

# Static Web Development

# Inleiding

**html:** markup-taal die inhoud structureert en definieert

- semantisch (inhoud & betekenis, niet design)

**css:** stijlblad met een opsomming van eigenschappen (*properties*)

- definieert opmaak

## HTML

HTML 1.0: ontwikkeld door Tim Berners Lee in 1991

## HTML 4.01: in 1999

XHTML (eXtensible markup): identieke content-elementen, andere syntax (volgens XML-regels)

- begin van semantisch web met CSS
- W3C Web Standard (Best Practices)

2004:

- W3C wilt XHTML → beseft in 2006 dat het niet werkt
- WHATWG wilt opvolger van HTML → HTML5

2009: idee voor XHTML2 wordt geschrapt, WHATWG en W3C werken samen aan HTML5

2014: HTML5 is nog niet klaar, maar we kunnen het wel al gebruiken (backwards compatible)

CSS

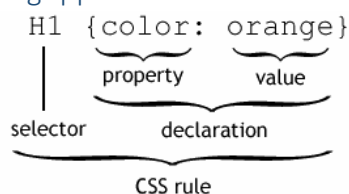
= Cascading Style Sheets

- sinds introductie in 1994 → steeds meer scheiding tussen markup (html) en design (css)
- browserimplementatie is lang een struikelblok geweest

## CSS toepassen in een pagina

- **inline** (als attribuut bij een element)  
`<tag style="...">voorbeeld</tag>`
- **embedded** (in de head van een pagina)  
`<style type="text/css">`  
...  
`</style>`
- **extern** (link in de head naar een afzonderlijk CSS-bestand)  
`<link rel="stylesheet" type="text/css" href="styles.css" />`

## Begrippen



## Best Practices

- Hou je code leesbaar:
  - Schrijf elementen in kleine letters (lowercase)
  - Sluit container-tags steeds af
  - Gebruik quotes (" of ') rond attributen
- Gebruik externe i.p.v. interne/embedded stylesheets

## HTML

### Basis-opbouw

<code>&lt;!doctype html&gt;</code>	= doctype
<code>&lt;html lang="nl"&gt;</code>	= html-element + taal
<code>&lt;head&gt;</code>	
<code>&lt;meta charset="utf-8"&gt;</code>	= character encoding
<code>&lt;meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1"&gt;</code>	= viewport
<code>&lt;title&gt;Title&lt;/title&gt;</code>	= paginatitel
<code>&lt;link rel="stylesheet" href="styles.css"&gt;</code>	= css-stylesheet linken
<code>&lt;script type="text/javascript" src="main.js"&gt;&lt;/script&gt;</code>	= javascript linken
<code>&lt;link rel="icon" type="image/x-icon" href="favicon.ico"&gt;</code>	= favicon linken
<code>&lt;/head&gt;</code>	
<code>&lt;body&gt;&lt;/body&gt;</code>	= paginacontent
<code>&lt;/html&gt;</code>	

### Block-level/inline-elementen

#### Block-level: structuurelementen

- zal automatisch zijn parent-container vullen als geen width is gespecificeerd
- zal automatisch verticaal vergroten om alle child elements weer te geven
- kan *margins* en/of *padding* hebben
- zal automatisch onder vorige elementen geplaatst worden (als niet gefloat/gepositioned is)
- bv. `<p>`, `<div>`, `<header>`, `<nav>`, `<ul>`, `<li>`, `<h1>`

#### Inline: tekstelementen

- vloeit mee met tekst
- negeert *top/bottom margins*, maar niet *left/right margins* en *padding*
- negeert *width* en *height properties*
- verandert in een block-level element als het gefloat wordt
- bv. `<a>`, `<span>`, `<b>`, `<em>`, `<i>`

#### Inline-block

- vloeit mee met tekst
- kan *width* en *height properties* hebben
- kan *top/bottom margins* en *padding*s hebben
- gebruikt om verschillende block-level elementen op een lijn te plaatsen, zonder float te gebruiken
- werkt niet in IE6/IE7

## Basiselementen

- Heading `<h1>Title</h1>` → `<h6>Titel</h6>`
- Paragraaf `<p>inhoud</p>`
- Nieuwe lijn `<br>`
- Horizontale lijn `<hr>`
- Blockquote (citaat) `<blockquote>Citaat</blockquote>`
- Hyperlink `<a href="http://www.google.com">Google</a>`
  - attribuut `target="_blank"` opent de link in een nieuw venster
  - attribuut `href="#content"` verwijst naar een sectie van de pagina met een bep. ID
  - attribuut `href="vb.html#content"` verwijst naar een anker in een andere pagina

