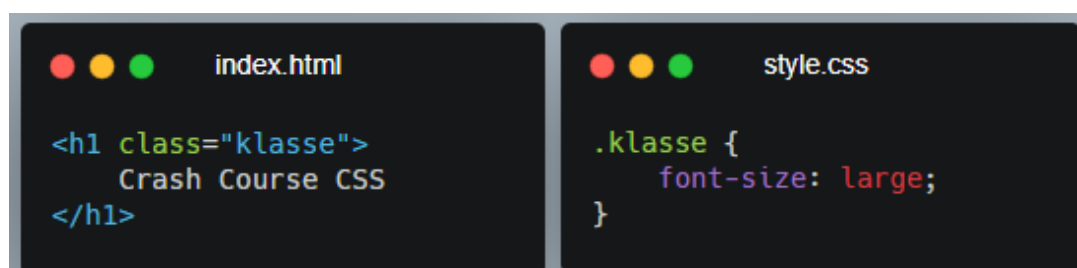
Om een element met een specifieke id te selecteren gebruik je een hashtag (#) gevolgd door de id van het element.



2.5.3 Klasse selector

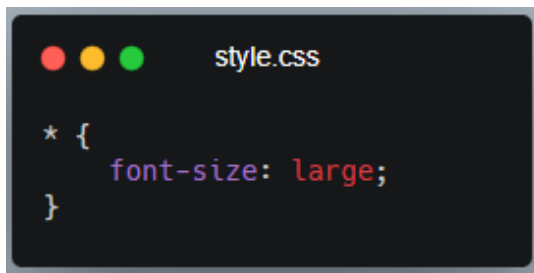
Deze selector maakt gebruik van het attribuut "class" van HTML-elementen. Een klasse kan aan zoveel elementen worden toegevoegd als je wilt.

Om een element te selecteren met een specifieke klasse gebruik je een punt (.) gevolgd door de naam van de klasse.



2.5.4 Universele selector

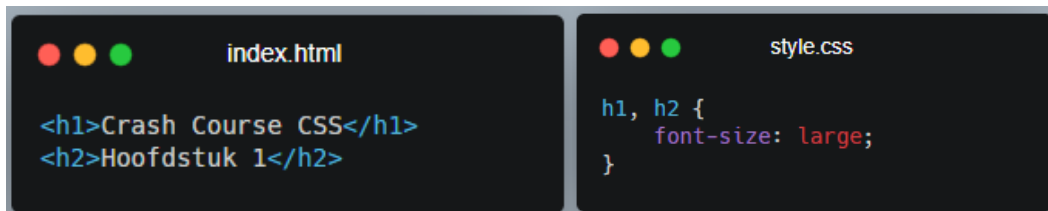
Deze selector selecteert alle HTML-elementen op de pagina en wordt aangeduid met een asterisk-teken (*).

A code editor window titled 'style.css' with a dark background. It contains the following CSS code:

```
* {  
    font-size: large;  
}
```

2.5.5 Komma combinatie

Als je voor meerdere elementen dezelfde stijl wilt toepassen kan je deze elementen selecteren met een komma (,) als scheidingsteken.

Two code editor windows side-by-side. The left window is titled 'index.html' and contains:

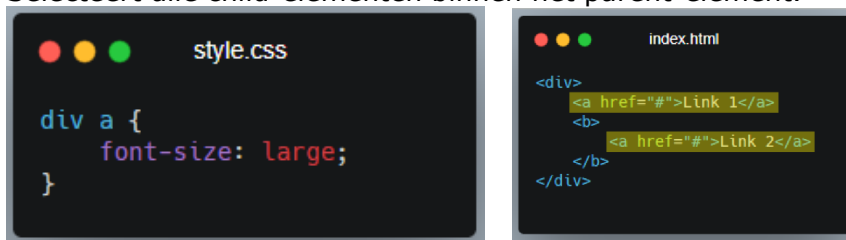
```
<h1>Crash Course CSS</h1>  
<h2>Hoofdstuk 1</h2>
```

The right window is titled 'style.css' and contains:

```
h1, h2 {  
    font-size: large;  
}
```

2.5.6 Descendant selector

Selecteert alle child-elementen binnen het parent-element.



2.5.7 Child selector

Selecteert alle directe child-elementen binnen het parent-element.



2.5.8 Adjacent sibling selector

Deze selector selecteert een element dat direct na een ander element komt.



2.5.9 General Sibling Selector

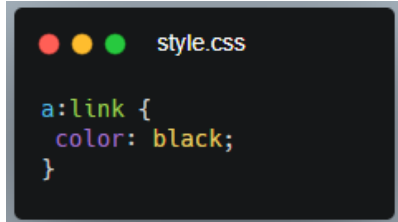
Hierdoor kan men het element selecteren dat een ander element opvolgt binnen hetzelfde parent-element.



2.5.10 Pseudo-klassen selectoren

Pseudo-klassen worden gebruikt om de stijl van een element te definiëren in een bepaalde staat, dit zijn selectoren met een dubbelpunt. Denk hieraan aan een link wanneer deze al bezocht is, hierover beweegt met je muis, ...

- Niet bezochte link



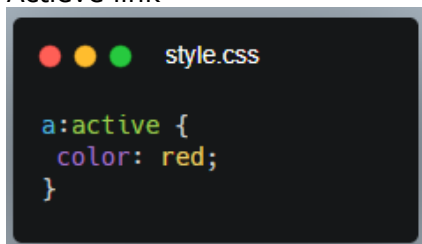
```
style.css
a:link {
  color: black;
}
```

- Bezochte link



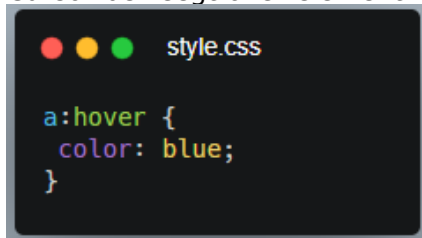
```
style.css
a:visited {
  color: white;
}
```

- Actieve link



```
style.css
a:active {
  color: red;
}
```

- Cursor beweegt over element



```
style.css
a:hover {
  color: blue;
}
```

Er zijn nog tal van aantal andere pseudo-klasse zoals:

- **:first-child**
- Selecteert het element dat het eerste kind is van een gespecificeerd element.
- **:last-child**
- Selecteert het element dat het laatste kind is van een gespecificeerd element.
- **:nth-child(n)**
- Selecteert het element dat het n-de kind is van een gespecificeerd element.
- ...

2.5.11 Pseudo elementen

Een pseudo-element wordt gebruikt om een specifiek gedeelte van een element een stijl te geven, dit zijn selectoren met twee dubbelpunten. Bijvoorbeeld de eerste letter van een element, inhoud toevoegen voor of na de inhoud van een element, ...

