

User Interface Design

Tuur Vanhoutte
Karel Bousson

13 mei 2020

Inhoudsopgave

1	HTML	1
1.1	Content models	1
1.2	HTML tags	1
1.2.1	Unclosed HTML tags	1
1.3	HTML attributes	2
1.3.1	HTML global attributes	2
1.3.2	HTML specifieke attributes	2
1.3.3	Semantic tags	2
1.3.4	article	3
1.3.5	section	3
1.3.6	header	3
1.3.7	footer	3
1.3.8	nav	4
1.3.9	aside	4
1.3.10	figure & figcaption	4
1.3.11	div	4
1.3.12	span	4
1.3.13	Block level element	4
1.3.14	Inline level element	5
1.4	The CSS Box Model	5
1.4.1	Default rendering: box-model	5
1.4.2	Box-sizing: border-box	5
1.5	Code conventions	5
2	CSS	6
2.1	Hoe werkt CSS?	6
2.2	DOM (Document Object Model)	6
2.3	CSS Syntax	7
2.4	Selectors	7
2.5	The Cascade	9
2.5.1	Source Order	9
2.5.2	Specificity	9
2.5.3	Importance	9
2.5.4	Rule mixin	10
2.5.5	Inheritance	10
2.6	Code conventions	10
3	CSS Defaults	11
3.1	User Agent Stylesheet	11
3.2	normalize.css	11
3.3	reset.css	11
3.4	Vendor prefixes	11
3.5	caniuse.com	12
3.6	Browser support	12
3.6.1	Desktop Browsers	12
3.6.2	Mobile Browsers	12
3.7	Progressive enhancement	13
3.8	Margin collapsing	13
3.9	Margin & Padding baseline system	13
3.10	Het verschil tussen margin en padding?	13

3.10.1 Margin & padding system gebaseerd op de baseline	13
3.11 Units of measurement: EM vs REM vs PX	14
3.11.1 Voorbeelden	14
4 CSS Architecture	14
4.1 CSS architecture	14
4.2 Specificity wars	14
4.3 ITCSS	15
4.4 Class naming conventions	16
4.4.1 Object classes .o-	16
4.4.2 Component classes .c-	16
4.4.3 Scoping classes .s-	16
4.4.4 Utility classes .u-	16
4.4.5 Javascript hooks .js-	17
4.5 BEM-notatie: Block element, modifier	17
4.5.1 Rules	18
4.6 Goede CSS selector	18
4.6.1 Selector intent	18
4.6.2 Overdraagbaarheid	18
4.6.3 Locatie onafhankelijk	18
4.6.4 Naming	18
4.7 The Single Responsibility Principle	19
4.8 The seperation of concerns	19
4.9 CSS selectors in MCT	19
5 Flexbox	19
5.1 Zelfstudie	19
5.2 CSS Flexible Box Layout Module	19
5.3 Properties voor de parent (flex container)	20
5.4 Display	20
5.5 Flex-direction	20
5.5.1 flex-direction: column-reverse	22
5.6 Flex-wrap	22
5.7 Justify-content	23
5.8 Align-items	23
5.9 Align-content	25
5.10 Properties voor de flex items	25
5.10.1 Order	25
5.11 Align-self	26
5.12 flex	26
5.12.1 Flex-grow	27
5.12.2 Flex-shrink	28
5.12.3 Flex-basis	28
5.12.4 Resources	28
6 Kleurtheorie	28
6.1 Hoe worden kleuren opgebouwd?	28
6.2 Additive color synthesis	29
6.2.1 Soorten kleuren	29
6.3 Hexadecimale codes	30
6.4 Color Properties	30
6.4.1 HSB: Hue, Saturation Brightness	30

6.5	Kleurenpalet samenstellen	30
6.5.1	Kleurenpalet	30
6.6	Kleurenpsychologie	32
6.7	Contrast	32
6.7.1	Contrast ratio	32
6.7.2	Non-text contrast	33
6.7.3	Interface elements	33
6.7.4	Tekst	33
6.8	Kleurenblindheid	33
6.9	Kleur in UI design	33
6.9.1	Tools	33
7	Layout & Grids	34
7.1	Layout	34
7.2	Alignment	34
7.3	Grids	34
7.3.1	Horizontal Grid	34
7.3.2	Vertical Grid	34
7.4	Grid systems	35
7.4.1	Responsive grid	35
7.4.2	Baseline grid	35
7.5	Spacing system	35
7.6	White space	36
8	Typografie	36
8.1	Lettertype vs Font	36
8.2	Classificatie	36
8.2.1	Serif	36
8.2.2	Sans-serif	37
8.3	Monospaced	38
8.3.1	Handwritten, Script & Decorative	39
8.4	Anatomie	39
8.4.1	Mean line & Baseline	39
8.4.2	x-height	40
8.4.3	Ascender & Descender	40
8.4.4	Width	40
8.5	Properties	41
8.5.1	Weight	41
8.5.2	Leading/line-height	41
8.5.3	Alignment	42
8.5.4	Tracking & Kerning	42
8.5.5	Ligatures	43
8.6	Leesbaarheid	44
8.6.1	De perfect paragraaf	44
8.6.2	Het perfecte UI lettertype	44
8.7	Lettertypes combineren	45
8.7.1	Lettertypes met dezelfde karakteristieken	45
8.7.2	Groot contrast	45
8.8	Lettertypes vinden	45
8.9	Hierarchie	46
8.9.1	Welke font-size kiezen?	46

9 Visual hierarchie	47
9.1 Gestalt theorie	47
9.2 Typografische hierarchie	47
9.3 Grootte en gewicht	48
9.4 Color	49
9.5 Alignment en position	49
9.6 The fold	50
9.7 Fixed elements	51
9.8 Rows & Containers	51
9.9 Icons	53
9.9.1 Geometrische en optische grootte	53
9.9.2 Bounding box	54
9.9.3 Pixel grid	54
9.9.4 User Interface icons	54
9.10 Resources	55
9.11 Tips 'n tricks	55
10 Document flow & stacking order	59
10.1 Document flow & stacking order	59
10.2 Position properties	59
10.2.1 top, right, bottom, left	59
10.2.2 Position: static	59
10.2.3 Position: relative	60
10.2.4 Position: absolute	60
10.2.5 Position: fixed	60
10.2.6 Position: sticky	60
10.3 z-index property	60
10.3.1 Z-index values	61
10.3.2 Z-index systeem	62
11 Responsive design	62
11.1 Origins	62
11.2 Requirements	62
11.3 One Web	62
11.4 Mobile first	63
11.4.1 Voorbeelden	63
11.5 Media queries	63
11.5.1 Media queries syntax	63
11.5.2 Media types	63
11.5.3 Media expressions	64
11.5.4 Media queries overview	64
11.5.5 Media queries in de praktijk	64
11.6 Breakpoints	65
11.7 Responsive images	65
12 Webfonts	66
12.1 @font-face	66
12.1.1 Verschillende formats	67
12.2 WOFF/WOFF2	67
12.3 SVG/SVGZ	67
12.4 EOT	67
12.5 Font-Family linking	68

12.6	Font-weight	68
12.7	Licenties	68
12.8	External Hosting Services	68
12.9	Fallback fonts	69
12.10	System UI fonts	69
12.11	Resources	69
13	SVG	69
13.1	Raster images (bitmaps)	70
13.2	Using SVG	70
13.2.1	Voorbeelden	70
13.3	viewBox	70
13.3.1	Voorbeelden	70
13.4	SVG shapes	71
13.4.1	Rectangle	71
13.4.2	Circle	71
13.4.3	Ellipse	71
13.4.4	Line	71
13.4.5	Polyline	72
13.4.6	Polygon	72
13.4.7	Path	72
13.5	Styling SVG content using CSS	72
13.5.1	Useful SVG attributes	72
13.6	Exporting SVG	73
13.7	Resources	73
14	Coole CSS properties & values	73
14.1	Float	73
14.2	Border-radius	73
14.3	Shape-outside	74
14.4	Filter	74
14.5	Background	74
14.6	box-shadow	74
14.7	text-shadow	75
14.8	Transform	75
14.8.1	Skew	75
14.8.2	Rotate	75
14.8.3	Translate	75
14.8.4	Scaling	75
14.8.5	Perspective	75
14.9	Transition	75
14.9.1	Transition-timing-function	76
14.10	Animation	76
14.11	Calc	76
14.12	v value	77
14.13	currentColor keyword	77
14.14	Rgba and HSLA value	77
15	Labo	77
15.1	Objects	77
15.1.1	o-row	77
15.1.2	o-container	78

15.1.3 o-layout	78
15.2 Utilities	79
15.2.1 u-max-width	79
15.2.2 u-flex-basis, u-flex-grow & u-x-of-y	79

1 HTML

HyperText Markup Language

1.1 Content models

<https://html.spec.whatwg.org/multipage/dom.html#content-models>

1.2 HTML tags

```
1 <tag> content </tag>
```

PERIODIC TABLE OF HTML TAGS
[INFOGRAPHICS]

The Periodic Table of HTML Tags is a grid-based diagram where each cell contains an HTML tag name and its corresponding number. The tags are color-coded into groups:

- BASIC HTML** (Orange): Dt (1), Hl (2), Hr (8), Ttl (5), Cm (9).
- FORMATTING** (Yellow): By (4), Ar (10), Bo (14), De (18), Mk (24), Rt (30), Sub (36), Fm (42), On (48), Ot (54), Fgc (60), Vd (66).
- FORMS AND INPUT** (Red): Hh (5), As (11), Bq (15), Dn (19), Mr (25), Ry (31), Sup (37), It (45), Lbl (49), If (55), Fg (61), A (67), Dd (75), Th (80), Cl (86), Ft (92), Dg (98), Nst (105), Ed (106).
- FRAMES** (Pink): P (6), B (12), Ct (16), Em (20), Pre (26), S (32), Tm (38), Ta (44), Fs (50), Ig (56), Pr (62), Lk (68), Mu (76), Td (82), Sl (88), Sn (94), Da (100), Ot (107).
- IMAGES** (Blue): Br (7), Bi (13), Cd (17), I (21), Ps (27), Sp (33), U (39), Bn (45), Ld (51), Mp (57), Ao (63), Nv (69), Mm (77), Thd (85), Dv (89), Ae (95), Hd (101), Pm (108).
- AUDIO / VIDEO** (Light Blue): Sm (34), Vr (40), St (46), Ds (52), Aa (58), Sc (64), Ul (70).
- LINKS** (Dark Blue): Sg (35), Wr (41), Op (47), Kn (53), Cs (59), Tk (65), Ol (71).
- LISTS** (Green): Tables (Tables).
- TABLES** (Light Green): Li (72), Tl (78), Tbd (84), Spn (90), Ad (96), Be (103), Spt (104).
- STYLES AND SEMANTICS** (Light Green): Li (72), Tl (78), Tbd (84), Spn (90), Ad (96), Be (103), Spt (104).
- META INFO** (Purple): Programming.
- PROGRAMMING** (Purple): Mt (102).

Figuur 1: Overzicht HTML tags

1.2.1 Unclosed HTML tags

- Sommige HTML tags worden niet afgesloten
- Kunnen geen content bevatten

```
1 
2 <input type="text">
```

1.3 HTML attributes

- Definieert het gedrag van een tag
- Attributes moeten op de start tag
- Attributes kunnen 1 of meerdere values hebben
- Globale attributes
- Specifieke attributes

```
1 <element attribute_name="value">content</element>
```

1.3.1 HTML global attributes

Kunnen op elke HTML tag geplaatst worden

- class
- id
- <https://www.javatpoint.com/html-global-attributes>

```
1 <h1 id="value">content</h1>
2 <nav class="nav nav-main">content</nav>
```

1.3.2 HTML specifieke attributes

Kunnen op 1 specifieke HTML tag geplaatst worden

- href
- type
- src

```
1 <a href="/index">content</a>
```

1.3.3 Semantic tags

- Semantiek is de studie van de betekenis van woorden en bewoordingen in een taal.
- Semantische elementen = elementen met betekenis.
- Een semantisch element beschrijft duidelijk de betekenis ervan voor zowel de browser als de ontwikkelaar.
- Voorbeelden van niet-semantische elementen: <div> en - vertelt niets over de inhoud ervan.
- Voorbeelden van semantische elementen: <form>, <table> en <article>; definieert duidelijk de inhoud ervan.

1.3.4 article

- Een zelfstandig onderdeel
- Is 'standalone' nog steeds zinvol
- Kan genest worden
- Kan meerdere sections, een header of footer bevatten
- Voorbeelden:
 - Forum post
 - Blog post
 - Newspaper article

1.3.5 section

- Beschrijft een sectie van een document
- Gerelateerde inhoud groeperen, als een onderdeel van een lang artikel, een groot deel van de pagina
- Heeft meestal een header en soms een footer
- Kan articles en sections bevatten
- Een homepage kan bijvoorbeeld opgesplitst worden in een intro section, een main content section en een contact section

1.3.6 header

- Beschrijft een header van een document, section of article
- Container voor intro content
- Verschillende headers in 1 document
- Voorbeelden:
 - Header van een document met logo en navigatie
 - Header van een article met intro content

1.3.7 footer

- Beschrijft een footer voor een document, section of article
- Extra informatie over het element waar de footer in zit
- Verschillende footer in 1 document
- Voorbeelden:
 - Auteur, copyright informatie, extra links, contact informatie
 - Deelknoppen van social media (bv binnen een article)
 - Niet altijd onderaan!

1.3.8 nav

- Grote belangrijke navigatie links
- Primaire en secundaire informatie
- Enkel voor menu's
- Niet voor alle links
- Horizontaal of verticaal
- Bevat meestal een omdat het een lijst is van links

1.3.9 aside

- Beschrijft content aside van een page/section content
- Binnenin section bvb poll, samenvatting
- Buiten article bvb commentatren, tweets, aside

1.3.10 figure & figcaption

- figcaption = voegt een visuele beschrijving toe aan een image
- en <figcaption> zitten gegroepeerd in <figure>

1.3.11 div

- W3C: *generic container for flow content that by itself does not represent anything*
- Niet semantisch
- Enkel voor opmaak met css

1.3.12 span

- W3C: *generic container for phrasing content that by itself does not represent anything*
- Niet semantisch
- Enkel voor opmaak met css

1.3.13 Block level element

- Start altijd op een nieuwe lijn
- Neemt de volledige breedte in die mogelijk is
- Rekt zo ver mogelijk links en rechts uit
- Is even breed als de breedte van het bovenliggende element/scherm/viewport

Voorbeelden:

```
<address> <article> <aside> <blockquote> <canvas> <dd> <div> <dl> <dt>
<fieldset> <figcaption> <figure> <footer> <form> <h1>–<h6> <header> <hr>
<li> <main> <nav> <noscript> <ol> <output> <p> <pre> <section> <table>
<tfoot> <ul> <video>
```

1.3.14 Inline level element

- Start niet op een nieuwe lijn
- Neemt enkel de breedte in die het nodig heeft
- Sommige CSS properties (width, height, padding, ...) doen niets.

1.4 The CSS Box Model

- Alle HTML elementen zijn boxes
- Box-model bepaalt hoe de oppervlakte van een box worden berekend.
- De oppervlakte: (van binnen naar buiten)
- Content: de content van de box. Waar text en images komen.
- Padding: Ruimte rond de content. Is altijd transparant
- Border: een border rond de padding en de content. Kan wel een kleur hebben
- Margin: Extra ruimte buiten de border. Is altijd transparant

1.4.1 Default rendering: box-model

- Width en height bepalen de grootte van de content
- Padding en border worden er niet bij opgeteld
- Margin duwt de box naar boven, rechts, beneden, of links waardoor de totaal ingenomen plaats ook beïnvloed wordt

1.4.2 Box-sizing: border-box

- Width en height bepalen de grootte van de content
- Padding en border worden vanbinnen toegevoegd en beïnvloeden dus niet langer de grootte van een element.
- Margin duwt de box naar boven, rechts, beneden of links waardoor de totale ingenomen plaats ook beïnvloed wordt.