## Inline-elementen

- Afkorting (*abbreviation*) `abbr`
- Vetgedrukte tekst `b`
- Titel van een werk `cite`
- Benadrukte tekst `em`
- Schuingedrukte tekst `i`
- Gehighlightte tekst `mark`
- Kleinere tekst `small`
- Belangrijke tekst `strong`
- Sub/Superscript-tekst `sub` / `sup`

## Lijsten

- Geordende lijst `<ol> </ol>`
- Ongeordende lijst `<ul> </ul>`
- Lijstitems `<li> </li>`
- Omschrijvingslijst `<dl> </dl>`
- Definieert woorden `<dt> </dt>`
- Omschrijft woorden `<dd> </dd>`

## Speciale karakters

- Commentaar `<!-- commentaar -->`
- Entiteiten o.a. `&euro` (€), `&copy` (©)

## Structuurelementen

- Betekenisloos element, dient om elementen te bundelen met oog op stijl `<div> </div>`
- Kop van de pagina of van een sectie van de pagina `<header> </header>`
- Headers groeperen `<hgroup> </hgroup>`
- Voettekst van de pagina `<footer> </footer>`
- Navigatiesectie v.d. pagina `<nav> </nav>`
- Main content van de pagina `<main> </main>`
- Inline equivalent van `<div>` `<span> </span>`
- Secties in een document `<section> </section>`
- Onafhankelijke inhoud `<article> </article>`
- Extra informatie `<aside> </aside>`

## Tabellen

- Tabel: `table`
- Caption: `caption`
- Table header: `thead`
- Table body: `tbody`
- Table footer: `tfoot`
- Table heading cell: `th`
- Table data cell: `td`
- Cellen die verschillende kolommen overlappen: `<th colspan="2"></th>`
- Cellen die verschillende rijen overlappen: `<th rowspan="2"></th>`
- td-elementen linken aan th-elementen: `<th id="name">Name</th>`  
`<td headers="name">John Doe</td>`

## CSS

- Ruimte tussen cellen: `border-spacing`
- Verbonden borders: `border-collapse: separate | collapse`

## Formulieren

- Form: `form`
- Invoerveld: `input`
  - ↳ `<input type="text">` Gewone tekst
  - ↳ `<input type="button">` Knop
  - ↳ `<input type="checkbox">` Checkbox (meerdere tegelijk mogelijk)
  - ↳ `<input type="radio">` Radio button (slechts één tegelijk mogelijk)
  - ↳ `<input type="email">` E-mailadres
  - ↳ `<input type="submit">` "Verzenden"-knop
  - ↳ `<input type="reset">` Reset-knop
  - ↳ `<input type="url">` URL
  - ↳ `<input type="hidden">` Verborgen invoerveld
  - ↳ `<input type="password">` Wachtwoord
  - ↳ `<input type="select">` Drop-down keuzemenu
    - `<option></option>` Verschillende opties in het keuzemenu
  - ↳ `<input placeholder="Naam">` Placeholder die verdwijnt bij typen
  - ↳ `<input required>` Invoerveld moet verplicht ingevuld worden
- Groot (multi-line) invoerveld: `textarea`
  - ↳ `<textarea rows="4">` Aantal rijen
  - ↳ `<textarea cols="50">` Aantal kolommen
- Label voor invoerveld: `label`
  - ↳ Label binden aan een input-element: `<label for="male"> </label>`  
`<input type="radio" id="male">`
  - ↳ Label binden aan een andere form: `<label form="sex" for="male"> </label>`  
`<input form="sex" type="radio" id="male">`
- Invoervelden groeperen: `fieldset`
  - ↳ Bijschrift bij een fieldset: `<legend>Personalia:</legend>`

## Afbeeldingen

### Formaten

**JPEG:** geschikt voor foto's

- ↳ lossy (verlies kwaliteit)

**GIF:** geschikt voor logo's, icons, bewegende beelden

- ↳ lossless (geen verlies kwaliteit)
- ↳ 256 kleuren
- ↳ transparantie (0 of 1, ≠ alpha-kanaal)

**PNG:** geschikt voor logo's, icons

- ↳ lossless
- ↳ miljoenen kleuren
- ↳ transparantie (alpha-kanaal)

