



**NEW MEDIA &  
COMMUNICATION  
TECHNOLOGY**

# **Front End Web Development**



**NEW MEDIA &  
COMMUNICATION  
TECHNOLOGY**

**CSS, alles wat je moet weten 😊**

# CSS wordt meestal ervaren als moeilijk.

Is het ook 😊

Omdat het  
uitgebreid is

Omdat men het meestal  
verkeerd aangeleerd krijgt

Omdat CSS  
misbruikt

# Cascading

Waterval effect  
Overervig  
Herbruiken

# Box Model

# Denk in blokken

Om met CSS te werken moet je denken dat er een soort onzichtbare box is rond elk element

- Deze blokken moet je dan een opmaak geven

**Lorem ipsum dolor sit amet**

Consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Aenean nec lorem. In porttitor. Donec laoreet nonummy augue. Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.

# Blokken selecteren

Een opmaak in CSS maken, bestaat door een aantal rules aan een geselecteerde blok toe te kennen

Een rule bestaat uit 2 delen

1. De selector
2. De declaratie (bestaat uit property en value)

# Selectors



Selector



p

{

font-family: Verdana;

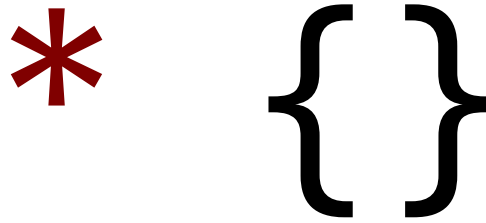
}

Declaration

p {  
    font-family: Verdana;  
}

The diagram illustrates the components of a CSS rule. A red bracket above the text 'font-family' is labeled 'Property' in a red box. A blue bracket below the text 'Verdana' is labeled 'Value' in a blue box.

# Universal Selector



Selecteert elk element in het document

# Type Selector

**h1** {}

Selecteert elk element dat matched met de naam

# Type Selector

**h1, h2, h3** {}

Selecteert elk element dat matched met de naam

## Class Selector

`.active {}`

Selecteert elk element dat matched met de waarde van het class attribuut

# ID Selector

#intro {}

Selecteert elk element dat matched met de waarde van het ID attribuut

# Child Selector

`li > a`

Selecteert elke `<a>` die `<li>` als rechtstreekse parent heeft



# Descendant Selector

p a { }

Selecteert elke <a> die ergens voorkomt in een paragraaf

# Adjacent Sibling Selector

**h1 + p**{ }

Selecteert de eerste <p> die achter een <h1> tag komt

# General Sibling Selector

**h1 ~ p{ }**

Selecteert elke <p> die achter een <h1> tag komt, maar op het eerste niveau

# Attribuut Selector

[title]

Selecteert alle elementen die het title attribuut bevatten

## Attribuut Selector

[target=\_blank]

Selecteert alle elementen die de waarde blank in het target attribuut hebben

## Attribuut Selector

[href~=.jpg]

Selecteert alle elementen waarvan jpg in het href attribuut voorkomt

## Attribuut Selector

[href\*=.jpg]

Selecteert alle elementen waarvan jpg in het href attribuut voorkomt

## Attribuut Selector

[href|=http]

Selecteert alle elementen waarbij de waarde van href start met http



## Attribuut Selector

`[href^=http]`

Selecteert alle elementen waarbij de waarde van href start met http

## Attribuut Selector

[href\$.jpg]

Selecteert alle elementen waarbij de waarde van het href attribuut eindigt met jpg

# Pseudo class Selector

**:link**

Selecteert alle elementen, meestal a, waarop nog niet geklikt is

## Pseudo class Selector

**:visited**

Selecteert alle elementen, meestal a, waarop wel geklikt is

## Pseudo class Selector

**:active**

Selecteert alle elementen die momenteel actief zijn

## Pseudo class Selector

**:hover**

Selecteert alle elementen waarop de muiscursor staat

## Pseudo class Selector

**:focus**

Selecteert alle elementen, meestal input, die focus hebben

## Pseudo class Selector

**:first-letter**  
p:first-letter

Selecteert elke eerste letter



## Pseudo class Selector

**:first-line**

p:first-line

Selecteert elke eerste lijn

## Pseudo element Selector

**:first-child**

`div:first-child`

Selecteert het eerste element van een div

# Pseudo element Selector

:before  
div:before

Voegt content toe voor elk <div> element

# Pseudo element Selector

:after  
div:after

Voegt content toe achter elk <div> element

## Pseudo element Selector

**:nth-child(n)**  
**P:nth-child(2)**

Selecteert elke <p> die het tweede kind is van zijn parent

[http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)