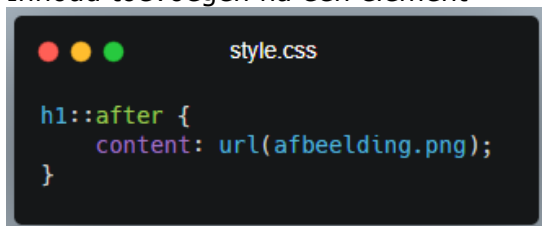
- Eerste letter



- Inhoud toevoegen voor een element



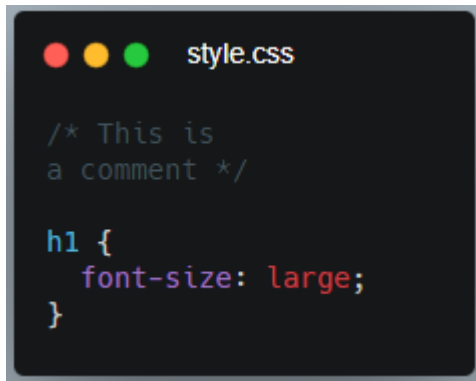
- Inhoud toevoegen na een element



2.6 Commentaar

Commentaar wordt gebruikt voor code toe te lichten, dit kan je later helpen wanneer je de code aanpast. Deze commentaarlijnen worden genegeerd door browsers.

In CSS start commentaar met `/*` en eindigt met `*/`.



2.7 Kleuren

In CSS heb je verschillende manieren om de waarden van een kleur te declareren. De meest gebruikte zijn:

- **Naam**
Dit zijn enkele namen van kleuren die je kan specificeren en die elke browser zal begrijpen. Voorbeelden hiervan zijn: red, green, cyan, magenta, lightpink, ... De hele lijst kan je [hier](#) vinden.
- **Hexadecimale-waarde**
Een hexadecimale kleur wordt gespecificeerd met #RRGGBB. RR (rood), GG (groen) en BB (blauw) zijn hexadecimale cijfers tussen 00 en FF die de sterkte van die kleur bepaald. Bijvoorbeeld zal de waarden #FF0000 de kleur rood tonen en de waarden #000000 de kleur zwart.
- **RGB-waarden**
Een RGB-kleur waarden wordt gespecificeerd met rgb(<RED>, <Green>, <Blue>) elke parameter is een integer tussen de waarden 0 en 255 die de sterkte van die kleur bepaald.
- **RGBA-waarden**
Dit is een uitbreiding van de RGB-waarden waarbij er een extra parameter is: rgba(<RED>, <GREEN>, <BLUE>, <ALPHA>). De alpha parameter bepaald de transparantie. Helemaal transparant (onzichtbaar) = 0.0, en niet transparant = 1.0.

Een makkelijke tool om zelf een kleur te kiezen is de [Google color picker](#).

2.7.1 Color

Beschrijving:

Met behulp van deze eigenschap kan je de voorgrond kleur wijzigen.

Syntax:

color: <kleur>;

Voorbeeld(en):

2.7.2 Background-color

Beschrijving:

Met behulp van deze eigenschap kan je de achtergrond kleur wijzigen.

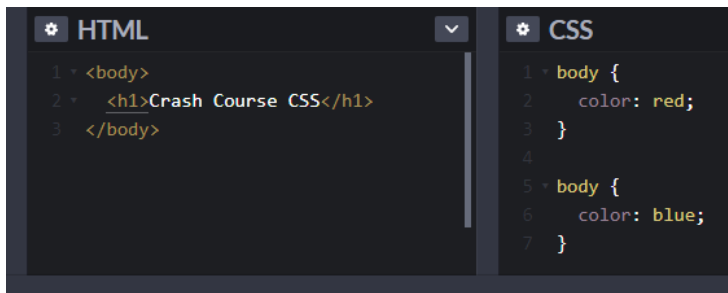
Syntax:

Background-color: <kleur>;

Voorbeeld(en):

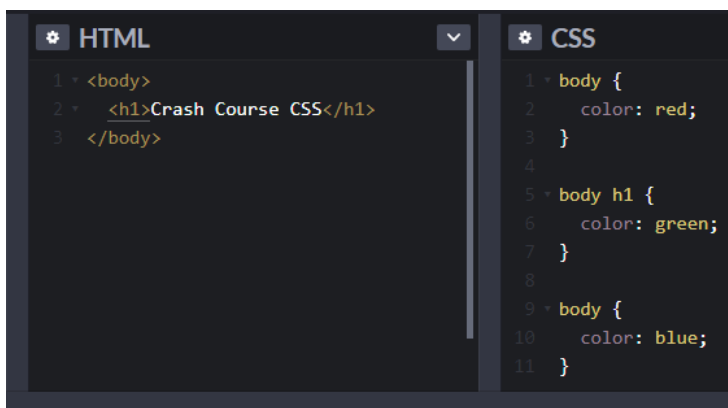
2.8 Overriding

Merk op dat als je eenzelfde element 2x specificeert met dezelfde eigenschap dat de laatste eigenschap zal overheersen.



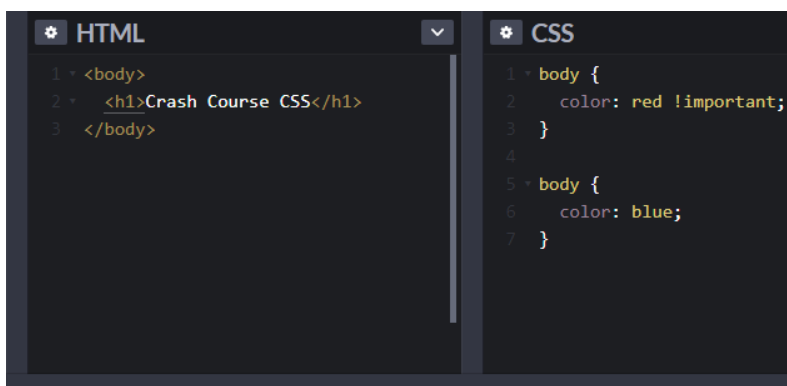
Crash Course CSS

Merk op hoe specifieker je een element specificeert dat deze eigenschap zal overheersen.



Crash Course CSS

Wil je toch dat een eigenschap die eerder is gespecificeerd zal overheersen over een eigenschap die later werd gespecificeerd, maak dan gebruik van de eigenschap !important.



Crash Course CSS

2.9 Lettertype wijzigen

2.9.1 Font-family

Beschrijving:

Met behulp van deze eigenschap kan je het lettertype aanpassen.

Voor de hele pagina een bepaald lettertype te geven declareer je dit op de <body>-tag.

Lettertypes worden vaak opgedeeld in 2 soorten: screef (serif) en schreefloos (sans serif).

Best practice is om een schreefloos lettertype op webpagina's te gebruiken. Zo wordt de pagina door de meeste surfers vlotter te lezen.