### Inline

= in de tekst, maakt deel uit van content → HTML

```

```

- `src` = bronbestand (absoluut/relatief pad)
- `alt` = alternatieve tekst
- `height` = hoogte van de afbeelding
- `width` = breedte van de afbeelding

```
<figure>
  
  <figcaption>Wolf</figcaption>
</figure>
```

- `figcaption` = bijschrift bij de afbeelding

### Responsive

```
<div class="bgimage" role="img" aria-label="beschrijving"></div>
```

```
.bgimage {
  background: url(img/banner.jpg) no-repeat center;
  background-size: cover;
}
```

```
@media screen and (max-width: 530px) {
  .bgimage {
    background-image: url(img/banner-small.jpg);
  }
}
```

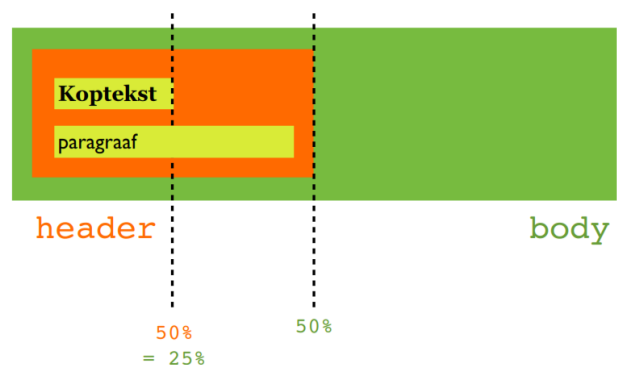
## CSS

### Selectoren

- <code>.class</code>	Klasse
- <code>#id</code>	Id
- <code>element</code>	Bepaald element
- <code>vb, vb</code>	Verschillende elementen tegelijk
- <code>vb vb</code>	Elementen binnen andere elementen
- <code>vb1 &gt; vb2</code>	Alle elementen vb2 met als <i>parent</i> vb1
- <code>vb1 + vb2</code>	Alle elementen vb2 die direct achter vb1 elementen staan
- <code>vb1 ~ vb2</code>	Alle elementen vb2 waar een vb1 element aan voorafgaat
- <code>[attribute]</code>	Alle elementen met een bepaald attribuut
- <code>[attribute="value"]</code>	Alle elementen met een bepaald attribuut en waarde
- <code>vb::after</code>	Plaats content na elk vb element
- <code>vb::before</code>	Plaats content voor elk vb element
- <code>vb::first-letter</code>	De eerste letter van alle vb elementen
- <code>vb-first-line</code>	De eerste regel van alle vb elementen
- <code>:not(vb)</code>	Alle elementen behalve vb elementen
- <code>:first-child</code>	Elk element dat de eerste <i>child</i> is van zijn <i>parent</i>
- <code>:nth-child(n)</code>	Elk element dat een zoveelste <i>child</i> is van zijn <i>parent</i>
↳ <code>:nth-child(odd)</code>	Elk oneven element dat een <i>child</i> is van zijn <i>parent</i>
↳ <code>:nth-child(even)</code>	Elk even element dat een <i>child</i> is van zijn <i>parent</i>
- <code>:first-of-type</code>	Elk element dat de eerste <i>child</i> van een bep. type van zijn <i>parent</i> is
- <code>:nth-of-type(n)</code>	Elk element dat een zoveelste <i>child</i> van een bep. type van zijn <i>parent</i> is
↳ <code>:nth-of-type(odd)</code>	Elk oneven element van een bep. type dat een <i>child</i> is van zijn <i>parent</i>
↳ <code>:nth-of-type(even)</code>	Elk even element van een bep. type dat een <i>child</i> is van zijn <i>parent</i>

### Eenheden om grootte uit te drukken

- **em**
  - ↳ schaaft mee met de browser
  - ↳ 1 em = huidige font size
  - ↳ 2 em = dubbele font size
- **px** (scherpixels)
  - ↳ niet goed te herschalen
- **pt** (punt/point)
  - ↳ gebruikt in print media (bv. print.css)
  - ↳ is net als px niet goed te herschalen
- **%** (percent)
  - ↳ werkt zoals em, maar toch verschillen
  - ↳ bij fonts → 1em = 100%
  - ↳ bij block elements → 100% = breedte v.h. *parent element*