**Specificiteit**

Elke goede CSS'er kan over specificiteit  
meespreken!!

Is wel een van de moeilijkste zaken in CSS





**Specificiteit:** bepaalt welke selector voorrang krijgt

# Specificity

Bepaalt welke rule toegepast wordt

- Aan elke rule wordt een gewicht gehangen
- Soort formule die de uitkomst bepaalt

Pas op: de specificiteit kan per CSS versie anders zijn

Elke selector heeft zijn plaats in de hiërarchie

Als 2 selectoren hetzelfde element refereren dan wint degene met de hoogste specificiteit

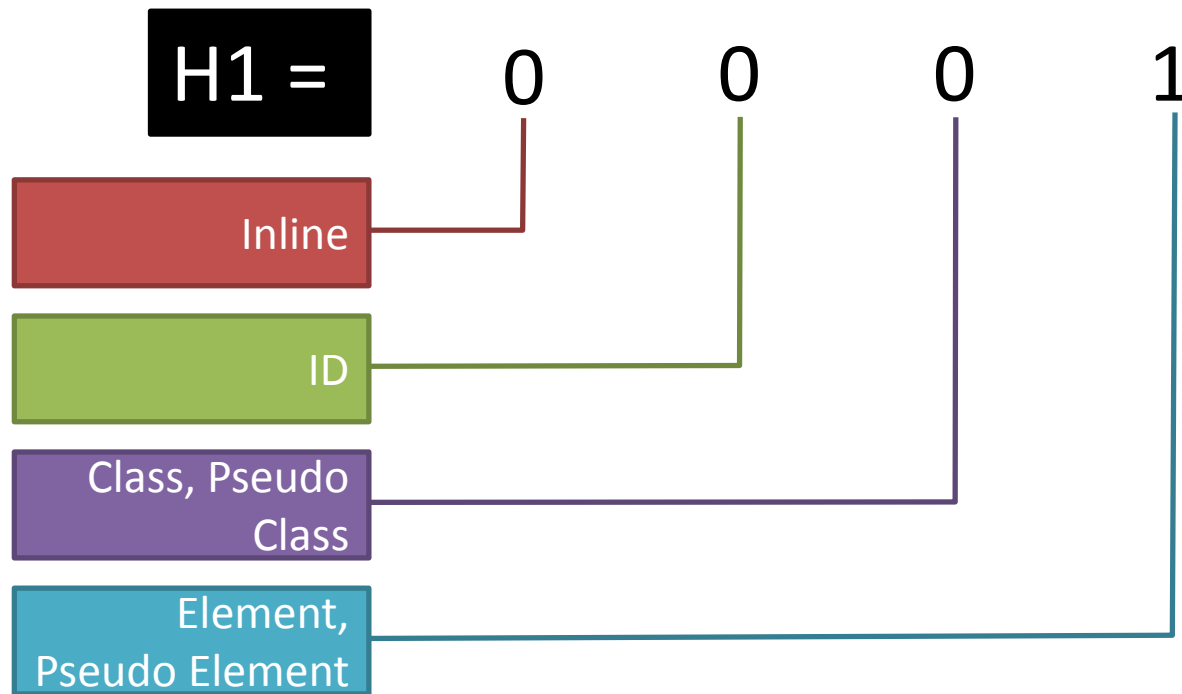
Wanneer 3 selectoren dezelfde specificity hebben, dan wint de laatste