1.5 Code conventions

- Geen ID's voor styling
- Lowercase tags
- Tags afsluiten
- Lowercase attribute names
- Attributen tussen dubbele aanhalingstekens

- Image attributes
- Indent

2 CSS

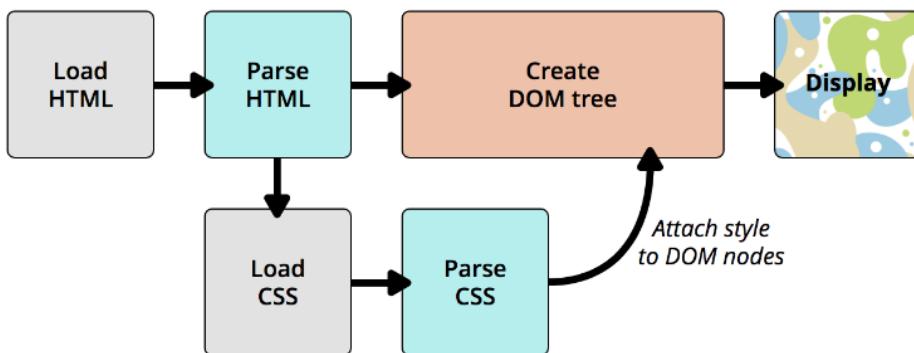
- Cascading Style Sheet
- Taal om te specificeren hoe documenten (HTML) worden gerepresenteerd
- Browsers passen CSS rules uit de stylesheet toe op elementen van een document
- CSS rules bestaan uit:
 - Selector
 - Set van properties

2.1 Hoe werkt CSS?

- Een browser interpreteert een HTML document in 2 stadia:
 - De browser converteert HTML en CSS in de Document Object Model (DOM). De DOM combineert de content met de stijl.
 - De browser toont de content van de DOM

2.2 DOM (Document Object Model)

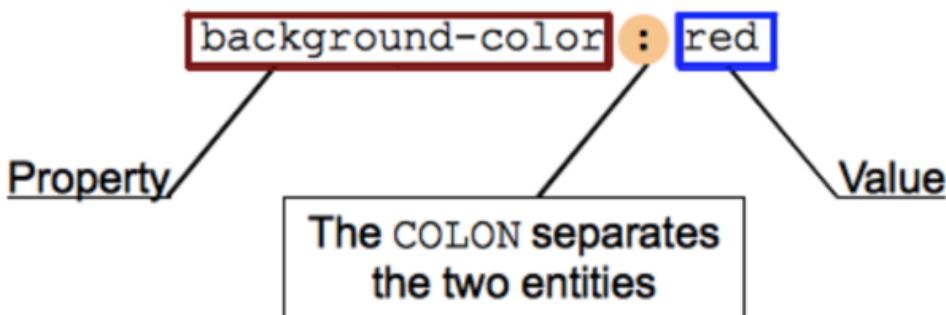
- De DOM represeneert het document in het geheugen van de computer
- DOM is where your CSS and the document's content meet up
- Boom structuur
- Elk element, attribuut en stukje tekst wordt een DOM node in de boom structuur
- Nodes worden gedefinieerd door hun relatie tot andere DOM nodes
- Parents & Children



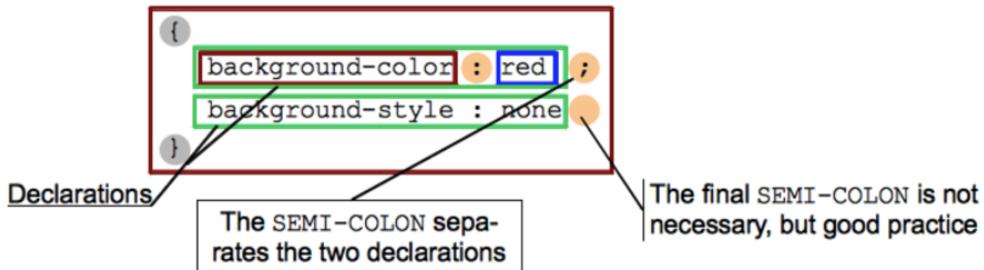
2.3 CSS Syntax

- CSS declaration
 - **Properties:** Wat?
 - **Values:** Hoe?
- CSS Declaration blocks
- CSS Ruleset/rule

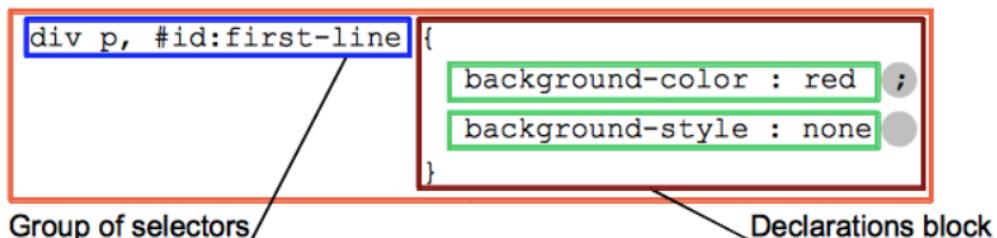
A CSS declaration :



A CSS declarations block:



A CSS ruleset (or rule):



2.4 Selectors

- Simple selectors
 - Type selectors (=element selectors): body, h1, p, ul, ...

- Class selectors: .class-name
- ID selectors: #id-name
- Universal selector: *
- Attribute selectors
 - CSS rules worden toegepast op elementen met een bepaald attribuut en attribuut waarde.
 - Kan je herkennen aan de square brackets: [attr]
 - Presence and value attribute selectors
 - Substring value attribute selectors
 - Meer info:

https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Attribute_selectors

- Pseudo selectors
 - pseudo-classes: selector:**:hover**
 - pseudo-elements: selector:**::before**

:active	:hover	:only-child
:any	:indeterminate	:only-of-type
:checked	:in-range	:optional
:default	:invalid	:out-of-range
:dir()	:lang()	:read-only
:disabled	:last-child	:read-write
:empty	:last-of-type	:required
:enabled	:left	:right
:first	:link	:root
:first-child	:not()	:scope
:first-of-type	:nth-child()	:target
:fullscreen	:nth-last-child()	:valid
:focus	:nth-last-of-type()	:visited
:focus-within	:nth-of-type()	

- Combinators
 - Group of selectors: A, B
 - Descendant selector: A B
 - Child selector: A > B
 - Adjacent sibling selector: A + B
 - General sibling selector: A ~ B
- Multiple Selectors

2.5 The Cascade

- Cascading: trapsgewijs
- Welke CSS rule wint? De volgorde:
 - Source order
 - Specificity
 - Importance

2.5.1 Source Order

- Volgorde van CSS rules in de CSS file
- Van boven naar beneden
- Wat onderaan in de CSS staat wint van wat er boven staat
- Tenzij de specificity of importance hoger is.

2.5.2 Specificity

Hoe specifiek is een selector? De rangorde der selectors:

1. Element selectors
2. Class selectors
3. ID selectors

Berekening van een waarde door middel van 4 scores:

- **1000:** inline styles
- **0100:** voor elke id er in de selector voor komt
- **0010:** voor elke class, attribute of pseudo class er in de selector voor komt
- **0001:** voor elk element of pseudo element er in de selector voor komt

Selector	Thousands	Hundreds	Tens	Ones	Total specificity
h1	0	0	0	1	0001
#important	0	1	0	0	0100
h1 + p::first-letter	0	0	0	3	0003
li > a[href*="en-US"] > .inline-warning	0	0	2	2	0022
No selector, with a rule inside an element's <code>style</code> attribute	1	0	0	0	1000

2.5.3 Importance

Speciale syntax die al hetgeen hiervoor overschrijft: **!important**

```
1 selector {  
2     property: value !important;  
3 }
```

Zo weinig mogelijk gebruiken!

Hangt ook af van welke stylesheet de declaration in staat:

1. Declarations in de user agent style sheets (browser default styles)
2. Normale declarations in user style sheets (custom styles instellingen door een gebruiker).
3. Normale declarations in author style sheets (onze stylesheets).
4. Important declarations in author style sheets
5. Important declarations in user style sheets

2.5.4 Rule mixin

- Properties overschrijven andere properties
- Niet de volledige CSS rule

2.5.5 Inheritance

- Sommige property values die toegepast worden op een parent element worden overgenomen door de children van dat element.
 - font-family
 - color
 - ...
- Sommige niet
 - margin, padding, border, background image
- Welke wel en welke niet is redelijk logisch
- Volledig overzicht: <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

Je kan dat zelf forceren door:

```
1 selector {  
2     property: value;  
3 }  
4  
5 selector child {  
6     property: inherit;  
7 }
```

2.6 Code conventions

- Geen ID's gebruiken om te stylen
- White space!
- Laatste declaration ook een semi-colon geven;
- Comments
- Shorthand properties

- Lowercase selectors
- Maximum 3 niveaus diep gaan met selectors
- Nog beter: zo veel mogelijk classesz

Zie Theorie 4.2 - CSS.pdf voor voorbeelden

3 CSS Defaults

3.1 User Agent Stylesheet

- Elke browser heeft een standaard stylesheet
- Om de pagina toch iets van stijl te geven
- Verschilt van browser tot browser
- Tegenwoordig heel gelijkaardig
- Maar andere manier van interpretatie

3.2 normalize.css

- Betere consistentie tussen browsers
- Creeert een standaardstijl voor HTML-elementen over alle browsers heen
- Alternatief voor reset.css
- Best practice
- Behoud nuttige browser-standaardinstellingen ipv ze te wissen
- Normaliseert stijlen voor een breed scala aan HTML-elementen
- Corrigeert bugs en algemene browser inconsistenties
- Verbetert de usability met subtiele verbeteringen
- Verklaart de code met behulp van comments en gedetailleerde documentatie

3.3 reset.css

- Reset letterlijk alle elementen
- Alle elementen krijgen dezelfde kleur, dezelfde font-size, margin
- Opnieuw beginnen

3.4 Vendor prefixes

- Vendor = browser maker
- Prefix = toevoeging aan de CSS-property
- CSS browser prefixes
- Support van CSS-features voor dat ze standaard worden. Bv: CSS-grid
- Geldt voornamelijk nog voor oudere browsers zoals IE10

- Prefixed properties eerst, normal properties laatst
- shouldiprefix.com

Prefixes:

Browser	prefix
Android	-webkit-
Chrome	-webkit-
iOS	-webkit-
Safari	-webkit-
Firefox	-moz-
Internet Explorer	-ms-
Opera	-o-

3.5 caniuse.com

- Up-to-date browser support
- Front-end web technologieën
- Desktop en mobiele browsers
- <https://caniuse.com/flexbox>

3.6 Browser support

- Hangt af van project tot project
- Check Google Analytics om te zien met welke browser en OS bezoekers komen
- IE11 usage:
 - Globaal: 1.55%
 - België: 2%

3.6.1 Desktop Browsers

Laatste versie van evergreen browsers - browsers die zichzelf updaten:

- Chrome
- Safari
- Firefox
- Opera

Laatste versie van Microsoft Edge en Internet Explorer

3.6.2 Mobiele Browsers

Standaard browser van de laatste versie van OS.

- iOS 13: **Safari**
- Android 9 & 10: **Chrome**
- IE mobile 11

3.7 Progressive enhancement

- Manier om te ontwikkelen
- Basis level van UX die nodig is voor je applicatie
- Meer geavanceerde functionaliteit als de browser het ondersteunt
- CSS fallbacks
 - = CSS die gebruikt wordt als geavanceerde functionaliteit niet wordt ondersteund door de browser

3.8 Margin collapsing

- Standaard krijgt elk typografisch element een margin-top en bottom
 - **Maar** die worden niet opgeteld
 - Margins van opeenvolgende elementen worden niet opgeteld
 - Margins van elementen in een parent met ook een parent worden niet opgeteld
 - Grootste margin = totale verticale margin
- = collapsing margins

3.9 Margin & Padding baseline system

Verticale white space tussen block level tekst elementen:

- single-direction margin declarations
- Alle margins duwen in dezelfde richting
- margin-bottom
- Veelvoud van de baseline

3.10 Het verschil tussen margin en padding?

- **Padding** voegt witruimte toe binnen de box
- **Margin** voegt witruimte toe buiten de box

3.10.1 Margin & padding system gebaseerd op de baseline

- Vertical white space tussen grote blokken: padding-top en bottom
- padding-top = padding-bottom x 2
- Horizontal white-space tussen columns en rand van de viewport of container: padding-left en -right
- Padding-left en -right: meervoud van de baseline

3.11 Units of measurement: EM vs REM vs PX

- pixels: absoluut. Een pixel is een pixel. Kan niet groter of kleiner worden
- Relatief:
 - em: 1em = 1x de font-size van de parent
 - rem = root em = 1x de font-size van de root (het html element)
 - Worden uiteindelijk ook berekend als pixels door de browser

3.11.1 Voorbeelden

```
1 .voorbeeld{  
2     font-size: 3rem;  
3     max-width: 20em;  
4 }
```

```
1 html {  
2     font-size: 100%; /* 16px */  
3 }  
4  
5 .klasse {  
6     font-size: 2em; /* 16*2 = 32px */  
7 }  
8  
9 .klasse p {  
10    font-size: 2em; /* 32*2 = 64px */  
11 }
```

Zie voorbeelden Theorie 5.1 - CSS defaults

4 CSS Architecture

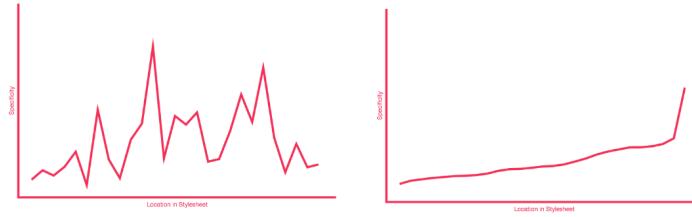
4.1 CSS architecture

- Hoe schrijven we CSS?
- Schaalbaar en uitbereidbaar
- Onderhoudbaar
- De cascade onder controle houden
- Specificity Wars vermijden

4.2 Specificity wars

- Welke selector wint hangt af van:
 - De locatie in de css file
 - De specificiteit van een selector
 - Important

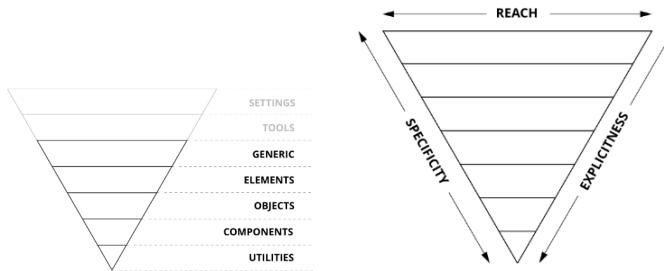
- Locatie en specificiteit door elkaar gebruiken zorgt voor specificity wars
- Issues wanneer zwaardere selectors voor lichtere komen



Figuur 2: Specificity graphs: rechts is beter

4.3 ITCSS

- = Inverted Triangle CSS
- Manier om CSS te schrijven
- In volgorde van specificiteit
 - Eerst element selectors rulesets
 - Daarna class selectors rulesets
- Bouwt verder op elkaar en voegt enkel toe of erft over van wat er voor komt
- Harry Roberts: CSS Wizardry
- <https://www.cssguidelin.es>



Figuur 3: Inverted triangle

- **Generic:** reset/normalize.css, universal selector * voor box sizing
- **Elements:** styling voor pure HTML elementen zoals H1, A, etc...
- **Objects:** herbruikbare class selectors die basis stijlen voor bepaalde patronen definieren (zoals een grid systeem)
- **Components:** herkenbare specifieke UI components met class selectors
- **Utilities:** Class selectors met 1 specifiek doel die alles kunnen overschrijven die in de voor-gaande layers zit. Kan zelfs !important bevatten

4.4 Class naming conventions

- Moeilijkste deel van css schrijven: naamgeving
- Altijd lowercase
- Woorden scheiden door koppelteken
- Namespacing system
- Baseren op ITCSS structuur

Types

- Object classes: .o-
- Component classes: .c-
- Scoping classes: .s-
- Utility classes: .u-
- Javascript hooks: .js-

4.4.1 Object classes .o-

- Herbruikbare class selectors die basis stijlen voor bepaalde patronen definieren (zoals een grid systeem)
- Kunnen voorgedefinieerd zijn in je css boilerplate, herbruikbaar voor ieder project
- .o-layout, .o-list

4.4.2 Component classes .c-

- Herkenbare specifieke UI components
- Zijn specifiek per project
- .c-nav
- .c-button
- .c-author

4.4.3 Scoping classes .s-

- Het is niet altijd mogelijk om aan elk element een class te geven
- CMS component
- Elementen scopen binnen een bepaalde context
- .s-content

4.4.4 Utility classes .u-

- Class selectors met 1 specifiek doel en die alles kunnen overschrijven die in de voorgaande layers zit
- Kunnen voorgedefinieerd zijn in je css boilerplate, herbruikbaar voor ieder project

- .u-text-center
- .u-max-width-large

4.4.5 Javascript hooks .js-

- Manier om duidelijk te maken in de HTML dat aan dit element javascript gekoppeld is. Verwijderen heeft dus geen impact
- Styling en functionaliteit scheiden
- .js-toggle-mobile-nav
- .js-show-next

4.5 BEM-notatie: Block, element, modifier

Manier van naamgeving om object en component classes te groeperen en modular te maken

- **Block:** Op zichzelf staande entiteit die op zich betekenisvol is. De parent node
- **Element:** Een deel van een block dat geen zelfstandige betekenis heeft en dat semantisch is verbonden met het block. De child node.
- **Modifier:** Voegt kleine veranderingen toe op blocks of elements bovenop hun bestaande properties
- **+ITCSS class naming**

In CSS:

- .block
- .block__element
- .block-modifier
- .block__element-modifier

```

1 .c-body{
2     ...
3 }
4
5 .c-body__arm {
6     ...
7 }
8
9 .c-body__arm--left {
10    ...
11 }
12
13 .c-body__arm--right {
14    ...
15 }
```

Real life examples:

- Navigatie
- Buttons

- Grid system
- ...