### Kleuren

- Kleurnaam	<code>p {color: red}</code>
- Hexadecimaal	<code>p {color: #FF0000}</code>
- Shorthand hex	<code>p {color: #F00}</code>
- Decimaal	<code>p {color: rgb(255, 0, 0)}</code>
- Dec. met α-kanaal	<code>p {color: rgba(255, 0, 0, 0.5)}</code>

- HSL `p {color: hsl(0, 100%, 50%)}`
- HSL met  $\alpha$ -kanaal `p {color: hsla(0, 100%, 50%, 0.5)}`

## Achtergrond

= achtergrond geven aan een element

- Achtergrondkleur `background-color: yellow | #00ff00 | rgb(255,0,255)`
  - ↳ Reikwijdte `background-clip: border-box|padding-box|content-box`
- Achtergrondaafbeelding `background-image: url("paper.gif")`
  - ↳ Startpositie `background-position: right top | 25% 50% | 20px 40px`
  - ↳ Herhaling `background-repeat: repeat | repeat-x | repeat-y | no-repeat`
  - ↳ Afmetingen `background-size: cover | contain | 50% 40% | 500px 200px`
  - ↳ Beginpunt `background-origin: padding-box|border-box|content-box`
- Gradient
  - ↳ Lineair `background: linear-gradient(angle, color1, color2);`
  - ↳ Lineair (herhalen) `background: repeating-linear-gradient`
  - ↳ Radiaal `background: radial-gradient(ellipse|circle, start-color, ..., last-color);`
  - ↳ Radiaal (herhalen) `background: repeating-radial-gradient`

## Meerdere achtergronden

```
background-image: url(images/sheep.png), url(images/bg1.png);
background-position: left, top right;
```

OF

```
background:
  url(img/sheep.png) no-repeat right,
  url(img/bg.jpg) repeat-x top;
```

## Lijsticoontjes

- `list-style-type` **property**
  - **unordered lists:** none, disc, circle, square
  - **ordered lists:** upper-alpha, lower-alpha, decimal, lower-roman
- `list-style-image: url(img/list-icon.png)` (eigen afbeeldingen)
- `list-style-position: inside | outside` (uitlopen onder icon | standaarduitlijning)

## Link states

Een link heeft verschillende 'states'

- `a:link` een niet bezochte link
- `a:visited` een bezochte link
- `a:hover` een link waar de muis boven staat
- `a:active` een link die aangeklikt wordt
- `a:focus` een link met cursor-focus ( met de tab-toets)

## Tekst

- Font instellen      `font-family: Arial`
- Font stack          `font-family: Arial, "Helvetica Neue", sans-serif`
- Font size            `font-size: 1.4em | 2% | 15px`
- Vette tekst         `font-weight: normal | bold | 100 → 900`
- Schuine tekst       `font-style: normal | italic | oblique`
- Hoofdlettergebruik `text-transform: none | capitalize | uppercase | lowercase`
- Regelhoogte        `line-height: 200% | 2em`
- Uitlijning          `text-align: left | right | center | justify (uitvullen)`
- Inspringing        `text-indent: 2em | 4px | 4pt | 10%`
- Schaduw            `text-shadow: horiz.-offset | vert.-offset | blur | color`  
                          `text-shadow: 0.2em 0.2em 0.3em #666`
  
- Custom font      

```
@font-face {
                        font-family: 'icomoon';
                        src: url('fonts/icomoon.eot');
                        src: url('fonts/icomoon.eot') format('embedded-opentype'),
                        url('fonts/icomoon.woff') format('woff'),
                        url('fonts/icomoon.ttf') format('truetype'),
                        url('fonts/icomoon.svg') format('svg');
                      }
```
- Google Fonts      `<link rel="stylesheet"
 href="http://fonts.googleapis.com/css?family=Fontnaam">`

## Sprites

= verschillende icons in één afbeelding

Voordeel :

- Browsers max 3 synchrone downloads / server
- DUS: minder afbeeldingen = véél kortere laadtijd

```
div {
  width: 16px;           // Breedte van één icoontje
  height: 16px;         // Hoogte van één icoontje
  background: url(img/spritesheet.png); // Spritesheet
  background-position: 32px 16px; // Positie van het gewenste icoon
  text-indent: -9999px   // Oorspronkelijke tekst verbergen
}
```

## Icon fonts

= icons bewaren in een font

Voordelen:

- net als bij spritesheet minder laadtijd
- fonts zijn vectorieel en dus schaalbaar
- het is tekst, geen achtergrondafbeelding