Principe van Cascading

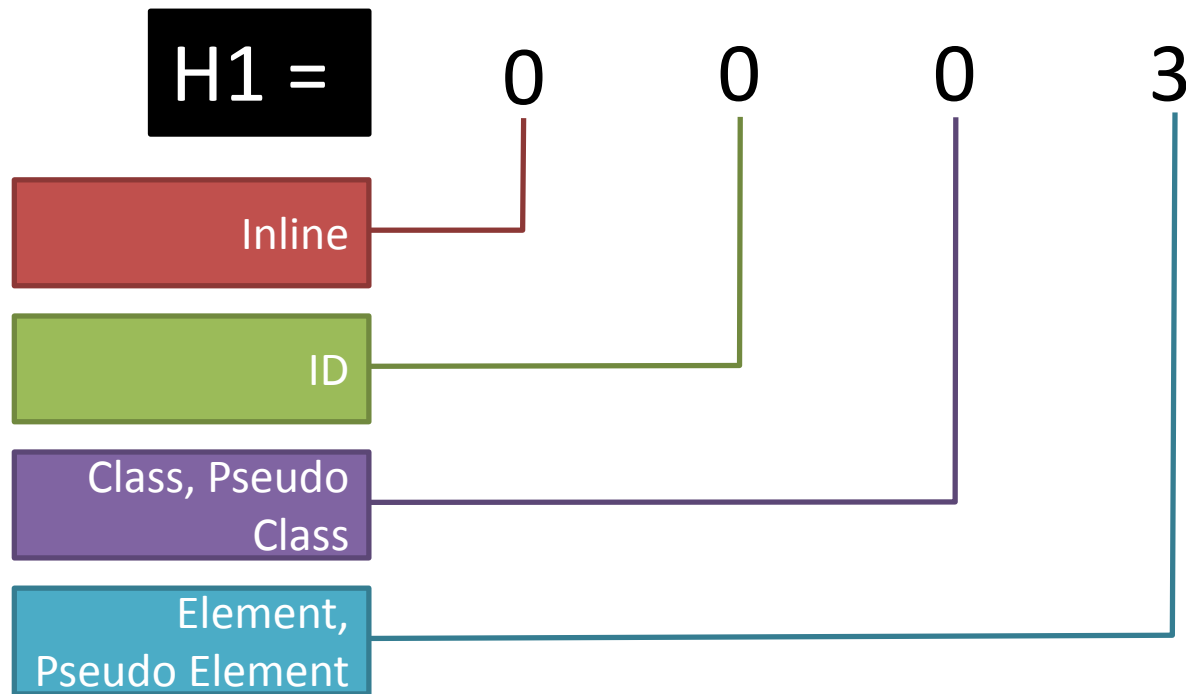
# Hierarchie

1. Inline styles
2. ID's
3. Classes en Pseudo Classes
4. Elementen en Pseudo Elementen

```
h1 {  
    font-size: 18px;  
}
```

