



# Les 6

Basis css

# Content

- De taal css
- De plaats van de CSS declaratie
- CSS verwerking door de browser
- Elementen benaderen met selectors
- Enkele css properties

# De taal `css`

# CSS3

- Cascading Style Sheets
  - Status: <https://www.w3.org/Style/CSS/current-work>
  - Opgebouwd uit modules: kleur, boxmodel,...
- 
- De referentie : <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

# CSS: Style Sheet

- Een style sheet is een verzameling van stijlregels, gekoppeld aan delen van je web pagina door middel van selectors. CSS is een declaratieve taal



– Commentaar `/*...*/`

# **De plaats van de CSS declaratie**

# De plaats van de CSS declaratie

- In een **external stylesheet**, een apart document. Heeft de voorkeur.
- In een **internal style block** in de head van een html-document
- Als **inline style** in de begin-tag van het element waarop de stijlregel betrekking heeft.

```
<head>  
  <title>CSS - basis</title>  
  <link rel="stylesheet" href="css/stijlen.css">  
</head>
```

```
<style>  
p {  
  font-family: Lato;  
  font-size: 20px;  
}  
</style>
```

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

# Het CSS bestand

- Commentaar : `/*` en `*/`
- Andere stijlbladen importeren: `@import url('bestand.css')`



# Ingebouwde stijlen aanpassen

- Elke browser heeft een **browser style sheet**. Deze browser style sheet verzorgt de 'default' opmaak van een html pagina. Soms heeft dit onverwachte gevolgen.
- Daarom is het aan te raden dat we bovenaan het CSS-bestand een blok toevoegen die de **browser styles reset**, ofwel download een reset stylesheet die dat voor jou doet
  - Reset css: <https://meyerweb.com/eric/tools/css/reset/>
  - HTML5 reset stylesheet: <http://html5doctor.com/html-5-reset-stylesheet/>
  - Normalize.css : <https://necolas.github.io/normalize.css/>

# De taal css - vervolg

# Waarden en eenheden

- Elke eigenschap heeft mogelijke waarden, vaak meer dan 1. De meest voorkomende:
  - sleutelwoorden: auto, content, flex-start, initial, block, ... => geen " gebruiken
  - getallen: getal of kommagetal (decimaal teken is . ). 0 wordt bij voorkeur weggelaten
  - lengte: getal met een eenheid. Geen spatie ertussen
  - percentage: getal tussen 0 en 100 met %. Geen spatie ertussen
  - url: absoluut of relatief
  - kleur: kleurnaam of numerieke waarde
  - globale waarden: initial, inherit, unset

# Eenheden

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

- Absolute eenheden
  - px: CSS pixel, een optische referentie-eenheid
- Font relatieve eenheden
  - em:
    - gelijk aan de berekende tekstgrootte van het ouderelement.
    - Elke browser heeft dezelfde ingebouwde basisgrootte van tekst: 16px
  - rem: de tekstgrootte van het rootelement
  - ch: breedte van tekst, meerbepaald breedte van het getal 0
- Viewport percentages
  - vw: 1% van de breedte van het venster
  - vh: 1% van de hoogte van het venster
  - vmin: de kleinste waarde van vh en vw
  - vmax: de grootste waarde van vh en vw