4.5.1 Rules

- Maximum 1 element diep
- Als een element kan opgedeeld worden in andere elementen: maak een nieuwe block aan
- Niet nesten met css als het niet nodig is
- Modifiers kunnen niet op zich staan

4.6 Goede CSS selector

- Is niet te specifiek
- Is overdraagbaar naar andere elementen
- Is herbruikbaar
- Is locatie onafhankelijk
- Is duidelijk maar toch vaag
- Niet nesten als het niet nodig is

4.6.1 Selector intent

- Wat is het doel van je selector
- Selecting the right things for the right reasons
- Wat wil je stijlen en hoe ga je dat doen?
- **Bijvoorbeeld:** stijlen van de primaire navigatie van een website

4.6.2 Overdraagbaarheid

- Geen **Qualified selectors** gebruiken
- Overdraagbaar naar andere elementen

4.6.3 Locatie onafhankelijk

- Geen stijlen toepassen op elementen op basis van waar ze zich bevinden
- Maar op basis van wat ze zijn
- Zo worden ze locatie onafhankelijk

4.6.4 Naming

- Kies namen die gemakkelijk te onderhouden zijn
- In plaats van heel specifieke namen
- Die duidelijk zijn maar toch vaag

4.7 The Single Responsibility Principle

- Development in het algemeen
- Elk stukje code focust zich op 1 iets
- Kleine CSS classes die zich focussen op 1 iets specifiek
- UI's opdelen tot in de kleinste component die 1 verantwoordelijkheid heeft
- Combinatie van die componenten tot grote complexe constructies

4.8 The separation of concerns

- Lijkt heel goed op *the single responsibility principle*
- HTML & CSS opdelen in duidelijk te onderscheiden section
- Individuele componenten die 1 taak verzorgen
- Samenstellen van al die verschillende componenten vormt de volledige UI
- Bijvoorbeeld een grid system voor layout

4.9 CSS selectors in MCT

- Elements en classes
- Opdelen in objecten en componenten, utilities om heel specifieke zaken te overschrijven
- Namespacing prefixes: .o-, .c-, .u- voor class selectors
- BEM notatie voor class selectors
- Makkelijk te onderhouden en uitbereidbaar
- Specificity wars voorkomen

5 Flexbox

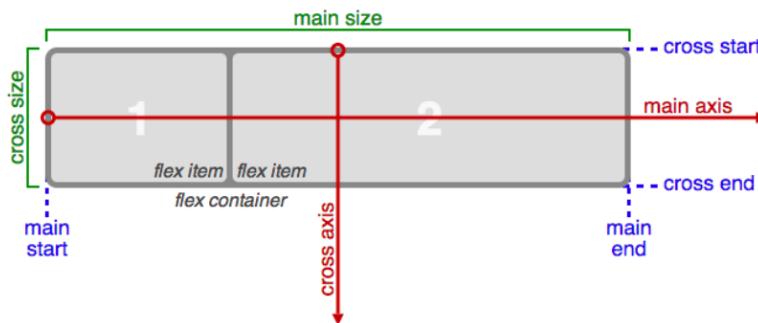
5.1 Zelfstudie

- Interactieve tutorial: <https://scrimba.com/g/gflexbox>
- Demo 1: <https://codepen.io/simoncoudeville-nmct/pen/zYGpJjz>
- Demo 2: <https://codepen.io/simoncoudeville-nmct/pen/bGdLJGj>

5.2 CSS Flexible Box Layout Module

- Creëert een Flexible box zoals een Inline box of een Block box.
- `display: flex;`
- Methode voor het positioneren van elementen in horizontale of verticale stacks.
- Efficiënte manier om elementen naast elkaar te plaatsen, te spreiden en te aligneren
- Voor kleine, 1 dimensionale layouts.
- CSS Grid voor grotere, 2 dimensionale layouts.

- Met de flex layout model kunnen directe children van een flex container in gelijk welke richting gelayout worden.
- Flex container: parent - eigen properties
- Flex items: direct children - eigen properties
- flex-flow directions
- Main-axis
- Cross-axis



Figuur 4: Flex container

5.3 Properties voor de parent (flex container)

- **display:** flex | inline-flex
- **flex-direction:** row | row-reverse | column | column-reverse;
- **flex-wrap:** nowrap | wrap | wrap-reverse;
- **flex-flow:** <'flex-direction'> || <'flex-wrap'>;
- **justify-content:** flex-start | flex-end | center | space-between | space-around | space-evenly;
- **align-items:** flex-start | flex-end | center | baseline | stretch;
- **align-content:** flex-start | flex-end | center | space-between | space-around | stretch;

5.4 Display

Creëert een block of inline element voor de container en een flex layout voor de children.

```

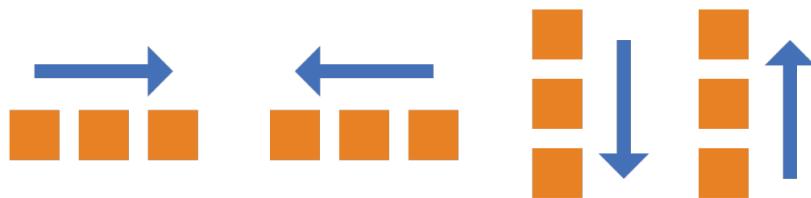
1 display: flex           =   display: block flex;
2 display: inline-flex    =   display: inline flex;

```

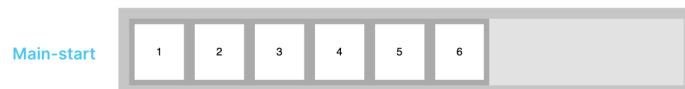
5.5 Flex-direction

- Bepaalt de direction van de main-axis
- Bepaalt de positie van de main-start

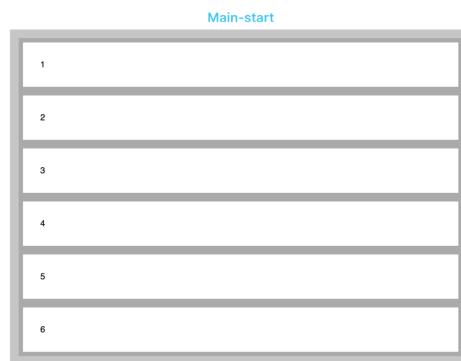
- Bepaalt waar de flow van de items start.
- Bepaalt zo de volgorde van de elementen.
- **Values:**
 - row
 - row-reverse
 - column
 - column-reverse



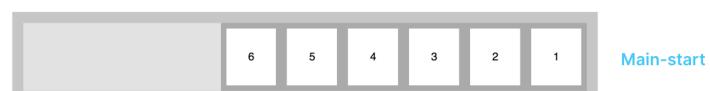
Figuur 5: Flex direction



Figuur 6: Flex direction: row



Figuur 7: Flex direction: column



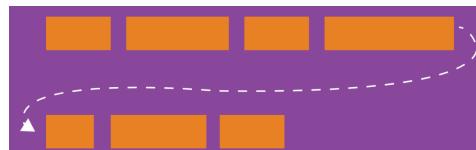
Figuur 8: Flex direction: row-reverse

5.5.1 flex-direction: column-reverse

= identiek aan row-reverse, maar met een column

5.6 Flex-wrap

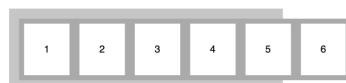
- Bepaalt of een flex-container single-line of multi-line is.
- Bepaalt richting van de cross-axis, in welke richting nieuwe lines lopen.



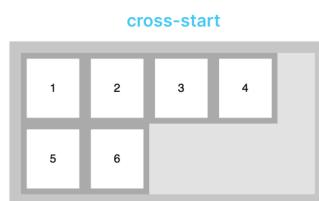
Figuur 9: Flex-wrap

Values:

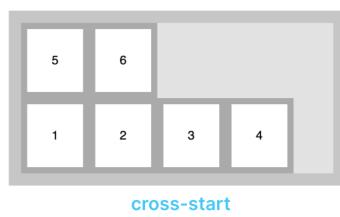
- no-wrap: Items worden altijd naast elkaar gezet. (default)
- wrap: items komen onder elkaar indien nodig
- wrap-reverse: items komen onder elkaar en de cross axis wordt omgedraaid.



Figuur 10: Flex-wrap: no-wrap (default)



Figuur 11: Flex-wrap: wrap



Figuur 12: Flex-wrap: wrap-reverse

5.7 Justify-content

Bepaalt de alignment van de items op de main-axis. Heeft alleen effect als de breedte van de container breder is dan alle items.

Values:

- flex-start: main-start (default)
- flex-end: main-end
- center: in het midden van de main-axis
- space-around, space-between, space-evenly



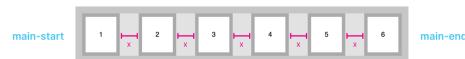
Figuur 13: Justify-content: flex-start



Figuur 14: Justify-content: flex-end



Figuur 15: Justify-content: center



Figuur 16: Justify-content: space-between



Figuur 17: Justify-content: space-around



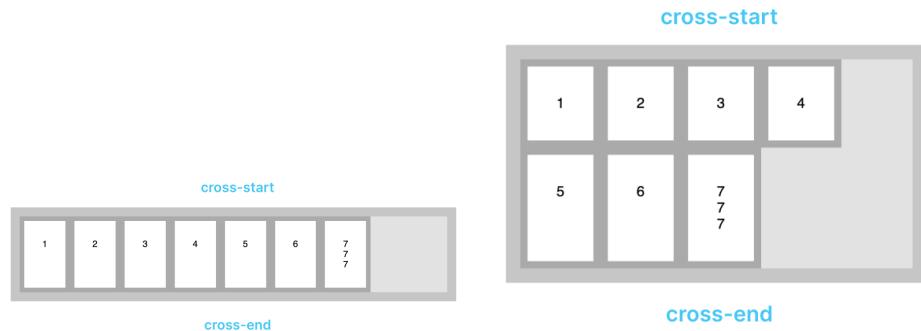
Figuur 18: Justify-content: space-evenly

5.8 Align-items

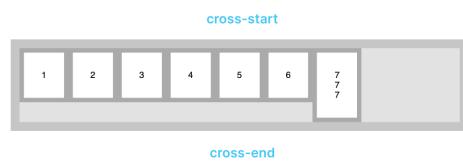
Bepaalt de alignment van de items op de cross-axis

Values:

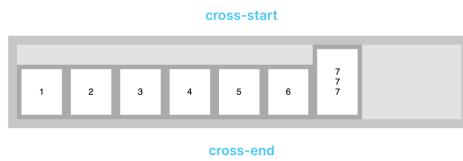
- stretch: maakt items even hoog (default)
- flex-start: cross-start
- flex-end: cross-end
- center: in het midden van de cross-axis
- baseline



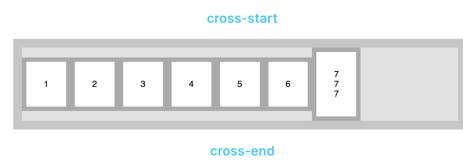
Figuur 19: align-items: stretch (default)



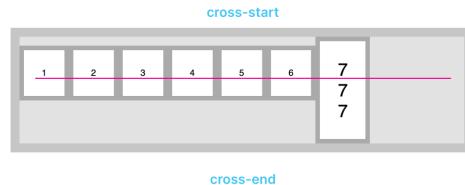
Figuur 20: align-items: flex-start



Figuur 21: align-items: flex-end



Figuur 22: align-items: center



Figuur 23: align-items: baseline

5.9 Align-content

Bepaalt de container van de items op de cross-axis met meerdere lijnen tov de container.

- Werkt enkel als de container hoger is dan alle items samen
- Er moeten meerdere lijnen zijn

Values

- stretch: items op dezelfde lijn worden even hoog (default)
- flex-start: cross start
- flex-end: cross end
- center: in het midden van de cross-axis
- space-around, space-between & space-evenly

5.10 Properties voor de flex items

```

1 order: <integer>;
2 align-self: auto | flex-start | flex-end | center | baseline |
3 stretch;
4 flex-grow: <number>; /* default 0 */
5 flex-shrink: <number>; /* default 1 */
6 flex-basis: <length> | auto; /* default auto */
7 flex: none | [ '<'flex-grow'> '<'flex-shrink'> || '<'flex-basis'>' ] */
8 default 0 1 auto*/

```

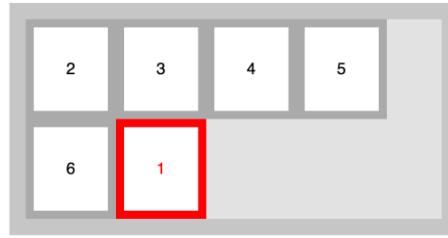
5.10.1 Order

- Bepaalt de visuele order van 1 item.
- Als alle andere items standaard op order: 0 staan wordt het item helemaal naar achter verplaatst.

```

1 .c-layout__item:nth-child(1) {
2   order: 1;
3   border-color: red;
4   color: red;
5 }

```



Figuur 24: Voorbeeld Order

5.11 Align-self

Hierdoor kan de alignment die is gespecificeerd door align-items worden overschreven voor individuele flex items.

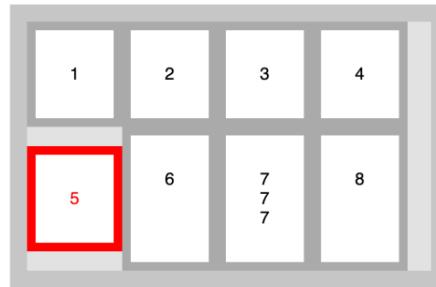
Values:

- auto /* default: volgt align-items */
- flex-start
- flex-end
- center
- baseline
- stretch

Deze values werken exact zoals align-items en align-content, maar dan op individuele flex items.

Bv: align-self: center:

```
1 .c-layout__item:nth-child(5) {
2     align-self: center;
3 }
```



Figuur 25: Voorbeeld align-self

5.12 flex

Hoe items flexen is afhankelijk van:

- Aantal items

- Breedte van de container
- **Content van de items**



Figuur 26: Flex is afhankelijk van de content van de items

```
1 flex: 0 1 auto
```

Standaard voor alle flex-items

Shorthand voor:

```
1 flex-grow: 0
2 flex-shrink: 1
3 flex-basis: auto
```

5.12.1 Flex-grow

```
1 flex-grow: 0
```

- Items worden standaard niet groter om de beschikbare plaats in te nemen op de main-axis.
- Negatieve nummers kunnen niet



Figuur 27: flex-grow: 0 vs flex-grow: 1





Figuur 28: Alle elementen hebben een flex-grow van 1, het 2de element heeft een flex-grow van 2

5.12.2 Flex-shrink

- flex-shrink: 0;
 - Items worden niet kleiner, er ontstaat overflow
- flex-shrink: 1;
 - Items zullen zo klein mogelijk worden voor er overflow gebeurt.
 - Negatieve nummers kunnen niet.

5.12.3 Flex-basis

Bepaalt de default size van een element voor de overgebleven ruimte wordt verdeeld.

- flex-basis: auto; (default)
 - Zorgt voor de best mogelijke layout.
 - Groot genoeg om de content te laten passen.
 - Hangt af van de flex-grow property.
 - Hangt ook af of items wrappen of niet.
- flex-basis: 0;
 - items worden maar zo groot als het langste woord van de content.
- flex-basis: 200px;
 - een item zal nooit kleiner worden dan 200px, eventueel wel groter
- flex-basis: 50%:
 - een item zal nooit kleiner worden dan 50% van de container, eventueel wel groter
- Geen negatieve waarden mogelijk

5.12.4 Resources

- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://scrimba.com/g/gflexbox>
- <https://philipwalton.github.io/solved-by-flexbox/>

6 Kleurtheorie

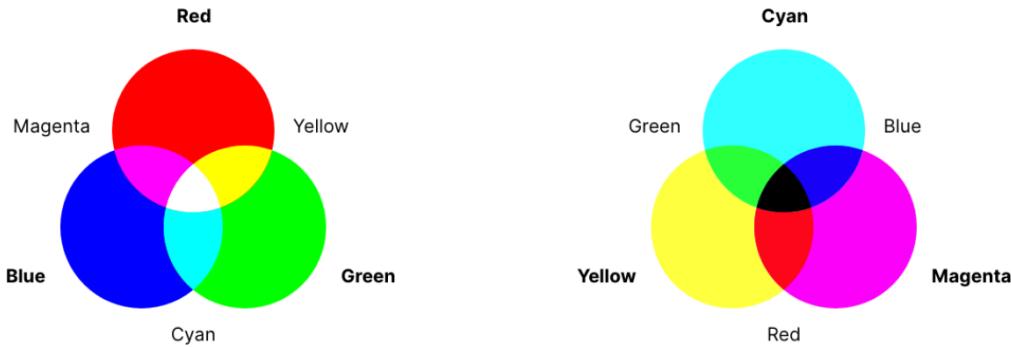
6.1 Hoe worden kleuren opgebouwd?