Syntax:

font-family: <letter type> (, <backup lettertype>) (, <generiek lettertype>);

Mogelijke waarden:

- <letter type>
 - Je hebt enkele lettertypes die je standaard binnen CSS kan gebruiken, je kan een lijst hiervan [hier](#) terug vinden.
 - Lettertypes die niet standaard binnen CSS zitten moet je importeren, [Google fonts](#) is een vaak gebruikte site hiervoor.
- <generiek lettertype>
 - sans-serif
 - serif
 - monospace
 - cursive
 - fantasy

Voorbeeld(en):

```

style.css

body {
  font-family: Verdana, Arial, sans-serif;
}
```

2.9.2 Font-size

Beschrijving:

Met behulp van deze eigenschap kan je de grootte van het lettertype aanpassen.

Syntax:

font-size: <waarden>;

Mogelijke waarden:

- xx-small , x-small, small, medium, large, x-large, xx-large, larger, smaller
- [<grootte>](#)
- [<percentage>](#)

Voorbeeld(en):

```

style.css

body{
  font-size: x-large;
}

```

```

style.css

p {
  font-size: 2em;
}

```

2.9.3 Font-weight

Beschrijving:

Met behulp van deze eigenschap kan je de dikte van je lettertype aanpassen

Syntax:

font-weight: <waarde>;

Mogelijke waarden:

- Standaard waarden
 - Normal = komt overeen met een waarde van 400.
 - Bold = komt overeen met een waarde van 700.
 - Bolder = dikker dan de waarde van het parent element.
 - Lighter = dunner dan de waarde van het parent element.
- <waarden>
 - Dit is een waarden tussen de 100 en 900 in stappen van 100. Dus 100, 200, 300, ..., 900. Hoe hoger het getal hoe dikker het lettertype.

Voorbeeld(en):

```

style.css

p {
  font-weight: bold;
}

```

```

style.css

p {
  font-weight: 600;
}

```

2.9.4 Tekst decoration

Beschrijving:

Met behulp van deze eigenschap kan je tekst onderlijnen of doorstrepen.

Ook hierbij kan je de lijn een andere kleur geven of deze onderbroken maken. Zie [borders](#) hiervoor.

Syntax:

Tekst-decoration: <waarden> (<stijl>) (<kleur>);

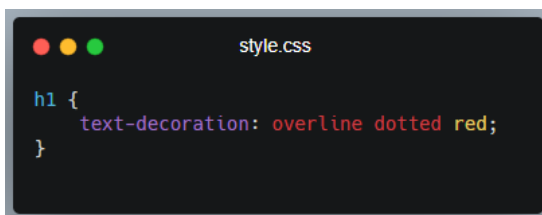
Mogelijke waarden:

- Standaard waarden
 - Underline = onderlijnen van geselecteerde tekst.
 - Line-through = doorstrepen van geselecteerde tekst.
 - Overline = lijn boven de geselecteerde tekst.

Voorbeeld(en):

A code editor window titled 'style.css' with a dark background. It contains the following CSS code:

```
h1 {  
  text-decoration: underline;  
}
```

A code editor window titled 'style.css' with a dark background. It contains the following CSS code:

```
h1 {  
  text-decoration: overline dotted red;  
}
```

2.10 Waarden van lengtes

2.10.1 Absolute lengte-eenheden

Deze eenheden staan op zichzelf. Ze zijn altijd hetzelfde.

Eenheid	Naam	Gelijkwaardig aan
Cm	centimeter	1cm = 38px
Mm	Millimeter	1mm = 0,1cm
Q	Kwart millimeter	1q = 1/4mm = 1/40cm
In	Inches	1in = 2.54cm
Pt	Points	1pt = 1.333px
Px	Pixels	1px = 1/38cm

De meeste van deze eenheden worden zelden tot nooit gebruikt. Meestal wordt de centimeter en millimeter eenheid gebruikt voor het stijlen van de webpagina wanneer deze afgedrukt moet worden.

Enkel de pixels eenheid wordt standaard gebruikt.

2.10.2 Relatieve lengte-eenheden

Deze eenheden zijn afhankelijk van andere absolute eenheden. Ze staan in relatie tot deze andere absolute getallen.

Hier bespreek ik enkel de relevante eenheden:

Eenheid	Relatief aan
Em	Relatief aan de font-size van het parent-element. (2em = 2x de grootte van de huidige font-size)
Rem	Relatief aan de font-size van het root-element (= <html>). (2rem = 2x de grootte van de huidige font-size)
Vw	Relatief aan 1% van de "viewport" breedte . (viewport = de ruimte waarin de webpagina weergegeven wordt)
Vh	Relatief aan 1% van de "viewport" hoogte .

Op [deze](#) website vind je enkele voorbeelden van deze eenheden.

2.10.3 Percentage

100% is gelijk aan de huidige grootte van het parent-element. Dus 150% is 1,5 keer groter dan de huidige grootte van het parent-element.

2.11 Border

Beschrijving:

Met behulp van deze eigenschap kan je het geselecteerde element een rand geven langs alle kanten.

Je kan voor een rand in te stellen ook apart gebruik maken van de eigenschappen "border-width", "border-style" en "border-color". Maar het is aangeraden deze manier te leren.

Syntax:

border: <border-width> <border-style> <border-color>;

Mogelijke waarden:

- border-width
 - Dit is een waarden in een van de voorafgaande [lengte-eenheden](#) (meestal px).
- border-style
 - none = standaard waarden. Er wordt geen border getoond.
 - dotted = gestippeld.
 - dashed = onderbroken.
 - solid = enkele lijn.
 - double = dubbele lijn.
 - ...
- border-color
 - Dit is een kleur in een van de voorafgaande [kleurwaarden](#).

Voorbeeld(en):

```

style.css

div {
  border-width: 2px;
  border-style: solid;
  border-color: black;
}
  
```

```

style.css

div {
  border: 2px solid black;
}
  
```

Je kan ook het element ook enkel een rand geven langs één of meerdere kanten. Dit doe je met behulp van de eigenschap "border-left", "border-top", ... Je kan ook hier apart gebruik maken van de eigenschappen voor de width, style en color: "border-left-width", "border-bottom-color", ...

2.12 Border-radius

Beschrijving:

Met behulp van deze eigenschap kan je de rand van het geselecteerd element afronden. Je moet zorgen dat je eerst een rand hebt gecodeerd voordat je deze eigenschap kan toepassen.

Voor het maken van een cirkel zet je deze eigenschap op 50%.