# **CSS verwerking door de browser**

# Waarden toewijzen

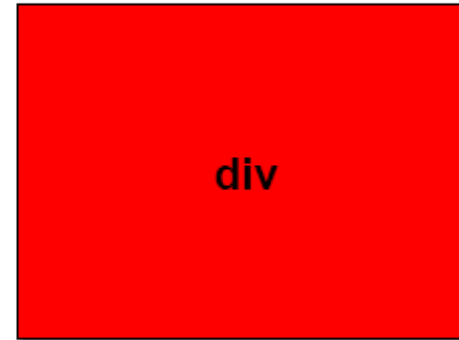
- Wat als er meerdere stijlregels van toepassing zijn op 1 element?

```
body > div
{
  background-color:red;
}

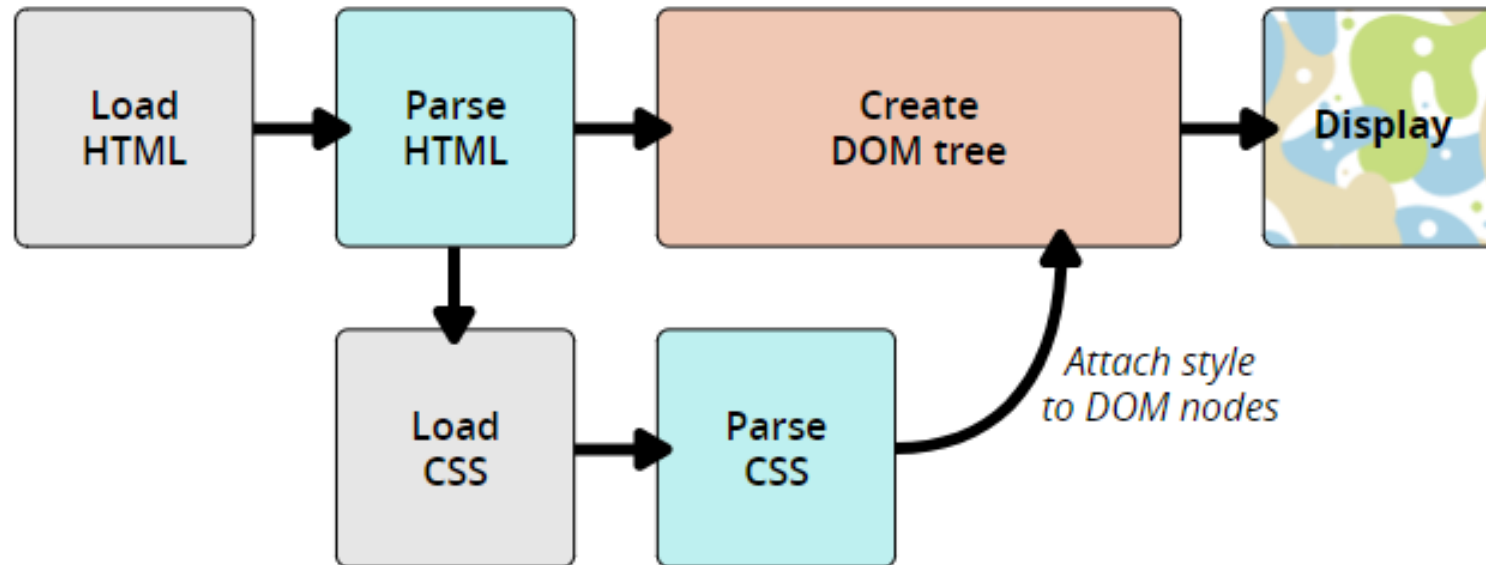
div
{
  background-color:black;
}

*
{
  background-color:green;
}
```

Why am I red?

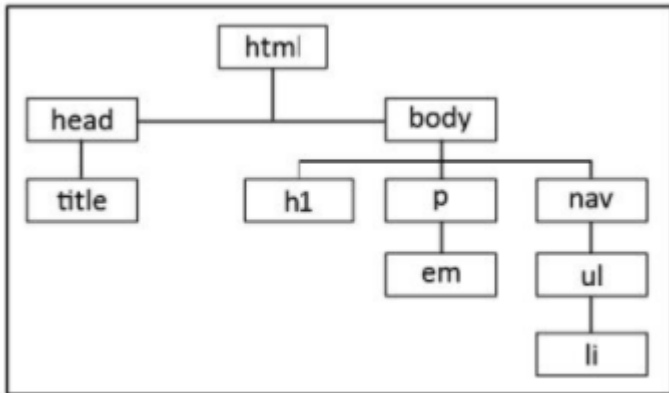


# Hoe werkt css?



# CSS verwerking door de browser

- De boomstructuur



```
<!DOCTYPE html>
<html lang="nl">
  <head>
    <title>HTML boomstructuur</title>
  </head>
  <body>
    <h1>Een kop</h1>
    <p>Een stuk tekst <em>nadruk</em> op een woord</p>
    <nav>
      <ul>
        <li>Home</li>
      </ul>
    </nav>
  </body>
</html>
```



