

# CSS

---

## Start van stylesheet

- **external** (apart document)

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

- **internal style block**

```
<style>
p {
}
```

- **inline style**

```
<p style="font-family: Lato; font-size: 20px">.. </p>
```

## Commentaar

```
/* op deze manier schrijf je commentaar */
```

## Pseudoklassen

---

VOOR DE : KOMT JE ELEMENT (bv: **p:first-child**)

- Hoofdelement

```
:root
```

- een element dat het eerste child is van zijn parent-element

```
:first-child
```

- een element dat het laatste child is van zijn parent-element

`:last-child`

- een element waarvan de ouder geen andere child element heeft

`:only-child`

- het eerste element van dat type

`:first-of-type`

- het laatste element van dat type

`:last-of-type`

- het enige element van dat type

`:only-of-type`

- elk zoveelste kinderelement

`:nth-child(n)`

- nu gerekend vanaf het laatste kind

`:nth-last-child(n)`

- elk zoveelste element van een type

`:nth-of-type(n)`

- nu gerekend vanaf laatste element

`:nth-last-of-type(n)`

- leeg element

```
:empty
```

## Toegang tot onderdelen van document waarvoor geen gewone elementen bestaan

- de eerste regel opgemaakte tekst van een element

```
::first-line
```

- de eerste letter

```
::first-letter
```

- voor de inhoud van een element

```
::before
```

- na de inhoud van een element

```
::after
```

## Toestand element

- niet bezocht

```
a:link { }
```

- bezocht

```
a:visited { }
```

- met cursor over gaan

```
a:hover { }
```

- actief

```
a:active { } (weinig gebruikt)
```

- handelingen van de gebruiker

```
:focus (element dat focus heeft)
```

- doel van een element

```
:target
```

## Basisselectors

- Voor selectie van specifieke elementen, gebruik attribuut class of id
- ID is uniek --> **selectie van 1 element**
  - waarde van id attribuut **moet uniek zijn** binnen pagina
  - een id-attribuut kan maar 1 waarde bevatten

```
<div id="zoek">
```

- in css: **naam van id selector begint met #**
- classes zijn niet uniek --> **selectie van meerdere elementen**
  - waarde van class mag onbeperkt voorkomen in pagina
  - element kan meerdere class waarden hebben. Deze worden gescheiden door spatie

```
<div class="widget rechts">
```

- in css: **naam van class selector begint met .**

## Boxmodel

---

(!! eerst wordt de code vernoemd, daaronder de mogelijke waarden !!)

- Ruimte rond inhoud
  - zorgt ervoor dat de rand (border) **niet aan inhoud vasthangt**
  - **ALTIJD transparant**, we zien **achtergrondkleur** van element waarop **padding** wordt toegepast

```
padding:  
  px,em, %  
  padding-top/right/bottom/left
```

- rand rond de padding

```
border:  
  5px solid red (border-width, border-style, border-color)  
  
border-style:  
  dotted, dashed, solid, double, groove, ridge, inset,  
  outset, none, hidden  
  
border-width:  
  px, %, em, rem, thin, medium, thick  
  
border-color:  
  naam van kleur
```

- breedte

```
width:  
  auto (box zo groot dat inhoud erin past)  
  
  lengte / %  
  
  max-content (afmeting van grootste item in box)  
  
  min-content (kleinst mogelijke  
  afmeting zonder overloop te veroorzaken)  
  
min-width / max-width:  
  zelfde eigenschappen als width  
  
min-height / max-height:  
  zelfde eigenschappen als width  
  
none: standaardwaarde
```

- Ingestelde width en height, enkel betrekking op **content**

```
box-sizing: content-box;
```

- Ingestelde width en height hebben betrekking op **content + padding + border**

```
box-sizing: border-box;
```

## Overflow

- Treedt op indien content te groot wordt voor box
  - **KAN ENKEL:** als breedte en hoogte is ingesteld

```
overflow:  
    scroll (zowel horizontaal als verticaal)  
  
    hidden (overige gedeelte niet zichtbaar)  
  
    auto (alleen verticale scroll)  
  
    visible (default)
```