Additief

- Licht
- RGB
- Alle kleuren samen zijn wit
- Digitaal

Subtractief

- Verf
- CMYK
- Alle kleuren samen zijn zwart
- Drukwerk



6.2 Additive color synthesis

- Gelijke verhouding van 2 primaire kleuren worden gecombineerd om de secundaire kleuren te creeren.
- Gelijke verhouding van 3 primaire kleuren worden gecombineerd om achromatische kleuren (wit, grijsinten en zwart) te creeren.

Alle tinten van het zichtbare lichtspectrum zijn mengsels van verschillende verhoudingen van 1, 2, of 3 van de primaire lichtkleuren. Een RGB-monitor synthetiseert kleuren additief door de rode, groene en blauwe 'phosphor-dots' van elke pixel selectief te verlichten met verschillende intensiteitsniveaus.

Het licht van de 3 phosphor-dots van een pixel smelt samen om 1 enkele kleur te genereren.

6.2.1 Soorten kleuren

- **Primary colors** : RGB
- **Secondary colors** : Bestaat uit gelijke proporties van 2 primaire kleuren
- **Achromatic colors** : Bestaat uit gelijke proporties van alle 3 primaire kleuren
- **Intermediate colors** : Bestaat uit gelijke proporties van opeenvolgende primaire en secundaire kleuren

6.3 Hexadecimale codes

Elke primaire kleur (RGB) heeft een bereik tussen 0 - 255. Voor de berekening van de hexadecimale waarde deel je de numerieke waarde van een kleur door 16 (=eerste deel). De rest van de deling vormt het tweede deel van de hexadecimale waarde. Als de quotiënt of de rest van de deling groter is dan 9, dan wordt deze aangeduid met een letter A tot F (A: 10, B:11, C:12, D:13, E:14, F: 15)

6.4 Color Properties

Chromatic colors : alle kleuren behalve wit, grijstinten en zwart - hebben 3 dimensies of eigenschappen:

1. **Hue** :

- Een waarneembare kleur die overeenkomt met een unieke, dominante golflengte van licht
- De kleur van een object zoals waargenomen door het oog, ontstaan doordat één of twee primaire RGB-kleuren overheersen.

2. **Value/Brightness** :

- De relatieve lichtheid of donkerheid van een kleur

3. **Chroma/Saturation** :

- De relatieve zuiverheid en intensiteit van een chromatic color.
- Minder saturation is het wegnemen van kleur.

6.4.1 HSB: Hue, Saturation Brightness

- Maakt het iets natuurlijker om een kleur te 'maken'
- **Bv:** voor lichtere, donkere en minder gesatureerde kleuren.

6.5 Kleurenpalet samenstellen

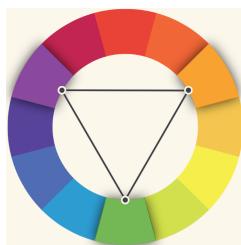
- Op gevoel
- Kleur samples
- Gebruik kleurenregels
- Tools

6.5.1 Kleurenpalet

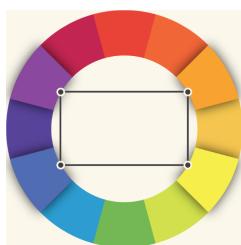
- **Monochromatic of eenkleurig:** 1 kleur kiezen en saturation en brightness aanpassen om meerdere kleuren te maken
- **Analogus of analoog:** naast elkaar liggende kleuren kiezen
- **Complementary of complementair:** tegenovergestelde kleuren die elkaar aanvullen gebruiken
- **Split complementary**



- **Triadic**



- **Tetradic**



6.6 Kleurenpsychologie

Kleur	Betekenis	Associaties volgens Leonardo da Vinci
Blauw	Waarheid, helderheid, waardigheid, macht	Oneindigheid, de hemel, ruimte, reizen, geestelijke liefde, meditatie, eeuwigheid
Bruin	Mannelijkheid, stabiliteit, gewichtigheid	Soliditeit, materie, geborgenheid, ondergang, naderend einde, herfst
Geel	Energie, vreugde, lichtheid van het bestaan	Tijd, zon, maan en sterren, zomer, rijpe oogst, goud, gewin, verstand
Geelgroen	Giftig	Verdorvenheid, verloren zuiverheid, valsheid, haat, afgunst, ziekte, afzondering, verraad
Grijs	Saai	Armoede, ontmoediging, ouderdom, theorie
Groen	Vruchtbaarheid, vrede, natuur aarde	Kracht van ontkiemend zaad, lente, jeugd, onervarenheid, macht, vrede en welvaart, hoop, vredige rust, autonomie
Oranje	Feest	Gezelligheid, gretigheid, weelde, feest
Paars	Leden van het koninklijk huis, rijkdom, verfijning	
Rood	Liefde, passie, warmte, vlammen, vrouwelijk geslacht	Onrust, oorlog, revolutie, bloed, hartstocht, offer, ego, energie, beweging
Violet		Waaierigheid, gezag, rouw, tweeslachtigheid, geheim, conflict, labilititeit
Wit	Licht, zuiverheid, netheid, leegte, dood	Ongerepteheid, volmaaktheid, goddelijke reinheid, onschuld, licht van geest, vrede en leegte
Zwart	Dood, rebellie, duisternis, elegantie	Macht van duisternis, dood, rouw, verderf, vernietiging

- Rood: Bloed, liefde, oorlog
- Oranje: Speels, vrolijk voor kinderen
- Geel: vriendelijkheid, geluk, attentie, waarschuwingsborden
- Groen: groei, natuur, geld
- Blauw: betrouwbaar
- Paars: luxe
- Roze: vrolijkheid, jeugd, onschuld

6.7 Contrast

Zorg dat er een groot genoeg verschil tussen **saturation** en **brightness** is.

6.7.1 Contrast ratio

- Volg accessibility standards outlined in de WCAG, de Web Content Accessibility Guidelines van W3C.
- Contrast ratio voor kleine tekst en grote tekst
- **Kleine tekst:** kleiner dan 24px of 19px bold
- **Grote tekst:** groter dan 24px of groter dan 19px bold
- 2 levels: AA and the more strict AAA
- Geen perfecte berekening: soms fouten

AA

- kleine tekst: minstens 4.5
- grote tekst: minstens 3

AAA

- Strikter dan AA
- kleine tekst: minstens 7
- grote tekst: minstens 4.5

6.7.2 Non-text contrast

Componenten moeten een contrast ratio van minimaal 3: 1 hebben t.o.v. hun achterliggende achtergrond.

6.7.3 Interface elements

Tekst t.o.v. achtergrond: minstens contrast ratio van 3:1 Background t.o.v. achterliggend element: minstens contrast ratio van 3:1 Als het kan:

- Kleine tekst bold zetten
- Background kleur donkerder zetten
- Font size vergroten

6.7.4 Tekst

- minstens 4.5:1
- Grote tekst 3:1

6.8 Kleurenblindheid

- "Druk op groen om door te gaan"= fout
- Zorg ervoor dat kleuren niet de enige manier zijn om belangrijke informatie over te brengen.

6.9 Kleur in UI design

- Donkere kleur variaties worden gemaakt door de helderheid te verlagen en de verzadiging te verhogen
- Lichtere kleur variaties worden gemaakt door de helderheid te verhogen en de verzadiging te verlagen.
- We behouden zo veel mogelijk HUE.

6.9.1 Tools

- <http://color.adobe.com/nl/create/color-wheel>
- <http://coolors.co/>

7 Layout & Grids

7.1 Layout

- De manier waarop een pagina wordt ingedeeld
- Hoe componenten op een pagina worden gepositioneerd
 - Logo
 - Navigatie
 - Verschillende titels en tekst
 - ...

7.2 Alignment

Er voor zorgen dat elk element correct is gepositioneerd t.o.v. andere elementen.

7.3 Grids

- Vormen de basis van een layout
- Zorgen voor orde en consistentie
- Versnellen het design proces
- Horizontal grid
- Vertical grid

7.3.1 Horizontal Grid

- Van links naar rechts
- Columns
 - Percentueel
 - Afhankelijk van de breedte van de container
- Breedtes
- Horizontale whitespace
- Verdeeld layout in grote stukken
- Container
 - Smartphone scherm of browser of de maximum breedte die je instelt
- Gutters (= whitespace tussen 2 items)
 - Vaste waarde (24px, 30px, ...) afhankelijk van spacing systeem

7.3.2 Vertical Grid

- Van boven naar onder
- Rows

7.4 Grid systems

- 12-column grid
 - Deelbaar door 2,3 en 4
 - Minder kolommen beschikbaar
- 16-column grid
 - Deelbaar door 2 en 4
- Niet te veel op vastpinnen: afwijken mag, met CSS kan alles
- $8/16$ of $6/12 = 50\%$
- $3/12$ of $4/16 = 1/4 = 25\%$
- $4/12 = 1/3 = 33.33\%$
- Dus als je in een pagina 3 kolommen nodig hebt dan kan dat, ongeacht het grid system.

7.4.1 Responsive grid

- Default van links naar rechts
- Wat er rechts niet meer bij kan verschuift bij een bepaald breakpoint naar onder

7.4.2 Baseline grid

- Vertical grid
- Van boven naar onder
- Rows
- Hoogte
- Vertical white space boven en onder componenten
- Vertical rhythm
- 1 virtuele lijn die zich herhaald om de X aantal pixels
- Vermenigvuldigen om ...:
 - ... hoogtes te bepalen
 - ... vertical white space boven en onder elementen te bepalen
- In plaats van lukraak iets te plaatsen

7.5 Spacing system

- 1 basis unit waarmee je kan vermenigvuldigen
- Basis voor horizontale white space zoals grid gutters
- Basis voor de verticale white space boven en onder componenten
- **Voorbeeld:**
 - Basis unit = 8px

- Column gutter = $3 * 8px = 24px$
- Vertical margin = $3 * 8px = 24px$
- $8px - 16px - 24px - 32px - \dots$
- Eens dit vast staat, overal in het design gebruiken voor alles wat met groottes en afstand te maken heeft

7.6 White space

- De lege plaats tussen de items
- Padding
- Margin
- <https://scrimba.com/p/prpYaAy/c9BGgbC4>

8 Typografie

- Typography transforms language into a decorative visual element.
- Web design is 95% typografie

8.1 Lettertype vs Font

Lettertype = volledige design met alle varianten (zoals een album) Font = 1 variant ervan (zoals een nummer op een album)

8.2 Classificatie

- Serif
- Sans-serif
- Monospaced
- Decorative
- <https://betterwebtype.com/assets/pdf/better-web-type-font-styles-cheat-sheet.pdf>

8.2.1 Serif



- Schreef
- Zijn karakters met afsluitende streeppjes (schreven) aan het eind van de letterbalken
- Vormen een denkbeeldige horizontale lijn, dat de leesbaarheid verbeterd
- Traditioneel

- Drukwerk: boeken, kranten
- Digitaal gebruiken voor titels en body copy
 - Blogposts
 - Nieuwsartikel
- Niet voor kleine UI elementen

Types:

- Old style
- Transitional
- Neoclassical & Didone
- Slab
- Clarendon
- Glyptic



8.2.2 Sans-serif

- Schreefloze letter
- Letters zonder afsluitende streeppjes aan het eind van de letterbalken
- Nieuwer innovatiever
- Perfect voor alles van tekst
- Perfect voor kleine UI elementen

Types:

- Grotesque & Neo Grotesque
- Square
- Humanistic
- Geometric



8.3 Monospaced

- Alle karakters nemen even veel plaats in
- Neemt dus meer plaats in
- Veel gebruikt om code te schrijven
- Estetisch aantrekkelijk of niet? Kan over gediscussieerd worden
- Voor development: text editors
- In webdesign: kan, maar is een expliciete stijl
- Subtiel gebruik voor details
- Is niet gemaakt om supergroot weer te geven

PROPORTIONAL

Hell just froze over. After seven years of offering no font options to write, iA Writer now comes with a choice. Next to the monospace Nitti you'll find a brand new duospace font.

MONOSPACED

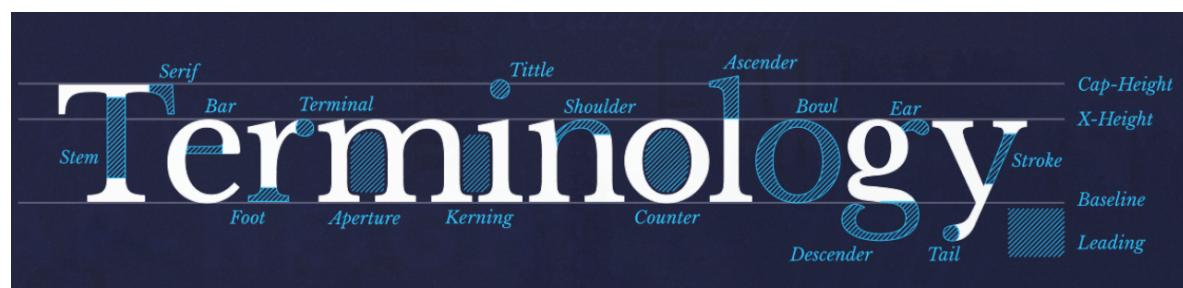
Hell just froze over. After seven years of offering no font options to write, iA Writer now comes with a choice. Next to the monospace Nitti you'll find a brand new duospace font.

Figuur 29: Proportional vs Monospaced

8.3.1 Handwritten, Script & Decorative

- Decoratieve karakters
- Beste voor kleine hoeveelheden tekst
- Titels en headers en meer grafische ontwerpen
- Niet voor UI-elementen

8.4 Anatomie



Figuur 30: [urlhttps://www.supremo.co.uk/typeterms/](https://www.supremo.co.uk/typeterms/)

8.4.1 Mean line & Baseline

- Top en bottom van de karakters
- Virtuele lijnen die ervoor zorgen dat letters even hoog zijn



Figuur 31: Mean line & baseline

8.4.2 x-height

- De hoogte van elk lettertype
- De body van elk karakter is gebaseerd op de body van de lowercase x
- Bepaalt de visuele grootte



Figuur 32: x-height

8.4.3 Ascender & Descender

- Ascender is het gedeelte van bepaalde karakters die boven de mean line uitkomen
- Descender is het gedeelte van bepaalde karakters die onder de baseline uitkomen
- Bepalen de visuele size van een lettertype
- Lettertypes met dezelfde font-size kunnen er daardoor kleiner/groter uitzien

8.4.4 Width

- Condensed: smaller
- Wide: breder
- Variant van een lettertype
- Niet voor body copy
- Zowel serif als sans-serif varianten

8.5 Properties

- font-size
- Weight
- Leading/line-height
- Alignment
- Tracking
- Kerning
- Style
- Ligatures

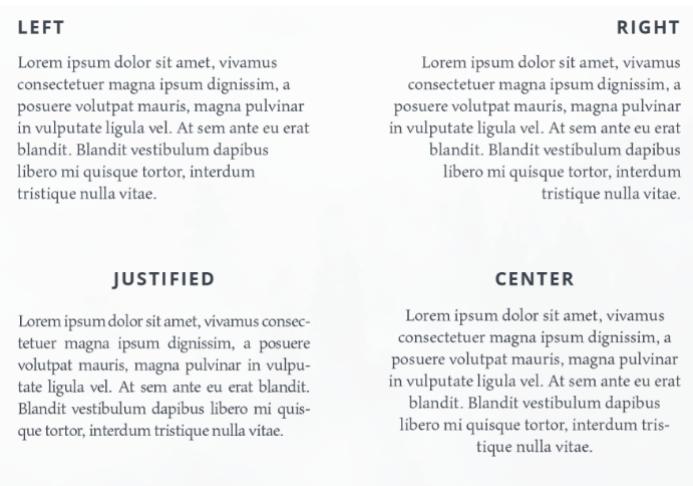
8.5.1 Weight

- De dikte van de karakters
- Varianten van een lettertype
- Ultra-thin - thin - extra-light - light - regular - medium - semi-bold - bold - black - fat
- font-weight: bold
- font-weight: 600

8.5.2 Leading/line-height

- De hoogte van 1 lijn tekst
- Bepaalt de hoogte van een blok tekst
- Line-height in CSS
- Richtlijn voor body tekst
 - 1.5x de font-size
 - font-size: 16px; line-height: 24px;
 - font-size: 20px; line-height: 30px;
- Hangt ook af van de x-height van een lettertype
- Line-height mag nooit kleiner zijn dan font-size

8.5.3 Alignment



Figuur 33: Types alignment

- Left: standaard voor left-to-right text directions
- Right: niet gebruiken voor tekstblokken
- Center: niet gebruiken voor lange tekst
- Justify: opvullen
 - Niet gebruiken op het web: de technologie om woorden af te breken is nog niet goed genoeg: er ontstaan grote gaten

8.5.4 Tracking & Kerning



Figuur 34: Tracking vs Kerning

Tracking	Kerning
De witruimte tussen ieder karakter groter of kleiner maken	De witruimte tussen 2 bepaalde karakters groter of kleiner maken
Bevordert leesbaarheid zeker bij uppercase	Om de witruimte tussen 2 karakters te optimaliseren

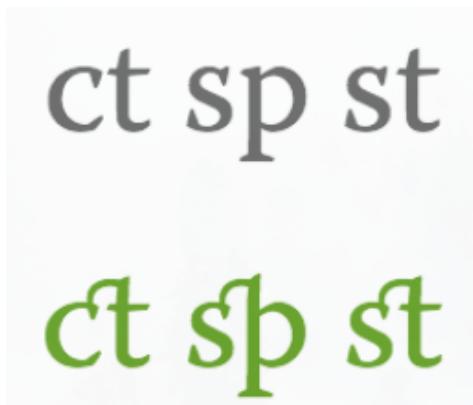
8.5.5 Ligatures

Een ligatuur is een teken dat gevormd is door twee of meer lettervormen in één vorm te schrijven of te drukken. <https://helpx.adobe.com/fonts/using/open-type-syntax.html> <https://practice.typekit.com/lesson/caring-about-opentype-features>



fi fi fi fi fi fi

Figuur 35: Common/standard ligatures



ct sp st
ct sp st

Figuur 36: Discretionary ligatures



room
room

Figuur 37: Contextual alternates

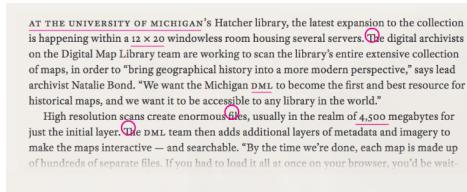


ABCDEFGHIJKLMNOPQRSTUVWXYZ	ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz	abcdefghijklmnopqrstuvwxyz
0123456789!?.	0123456789!?.