# Waarden toewijzen

- Als de boomstructuur is opgebouwd, **krijgen alle eigenschappen van alle elementen in de boomstructuur waarden toegewezen**
  - Eerst wordt gekeken of een waarde voor de eigenschap in de **cascade** aanwezig is
    - Een stijlblad van de auteur
    - Een stijlblad van de gebruiker
    - Een stijlblad van de browser
  - Zo niet, wordt gekeken of de opmaakinformatie kan worden overgenomen van een ouder (**inheritance**)
  - Kan dat ook niet, dan wordt de **defaultwaarde** van de eigenschap gebruikt

# De cascade

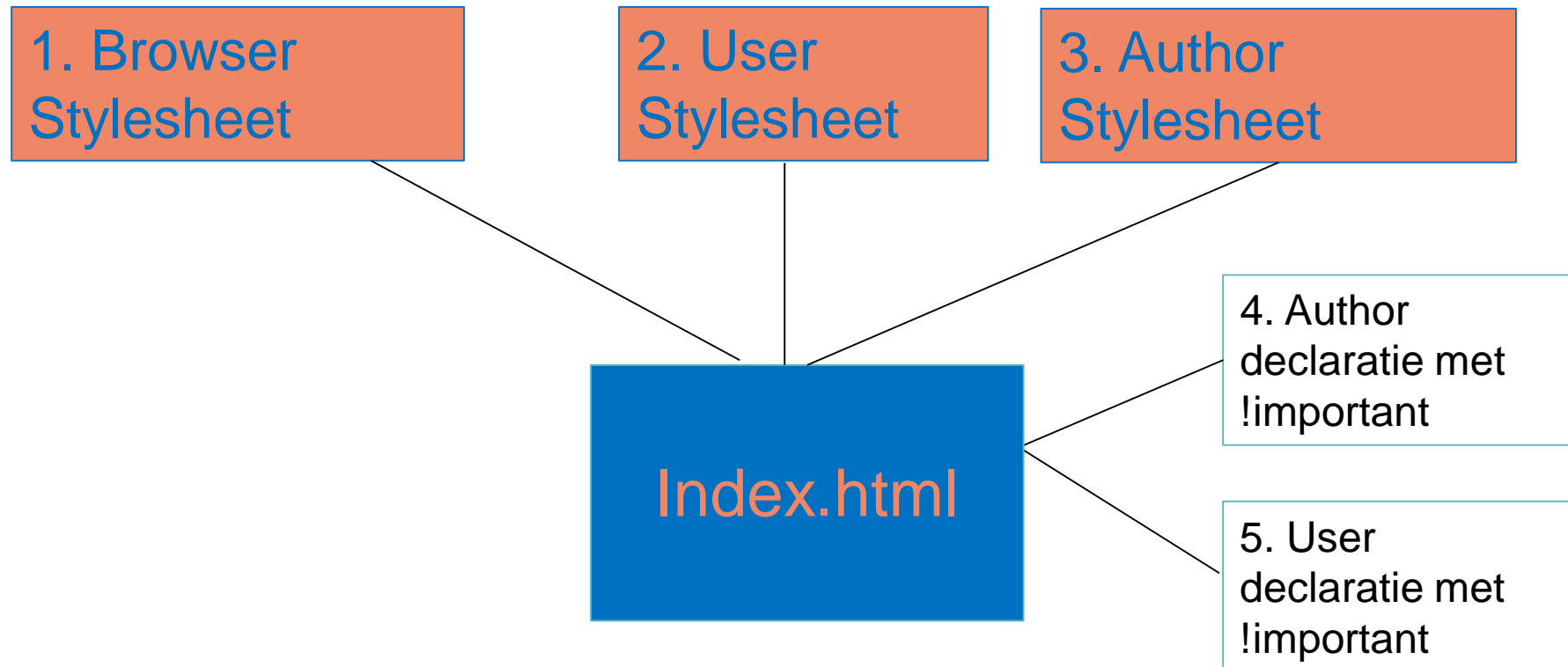
1. Zoek voor een element alle geldige declaraties op
  - selector komt overeen met het element en het weergavemedium komt overeen met de eigenschap media
2. Sorteer alle declaraties naar **belangrijkheid** (importance). Bronnen in oplopende volgorde van belangrijkheid:
  - Declaraties in de browser
  - Declaraties van de gebruiker
  - Declaraties van de auteur
  - Declaraties van de auteur met kenmerk !important
  - Declaraties van de gebruiker met kenmerk !important