## Display

- Verdeling van boxen over pagina: lay-out
- Delen naast elkaar plaatsen of onder elkaar --> door type box

```
display:  
    display-outside (gedrag tov andere boxen)  
        block, inline, run-in  
  
    display-inside (gedrag van afstammelingen van box)  
        flow, flow-root, table, flex, grid, list-item  
  
    display-box (parent maakt geen box aan)  
        none(children maken ook geen box aan, niet gerenderd)  
  
        contents(children maken boxen aan, vb grid en flex)  
  
    display-legacy (waarden kunnen ook gekoppeld worden)  
        inline-block, inline-grid, inline-flex
```

## Visibility

- Elementen verbergen

- worden **niet weergegeven** in browser, wel **ruimte voor box gereserveerd op scherm**

```
visibility:  
  visible (default)  
  hidden
```

## Positionering

- **Position en outline (offset)**

```
position:  
  static (default)  
  
  relative (relatief ten opzichte van normale positie)  
    left, right, top, bottom  
  
  absolute  
  
  fixed (staat vast tijdens scrollen)  
  
outline: (voegt ruimte tussen omtrek en rand van element)  
  4x solid red  
  
outline-offset:  
  px
```

- **Float en clear**

- haal je blok uit normale flow en laat inhoud van onderliggende blokken erom heen lopen

```
float:  
  right, left, none, initial
```

Wil je niet dat de blokken eromheen lopen?

```
clear:  
  left, right, none, both(links en rechts)
```

--> werd veel gebruikt vroeger, nu meer **flex** en **grid**

## Media queries

- **doel:** opmaak koppelen aan eigenschappen van device

- **media queries bestaan uit:**

- apparaat type (all, screen, print, speech)
- media kenmerken (breedte, orientatie): **breakpoints** een blok { } van stijlregels (geldig voor gedeclareerde apparaat type en media kenmerken)

```
@media apparaat and (media feature) {  
  selector {  
    declaraties;  
  }  
}
```

**Voorbeeld:**

- Indien scherm minimaal 1024px breed is --> twee kolommen met 66% en 34% van breedte van het scherm

```
@media screen and (min-width: 1024px) {  
  main {  
    display: grid;  
    grid-template-columns: 66% 34%;  
  }  
}
```

## Viewport

- Viewport is onafhankelijk van scherm
  - **op mobiele toestellen:** viewport is hetzelfde, ruimte die wordt gebruikt om webpagina te tonen, ongeacht de grootte van het scherm

ZOEK NAAR VIEWPORTS:

- <https://mediag.com/blog/popular-screen-resolutions-designing-for-all/>

Viewport toevoegen --> width gelijk aan device width

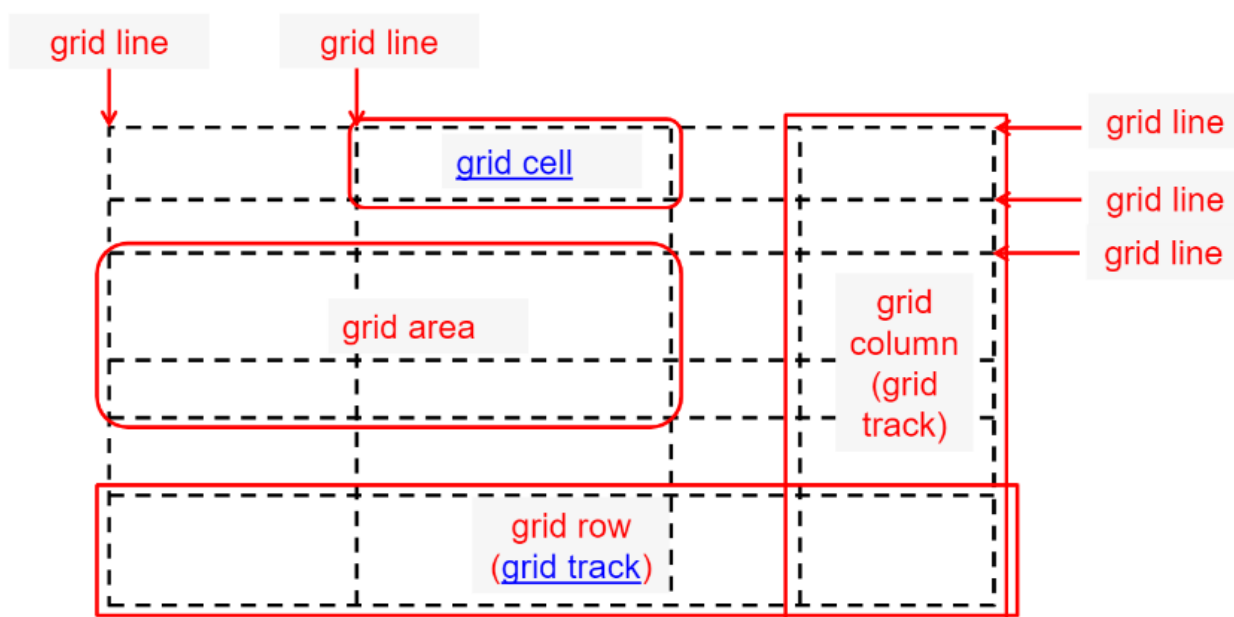
```
<meta name="viewport" content="width=device-width", initial-scale = 1.0, user-scalable=yes">
```

## Multi-column



```
/* gewenste aantal kolommen */  
column-count:  
    number, auto, initial  
  
/* gewenste breedte */  
column-width:  
    auto, rem, %, px  
  
/* overspannen van kolommen: none/all */  
column-span:  
    none, all  
  
/* breedte van ruimte tussen kolommen */  
column-gap:  
    px, normal  
  
/* lijn tussen kolommen (color/style/width) */  
column-rule:  
    4px double #ff00ff;  
  
/* opvullen van de kolommen - gebalanceerd of niet */  
column-fill  
    auto, balance
```

## Grid lay-out



- Creëer een grid container element

```
display:
  grid;
display:
  inline-grid;
```

=> grid is een block-level-element en inline-grid is een inline-level-element

- Definieer **grid en plaats grid items** op de grid
  - grid zelf, aantal rijen en kolommen + breedte/hoogte, definieer je via:

```
grid-template-rows:
  100px 300px

grid-template-columns:
  none, auto, max-content, min-content, px

grid-template-areas:
  'header header header header'
  'menu main main right'
  'menu footer footer footer';
```

## Repeat function

- track list herhalen
  - bv:

```
grid-template columns: 1fr 1fr 1fr;
```

korter schrijven als:

```
grid-template columns: repeat(3, 1fr);
```

Een meer uitgebreid voorbeeld:

```
grid-template columns: 20px repeat(4, 1fr 2fr);
```

is hetzelfde als

```
grid-template columns: 20px 1fr 2fr 1fr 2fr 1fr 2fr 1fr 2fr;
```

- auto-fill of auto-fit

```
grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));  
/* geen vaste kolommen maken, browser laten doen (autofill) */
```

==> zelfde functie, behalve als lege tracks zijn

Link: <https://gridbyexample.com>

## Track overspannen: span

- In plaats van 'end' lijnnummer op te geven, aantal tracks opgeven dat je wilt overspannen

```
grid-row-start: 1; grid-row-end: 4;  
/* korter schrijven als */  
grid-row-start: 1; grid-row-end: span 3;
```

## Ruimte tussen tracks

- ruimte tussen tracks creëren via **gutters**

```
column-gap:  
    px, %, em  
row-gap:  
    px, %, em  
gap:  
    px, %, em
```

## Mogelijke afmetingen instellen voor tracks

- Elke track heeft **base-size** en **growth limit**(maximum size)
- Als track size: 200px is base-size en growth limit beide 200px Als track-size: minmax(100px, 200px), base-size is 100px en growth-limit 200px
- lengtewaarde: px - % - fr
- keywords:

```
max-content  
min-content  
minmax(min, max)
```

```
fit-content(value)
```

```
auto
```