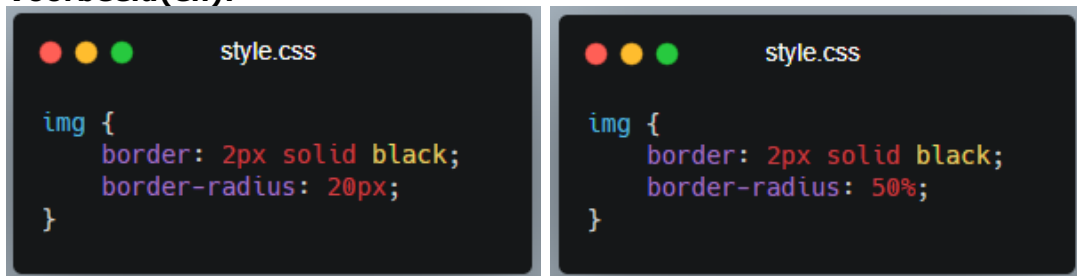
Syntax:

Voor alle hoeken: Border-radius: <waarden>;

Of

Voor individuele hoeken: Border-radius: <linksboven waarden> <rechtsboven waarden> <rechtsonder waarden> <linksonder waarden>;

Voorbeeld(en):



2.13 Positionering

Positionering in CSS is een redelijk uitgebreid thema wil je het helemaal onder de knie krijgen. Voor deze cursus laat ik je zien hoe je een element vrij binnen een ander element kan positioneren.

Allereerst geef je het element waar je het child-element in wilt positioneren de eigenschap `position: relative`.

Vervolgens geef je het element of de elementen dit je wilt positioneren de eigenschap `position: absolute`.

Nu kan je deze elementen vrij binnen het parent-elementen positioneren door middel van volgende attributen:

- Top: <waarden>
- Bottom: <waarden>
- Left: <waarden>
- Right: <waarden>

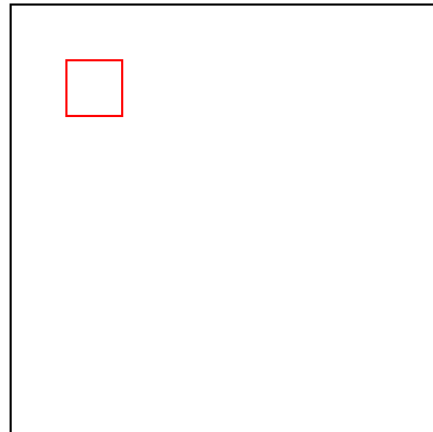
Let op! Steeds de linker bovenhoek van dit element wordt gepositioneerd op deze plaats.

```

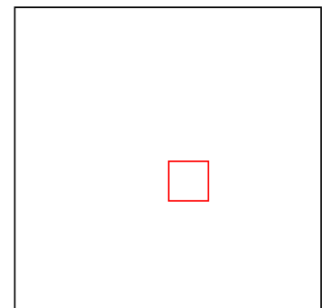
HTML
1 <div id="parent">
2   <div id="child"></div>
3 </div>

CSS
1 #parent {
2   border: 2px solid black;
3   width: 400px;
4   height: 400px;
5
6   position: relative;
7 }
8
9 #child {
10  border: 2px solid red;
11  width: 50px;
12  height: 50px;
13
14  position: absolute;
15  top: 50px;
16  left: 50px;
17 }

```



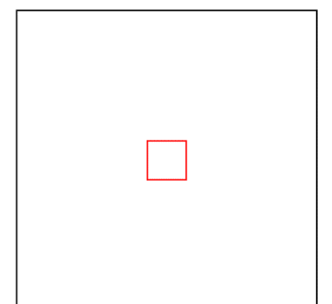
Mocht je het element willen centreren binnen de parent div dan kan je gebruik maken van de properties `top: 50%`; en `left: 50%`.



Merk op dat het element zijn linkerbovenhoek gecentreerd wordt, we kunnen het element zelf centreren door het volgende toe te voegen:

```
transform: translate(-50%, -50%);
```

Het element zal 50% van zijn breedte verzet worden naar links en 50% van zijn hoogte verzet worden naar boven.

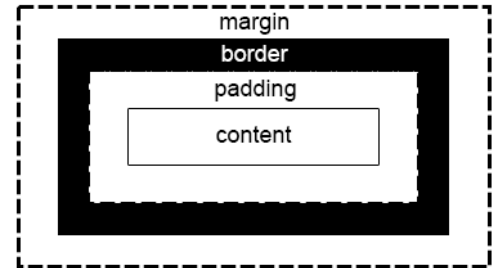


2.14 Margin en Padding

Beschrijving:

Met behulp van deze eigenschappen kan je tussen het geselecteerde element van wat ruimte voorzien.

Belangrijk hierbij is het "CSS Box model" dit is een box dat zich rond elk HTML-element omhuld. Het bestaat uit de marge, een border, een padding en de inhoud zelf. Dit box model zal je ook terug vinden in je browser zijn developer tool.



Goed om te weten is dat een achtergrondkleur dat wordt toegepast op een element ook toegepast wordt op de padding. Dus binnen de border wordt de achtergrondkleur toegepast.

Syntax:

margin: <waarden>;

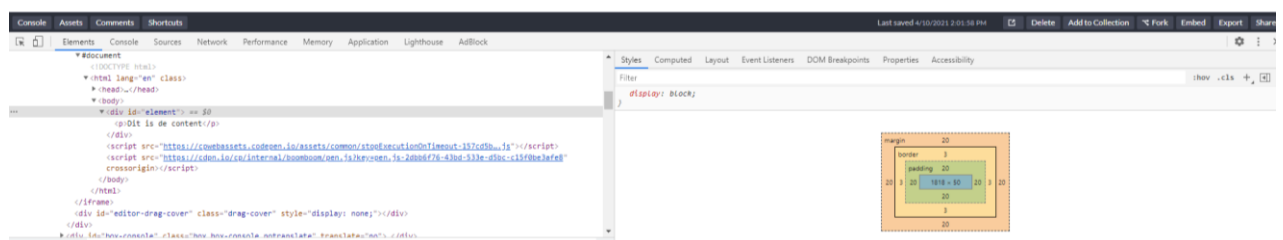
padding: <waarden>;

Voorbeeld(en):

```
style.css

#element {
  background-color: green;

  padding: 20px;
  border: 3px solid red;
  margin: 20px;
}
```



2.15 Width en Height

Beschrijving:

Met behulp van deze eigenschappen kan je de hoogte en breedte van een element vastleggen.

Dit kan erg handig zijn mocht je een knop een bepaalde hoogte of breedte willen geven of als je wilt dat een element maar de helft van je scherm inneemt.

Ook aangeraden is dat je afbeeldingen een percentage voor de breedte geeft zodat deze mee wordt geschaald met de grootte van je scherm.

Syntax:

width: <waarden>;

height: <waarden>;

Voorbeeld(en):



2.16 Verschillende opmaak op verschillende schermgroottes

Aangezien er tegenwoordig zoveel toestellen zijn die erg variëren in grootte van scherm moeten we onze pagina responsief maken. Hiermee bedoelen we dat de inhoud van de pagina zichtbaar is op verschillende groottes van schermen.


Dit doen we door eigenschappen van elementen te veranderen op verschillende groottes van het scherm.

Een algemene best practice is om mobile-first te programmeren. Hiermee bedoelt men dat je eerst de webpagina gaat stijlen voor een klein (smartphone) scherm.

