

KISS MY CSS

FRONTEND DEVELOPMENT LES 2

VORIGE WEEK

- Geschiedenis van HTML
- Nieuwe, semantische elementen
- Formulieren

Even heel kort samenvatten waar we het ook al weer over hebben gehad vorige week. Niet te lang bij stilstaan want er is heel veel materiaal te behandelen.

CSS

Cascading Stylesheets

The process of combining several style sheets and resolving conflicts between them.

Håkon Wium Lie

Wat is een cascade? Volgens woordenboeken is het een waterval. Een opeenvolging van dingen. Vooral één richtingsverkeer dus. De code wordt lineair gelezen. Lekker simpel. De definitie van Håkon Wium Lie, een van de bedenkers van CSS is lekker duidelijk. Cascading heeft een paar toffe eigenschappen tot gevolg:

OVERERVING

```
body {  
  color: green;  
  border: 1px solid green;  
}  
p {  
  /* Kleur is vanzelf green,  
  Border is niet vanzelf 1px solid green */  
  border: inherit;  
}
```

Sommige eigenschappen worden vanzelf overgeërfd, andere niet. Het lijkt dat vooral eigenschappen om tekst en tabellen te stijlen overgeërfd worden. Met het woordje 'inherit' kan je het forceren.

CSS IS TOLERANT

```
bdoy {  
  colour: #gghhjj;  
}
```

Een andere fijne eigenschap is het feit dat code die niet begrepen wordt gewoon genegeerd wordt. Dit lijkt een *minor detail*, maar dit is essentieel om te kunnen innoveren:

CSS IS DUS HANDIG

```
body {  
  background: green; /* oude browsers */  
  background: hsla(0,50%,50%,.5); /* nieuwe browsers */  
}
```

Zonder deze eigenschap hadden we nooit nieuwe CSS kunnen gebruiken. Wat niet begrepen wordt wordt genegeerd, wat wel begrepen wordt overschrijft datgene wat er aan vooraf ging. Hoera, vooruitgang!

CSS IS DUS HANDIG

```
body {  
  background: green;  
}  
body {  
  background: white;  
}
```

```
.night-modus {  
  background: black;  
}
```

Door een class toe te voegen kunnen we er dus voor zorgen dat dingen er anders uit gaan zien.

BLING!

De overgang van de vorige slide naar deze is met CSS gedaan.

VROEGER

```
body {  
  font: 100%/1.5 helvetica, arial, sans-serif;  
  color: papayawhip;  
  border: 1em solid olivedrab;  
}
```

CSS is ooit ontwikkeld om tekstdocumenten mee vorm te geven. Het is in de eerste 15 jaar eigenlijk nauwelijks geüpdatet. Omdat we toch ook mooie dingen wilden maken gingen we hacken. Ook bij CSS3 gold: mensen willen dit doen, laten we er voor zorgen dat dat dan ook kan.

DUS

```
div {  
  border-radius: .3em;  
}
```

Mensen willen ronde hoekjes? Dat kan.

OPACITY

```
opacity: 1;
```

LIVE CODING! De CSS die je in deze box typt beïnvloedt de box! Laat hier zien dat je ook `hsla(0,50%,50%,0.5)` kunt doen om alleen de achtergrond transparant te maken. Toon het duidelijke verschil met `opacity`. Als je zin hebt kun je ook nog `background: currentColor;` laten zien.

BORDER-RADIUS

```
border-radius: 20px;
```

Laat zien dat je alle hoeken apart kunt aanpassen door bijvoorbeeld 20px 80px 40px 0 te typen. Laat ook horizontale en verticale border radius zien door dingen als '20px / 80px;' of 20px 100px 0 100% / 80px of percentages te laten zien. Hier gaan ze mee oefenen, niet te lang op doorgaan dus!

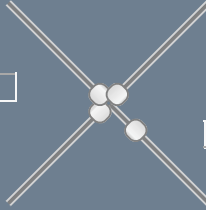
BOX-SHADOW

```
box-shadow: 5px 5px 20px black;
```

Laat zien waar elke waarde voor dient. Voeg ook het vierde cijfer, spread, toe. Laat ook zien dat hsla() of rgba() voor mooiere resultaten kan zorgen. Toon wat het woordje 'inset' doet: box-shadow: inset 0 0 15px 14px white; Laat ook multiple backgrounds zien. bijvoorbeeld:

```
box-shadow: 0 0 0 2px black, 0 0 0 4px green, 0 0 0 6px red, 0 0 0 8px yellow, 0 0 0 10px olivedrab, 0 0 0 12px purple, 0 0 0 14px white;
```

hsl(0,0%,100%)