## Expliciet en impliciet raster

- **Expliciet raster:**
  - columns en rijen definiëren met **grid-template-\*** eigenschap
- **Impliciet raster**
  - columns en rijen standaard 'auto-sized', aanpassen via:

```
grid-auto-rows
```

```
grid-auto-columns
```

## Plaatsen van elementen op raster

- Rij

```
grid-row-start:  
  2 (geef aan waar je wilt starten), auto  
  
grid-row-end:  
  span 3 (over 3 rijen zetten)  
  
grid-row:  
  1 / span 2 (start op rij 1, over 2 rijen zetten)
```

- Kolommen

```
grid-column-start:  
  2 (geef aan waar je wilt starten), auto  
  
grid-column-end:  
  span 3 (over 3 rijen zetten)  
  
grid-column:  
  1 / span 2 (start op rij 1, over 2 rijen zetten)
```

- Rij en kolommen samen:

```
grid-area:
  2 / 1 / span 2 / span 3 (start op rij 2 column 1, over 2 rijen
  zetten en 3 kolommen)
```

## Flexbox

---

- Uitlijnen van grid in zijn container

```
justify-content:
  start, center, space-between, space-around, space-evenly

align-content:
  start, center, space-between, space-around
```

- Uitlijnen van item in een grid-cell/track/area

```
/* Alle items in grid(flexbox), declaratie in container element */
justify-items:
  flex-start, flex-end, center, space-between,..

align-items:
  stretch, center, flex-start, flex-end, baseline,..

/* 1 item, declaratie in grid-item */
justify-self:
  auto, flex-start, flex-end, center, space-between,..

align-self:
  auto, stretch, center, flex-start, flex-end, baseline,..
```

- Om items uit te lijnen (verticaal): gebruik **align-items** en **align-self**
- Om items uit te lijnen (horizontaal): gebruik **justify-items** en **justify-self**

## Flex

- Verdeelt inhoud in 1 richting
- Creëer een flex container-element

```
display:
  flex

display:
  inline-flex
```

## Richting items

- Flex containers: **main axis** en **cross axis**
  - standaard:
    - main axis van links naar rechts
    - cross axis van boven naar onder
    - **AANPASSEN** met:

```
flex-direction:  
    row, row-reverse, column, column-reverse
```

- container wordt opgevuld langs main axis
- Overloop van inhoud regelen:

```
flex-wrap:  
    nowrap, wrap, wrap-reverse, initial
```

- Shorthand(direction wrap)

```
/* flex-flow: flex-direction flexwrap*/  
flex-flow:  
    row-reverse wrap;
```

## Uitlijnen binnen container

(herhaling van voorgaand stukje)

```
/* langs main-axis */  
justify-content:  
    start, center, space-between,..  
  
/* langs cross-axis */  
align-content:  
    start, center, space-between,..
```

## Uitlijnen items binnen container

```
/* uitlijnen alle items langs cross-axis */  
align-items:  
    stretch, center, start, end, ..  
  
/* uitlijnen individueel flex-item langs cross-axis */
```

```
align-self:
  stretch, center, start, end, ..
```

## Volgorde items - grootte

- Volgorde wijzigen (**order**)

```
div#myRedDIV {order: 2;}
div#myBlueDIV {order: 4;}
```

- Initiële waarde van flex-item

```
flex-basis:
  px, auto
```

- Flex-items kunnen groeien of krimpen als er extra plaats is in container (**flex-grow en flex-shrink**)

```
div:nth-of-type(1) {flex-grow: 1;}
div:nth-of-type(2) {flex-shrink: 3;}
```

- Waarde: getal
  - **0**: niet groeien/krimpen
  - **positief**: groei/krimp in verhouding met andere items
- Short: **flex**