Dus, hoe kan je een andere opmaak toevoegen voor een andere grootte van scherm?

Dit is makkelijk te doen met de media-tag binnen CSS. Deze tag laat ons toe te speciëren van wanneer de stijl binnen deze tag toegepast moet worden.


In deze media tag specificeer je ook vanaf welke scherm breedte deze opmaak moet toegepast worden. Je kan meerdere media tags hebben binnen eenzelfde CSS-file.



```
style.css

@media (min-width: ...px) {
  /* stijl vanaf scherm breedte van ...px */
}
```

Een voorbeeld ter verduidelijking:



```
div {
  background-color: lightgreen;
}

@media (min-width: 800px) {
  div {
    background-color: lightblue;
  }
}

@media (min-width: 1600px) {
  div {
    background-color: lightpink;
  }
}
```

De div zal oorspronkelijk een licht groene achtergrond hebben (mobile first). Als het scherm groter is dan 800px zal deze div een licht blauwe achtergrond hebben en wanneer het scherm groter is dan 1600px zal de div een licht roze achtergrond hebben.

2.17 Flexbox

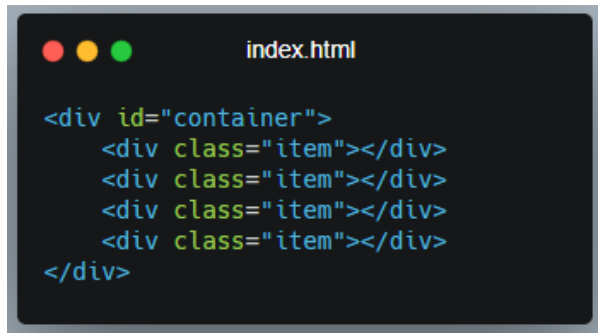
De flexbox opmaak zorgt voor een efficiëntere manier om de positie, uitlijning en toewijzing van ruimte tussen items in een container in te stellen.

Hierdoor kunnen we bijvoorbeeld makkelijk een navigatie maken of afbeeldingen positioneren langs elkaar.

Aangezien flexbox een redelijk uitgebreid topic is gaan we hier niet te diep op in.

Voor een flexbox te maken heb je een container nodig en items. Deze container kan bijvoorbeeld een div-element of ul-element zijn en de items kunnen bijvoorbeeld images, list-items, buttons, div's, ... zijn.

In dit voorbeeld zal ik werken met een div als container en div's als items.



```
index.html

<div id="container">
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
</div>
```

Voor het overzichtelijk te maken geef ik de container een groene achtergrond kleur en neemt deze container de hele hoogte en breedte in van de pagina. De items krijgen een blauwe achtergrond en een border. Merk op dat ik de default margins en paddings verwijder en ik de body- en html-tag een hoogte geef van 100%.



```
style.css

* {
  margin: 0px;
  padding: 0px;
}

html, body {
  height: 100%;
}

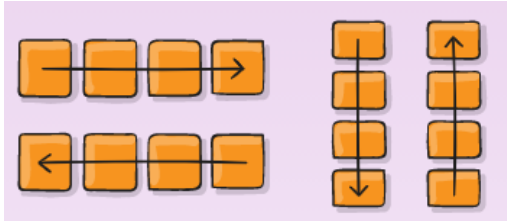
#container {
  width: 100%;
  height: 100%;
  background-color: lightgreen;
}

.item {
  background-color: lightblue;
  border: 1px solid black;

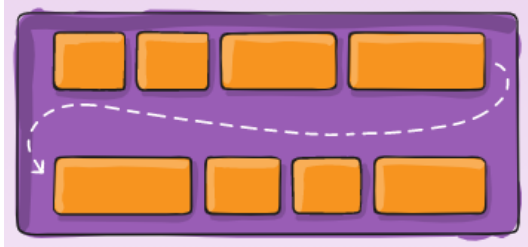
  width: 25%;
  height: 25%;
}
```

Allereerst zijn er enkele eigenschappen die we op de container moeten definiëren:

- **Display: flex;**
Hiermee definieer je dat het element een flex container is.
- **Flex-direction: row | row-reverse | column | column-reverse;**
Hiermee specificeer je hoe de items in de flex container worden geplaatst. Een row plaatst de items horizontaal, een column plaatst de items verticaal.



- **Flex-wrap: wrap;**
Door middel van deze eigenschap worden de items die niet meer op de lijn passen verplaatst naar de volgende lijn.



Als we de items horizontaal naast elkaar willen plaatsen kunnen we dit op de volgende manier doen:

```

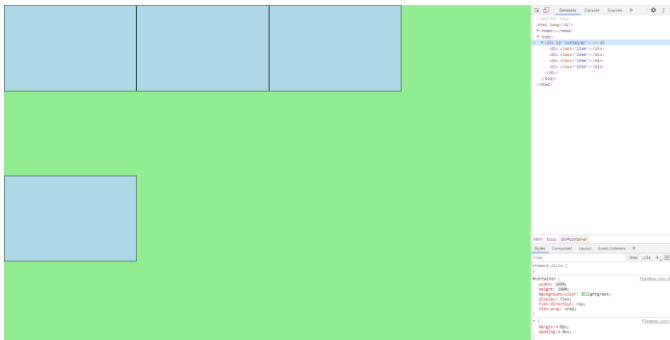
style.css

#container {
  width: 100%;
  height: 100%;
  background-color: lightgreen;

  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
}

```

Het resultaat echter is niet zoals verwacht:



Het laatste item staat op de volgende lijn, nochtans als we 4x 25% pakken hebben we 100%.

2.18 Icoontje

Dit onderdeel heeft niet echt iets met CSS te maken, maar is wel een leuke toevoeging voor vele websites.

Er bestaan enkele "icon libraries" op het internet die over icoontjes beschikken waarvan je gebruik kunt maken. Zo is [Font Awesome](#) hier een van.

Wanneer je naar de website navigeert zie je een hoop icoontjes waarvan je gebruik kunt maken. Helaas zijn enkele icoontjes enkel voor betalende gebruikers, maar er is nog veel keus voor niet-betalende gebruikers.