## De cascade

3. Sorteer de declaraties uit dezelfde bron en met dezelfde belangrijkheid op **specificiteit**. De hoogste specificiteit wint
4. Sorteer op de volgorde waarin de declaratie is aangegeven. Als twee regels dezelfde belangrijkheid en specificiteit hebben wint de laatst gespecificeerde.

# Cascade: belangrijkheid

- De bronnen worden in een vaste volgorde overlopen
  - Regels uit belangrijker bronnen hebben voorrang op regels uit minder belangrijke bronnen (1=minst belangrijk, 5 belangrijkste)



# Cascade: specificiteit

- Hoe wordt specificiteit van een regel bepaald?
  - Wordt uitgedrukt in een reeks van 4 getallen
  - De specificiteit neemt af van links naar rechts

0, 0, 0, 0

Aantal elementnamen of pseudo elementen

Aantal class attributen, elk ander attribuut of pseudo class

Aantal id attributen

Inline style attribuut?

# Cascade: specificiteit

Selector	Specificiteit	Toelichting
div ul ul li	0,0,0,4	4 elementselectors
div.klasse ul li	0,0,1,3	1 klasseselector, 3 elementselectors
a::before	0,0,0,2	1 pseudoklasse, 1 elementselector
a.klasse:hover	0,0,2,1	1 klasse, 1 pseudoklasse, 1 elementselector
p.eerste	0,0,1,1	1 klasseselector, 1 elementselector
#id p.eerste	0,1,1,1	1 id-selector, 1 klasseselector, 1 elementselector

# Cascade: specificiteit

- Als er meerdere conflicterende stijlregels uit 1 bron van toepassing zijn op een element, bereken specificiteit stijlregel

```
<li id="content" class="red">...</li>
```

```
*{ color: ■green; }  
li{color: ■red;}  
ul li{color: ■orange;}  
li.red{color: ■yellow;}  
#content{color: ■blue;}
```

# Cascade: specificiteit

- Voorbeeld gesorteerd by specificity

```
<li id="content" class="red">...</li>
```

## Specificity Calculator

A visual way to understand [CSS specificity](#). Change the selectors or paste in your own.

\*

0

Inline styles

0

IDs

0

Classes, attributes  
and pseudo-classes

0

Elements and  
pseudo-elements

+ Duplicate

li

0

Inline styles

0

IDs

0

Classes, attributes  
and pseudo-classes

1

Elements and  
pseudo-elements

+ Duplicate

<https://specificity.keegan.st/>

ul li

Sort by specificity

0

Inline styles

0

IDs

0

Classes, attributes  
and pseudo-classes

2

Elements and  
pseudo-elements

+ Duplicate

li.red

0

Inline styles

0

IDs

1

Classes, attributes  
and pseudo-classes

1

Elements and  
pseudo-elements

+ Duplicate

#content

0

Inline styles

1

IDs

0

Classes, attributes  
and pseudo-classes

0

Elements and  
pseudo-elements



# Overerving

- Een opmaakeigenschap hoeft niet rechtstreeks een waarde te krijgen
- De waarde kan worden overgeerd van het ouderelement

```
h1{color:blue;}
```

```
<h1>Hey, <em>look</em> at that sheepdog</h1>
```

Maar

```
h1{color:blue;}
```

```
em{color:orange;}
```

```
<h1>Hey, <em>look</em> at that sheepdog</h1>
```

# Overerving

- Sommige eigenschappen zijn niet erfelijk, zoals: padding, width, height, background-color, border en float,.... Je kan dit toch forceren door **inherit** mee te geven als waarde.