Figuur 38: Stylistic alternates / stylistics sets

01 02 03 04 05 06 07 08 09 00	01 02 03 04 05 06 07 08 09 00
11 12 13 14 15 16 17 18 19 10	11 12 13 14 15 16 17 18 19 10
21 22 23 24 25 26 27 28 29 20	21 22 23 24 25 26 27 28 29 20
31 32 33 34 35 36 37 38 39 30	31 32 33 34 35 36 37 38 39 30
41 42 43 44 45 46 47 48 49 40	41 42 43 44 45 46 47 48 49 40

Figuur 39: Tabular numbers



Figuur 40: Features for body text



Figuur 41: Features for display text

8.6 Leesbaarheid

8.6.1 De perfect paragraaf

- font-size: 16px
- Line-height: 1.5x font-size (afhankelijk van de x-height van het lettertype)
- Aantal karakters tussen de 50 en de 75 om de afstand die het oog moet afleggen naar het begin van de nieuwe regel te verminderen

8.6.2 Het perfecte UI lettertype



Figuur 42: Helvetica vs Verdana

- Heeft een hoge x-height: herkenbaar van ver en klein (klein = relatief ver)
- Heeft een duidelijk onderscheid tussen moeilijk te onderscheiden karakters (l en l)
- Android: Roboto

- Apple: San Francisco

8.7 Lettertypes combineren

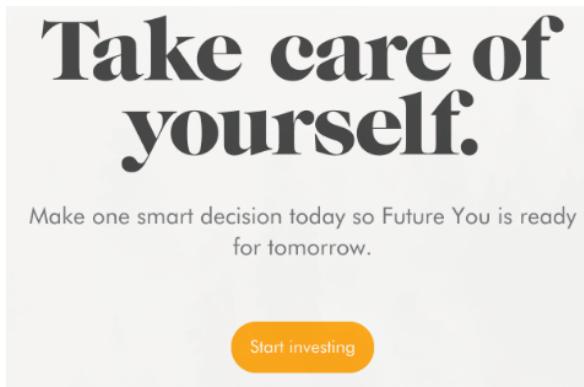
- Gebruik niet te veel verschillende lettertypes in 1 ontwerp
- Voor websites: maximum 2 a 3 lettertypes
- Voor apps: afvragen of het nodig is. Wel spelen met verschillende weights
- Is dus niet altijd nodig
- Lettertypes met dezelfde properties
- Groot contrast
- Families
- Fonts van dezelfde makers
- Op gevoel: geen exacte wetenschap

8.7.1 Lettertypes met dezelfde karakteristieken



Figuur 43: Groot contrast

8.7.2 Groot contrast



Figuur 44: Groot contrast

8.8 Lettertypes vinden

- <http://fonts.adobe.com/>
- <https://fonts.google.com/>
- <https://www.typewolf.com/not-on-google-fonts>

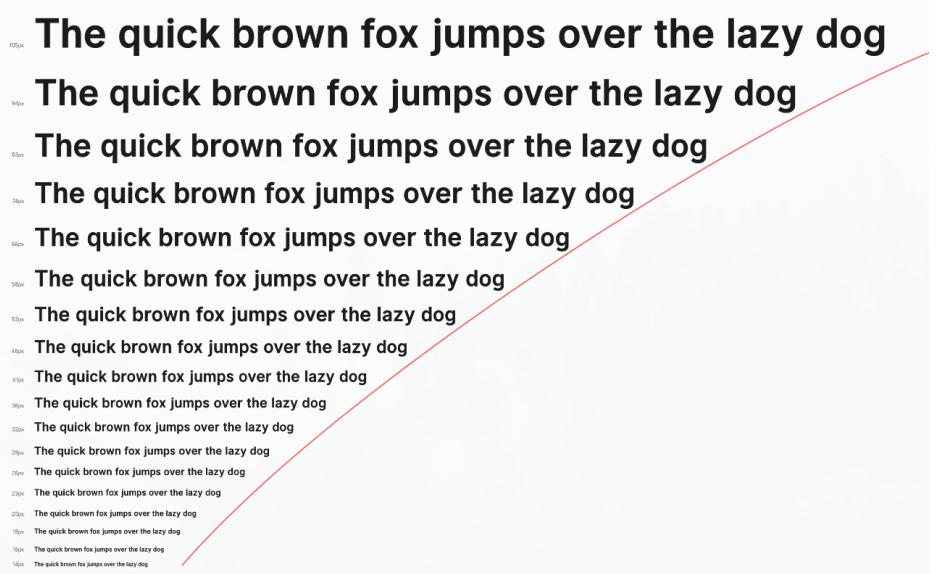
- <http://open-foundry.com/hot30>
- <http://www.fontsquirrel.com/>
- <https://www.theleagueofmoveabletype.com/>
- <https://fontpair.co/index.html>
- Whatfont chrome extension
- Vermijd illegale lettertypes

8.9 Hierarchie

- Opbouw van een pagina/scherm
- Structuur: welke elementen moeten eerst opvallen
- Verschillen tussen:
 - Titels, subtitels
 - Tekst
 - Caption
 - Buttons
 - ...
- Font-size
 - Biggest type first
- Contrast:
 - Width
 - Weight
 - Stijl (classificatie)

8.9.1 Welke font-size kiezen?

- Gebruik een modular scale
- Een reeks van nummers die op een betekenisvolle manier aan elkaar gelinkt zijn
- In plaats van random een cijfer te kiezen
- Minder keuze
- <http://www.modularscale.com/?16&px&1.125>
- Genoeg contrast in:
 - Size
 - Weight
 - Style
 - Color



Figuur 45: Fonts van groot naar klein

9 Visual hierarchie

9.1 Gestalt theorie

9.2 Typografische hierarchie

- 3 levels
 - Contrast tussen levels
 - Size
 - Weight
 - Style
 - Color
1. Primair
 - Headline
 - Core informatie pagina/scherm
 2. Secundair
 - Subtitles
 - Scanbaar maken
 3. Tertiair
 - Body text
 - Captions
 - Smaller UI-elements

Balancing Line Length And Font Size In Responsive Web Design



Simon Coudeville
User Interface designer

Level 3

As we refine our methods of responsive web design,* we've increasingly focused on measure (another word for "line length") and its relationship to how people read. The popularization of the "ideal measure" has led to advice such as "Increase font size for large screens and reduce font size for small screens." While a good measure does improve the reading experience, it's only one rule for good typography. Another rule is to maintain a comfortable font size.

Level 2

How People Read

People read online text to serve their own needs: to find the information they seek, to discover new ideas and to confirm their notions about life.

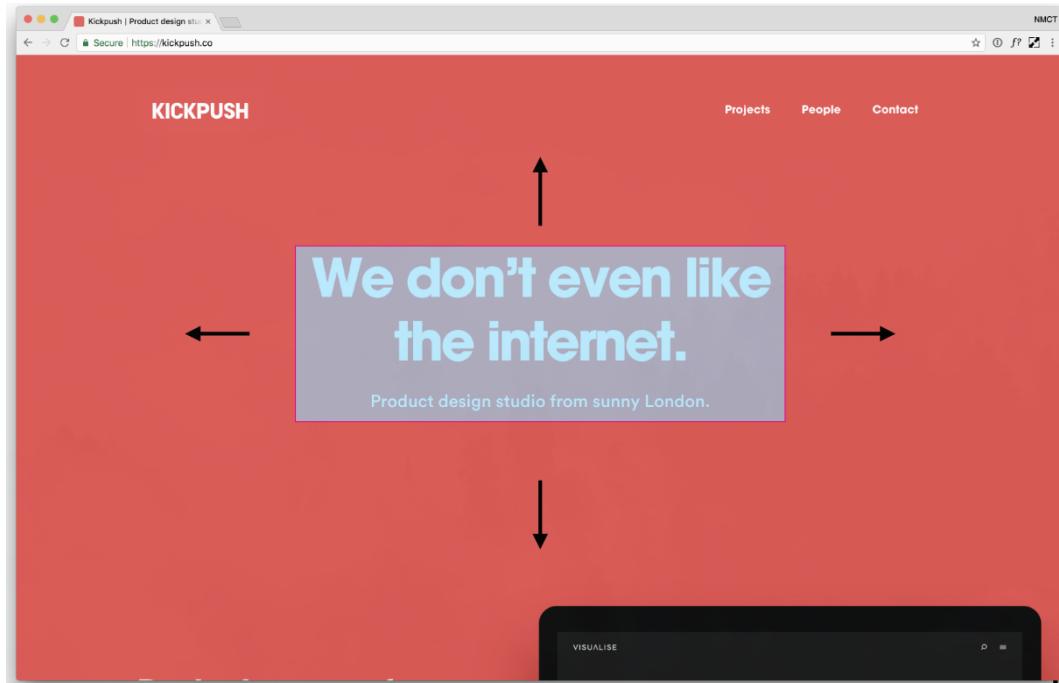
People read in three ways

In 2006, the [Nielsen Norman group](#) released images of heat maps from eye-tracking studies. The areas where people looked at the most while reading are red, areas with fewer views are yellow, and the least-viewed areas are blue. As you can see below, the red and yellow areas form three variations of an F-shaped pattern. These variations aren't surprising because people read in three different ways.

Figuur 46: Voorbeeld typografische hierarchie

9.3 Grootte en gewicht

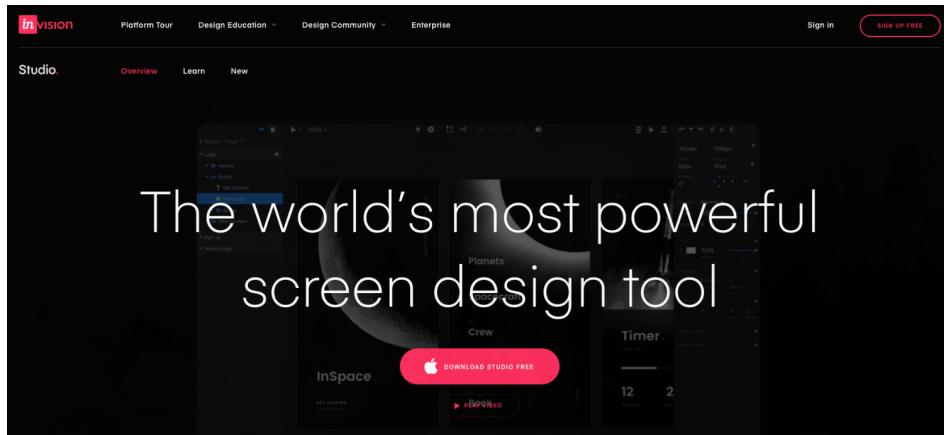
- **Horizontaal:** hoeveel kolommen neemt een element in
- **Verticaal:** de hoogte van een element
- **Whitespace:** hoeveel ruimte krijgt een element



Figuur 47: Voorbeeld grootte en gewicht

9.4 Color

- Felle kleuren voor **interactie en feedback**
- Neutrale, zachte kleuren voor achtergrond, tekst, ...

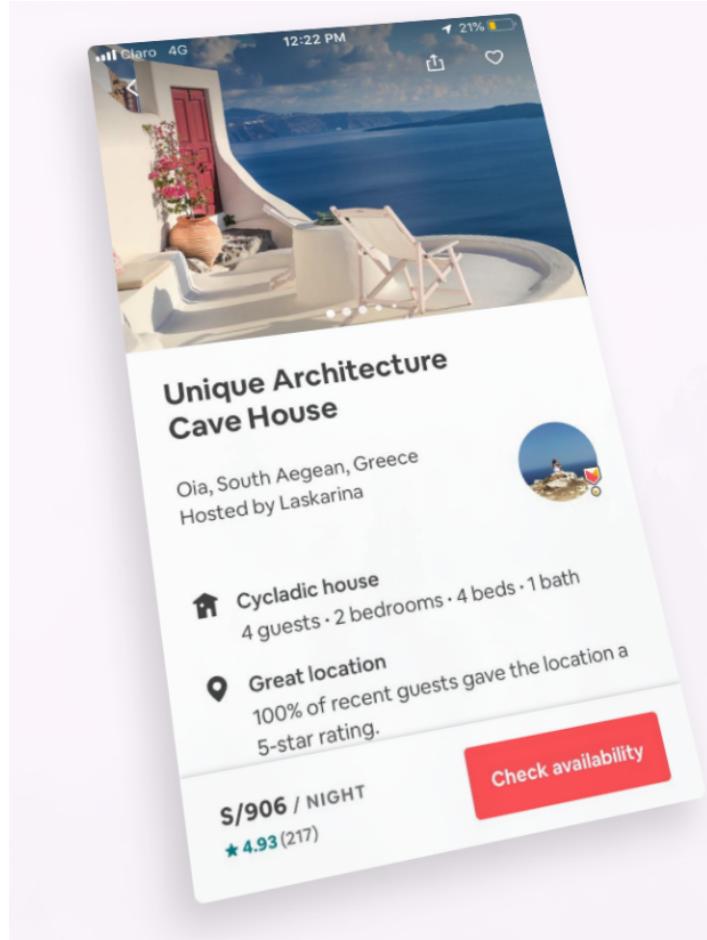


Figuur 48: Voorbeeld kleur

9.5 Alignment en position

- t.o.v. de volledige pagina
 - Een element dat zich afzondert van de globale alignment zal aandacht krijgen

- Above The Fold: wat is er eerst zichtbaar
- t.o.v. het scherm
- Fixed position: elementen die blijven staan

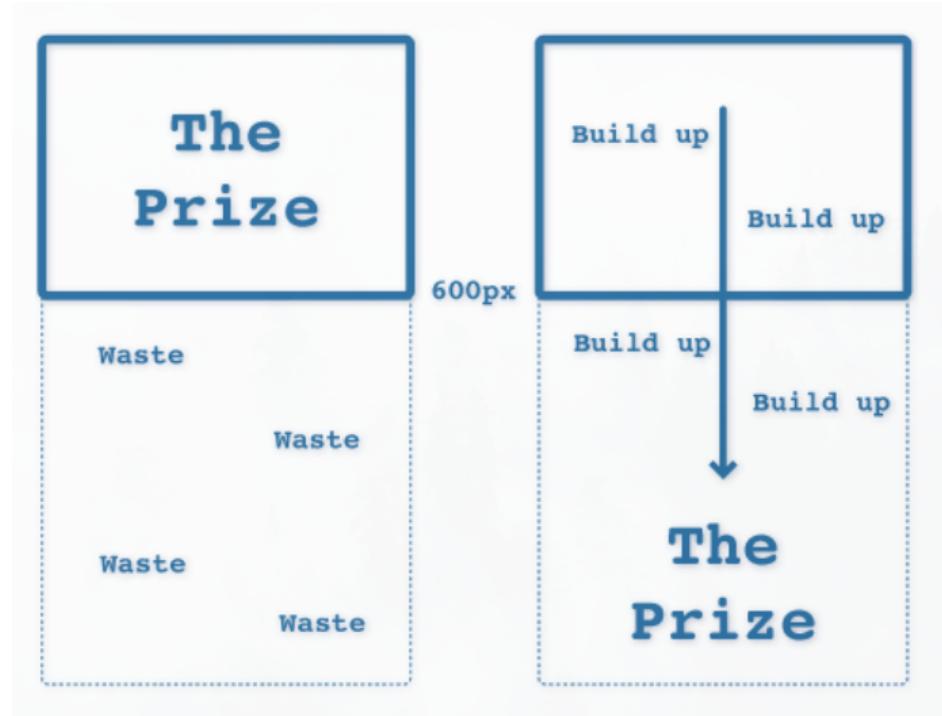


Figuur 49: Voorbeeld position en alignment

9.6 The fold

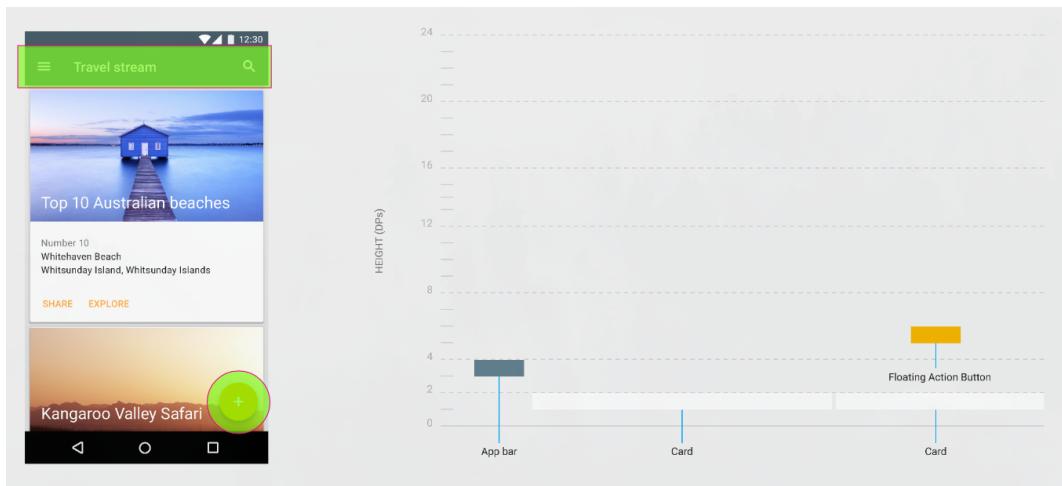
- Naam komt van kranten: als je kranten plooit, moet het belangrijkst artikel voor de fold komen
- Digitaal:
 - het gedeelte van een pagina dat onmiddellijk zichtbaar is
 - De viewport
- Is nog steeds belangrijk
- Maar: iedereen scrollt
- Verkenning simuleren
 - Indicators