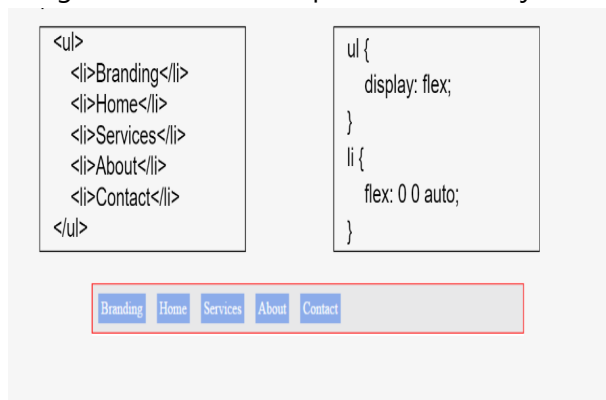
```
flex-grow: 2;      )
flex-shrink: 1;    ) = flex: 2 1 auto;
flex-basis: auto;  )
```

## Absolute en relatieve flex

- **Absolute flex items**: ingenomen ruimte enkel bepaald door Flexbox
  - flex: 1 1 0;
  - item mag groeien, mag verkleinen, en er wordt geen ruimte bepaald op voorhand
- **Relative flex items**: ingenomen ruimte enkel bepaald door grootte inhoud
  - flex: 1 1 auto;
  - item mag groeien, mag verkleinen, maar ruimte wordt eerst automatisch bepaald door inhoud

## Flexbox en margin: auto

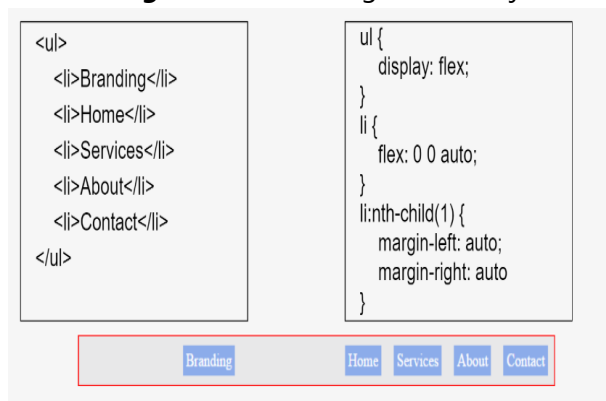
- margin: **auto** instellen op een item zal vrije ruimte "verplaatsen"



- Rechtermargin** instellen zorgt dat daar de vrije ruimte geplaatst wordt



- Beide margins** instellen zorgt dat de vrije ruimte langs beide kanten verspreid wordt



## Eigenschappen met lettertype (font)

- geprioritiseerde lijst van lettertypes

```
font-family:
  naam lettertype
```

- de grootte van het lettertype



```
font-size:  
  px, %, large
```

- om tekstgrootte gelijk te houden wanneer wordt teruggevallen op 2de keus in font-family

```
font-size-adjust:  
  0.58
```

- regelafstand, hoeveelheid wit tussen regels tekst

```
line-height:  
  %, normal, 1.6
```

- de vetheid, 100-900 of normal(400), bold(700), bolder, lighter

```
font-weight:  
  waardes 100-900, bolder, lighter
```

- weergave tekst

```
font-style:  
  normal, italic, oblique(schuin)
```

- kleinkapitaal (weinig gebruikt)

```
font-variant:  
  normal, small-caps
```

- uitrekken of indrukken

```
font-stretch:  
  expanded, condensed
```

- verzameleigenschap font-style font-variant font-weight ,...

```
font:  
  italic normal bold 12px Georgie, serif;
```

# Eigenschappen voor tekst

---

- kleur

```
color:  
  red, #00f00, rgb(0,0,255)
```

- uitlijnen

```
text-align:  
  left, right, center, justify
```

- uitlijnen laatste regel

```
text-align-last:  
  left, right, center, justify,
```

- lijneffect

```
text-decoration:  
  red 5px underline  
  
text-decoration-line:  
  none, underline, overline, line-through, initial, inherit  
  
text-decoration-style:  
  solid, wavy, double  
  
text-decoration-color:  
  red
```