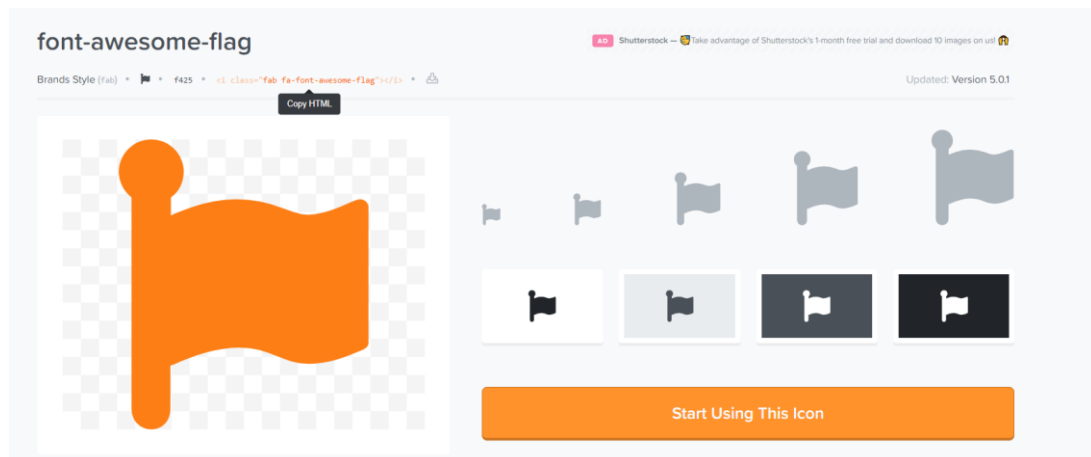
Om deze icoontjes te kunnen gebruiken moet je de volgende lijn toevoegen in je head-tag van je HTML-code op je webpagina.

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" integrity="sha512-iBBXm8fW90+nuLcSKlbmrPcLa00T92xO1BIsZ+ywDWZCvqsWgccV3gFoRBv0z+8dLJgyAHlR35VZc2oM/gI1w==" crossorigin="anonymous" />
```

Deze link verwijst naar een file op een webserver waar deze icoontjes te vinden zijn, info over CDN's volgt in het deel over Bootstrap.

Nu kan je gebruik maken van eender welk gratis icoontje op deze site. Dit doe je door in de lijst van icoontjes een gepast icoontje te kiezen en hierop te klikken.

Vervolgens zie je bovenaan de pagina de HTML-code voor dit icoontje.



Kopieer deze code en plak deze in je HTML-code van je webpagina op de gewenste plaats.

3 BOOTSTRAP

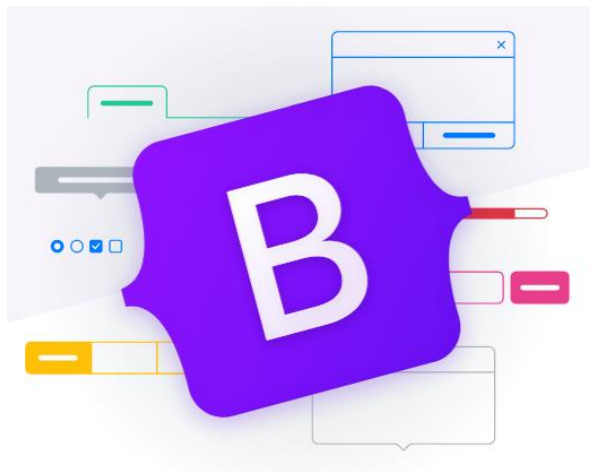
3.1 Inleiding

Bootstrap is een front-end framework bestaande uit een groot aantal CSS classes. Dankzij bootstrap kunnen we de lay-out en bepaalde elementen van een webpagina veel sneller en makkelijker opmaken. Bootstrap heeft ook bepaalde ingebouwde interactieve elementen die werken aan de hand van een Javascript bestand.

Bootstrap is ontstaan bij Twitter als framework om alles een consistente look te geven. Na een tijd besliste men om de code open source te maken aangezien het meer begon te worden dan een simpel intern framework. Bootstrap is beschikbaar op GitHub waar iedereen veranderingen en verbeteringen kan voorstellen.

Out-of-the-box ziet Bootstrap er al heel goed uit, maar er zijn ook een hele reeks opties om het framework te customizen. Dit kan gebeuren op simpele manieren zoals kleine aanpassingen maken in de lokale CSS file van Bootstrap. Maar het kan ook heel ver gaan zoals globale variabelen en waarden aanpassen met behulp van SASS. In deze cursus beperken we ons echter tot de standaard-look van Bootstrap.

Doorheen de cursus zullen we veel verwijzen naar de officiële site en documentatie van bootstrap. Hou deze [site](#) dus zeker bij de hand.



3.2 Installatie / integratie van Bootstrap

3.2.1 Installeren via pre-compiled bestanden

Bootstrap kan op twee manieren aan een site toegevoegd worden.

De eerste optie is om de volledige gecompileerde CSS file te downloaden en hiernaar te verwijzen in de head-tag van je HTML bestand.



```
<link rel="stylesheet" href="bootstrap.min.css">
```

Voordelen:

- Het staat lokaal op je eigen server, je bent niet afhankelijk van externe hosting.

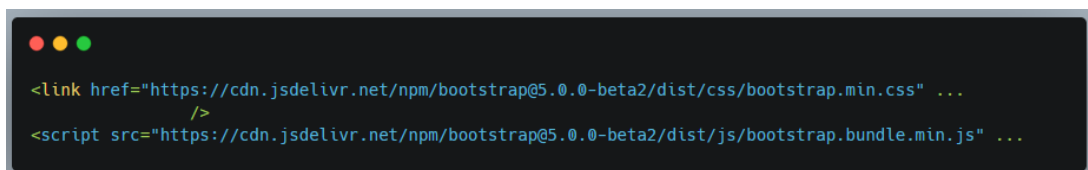
Nadelen:

- Neemt plaats en bandbreedte in
- Veel verschillende bestanden
- Moeilijker te linken in een complexe mappenstructuur

3.2.2 Integreren via een CDN

De tweede optie is om gebruik te maken van een zogenaamde CDN, wat staat voor Content Delivery Network.

De bestanden die nodig zijn voor bootstrap worden dan door browser van de bezoeker van de site gedownload via een externe server.