- In plaats van alles bovenaan te plaatsen: opbouwen



Figuur 50: Voorbeeld The Fold

9.7 Fixed elements

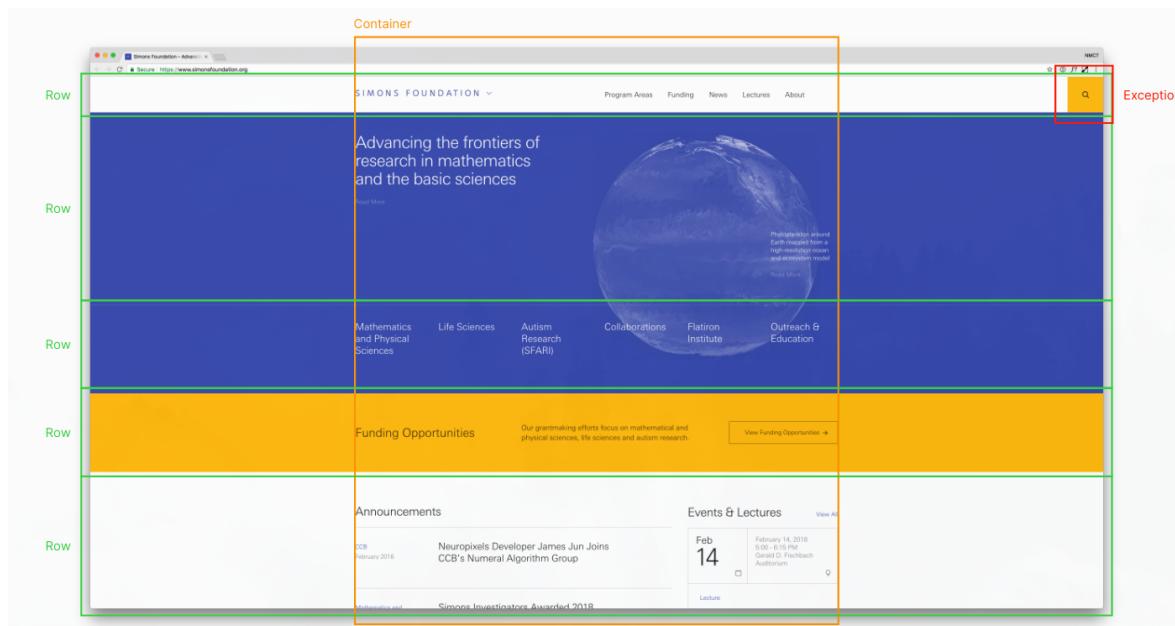


Figuur 51: Voorbeeld Fixed elements

9.8 Rows & Containers

- Elke pagina of scherm bestaat uit rows en containers

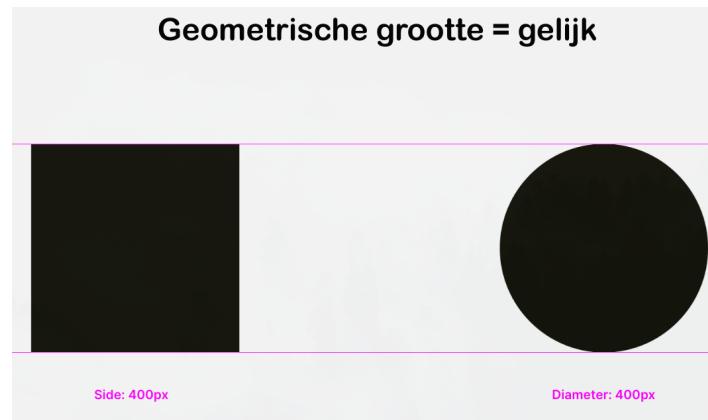
- Rows
 - Horizontaal
 - Grote sectie van een pagina
 - Onderscheid door background & borders
- Container
 - Verticaal
 - Zitten in rows
 - Max-width voor grote resoluties



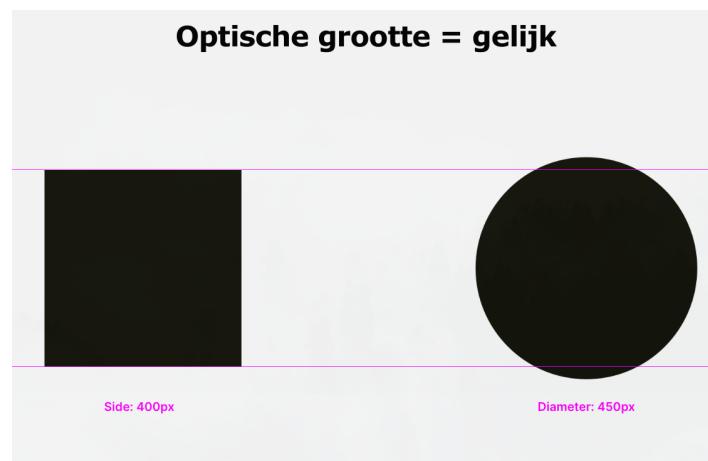
Figuur 52: Voorbeeld Rows & Containers. Let op de exception

9.9 Icons

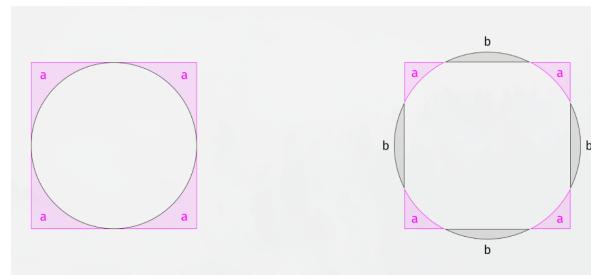
9.9.1 Geometrische en optische grootte



Figuur 53: Geometrische grootte



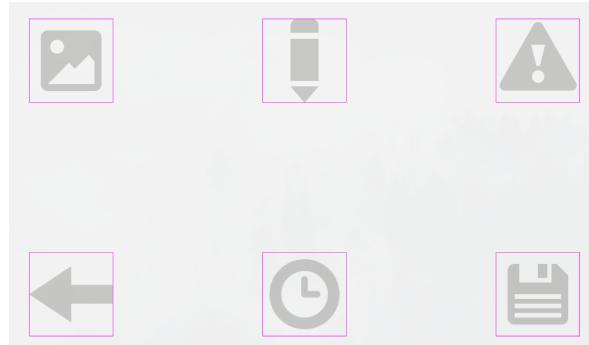
Figuur 54: Optische grootte



Figuur 55: Vierkant vs cirkel

9.9.2 Bounding box

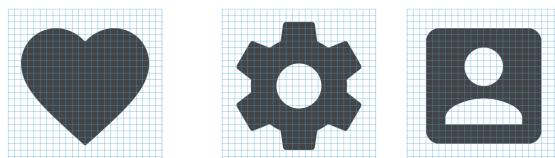
- = De rand die rond elk icoon staat zodat elk icoon dezelfde grootte heeft
- De vorm staat horizontaal en verticaal gecentreerd in de bouding box



Figuur 56: Bounding boxes bij icons

9.9.3 Pixel grid

- Bepaalt de bounding box, dus de grootte van alle iconen in een set
- De basis waar elk icoon wordt op getekend



Figuur 57: Pixel grid

9.9.4 User Interface icons

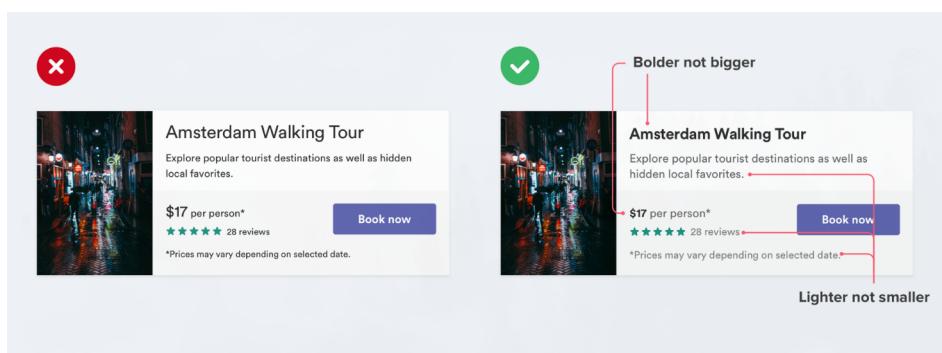
- 1 kleur
- Gebruik altijd icons uit dezelfde set
 - Met dezelfde bounding box
 - Met dezelfde pixel grid
- Altijd SVG icons, geen raster icons: als de icons worden vergroot verliest je zo geen kwaliteit
 - Scalable Vector Graphics
 - Vector gebaseerd ⇒ oneindig groot
 - 1 kleur
 - Basisvormen
 - Makkelijk te importeren in design software
 - Makkelijk te importeren in HTML

9.10 Resources

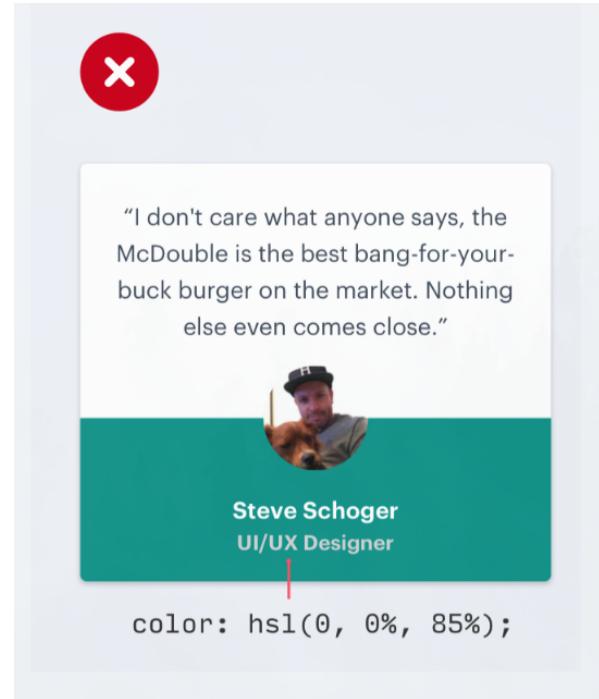
- material.io/icons
- zondicons.com
- fontawesome.com/
- www.entypo.com/
- <https://icons8.com/line-awesome>
- <https://iconsvg.xyz/>
- <https://akveo.github.io/eva-icons/>
- <https://feathericons.com/>
- <https://ikonate.com/>
- <https://iconscout.com/unicons>

9.11 Tips 'n tricks

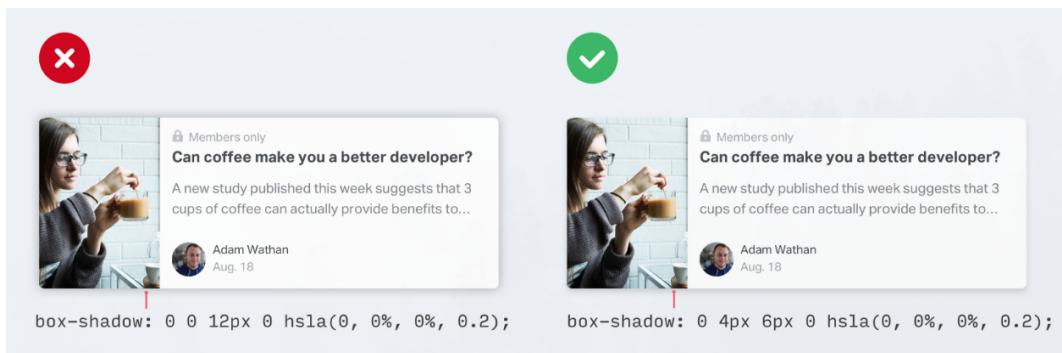
- Gebruik kleur, contrast en weight om hierarchie te creeren ipv size
- Gebruik geen grijze tekst op gekleurde achtergronden
- Maak natuurlijke shadows & borders
- Reduce visual noise: the simplest solution is most likely the right one. Less = more
- Gebruik gekleurde borders in een anders eentonige interface
- Vergroot geen icons die bedoeld zijn voor kleine weergaven
- Niet elke button heeft een achtergrondkleur nodig



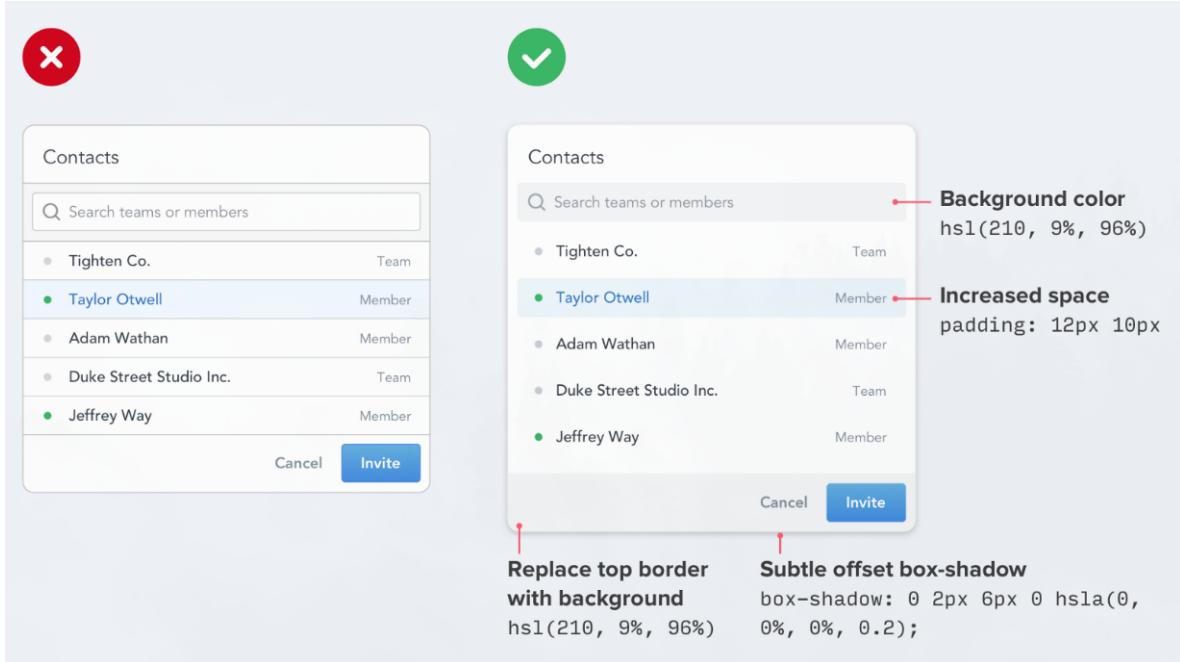
Figuur 58: Gebruik kleur, contrast en weight om hierarchie te creeren ipv size