- inspringen van de eerste regel van een tekstblok

```
text-indent:  
  px, em, %
```

- horizontale of verticale tekst

```
text-orientation:
  mixed, upright, sideways
```

- overlopende tekst, bepaald welk visueel effect gebruiker krijgt als er meer tekst is dan hij ziet

```
text-overflow: ellipsis

(hoort bij)
white-space:
  nowrap;
overflow:
  hidden
```

- schaduw (horizontale verschuiving, vertical verschuiving, vervaging,..)

```
text-shadow
  2px 2px #FF0000
```

- hoofdletters, kleine letters

```
text-transform
  uppercase, lowercase, capitalize, none
```

- witruimte al dan niet behouden

```
white-space
  pre, normal, nowrap
```

- ruimte tussen letters

```
letter-spacing
  px
```

- ruimte tussen woorden

```
word-spacing
  px
```

- automatisch afbreken

```
word-break:
  break-word
```

- afbreken van een woord toegestaan

```
overflow-wrap
  anywhere, normal, break-word
```

## Opmaken van lijsten

- opsommingsteken

```
list-style-type:
  circle, square, upper-roman, ..
```

<b>disc</b> <ul style="list-style-type: none"> <li>• item 1</li> <li>• item 2</li> <li>• item 3</li> </ul>	<b>circle</b> <ul style="list-style-type: none"> <li>◦ item 1</li> <li>◦ item 2</li> <li>◦ item 3</li> </ul>	<b>square</b> <ul style="list-style-type: none"> <li>▪ item 1</li> <li>▪ item 2</li> <li>▪ item 3</li> </ul>
<b>decimal</b> <ol style="list-style-type: none"> <li>1. item 1</li> <li>2. item 2</li> <li>3. item 3</li> </ol>	<b>decimal-leading-zero</b> <ol style="list-style-type: none"> <li>01. item 1</li> <li>02. item 2</li> <li>03. item 3</li> </ol>	<b>upper-alpha</b> <ol style="list-style-type: none"> <li>A. item 1</li> <li>B. item 2</li> <li>C. item 3</li> </ol>
<b>lower-alpha</b> <ol style="list-style-type: none"> <li>a. item 1</li> <li>b. item 2</li> <li>c. item 3</li> </ol>	<b>upper-roman</b> <ol style="list-style-type: none"> <li>I. item 1</li> <li>II. item 2</li> <li>III. item 3</li> </ol>	<b>lower-roman</b> <ol style="list-style-type: none"> <li>i. item 1</li> <li>ii. item 2</li> <li>iii. item 3</li> </ol>

- afbeelding als opsommingsteken

```
list-style-image:
  url('naam.gif');
```

- plaatsing opsommingsteken

```
outside, inside
```

- shorthand

```
list-style:
  square inside url("naam.gif");
```

## Kleuren

---

- 4 manieren om te schrijven
  - **rgb-code:** rgb(a)(145,233,12)
  - **verkorte hex-code:** #1e85af
  - **keyword:** green, red
  - **hsl-code:** hsl(a)
- Transparantie

```
opacity:
  1.5
```

- mengen van achtergrondsafbeeldingen en achtergrondskleur element

```
background-blend-mode:
  lighten, color, darken, normal, ..
```

- inhoud van element mengen met zijn achtergrond en inhoud ouderelementen

```
mix-blend-mode:
  lighten, color, darken, normal,..
```

Color picker: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Colors/Color\\_picker\\_tool](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Colors/Color_picker_tool)

## Achtergrond

- Elke box van element heeft achtergrond (**default transparant**)
- instellen achtergrondkleur:

```
background-color:
  red, #1e85af
```

- instellen afbeelding

```
background-image:  
  url("paper.gif");
```

- herhalen van afbeelding tot alle ruimte in blok is gevuld

```
background-repeat:  
  repeat-y, repeat-x
```