```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" ... />
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js" ...
```

Voordelen:

- Je eigen server gebruikt minder bandbreedte.
- Bootstrap neemt geen plaats in op je server
- Veel sneller en makkelijker te integreren
- Veel CDN's beschikbaar

Nadelen:

- Wanneer het CDN down is, werkt bootstrap niet op je site.
- Je hebt een internet connectie nodig.

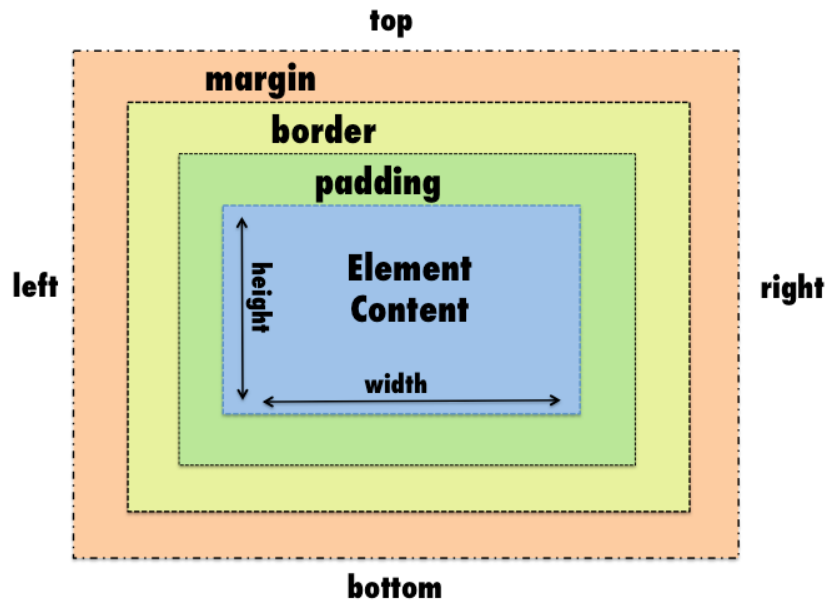
In deze cursus zullen wij werken met een CDN aangezien dit de snelste en makkelijkste optie is.

3.3 Bootstrap Utilities

Bootstrap heeft een groot aantal zogenaamde utilities om de opmaak en layout van de pagina makkelijker en sneller te maken. Deze Utilities bestaan uit een reeks classes die elk hun eigen functie hebben. We bekijken in dit onderdeel de belangrijkste Utilities.

3.3.1 Margin en padding

Jullie kennen het concept van margin en padding reeds uit het CSS deel. Bootstrap is voorzien van standaard classes om deze toe te voegen aan elementen.



De syntax van deze classes werkt als volgt:

{Eigenschap}{Kant}-{Grootte}

Eigenschap is oftewel "m" voor margin of "p" voor padding.

Kant is "r" voor right, "t" voor top, enzovoort. We kunnen ook met "x" en "y" werken om beide horizontale of beide verticale kanten te kiezen.

Grootte is een waarden van 0 tot en met 5 om een bepaalde vastgelegde grootte te selecteren. We kunnen ook "auto" gebruiken om een element te centreren.

Enkele voorbeelden:

```
<div class="pt-2">
  <p>Dit is een div met padding bovenaan.</p>
</div>
```

```
<div class="mx-1">
  <p>Dit is een div met margin aan de linker- en rechterkant.</p>
</div>
```

```
<div class="m-auto">
  <p>Dit is een automatisch gecentreerde div.</p>
</div>
```

3.3.2 Width

We kunnen de breedte van een element heel eenvoudig bepalen met bootstrap.

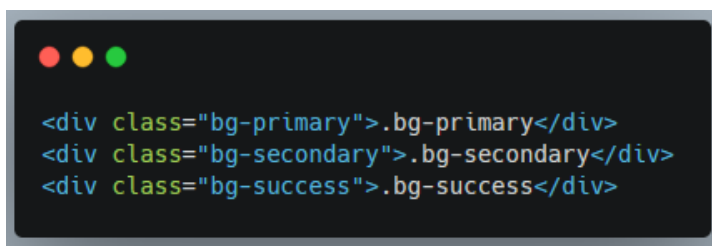
Om een element een bepaalde breedte te geven ten opzichte van het **bovenliggende element** gebruiken we deze classes:

```
<div class="w-25">Width 25%</div>
<div class="w-50">Width 50%</div>
<div class="w-75">Width 75%</div>
<div class="w-100">Width 100%</div>
<div class="w-auto">Width auto</div>
```

3.3.3 Achtergrondkleur

We kunnen elementen enkele standaard achtergrondkleuren geven door middel van ingebouwde classes.

Bijvoorbeeld:

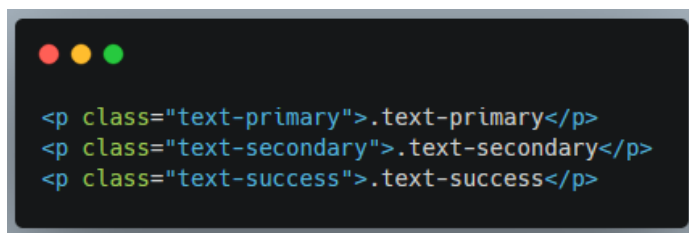


3.3.4 Tekstkleur

We kunnen niet enkel de achtergrondkleur maar ook de tekstkleur aanpassen op een heel gelijkaardige manier.

Bijvoorbeeld:

`.text-primary`
`.text-secondary`
`.text-success`



3.4 Bootstrap Grid

In 2021 is het heel belangrijk dat onze site er goed uit ziet op elk soort scherm. Sites worden weergegeven op computerschermen met verschillende resoluties, smartphones, tablets, TV's, en ga zo maar verder. Dit uitwerken in CSS is een hele uitdaging maar dankzij bootstrap is dit een stuk makkelijker aan de hand van het grid-systeem.

3.4.1 Mobile first

"Mobile first" is een algemene regel binnen webdesign. We ontwikkelen tegenwoordig altijd websites met mobile devices in het achterhoofd. Om ervoor te zorgen dat onze website goed werkt op mobiele devices voegen we steeds de volgende regel toe in de head-tag van onze pagina:

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The code is written in a light blue/cyan color and is as follows:

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
```

Het toevoegen van deze regel is ook belangrijk voor een goede werking van het bootstrap grid-systeem.

3.4.2 De werking van het grid systeem

Het bootstrap grid-systeem deelt een volledige pagina, of een onderdeel ervan, op in meerdere delen.

- Het geheel van deze delen noemen we een container.
- Een container is opgedeeld in meerdere rows.
- Elke row bestaat uit 12 columns.

Bekijk het [voorbeeld](#) om dit in werking te zien.