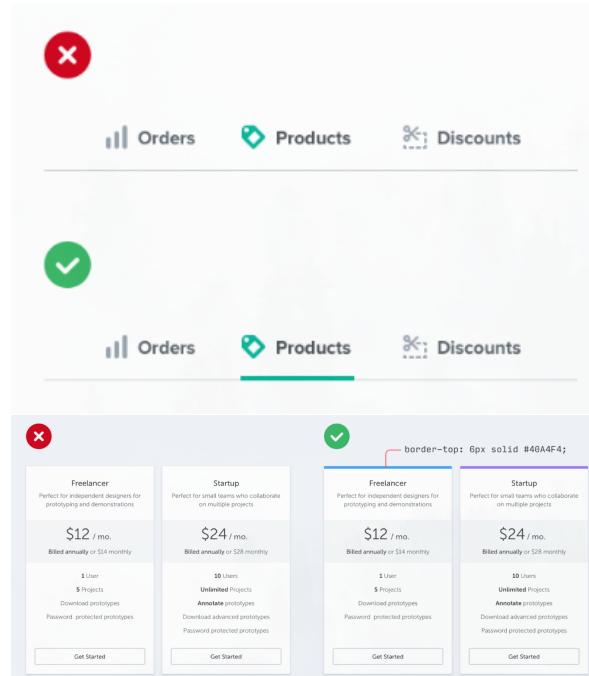
Figuur 59: Gebruik geen grijze tekst op gekleurde achtergronden



Figuur 60: Maak natuurlijke shadows & borders



Figuur 61: Reduce visual noise



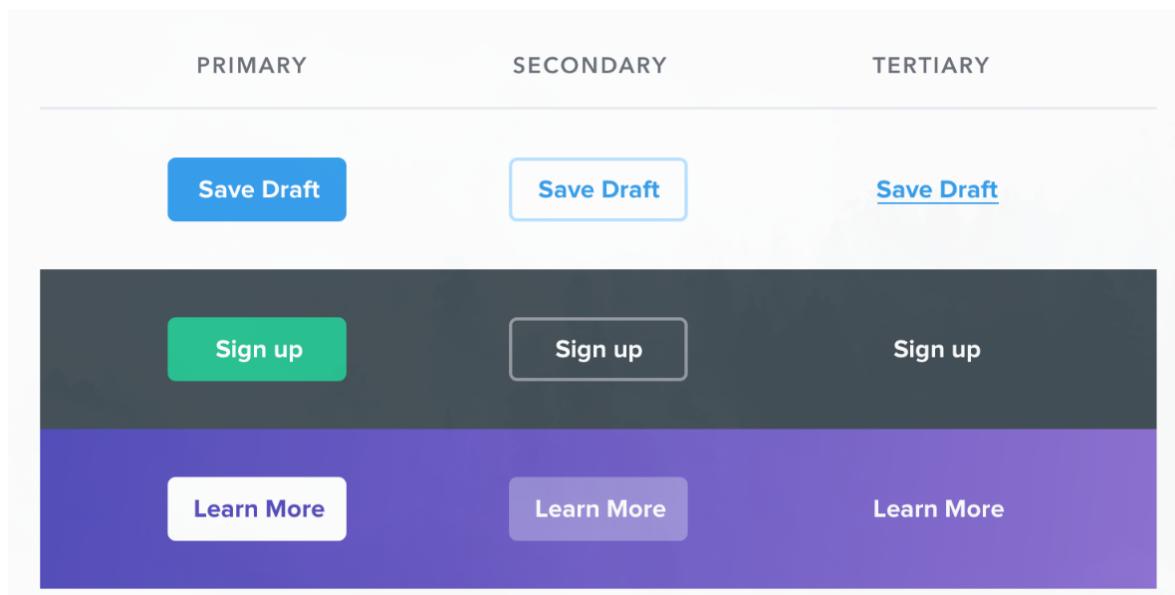
Figuur 62: Gebruik gekleurde borders



Figuur 63: Vergroot geen icons die bedoeld zijn voor kleine weergaven



Figuur 64: Omgekeerd ook niet: er zit te veel detail in om ze te verkleinen



Figuur 65: Niet elke button heeft een achtergrondkleur nodig

10 Document flow & stacking order

10.1 Document flow & stacking order

- Block elementen volgen elkaar op van boven naar beneden en duwen elkaar naar beneden: document flow
- Elk element kan voor of achter een ander element liggen
- Default: hetzelfde als de volgorde in HTML
- Vormen lagen: **stacking order**
- Hoe lager in de markup, hoe hoger in stacking order
- Laatste element “ligt bovenaan”
- Tenzij het een andere position value dan “static” krijgt
- Tenzij je de het verandert van stacking order met z-index

Voorbeeld default stacking order met schaduwen: <https://codepen.io/simoncoudeville-nmct/pen/wZQbZZ>

10.2 Position properties

position:

- static (default)
- relative fixed
- absolute
- sticky

10.2.1 top, right, bottom, left

- Zoals margins, maar dan zonder de rest van de flow te beïnvloeden.
- Bepalen de positie t.o.v. de parent.
- top: 24px;
- right: -12px;
- bottom: 0;
- left: 0;

10.2.2 Position: static

- Default
- Beteekt niet veel. Het element flowt gewoon mee met de pagina zoals het normaal zou doen.
- Static positioned elements worden niet beïnvloedt door de top, bottom, left, right & z-index properties
- Moet je normaal nooit definiëren

10.2.3 Position: relative

- Relatief t.o.v. zichzelf
- Kunnen wel worden beïnvloed door de top, bottom, left, right en z-index properties.
- Staat automatisch hoger in de stacking order t.o.v. elementen met position: static.
- Volgt anders gewoon de document flow
- Wordt een parent voor elementen met een andere position bv.: absolute

Voorbeeld Position: static vs relative: <https://codepen.io/simoncoudeville-nmct/pen/pQomQb>

10.2.4 Position: absolute

- Kan je plaatsen waar je wilt met de top, bottom, left, right properties
- Relatief t.o.v de parent:
 - Een element met 'position: relative' of als er geen zijn:
 - Het html element
- Staat automatisch hoger in de stacking order t.o.v. elementen met position: static.
- Verdwijnt uit de flow: heeft niet langer invloed op andere elementen
- Mee oppassen dus, flexibiliteit verdwijnt.

Voorbeeld Position: static vs relative vs absolute: <https://codepen.io/simoncoudeville-nmct/pen/bQGyQG>

10.2.5 Position: fixed

- Relatief t.o.v. de viewport
- Blijft staan als je scrollt
- Kan je plaatsen waar je wilt met de top, bottom, left, right properties.
- Staat automatisch hoger in de stacking order t.o.v. elementen met position: static.

Voorbeeld Position: static vs fixed: <https://codepen.io/simoncoudeville-nmct/pen/VVwOEG>

10.2.6 Position: sticky

- Mix tussen relative & fixed
- Relatief tot op een bepaald punt, dan wordt het fixed.
- Staat automatisch hoger in de stacking order t.o.v. elementen met position: static.

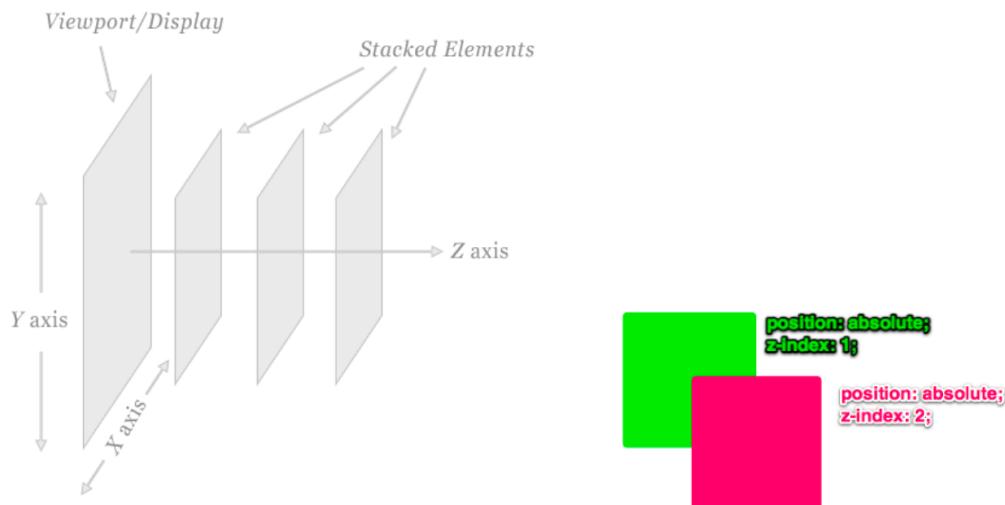
Voorbeeld Position: static vs sticky: <https://codepen.io/simoncoudeville-nmct/pen/dQyEgj>

10.3 z-index property

- Bepaalt de stacking order: in welke volgorde lagen elkaar overlappen
- Z-axis tov X & Y
- Overschrijft de volgorde van de markup

- Werkt enkel op elementen die niet position: static zijn
- Nesting speelt een rol: Als een element B boven element A staat kan een child element van A nooit hoger zijn dan element B.

Oefening Z-index: probeer het witte vak over de header & main te zetten: <https://codepen.io/simoncoudeville-nmct/pen/rQNgqm>



Figuur 66: Z-index



Figuur 67: Rode element kan nooit boven blauwe element komen want het is een child van een element met lagere z-index

10.3.1 Z-index values

- integer, zonder units
- 0 tot 2 147 483 647 (maximum value van een 32bits integer)
- Negatieve nummers zijn toegelaten: z-index: -1;

- Zorgt ervoor dat een child achter zijn parent verdwijnt
- Tenzij de parent ook een z-index heeft

10.3.2 Z-index systeem

- Gebruik iets anders dan gewoon 1, 2, 3, 4, ...
- Gebruik meervouden van 10 of 100 of 1000
- Op die manier heb je nog ruimte om nieuwe elementen 'er tussen te schuiven'
- Children krijgen automatisch dezelfde z-index
- Children krijgen een value die begint met hetzelfde eerste cijfer van hun parent

Voorbeeld Alles: <https://codepen.io/simoncoudeville-nmct/pen/RqwmqE>

11 Responsive design

11.1 Origins

- 2010
- Ethan Marcotte
- A list apart
- <http://alistapart.com/article/responsive-web-design/>
- <http://alistapart.com/d/responsive-web-design/ex/ex-site-larger.html>

11.2 Requirements

- Fluid grids
- Fluid media
- Media queries

9 basisprincipes responsive webdesign:

<http://blog.froont.com/9-basic-principles-of-responsive-web-design/>

11.3 One Web

- **Content parity:** Alle content op alle toestellen gelijk. Don't hide stuff.
- Flexibiliteit: the web is responsive by default. Omarm de fluidity van het web. Geen vaste grote width gebruiken **maar max-width**
- Device agnostic: Denk niet te veel in termen als mobile, tablet, desktop, ...
- Design Interfaces die er goed uitzien en goed functioneren op elk punt van het resolutie spectrum.
- Laat de content breakpoints bepalen
- <http://bradfrost.com/blog/post/beyond-squishy-the-principles-of-adaptive-design/>

11.4 Mobile first

- In plaats van Desktop down
- Respecteert de cascade
- Bouwt verder op in plaats van te overschrijven
- Met de **min-width** media-query
- UX: Focus op wat echt belangrijk is

11.4.1 Voorbeelden

- Desktop first: <https://codepen.io/simoncoudeville-nmct/pen/BGaeGX>
- Mobile first: <https://codepen.io/simoncoudeville-nmct/pen/gQ0JQy>

11.5 Media queries

- De basis voor responsive design
- Laat toe om bepaalde CSS rules te specificeren voor bepaalde omstandigheden
- Logical expression: true or false
- Een media query is true als het media type van de media query matcht met het media type van het device en als alle expressions in de media query true zijn.
- Voorbeelden mediaqueries:
 - mobile CSS rules
 - desktop CSS rules
 - print: wat als je pagina geprint wordt?
- Cascade: alle niet overschreven rules blijven gelden

11.5.1 Media queries syntax

```
1 @media type {...} // bv: print
2 @media (expression) {...}
3 @media type and (expression) {...}
4 @media type and (expression), type and (expression) {...}
```

11.5.2 Media types

- all: All devices listen to this (default)
- screen: Used primarily for color computer screens and smartphones.
- print: Used for paged material and for documents viewed on screen in print preview mode.
- braille: Used for braille tactile feedback devices.
- embossed: Used for paged braille printers.
- handheld: Used for handheld devices (Smartphones and tablets do NOT listen to this!).

- projection: Used for projected presentations, for example projectors.
- speech: Used for speech synthesizers.. (Whatever that may be)
- tty: Used for media using a fixed-pitch character grid (such as teletypes, terminals, or portable devices with limited display capabilities).
- tv: Used for television-type devices (low resolution, color, limited-scrollability screens, sound available).

11.5.3 Media expressions

- width: The width of the current window
- height: The height of the current window
- device-width: The width of the device
- device-height: The height of the device
- orientation: Either landscape or portrait
- aspect-ratio: The aspect ratio of the current window
- device-aspect-ratio: The aspect ratio of the device
- color: The number of color bits per color component
- color-index: The number of available colors on the device
- monochrome: The number of bits per pixel in a monochrome frame buffer
- resolution: The resolution of the device
- scan: Either progressive or interlace
- grid: Is the device grid-based?

Dark mode:

```
1 @media (prefers-color-scheme: dark) { ... }
```

11.5.4 Media queries overview

Check welke media queries van toepassing zijn op jouw toestellen:

<http://cssmediaqueries.com/overview.html>

11.5.5 Media queries in de praktijk

Meest courant:

```
1 @media (min-width: breakpointX) { ... }
```

Af en toe:

```
1 @media (max-width: breakpointY) { ... }
```

11.6 Breakpoints

- **Geen** device specifieke breakpoints
- **Wel** content specifieke breakpoints: een breakpoint als de content niet langer gemakkelijk te gebruiken is
- Wanneer het design op een punt gekomen is dat het breekt.
- Op die manier hoef je niks te herschrijven als er een nieuw device uitkomt met weer een nieuwe viewport
- <https://medium.freecodecamp.org/the-100-correct-way-to-do-css-breakpoints-88d6a5ba1862>

```
/* Extra small devices (portrait phones, less than 576px)
   No media query since this is the default in mobile first design
*/

/* Small devices (landscape phones, 576px and up) */
@media (min-width: 576px) {}

/* Medium devices (tablets, 768px and up) */
@media (min-width: 768px) {}

/* Large devices (landscape tablets, desktops, 992px and up) */
@media (min-width: 992px) {}

/* Extra large devices (large desktops, 1200px and up) */
@media (min-width: 1200px) {}

/* If needed extra extra large devices (extra extra large desktops, 1400px and up) */
@media (min-width: 1400px) {}
```

Figuur 68: Breakpoints

11.7 Responsive images

- Fluid media:

```
1 img { max-width: 100%; }
```

- Performantie & laadtijd
 - Afbeeldingen van verschillende groottes en filesize, zodat laden sneller gaat
 - Under development, nog geen standaard
 - The srcset attribute
 - The picture element

```
<picture>
    <source srcset="img/jpg/img-1.jpg" media="(min-width: 992px)">
    <source srcset="img/jpg/img-1-landscape@2x.jpg" media="(min-width: 576px)">
    
</picture>
```

Figuur 69: Het picture element

- Art direction
 - <http://usecases.responsiveimages.org/#proposed-solutions>



Figuur 70: Art direction

12 Webfonts

12.1 @font-face

- Laat toe om custom fonts in te laden op een webpagina
- Niet langer gelimiteerd door fonts die op de computer van de gebruiker staan
- Geeft de browser instructies om het font te downloaden waar het gehost staat
- Past het toe zoals gespecificeerd in CSS rules

```

@font-face {
    font-family: 'MyWebFont';
    src: url('myfont.woff2') format('woff2'),
         url('myfont.woff') format('woff');
}

html {
    font-family: 'MyWebFont', Fallback, sans-serif;
}

```

Figuur 71: @font-face CSS syntax

12.1.1 Verschillende formats

- .ttf
- .svg
- .woff / .woff2
- .eot (=embedded-opentype)

De meeste browsers ondersteunen woff(2).