Nadeel:

- geen veelkleurige icons mogelijk



## Box model

- **Hoogte** `height: 100px | 50%`
  - ↳ **Min. hoogte** `min-height: 100px | 50px`
  - ↳ **Max. hoogte** `max-height: 100px | 50%`
- **Breedte** `width: 100px | 50%`
  - ↳ **Min. breedte** `min-width: 100px | 50px`
  - ↳ **Max. breedte** `max-width: 100px | 50%`
- **Marge** `margin: top right bottom left`
  - ↳ **Top** `margin-top: 20px 10%`
  - ↳ **Bottom** `margin-bottom: 20px 10%`
  - ↳ **Left** `margin-left: 20px 10%`
  - ↳ **Right** `margin-right: 20px 10%`

→ **Horizontaal centreren binnen container:** `margin: 0 auto;`
- **Padding** `padding: top right bottom left`
  - ↳ **Top** `padding-top: 20px 10%`
  - ↳ **Bottom** `padding-bottom: 20px 10%`
  - ↳ **Left** `padding-left: 20px 10%`
  - ↳ **Right** `padding-right: 20px 10%`



`box-sizing: content-box | border-box`

- `content-box`: de width en height bevatten enkel de content
- `border-box`: de width en height bevatten de content, padding en border (niet de margin)

## Lay-out

`clear: left | right | both | none`

- voorkomen dat volgende elementen naast de float komen

`overflow: visible | hidden | scroll`

- `visible`: inhoud vloeit over andere elementen
- `hidden`: inhoud die overvloeit wordt verborgen
- `scroll`: scrollbar worden toegevoegd aan de container

`position: static | absolute | fixed | relative`

- `static`: elementen verschijnen volgens de document flow (default)
  - ↳ is niet gepositioneerd
- `absolute`: element is gepositioneerd relatief tot zijn containing block
- `relative`: element is gepositioneerd relatief tot zijn normale positie in de flow
- `fixed`: element is gepositioneerd relatief tot de viewport
  - ↳ blijft op dezelfde plaats, zelfs als het scherm wordt gescrold

`z-index: number`

- bepaalt de positie van een element in de stacking order
- enkel gepositioneerde elementen
- element met een hogere z-index ligt altijd boven een element met een lagere z-index

## Clearfix

een container met daarin een aantal floatende boxen (of andere elementen) float zelf niet en heeft dus geen body. Hij ziet immers de floatende boxen niet.

→ oplossing:

- ↳ floatende elementen clearen
- ↳ een clearfix-klasse toewijzen aan de container

```
.clearfix:before,  
.clearfix:after {  
    content: " "; /* 1 */  
    display: table; /* 2 */  
}  
  
.clearfix:after {  
    clear: both;  
}
```

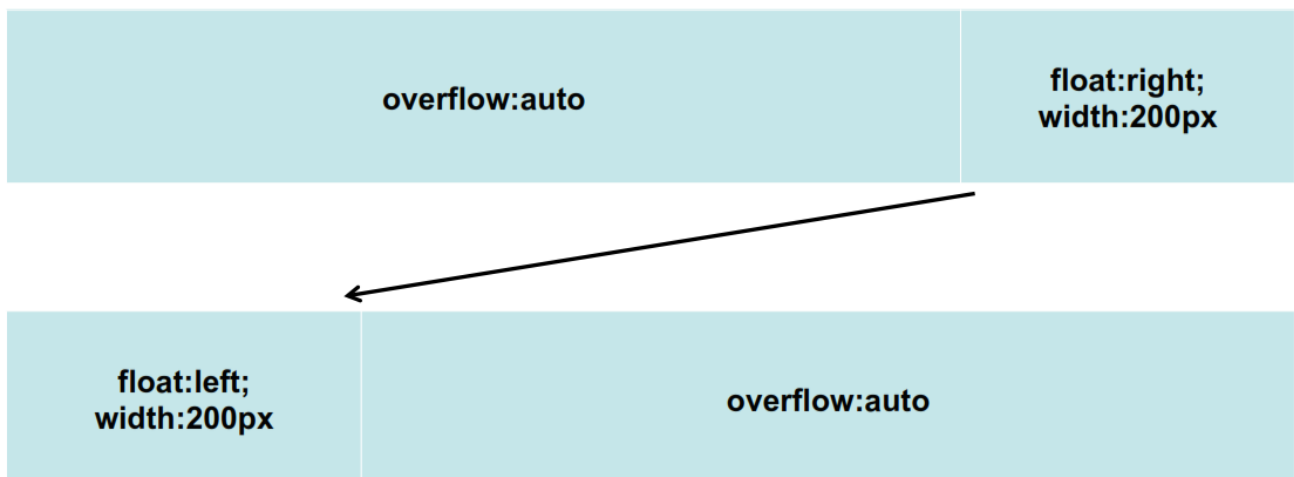
`<div>`  
Ow ow... deze afbeelding is groter dan het containing element en het is gefloat, dus het breekt uit de flow van de container!