# Overerving

- Welke properties worden overgeërfd?
    - Tekst – gerelateerde properties:  
font-family, font-size (zie verder!), font-style, font-variant, font-weight, font, letter-spacing, line-height, text-align, text-indent, tekst-transform, word-spacing.
    - List – gerelateerde properties:  
list-style-image, list-style-position, list-style-type, list-style
    - Kleuren:  
color
- Merk op: background-color en box eigenschappen (zie later) worden niet overgeërfd.

# Percentages werken door

```
div.inner {  
  height: 50%;  
  width: 50%;  
}  
<div>  
  <div class="inner">  
    <div class="inner"></div>  
  </div>  
</div>
```



# **Elementen benaderen met selectors**

# Selectors

- Selectors level 3 (W3C aanbeveling sinds November 2018)
- 4 groepen
  - Basiselectors
  - Combinatie selectors
  - Pseudo klassen
  - Pseudo elementen
- Basis : DOM

# Elementnamen of klassen of beide?

- Elementnamen en selectors
  - Links in menu in styles conference
    - `header > nav > ul > li > a{...}` /\*kindselector, maar onnodig specifiek\*/
    - Beter `header > nav > a{...}` of `nav:first-of-type a{...}`
  - Alinea in de lead section heeft een andere opmaak dan de rest van de pagina
    - `Header+section > p { ...}`
    - Nadeel : CSS is op deze manier volledig gekoppeld aan de HTML structuur

# Elementnamen of klassen of beide?

- Klassen
  - Links in menu in styles conference
    - `<nav class="pagemenu">`
    - `.pagemenu a{...}`
  - Alinea in de lead section heeft een andere opmaak dan de rest van de pagina
    - `<section class="lead">...</section>`
    - `.lead p{...}`



# Elementnamen of klassen of beide?

- En de Winnaar is ...
  - Geen sluitend antwoord
  - Voor Styles conference, maken we zoveel als mogelijk gebruik van elementnamen
  - Voor grote projecten, worden vaak klassen gebruikt en daarbij blijkt de BEM notatie favoriet (BEM=Block Element Modifier, <http://getbem.com>)



# Elementnamen of klassen of beide?

- BEM voorbeeld

```
<!DOCTYPE html>
<html>
<body>
  <header class="header">
    <h1 class="header__title header__title--big">Some Title of a Web Page</h1>
  </header>
  <div>
    <section>First Section</section>
    <nav>Nav Bar</nav>
  </div>
  <footer>
    <p>Only Footer</p>
  </footer>
</body>
</html>
```

```
.header__title {
  background: #bbbbbb;
  padding: 10;
  margin: 10;
}

.header__title--big {
  fontsize: 48;
}
```

# Basisselectors

- Type Selectors (of Element Selectors)
  - hebben dezelfde naam als het element dat ze ‘stylen’.
  - Zijn van toepassing op elk ‘optreden’ van dit element
    - `h1 {color:red;}`
- Universele selector: `*`
  - Regels zijn van toepassing op elk element in de pagina.
  - `* {color:red}`

# Basisselectors

- Voor selectie van specifieke elementen, gebruik het attribuut class of id
  - ID is uniek => selectie van 1 element
    - De waarde van een id attribuut moet uniek zijn binnen een pagina
    - Een id-attribuut kan maar 1 waarde bevatten  
`<div id="zoek">`
    - In css : naam van een id selector begint met een #
  - classes zijn niet uniek => selectie van meerdere elementen
    - De waarde van een class mag onbeperkt voorkomen in een pagina
    - Een element kan meerdere class waarden hebben. Deze worden gescheiden door een spatie.  
`<div class="widget rechts">`
    - In css : naam van een class selector begint met een . (punt)