12.2 WOFF/WOFF2

- Staat voor: **Web Open Font Format**
- Speciaal ontwikkeld voor webgebruik, door Mozilla
- Laden sneller dan andere formaten
- WOFF2 is de volgende generatie van WOFF en is nog beter gecomprimeerd

12.3 SVG/SVGZ

- Stands for: Scalable Vector Graphics (Font)
- Enige formaat dat wordt ondersteund door Safari 4.1 en lager op iOS
- Worden niet ondersteund door Firefox, IE of IE Mobile
- Firefox heeft ondersteuning volledige laten vallen om te focussen op WOFF
- SVGZ is de gezippte versie van SVG

12.4 EOT

- Stands for Embedded Open Type
- Ontwikkeld door Microsoft (de eerste support van @font-face)
- Werkt enkel op IE8 en lager

12.5 Font-Family linking

- 1 font name
- Verschillende font-weight
- Verschillende font-style
- Opletten voor **faux bold** of **faux italics**

```
@font-face {  
    font-family: "Droid Serif";  
    font-weight: 400;  
    src: url("../fonts/DroidSerif-Regular-webfont.woff") format("woff");  
}  
  
@font-face {  
    font-family: "Droid Serif";  
    font-weight: 400;  
    font-style: italic;  
    src: url("../fonts/DroidSerif-Italic-webfont.woff") format("woff");  
}  
  
@font-face {  
    font-family: "Droid Serif";  
    font-weight: 700;  
    src: url("../fonts/DroidSerif-Bold-webfont.woff") format("woff");  
}  
  
@font-face {  
    font-family: "Droid Serif";  
    font-weight: 700;  
    font-style: italic;  
    src: url("../fonts/DroidSerif-BoldItalic-webfont.woff") format("woff");  
}
```

Figuur 72: @font-face: verschillende styles en weights

12.6 Font-weight

- Browser defaults:
 - font-weight: 400 = normal
 - font-weight: 700 = bold
- Extra weights afhankelijk van font
- Thin, Extra-Light, Light, **Regular**, Medium, Semibold, **Bold**, Extra-Bold, Black
- Css: 100, 200, 300, **400**, 500, 600, **700**, 800, 900

12.7 Licenties

- Fonts maken kost **veel** tijd
- Daarom zijn sommige fonts niet gratis
- Kwaliteit ligt hoger op vlak van: gewichten, variaties, ligatures en andere font-features

12.8 External Hosting Services

- Doen het zware werk voor jou
- Embed/include code

- <http://www.fonts.google.com>
- <http://www.fonts.adobe.com> (betalend)
- <http://www.fonts.com> (betalend)

12.9 Fallback fonts

Als een font niet kan geladen worden, wordt het volgende font in de rij gebruikt:

```
1 font-family: "Droid Serif", Helvetica, sans-serif
```

12.10 System UI fonts

- Elk platform heeft zijn eigen system UI font
- Font stack die ervoor zorgt dat afhankelijk van het platform het default UI font gebruikt en niet het default van de browser
- Minder downloads
- Minder controle
- <https://css-tricks.com/snippets/css/system-font-stack/>

```
1 font-family: -apple-system, BlinkMacSystemFont, Helvetica, sans-serif
```

12.11 Resources

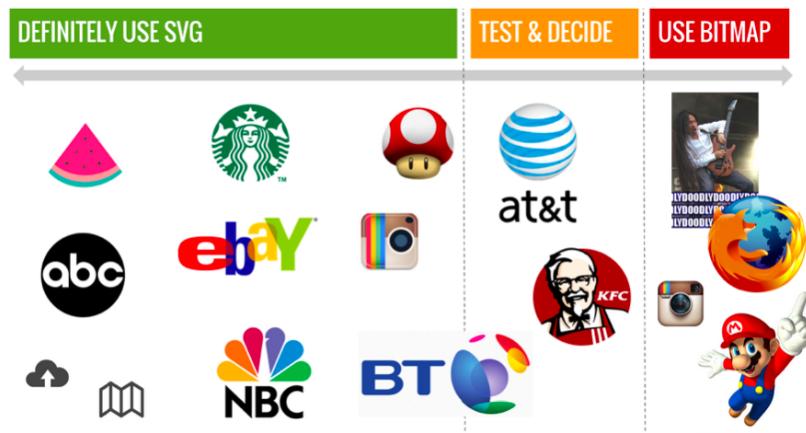
- <https://css-tricks.com/snippets/css/using-font-face/>
- <https://github.com/grillitype/web-fonts-guide>
- <https://www.typewolf.com/>
- <https://www.typewolf.com/google-fonts>
- <https://chrome.google.com/webstore/detail/whatfont/jabopobgcpjmedljpcaablpmlmfcogm?hl=en>

13 SVG

- Scalable Vector Graphic
- XML based markup
- Wiskundig geformuleerde instructies hoe een shape of een curve getekend moet worden
- Sharp & flexible (scalable)
- Shapes, icons, logo's, simpele illustraties
- Voorbeeld: <https://codepen.io/chriscoyier/pen/0243a0b2269653439308ae8ccadfaf23d0>

13.1 Raster images (bitmaps)

- Niet uitvergrootbaar zonder verlies aan detail (resolutie bepaalt scherpte)
- .jpg, .png, .gif
- Goed voor foto's



Figuur 73: SVG vs Bitmap: wanneer welk gebruiken?

13.2 Using SVG

-
- CSS background-image
- Inline SVG: beste manier, als je er voor de rest niets meer wil mee doen

13.2.1 Voorbeelden

- <https://codepen.io/chriscoyier/pen/7d5d25d7a93f10035b9b19c6f4fce516>
- <https://codepen.io/chriscoyier/pen/GAekv>

13.3 viewBox

- Definieert aspect ratio
- Definieert origin of SVG coordinate system
- Unitless in plaats van pixels
- viewBox="x, y, width, height"

13.3.1 Voorbeelden

- <https://codepen.io/simoncoudeville/pen/qovpv0>
- <https://codepen.io/simoncoudeville-nmct/pen/VRdpbj>

13.4 SVG shapes

- Rectangles
- Circle
- Ellipse
- Line
- Polyline
- Polygon
- Path
- <https://codepen.io/simoncoudeville-nmct/pen/qQBGLY>

13.4.1 Rectangle

- <rect x="60" y="10" rx="10" ry="10" width="30" height="30"/>
- x: The x position of the top left corner of the rectangle.
- y: The y position of the top left corner of the rectangle.
- rx: The x radius of the corners of the rectangle
- ry: The y radius of the corners of the rectangle

13.4.2 Circle

- <circle cx="25" cy="75" r="20"/>
- r: The radius of the circle.
- cx: The x position of the center of the circle.
- cy: The y position of the center of the circle.

13.4.3 Ellipse

- <ellipse cx="75" cy="75" rx="20" ry="5"/>
- rx: The x radius of the ellipse.
- ry: The y radius of the ellipse.

13.4.4 Line

- <line x1="10" x2="50" y1="110" y2="150"/>
- x1: The x position of point 1.
- x2: The x position of point 2.
- y1: The y position of point 1.
- y2: The y position of point 2.

13.4.5 Polylines

- Polyline's are groups of connected straight lines. Since that list can get quite long, all the points are included in one attribute:
- <polyline points="60 110, 65 120, 70 115, 75 130, 80 125, 85 140, 90 135, 95 150, 100 145"/>
- x & y coördinaten van 1 punt
- Komma's zijn niet verplicht

13.4.6 Polygon

- A lot like Polyline's but the path automatically returns to the first point for you at the end, creating a closed shape.
- <polygon points="50 160, 55 180, 70 180, 60 190, 65 205, 50 195, 35 205, 40 190, 30 180, 45 180"/>
- x & y coördinaten van 1 punt
- De laatste rechte lijn gaat in dit geval van x45 y180 naar x50 y160

13.4.7 Path

- Using a path element you can draw rectangles (with or without rounded corners), circles, ellipses, polylines, and polygons. Basically any of the other types of shapes, bezier curves, quadratic curves, and many more.
- <path d="M20,230 Q40,205 50,230 T90,230"/>
- d: A list of points and other information about how to draw the path.
- <https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Paths>
- <https://css-tricks.com/svg-path-syntax-illustrated-guide/>
- <https://codepen.io/chriscoyier/pen/NRwANp>

13.5 Styling SVG content using CSS

- Elements in an SVG document can be styled using CSS
- Most visual characteristics and some aspects of element geometry are controlled using CSS properties.
- Elk element kan een class attribuut krijgen
- + animeerbaar met CSS
- <https://codepen.io/simoncoudeville-nmct/pen/ZmENVe?editors=1100>

13.5.1 Useful SVG attributes

- Fill
- Stroke
- Stroke-width
- <https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute>

- Demo: <https://codepen.io/simoncoudeville-nmct/pen/ZmENVe>

13.6 Exporting SVG

- Niet de bedoeling dat je complexe illustraties of logo's zelf gaat typen
- Exporteren als svg met XD
- XD export is niet optimaal
- Optimaliseren met <https://jakearchibald.github.io/svgomg/> of handmatig optimaliseren
- <title> toevoegen

13.7 Resources

- <https://css-tricks.com/lodge/svg/>
- <https://svgontheweb.com/>
- <https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial>
- https://docs.google.com/presentation/d/1CNQLbqC0krocy_fZrM5fZ-YmQ2JgEADRh3qR6Rb00Gk/edit#slide=id.g334f2b22d_41
- <https://css-tricks.com/scale-svg/>
- <https://css-tricks.com/svg-path-syntax-illustrated-guide/>

14 Coole CSS properties & values

14.1 Float

- Elementen die na een floating element komen vloeien er rond
- Values: left, right
- Lange tijd de enige manier geweest om layouts te creeëren
- <https://codepen.io/simoncoudeville-nmct/pen/gQ0Jdy>

14.2 Border-radius

- The border-radius property defines the radius of the element's corners.
- Values: top-left, top-right, bottom-right, bottom-left
- Shorthand: 1 value voor alle corners of 2 values voor top-left/right en bottom-right/left
- Kan ook gebruikt worden om van een vierkant element een cirkel te maken door: border-radius: 50%
- <https://codepen.io/simoncoudeville-nmct/pen/VVZgON>
- <https://9elements.com/io/css-border-radius/>

14.3 Shape-outside

- Manier om content te wrappen rond complexe vormen in plaats van enkel rechthoekig
- <https://codepen.io/simoncoudeville-nmct/pen/bQGymG>
- Values: <https://developer.mozilla.org/en-US/docs/Web/CSS/shape-outside>
- Browser support! <https://caniuse.com/#search=shape-outside>

14.4 Filter

- De filter property voegt visuele effecten toe aan elementen. Zoals: blur, brightness, contrast, sepia, saturate.
- <https://codepen.io/gregh/pen/GrJNdJ>
- Instagram filters met CSS: <https://una.im/CSSgram/>

14.5 Background

- Background-image: png, jpg, or gradient
- Background-position
- Background-repeat
- Multiple background images
- <https://codepen.io/simoncoudeville-nmct/pen/KrPJYW>
- Background-attachment
- Background-size
 - <https://codepen.io/simoncoudeville-nmct/pen/YRKBgM>
- Background-clip
 - <https://codepen.io/gregh/pen/dNyWRP>

14.6 box-shadow

- Creeert een slagschaduw achter of binnen een element
- Values:
 - h-offset
 - v-offset
 - blur
 - spread (optional)
 - color
 - inset (optional)
- Multiple box-shadows
- <https://codepen.io/simoncoudeville-nmct/pen/EOYrML>

14.7 text-shadow

- Creeert een schaduw achter tekst
- Values:
 - h-offset
 - v-offset
 - blur
 - color
- Multiple text-shadows
- <https://codepen.io/chriscoyier/pen/urkCd>

14.8 Transform

= Visueel manipuleren

14.8.1 Skew

- <https://codepen.io/team/css-tricks/pen/d7be823bca05649502f63fc8490c6d93>
- <https://codepen.io/team/css-tricks/pen/cb9ab0a908241806e3eff88b57f15693>

14.8.2 Rotate

- <https://codepen.io/team/css-tricks/pen/d31be2118a19782d2c4fcfb048351ccf>
- rotateY: <https://codepen.io/gregh/pen/jy0rd0>

14.8.3 Translate

= verplaatsen

- <https://codepen.io/team/css-tricks/pen/b9df78cff3ab6b64b5925b10c1e52d9e>

14.8.4 Scaling

- <https://codepen.io/team/css-tricks/pen/d5d50055d95ceecbf94068e75a502c97>

14.8.5 Perspective

- <https://codepen.io/gregh/pen/wgBKJX>

14.9 Transition

- Animatie tussen 2 states als een property verandert van value
- Bijvoorbeeld on hover
- Transition values:
 - transition-property

- transition-duration
- transition-timing (easing)
- transition-delay
- Tutorial: <http://css3.bradshawenterprises.com/transitions/>

14.9.1 Transition-timing-function

- Easing: de manier waarop de transition gebeurt
- Lineair: altijd even snel
- Hebben een naam, bijvoorbeeld: ease-in.
- Kunnen ook beschreven worden als een cubic-bezier()
- <https://matthewlein.com/tools/ceaser>
- <http://easings.net/nl>
- Meer info over easing:

<https://medium.com/motion-in-interaction/animation-principles-in-ui-design-understanding-easing-bea0>

14.10 Animation

- Lopen door ongeacht of er iets verandert of niet
- Keyframes definieren met @keyframes
- Geef je animatie een naam
- Roep je animatie op met animation-name
- https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation3
- Oefeningen: https://www.w3schools.com/css/css3_animations.asp

14.11 Calc

- Value om complexe berekeningen te doen met verschillende units
- Laat de browser het zware werk doen
-
- Voorbeelden:
 - width: calc(50% - 24px);
 - width: calc(100% / 6); (16.6666667%)
- Nesting is mogelijk: width: calc(((1536px/12*7) + ((100vw - 1536px)/2)) + 104px);
- <https://codepen.io/simoncoudeville-nmct/pen/MzgLZv>

14.12 v value

- vh & vw
- Viewport-height & viewport-width
- Procent van de viewport
- bv: height: 100vh;
- <https://codepen.io/simoncoudeville-nmct/pen/MzgLZv>

14.13 currentColor keyword

- Soort van variabele in CSS
- Neemt de kleur over van de color property van zichzelf of een parent
- Voor elke value die color kan declareren
- Ook voor color values die niet kunnen overgeërfd worden.
- bv: box-shadow: 0 0 0 1px currentColor;
- <https://codepen.io/simoncoudeville-nmct/pen/JePxeB>

14.14 Rgba and HSLA value

- Andere manier om kleur te schrijven met CSS
- Extra value: alpha waarde: opacity
- Bv: background-color: rgba(255,255,255,.5); wit/50% transparant
- Bv: color: hsla(0,0%,0%,.5); zwart/50% transparant. (Pas op! Niet hetzelfde als HSB in XD.)
- <https://codepen.io/simoncoudeville-nmct/pen/JePxeB>

15 Labo

15.1 Objects

15.1.1 o-row

Maakt een horizontale rij die de volledige viewport vult en padding toevoegt rond kinderen (links, rechts en boven)

```
1 .o-row {  
2     position: relative;  
3     padding: 24px 24px 0;  
4     display: flow-root;  
5 }
```

15.1.2 o-container

Maakt een horizontaal gecentreerde container die de maximumbreedte bepaalt van de hele pagina. Containers zitten in rows of in andere containers.

```
1 .o-container {
2     margin-left: auto;
3     margin-right: auto;
4     width: 100%;
5     max-width: 90em; /* (90*16px = 1440px), als 16px de grootte van het standaardfont
6     is */
7 }
```

15.1.3 o-layout

Om via flexbox elementen naast elkaar te plaatsen in een grid. Met modifiers bepaal je de gutters.

```
1 .o-layout {
2     display: -webkit-flex;
3     display: -ms-flexbox;
4     display: flex;
5     flex-wrap: wrap;
6 }
7
8 .o-layout__item {
9     width: 100%;
10 }
```

Modifiers:

```
1 /* Dit haalt padding weg uiterst links en uiterst rechts,
2 zodat er enkel padding is TUSSEN de elementen */
3 .o-layout--gutter {
4     margin: 0 -12px;
5 }
6
7 /* Padding links en rechts van elk child met de klasse o-layout__item */
8 .o-layout--gutter > .o-layout__item {
9     padding: 0 12px;
10 }
11
12 /*
13 .o-layout--gutter-sm = 6px
14 .o-layout--gutter-lg = 24px
15 */
```

o-layout-row-reverse Plaatst de rij in omgekeerde volgorde.

```
1 .o-layout--row-reverse {
2     flex-flow: row-reverse;
3 }
```

15.2 Utilities

15.2.1 u-max-width

```
1 .u-max-width-sm {
2     max-width: 36em !important;
3 }
4
5 .u-max-width-md {
6     max-width: 45em !important;
7 }
8
9 .u-max-width-lg {
10    max-width: 60em !important;
11 }
12
13 .u-max-width-xl {
14    max-width: 75em !important;
15 }
16
17 .u-max-width-none {
18     max-width: none !important;
19 }
```

15.2.2 u-flex-basis, u-flex-grow & u-x-of-y

Met u-x-of-y kan de flex-basis (grootte) van een o-layout item aanpassen. De breedte is dan x/y van de volledige breedte (bv 4/5e)

```
1 .u-flex-basis-auto {
2     flex-basis: auto !important;
3 }
4
5 .u-flex-grow-1 {
6     flex-grow: 1 !important;
7 }
8
9 .u-1-of-2 {
10    flex-basis: calc(100% / 2) !important;
11 }
```

Je kan dit ook instellen vanaf bepaalde breakpoints. Items worden dus x/y van de volledige breedte vanaf een bepaalde min-width.

```
1 @media (min-width: 576px) {
2     .u-1-of-2-bp1 {
3         flex-basis: calc(100% / 2) !important;
4     }
5 }
6
7 /* bp1 = 576px */
8 /* bp2 = 768px */
9 /* bp3 = 992px */
10 /* bp4 = 1200px */
```