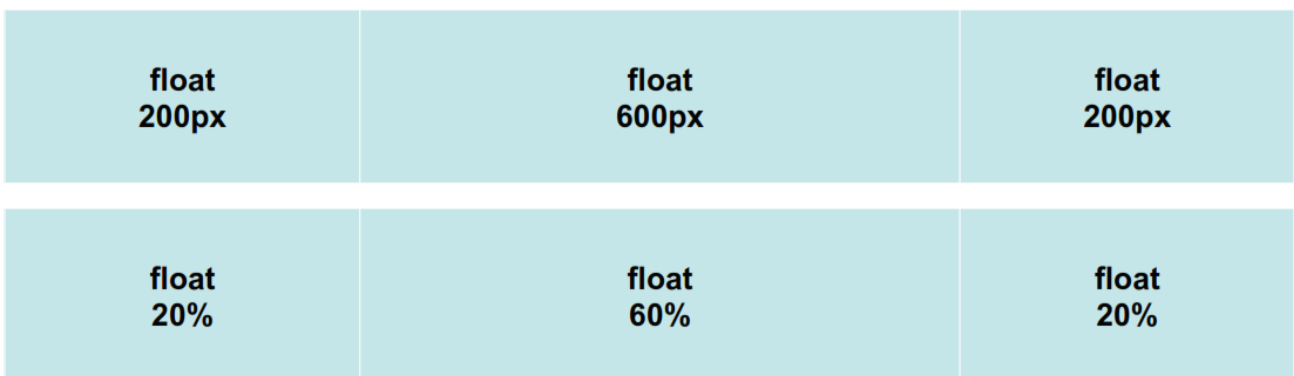
`<div class="clearfix">`  
Veel beter!



## 2 column layout



## Floating layout



## Border

- Border
  - ↳ Top `border-top-width: 20px 10%`
  - ↳ Bottom `border-bottom-width: 20px 10%`
  - ↳ Left `border-left-width: 20px 10%`
  - ↳ Right `border-right-width: 20px 10%`
- Afronding `border-radius: 2em`
  - ↳ Top left `border-top-left-radius: 2em`
  - ↳ Top right `border-top-right-radius: 2em`
  - ↳ Bottom right `border-bottom-right-radius: 2em`
  - ↳ Bottom left `border-bottom-left-radius: 2em`

→ zie *border-radius.com*
- Border image `border-image: source slice width outset repeat`  
`border-image: url(border.png) 30 30 30 round;`  
 → zie *border-image.com*
- Box shadow `box-shadow: horiz.offset vert.offset blur color`
  - ↳ Achter element `box-shadow: 5px 5px 5px #828282`
  - ↳ Binnen element `box-shadow: inset 5px 5px 5px #828282`

## Media queries

**@media:** gebruikt om verschillende CSS-code te schrijven voor

- verschillende mediatypes
  - ✓ @media screen only (*only* is voor compatibiliteit met oudere browsers)
  - ✓ @media print
  - ✓ @media speech
- verschillende features
  - ✓ min/max-width breedte van het weergavegebied (*viewport*)
  - ✓ min/max-height hoogte van het weergavegebied (*viewport*)
  - ✓ min/max-device-width breedte van het apparaat
  - ✓ min/max-device-height hoogte van het apparaat
  - ✓ min/max-resolution pixeldensiteit
  - ✓ orientation landschap/portretmodus
- retinaschermen:
  - @media
  - only screen and (-webkit-min-device-pixel-ratio: 2),
  - only screen and ( min--moz-device-pixel-ratio: 2),
  - only screen and ( -o-min-device-pixel-ratio: 2/1),
  - only screen and ( min-device-pixel-ratio: 2),
  - only screen and ( min-resolution: 192dpi),
  - only screen and ( min-resolution: 2dppx)