# Basisselectoren

- Combinatieselectors
  - Groupselectors: A,B
    - Lijst van selectoren waarop een stijl van toepassing is
  - Algemene afstamming: A B
    - selecteert de nakomelingen van een element (kind, kleinkind,...)
  - Kindselector: A>B
    - selecteert de directe kinderen van een element
  - Aangrenzend: A+B
    - Elk element B, op hetzelfde niveau als A en dat onmiddellijk op element A volgt
  - Op hetzelfde niveau: A~B
    - Elk element B, op hetzelfde niveau als A en dat op element A volgt

# Basisselectors

- Attribuutselectors

Teken	Beschrijving	Voorbeeld	Betekenis voorbeeld
		img[alt]	elk element img met attribuut alt
=	Waarde exact gelijk aan	img[alt="foto"]	elk img element waarvan alt attribuut gelijk aan de waarde foto
~=	Bevat waarde	img[class~="fotoboek"]	elk img element waarvan class attribuut meerdere waarden kan hebben waaronder fotoboek
=	Begint met waarde	[hreflang = "nl"]	elk element waarvan hreflang attribuut gelijk aan nl, of begint met nl- (nl, nl-NL, nl-BE)
^=	Begint met exacte waarde	[name^="keuze"]	name attribuut begint met het exacte woord "keuze" (bv. "keuze2" of "keuzevak")
\$=	Eindigt met exacte waarde	[href\$=".com"]	href attribuut eindigt op ".com" (bv. "www.google.com")
*=	Bevat exacte waarde	[href*="user"]	href attribuut bevat het exacte woord "user" (bv. "www.hogent.be/user/jan")

# Basisselectors

- Pseudoklassen
  - Duidt een toestand van een element aan
  - Link pseudo-classes (in de juiste volgorde)
    - 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](#)