3.4.3 Oefening op het grid systeem

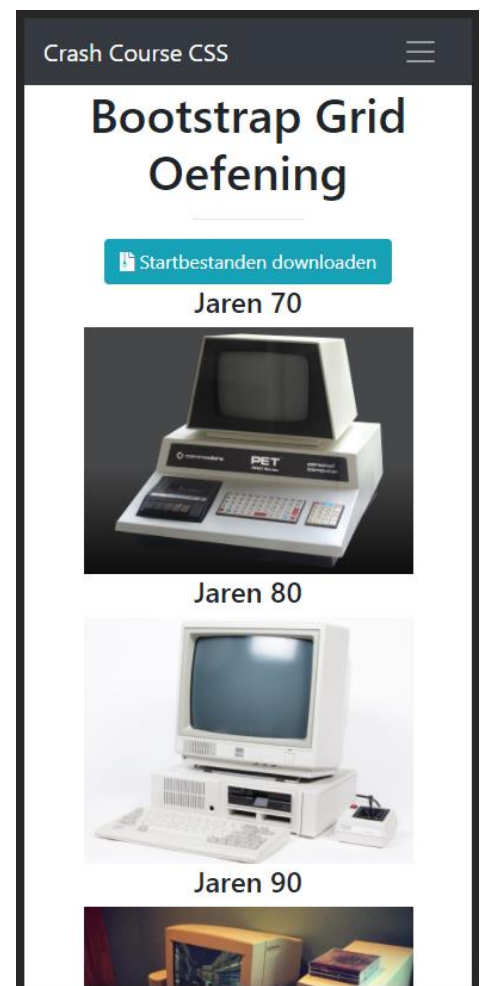
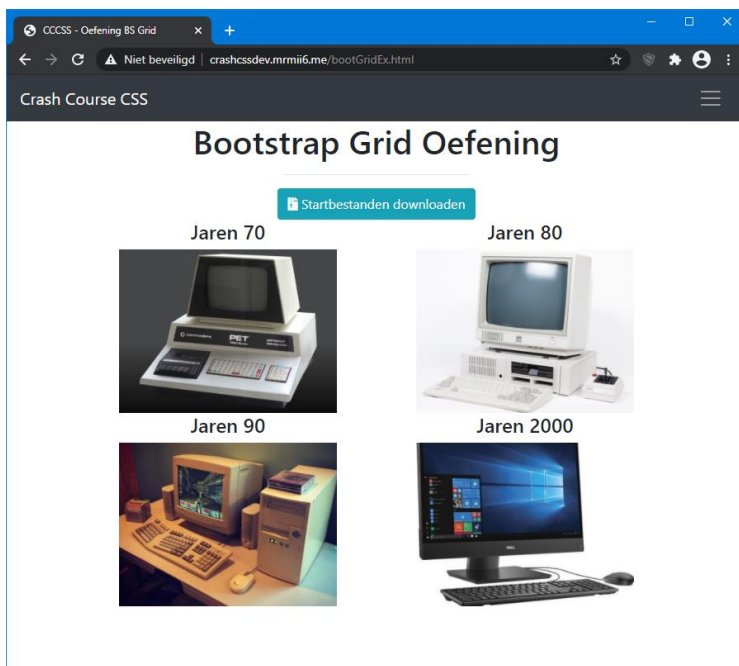
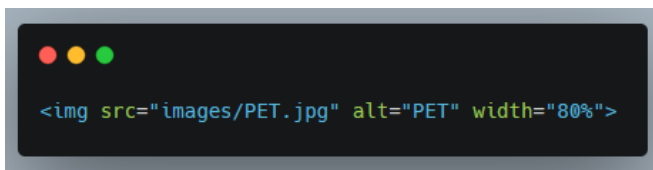
Bij deze oefening is het doel om een variant op het voorbeeld uit het vorige deel te maken. We willen een overzicht krijgen van de geschiedenis van de computer bestaande uit 4 afbeeldingen. Op een klein scherm komen de afbeeldingen onder elkaar te staan. Op een medium of groot scherm komen ze in een grid van 2x2 te staan.

Download de startbestanden en bekijk het voorbeeld [op de site](#).

Hou ook zeker de [bootstrap documentatie](#) bij de hand.

Concrete stappen:

- Bootstrap toevoegen aan de pagina met behulp van een CDN.
- De afbeeldingen toevoegen met een width van 75%
- Een grid opstellen van 2x2 op een medium of groot scherm. En de afbeeldingen onder elkaar tonen op een klein scherm.



3.4 Bootstrap componenten

3.4.4 Knoppen

Bootstrap bevat classes om knoppen een mooie opmaak te geven die je in een groot aantal situaties kan gebruiken.

Het aanmaken van deze knoppen begint altijd met een standaard HTML knop:

```
<button type="button">knoptekst</button>
```

Om de knoppen een bepaalde look te geven voegen we er Bootstrap-classes aan toe.

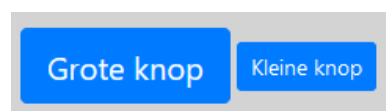
```
<button type="button" class="btn btn-primary">knoptekst</button>
```

Voorbeeld basisstijlen:



```
<p>Accepteerd u de cookies van deze website?</p>
<button type="button" class="btn btn-success">Natuurlijk</button>
<button type="button" class="btn btn-danger">Nope</button>
```

Voorbeeld groottes:



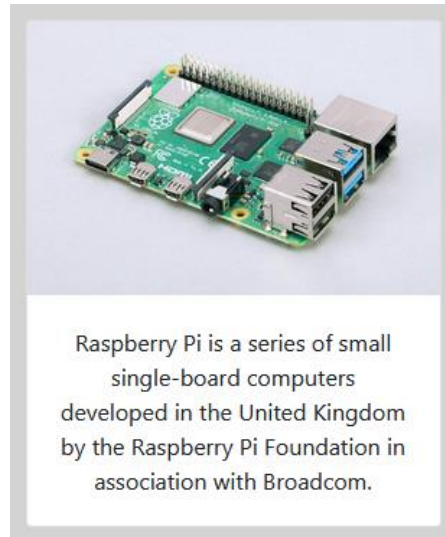
```
<button type="button" class="btn btn-primary btn-lg">Grote knop</button>
<button type="button" class="btn btn-primary btn-sm">Kleine knop</button>
```

3.4.5 Cards

De Bootstrap cards zijn een perfecte manier om informatie te groeperen. Ook zijn ze perfect voor het opmaken van een menu.

Er zijn veel verschillende soorten cards. Met of zonder afbeelding, met of zonder knoppen, verschillende groottes, enzovoort.

Hieronder zie je een eenvoudig voorbeeld:



```
<div class="card m-auto" style="width: 18rem;">
  
  <div class="card-body">
    <p class="card-text">Raspberry Pi is a series of small single-board computers developed in the
      United Kingdom by the Raspberry Pi Foundation in association with Broadcom.</p>
  </div>
</div>
```

3.4.6 Alerts

Alerts worden gebruikt om op een opvallende manier informatie door te geven aan de gebruiker. Deze alerts zullen bijna altijd gegenereerd worden en een taal zoals Javascript of PHP. In deze cursus zullen we ons echter beperken tot de opmaak ervan.

Net zoals bij de knoppen zijn er verschillende alerts voor verschillende situaties.

A simple primary alert—check it out!

A simple secondary alert—check it out!

A simple success alert—check it out!

A simple danger alert—check it out!

Je kan ook alerts toevoegen met een ingebouwde link, zoals je hier kan zien:

Bootstrap is enorm handig! **Bekijk zeker de site** voor meer informatie.

```
<div class="alert alert-info" role="alert">
  Bootstrap is enorm handig!
  <a href="https://getbootstrap.com/" class="alert-link">Bekijk zeker de site</a> voor meer informatie.
</div>
```


3.4.7 Oefening op componenten

Bij deze oefening gaan we een simpele pagina maken met enkele bootstrap componenten. We maken ook gebruik van enkele helper-classes voor de opmaak.

De pagina bestaat uit een alert dat de gebruiker verwelkomt. Vervolgens stellen we de vraag of de gebruikers cookies accepteert met "ja" en "neen" knoppen. Als laatste plaatsen we een card met een afbeelding en wat informatie.

Download de startbestanden en bekijk het voorbeeld [op de site](#).

Hou ook opnieuw de [bootstrap documentatie](#) bij de hand.