☐ wtf

all the way

3D

MOTHER EFFING TEXT-SHADOW

```
text-shadow: #ffffff 1px 1px, #ffffff 2px  
2px, #ffffff 3px 3px, #ffffff 4px 4px,  
#ffffff 5px 5px, #ffffff 6px 6px, #ffffff 7px  
7px, #ffffff 8px 8px, #ffffff 9px 9px,  
#ffffff 10px 10px, #ffffff 11px 11px;
```

My Presets [[save new preset \(#\)](#)]

Klik vooral even op de All the way button

Made by [Paul Irish](http://paulirish.com) (<http://paulirish.com>)

TRANSFORM

```
transform: rotate(1deg);
```

Laat zien: `scale(1.1)` en `skew(0deg, 20deg)` en `translate(10px, 10px)`. Laat ook combinaties zien, of leg ze uit dat combinaties mogelijk zijn. `transform-origin: 0 0`; is ook goed om te laten zien.

TRANSITIES

1

soms doet deze slide het niet. Open dan <http://vvg.gr/ih> in een nieuwe tab Je kan alleen een enkele propertie animeren, bijvoorbeeld

```
textarea { transition: color 1s; }
```

```
textarea:hover { background: green; color: yellow }
```

Je kan ook een transition delay toevoegen:

```
textarea { transition: color 1s 1s; }
```

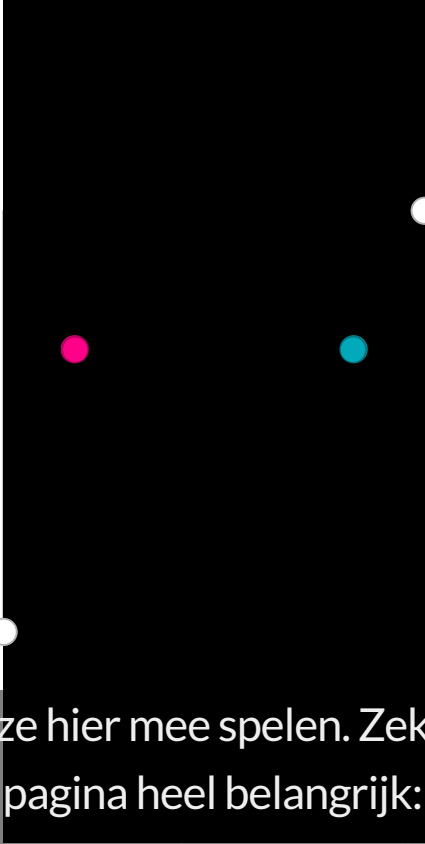
Voor easing kan je keywords gebruiken zoals linear, ease-in, ease-out.

TRANSITIES

DONATE

Made by Lea Verou with care ✨ About



PROGRESSION



cubic-bezier(.17, .67, .83, .67)

Preview & compare

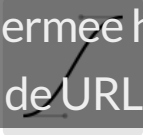

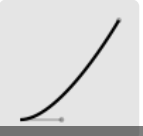


Duration: 1 second GO!



Library

IMPORT EXPORT

Click on a curve to compare it with the current one.



ease-in-out

Laat ze hier mee spelen. Zeker voor de mensen die motion design tof vinden is deze pagina heel belangrijk: hiermee hebben ze fine control over hoe een component animeert. Geef ze de URL cubic-bezier.com/

SELECTORS

UNIVERSAL SELECTOR

```
* {  
  color: green;  
  border: 1px solid green;  
}
```

Wat vooral belangrijk is is dat studenten begrijpen wát ze kunnen vormgeven. Het is interessant als een designer weet dat je een element wat na een ander element staat kunt stijlen (maar niet iets wat er voor staat). Of dat je de laatste drie dingen kunt stijlen (hoewel we daar pas volgende week mee aan de slag gaan).

TYPE SELECTOR

```
body {  
  background: white;  
}
```

DESCENDANT SELECTOR

```
body div {  
  background: white;  
}
```

DIRECT CHILD SELECTOR

```
body > div {  
  background: white;  
}
```

```
body div {  
  background: white;  
}
```

Wat is het verschil. Dat lijkt me een prima vraag om even in de klas te stellen.

ADJACENT SIBLING SELECTOR

```
label + input {  
  background: none;  
}
```

Dit is een hele krachtige waarmee je relaties kunt leggen.

ATTRIBUTE SELECTOR

```
[for="email"] {  
  background: slategray;  
}
```

```
a[href^="https://"] {  
  background: url(veiligelink.svg);  
}
```

```
img[src$=".jpg"] + p {  
  font-style: italic;  
}
```

Geen classes nodig, gewoon dingen stijlen aan de hand van attributen die ze al hebben. Kan je dingen *semantisch* stijlen. Veilig icoontje als een link met https// begint. De paragraaf die direct na een jpg staat anders stijlen dan die na een gif staat. ETC. Dit moet allemaal in een half uur passen.