- afbeelding schuift standaard mee met de rest van pagina, kan hem ook vastzetten

```
background-attachment:  
  fixed, scroll, local, initial, inherit
```

- plaatsing

```
background-position:  
  left, center, left-top, right,...
```

- weergave overloop:

```
background-clip:  
  border-box, padding-box, content-box
```

- plaatsing

```
background-origin:  
  border-box, padding-box, content-box
```

- schalen

```
background-size  
  px, em, %, auto
```

- **in 1 property**

```
background:  
  red url("img_tree.gif") no-repeat fixed center;
```

## Kleurverlopen

- lineair verloop

```
background-image:  
  linear-gradient(red, yellow)
```

- radial-gradient

```
background-image:  
  radial-gradient(red, yellow)
```

## Afgeronde hoeken

- Afronden van randen van element
- alle hoeken

```
/* 1 waarde, alle 4 hoeken */  
  
border-radius  
  15px  
  
/* 2 waarden, 1ste: top-left & bottomright, 2de: top-right &  
bottom-left */  
  
border-radius  
  15px 50px  
  
/* 3 waarden, 1ste: top-left , 2de: top-right &  
bottom-left, 3de: bottom-right*/  
  
border-radius  
  15px 50px  
  
/* 4 waarden, 1ste: top-left , 2de: top-right, 3de  
bottom-right, 4de: bottom-left*/  
  
border-radius  
  15px 50px
```

- Elke hoek afzonderlijk instellen

```
border-top-left-radius:
    px

border-top-right-radius:
    px

border-bottom-right-radius
    px

border-bottom-left-radius
    px
```

- Generator: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Background\\_and\\_Borders/Border-radius\\_generator](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Background_and_Borders/Border-radius_generator)

## Randafbeeldingen

---

- Beeld voor de rand van een element
- shorthand code

```
border-image:
    url(border.png) 30 round;
```

```
/* afbeelding */
border-image-source:
    url(border.png)

/* opdelen van beeld in regio's */
border-image-slice:
    %, 30(getallen)

/* breedte */
border-image-width:
    px, %

/* afstand waarmee randafbeelding wordt
uitgezet tov zijn randkader */
border-image-outset:
    px, %

/* hoe voelt het beeld de rand? */
border-image-repeat:
    repeat, round, stretch, space, initial
```

## Afbeeldingen uitsnijden



- definieert gebied/vorm en alles daarbuiten wordt weggeknipt

```
clip-path:  
  polygon(50%, 0, 100%, 50%, 50%, 100%, 0 50%)
```



## Beeldfilters

---

- toepassen van grafische effecten op afbeeldingen (blur, color shift)

```
filter:  
  grayscale(%)  
  blur(px)  
  brightness(%)  
  ...
```

## Schaduw

---

- verschuiven in x-of y-richting
- Generator: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Background\\_and\\_Borders/Box-shadow\\_generator](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Background_and_Borders/Box-shadow_generator)

## Functies

---

### Rekenen met calc

- +, -, \*, / in combinatie met alle eenheden

### CSS variabelen: de functie var()

- property value waar je naam aan toekent
- **globale variabelen** :root (kleurenschema van site)

```
:root {  
  --main-bg-color: pink;  
}  
  
body {  
  background-color: var(--main-bg-color);  
}
```

- **scoped variabelen:** koppelen aan id, class

```
#firstParagraph {  
  background-color: var(--first-color);  
  color: var(--second-color);  
}
```

## Bootstrap & SASS

---

- OPEN DE WORKSHOPS, er staat een duidelijke handleiding

## Overzicht handige links

---

Alle mogelijke codes:

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

Auto-fill / auto-fit:

- <https://gridbyexample.com>

Borders aanpassen:

- [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Background\\_and\\_Borders/Border-radius\\_generator](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Background_and_Borders/Border-radius_generator)

Color picker:

- [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Colors/Color\\_picker\\_tool](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Colors/Color_picker_tool)

W3schools:

- <https://www.w3schools.com/cssref/>

grid MDN:

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