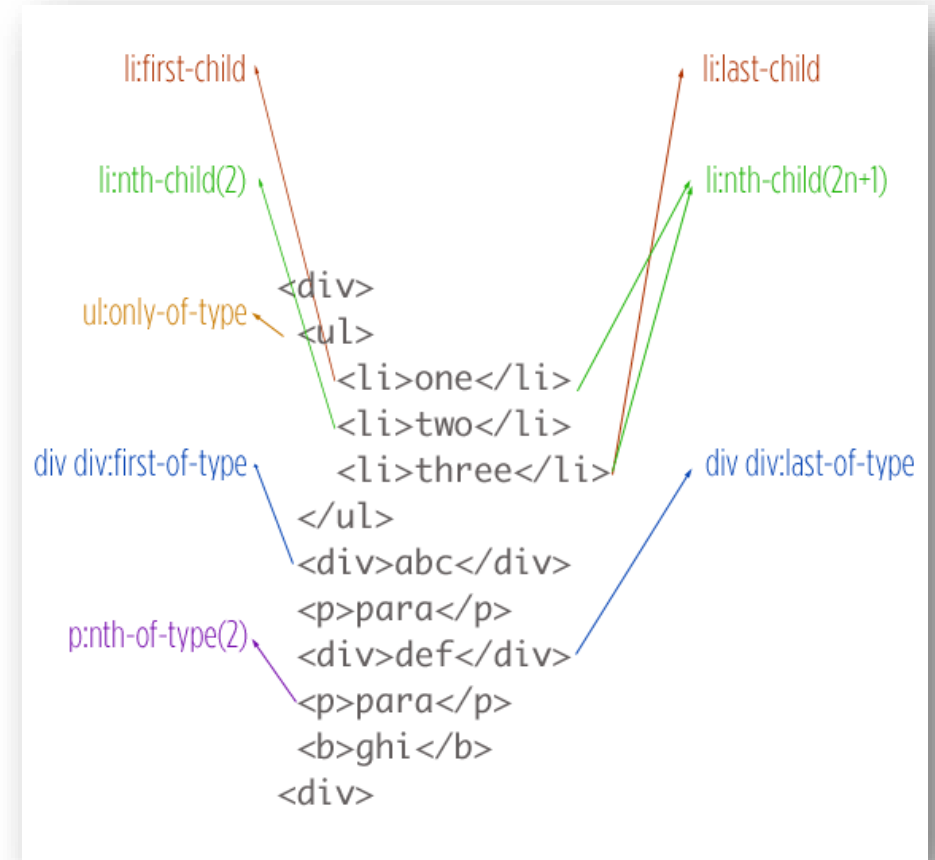
# Basisselectors

- Pseudoklassen
  - taal :
    - [:lang\(\)](#)
  - UI toestand
    - [:enabled](#) : elementen die beschikbaar zijn
    - [:disabled](#) : elementen die niet beschikbaar zijn
    - [:checked](#) : een keuzerondje of selectievakje die is ingeschakeld
  - ontkenning
    - [:not\(x\)](#)



# Basisselectors

- Structurele pseudoklassen
  - [:root](#): hoofdelement <html>
  - [:first-child](#): een element dat het eerste child is van zijn parent-element,
  - [:last-child](#): een element dat het laatste child is van zijn parent-element
  - [:only-child](#): een element waarvan de ouder geen andere child elementen heeft
  - [:first-of-type](#): het eerste element van dat type
  - [:last-of-type](#): het laatste element van dat type
  - [:only-of-type](#): het enige element van dat type
  - [:nth-child\(n\)](#): elk zoveelste kindelement
  - [:nth-last-child\(n\)](#): nu gerekend vanaf het laatste kind
  - [:nth-of-type\(n\)](#): elk zoveelste element van een type
  - [:nth-last-of-type\(n\)](#): nu gerekend vanaf laatste element
  - [:empty](#): leeg element



# Basisselectors

- `:nth-child(n)`, `:nth-last-child(n)`
  - Selecteert elk zoveelste child element
    - `nth-child`: start bij eerste child
    - `nth-last-child` : gerekend vanaf het laatste child
  - $n$  = getal of expression:  $an$ ,  $an+b$ ,  $an-b$ ,  $n+b$ ,  $-n+b$ ,  $-an+b$ 
    - $(a*n) \pm b$  met  $n$  is een multiplier startende vanaf 0 en  $b$  is startelement. Een negatieve  $a$  waarde moet gevolgd worden door een positieve  $b$  waarde. In dat geval bepaalt  $b$  hoever de telling gaat
    - `li:nth-child(2)` : 2<sup>de</sup> item
    - `li:nth-child(3n)` :  $3*0$ ,  $3*1$ ,  $3*2$ ,.... item
    - `li:nth-child(2n+3)` :  $(2*0)+3$ ,  $(2*1)+3$ ,  $(2*2)+3$ ,... item
    - `li:nth-child(6n-4)` :  $(6*0)-4$ ,  $(6*1)-4$  ... item
    - `li:nth-child(-3n+12)` : elke 3<sup>de</sup> item binnen de eerste 12 items
  - Even of oneven children selecteren: “even” ipv  $2n$  of “odd” ipv  $2n+1$

# Basisselectoren

- Pseudo elementen
  - Toegang tot onderdelen van een document waarvoor geen gewone elementen bestaan
    - [::first-line](#) : de eerste regel opgemaakte tekst van een element
    - [::first-letter](#) : de eerste letter
    - [::before](#): voor de inhoud van een element
    - [::after](#): na de inhoud van een element

# Enkele css properties

# Enkele css properties

- [background](#) : achtergrondskleur
- [color](#) : kleur tekst
- [font-family](#) : het lettertype
- [font-size](#) : tekstgrootte
- [font-weight](#) : het gewicht (de vetheid) van het lettertype
- [line-height](#) : de regelafstand, hoeveelheid wit tussen de regels tekst
- [margin-bottom](#)/top/left/right : ruimte tussen de rand van een element en de rand van het parent element of een aangrenzend element
- [padding-bottom](#)/top/left/right : de ruimte tussen de inhoud en de rand
- [text-align](#) : uitlijnen van de tekst
- [text-decoration](#) : lijneffecten
- [text-transform](#) : omzetten naar kleine/hoofdletters