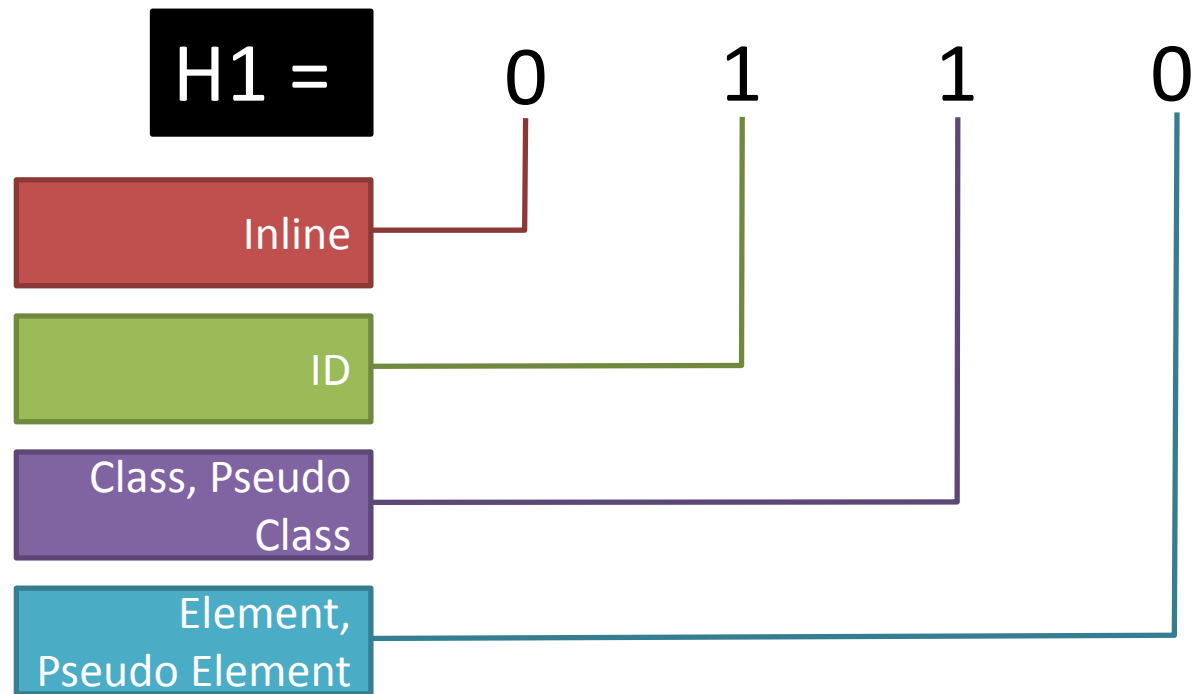


```
ul li a {  
  color: red;  
}
```

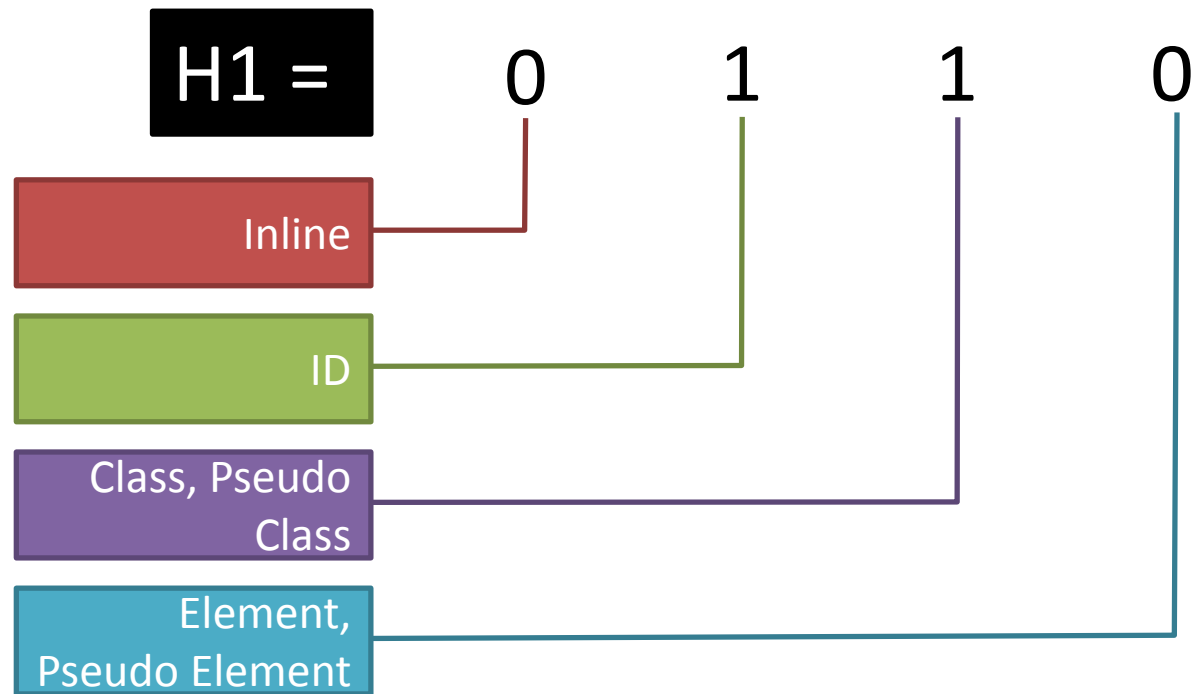




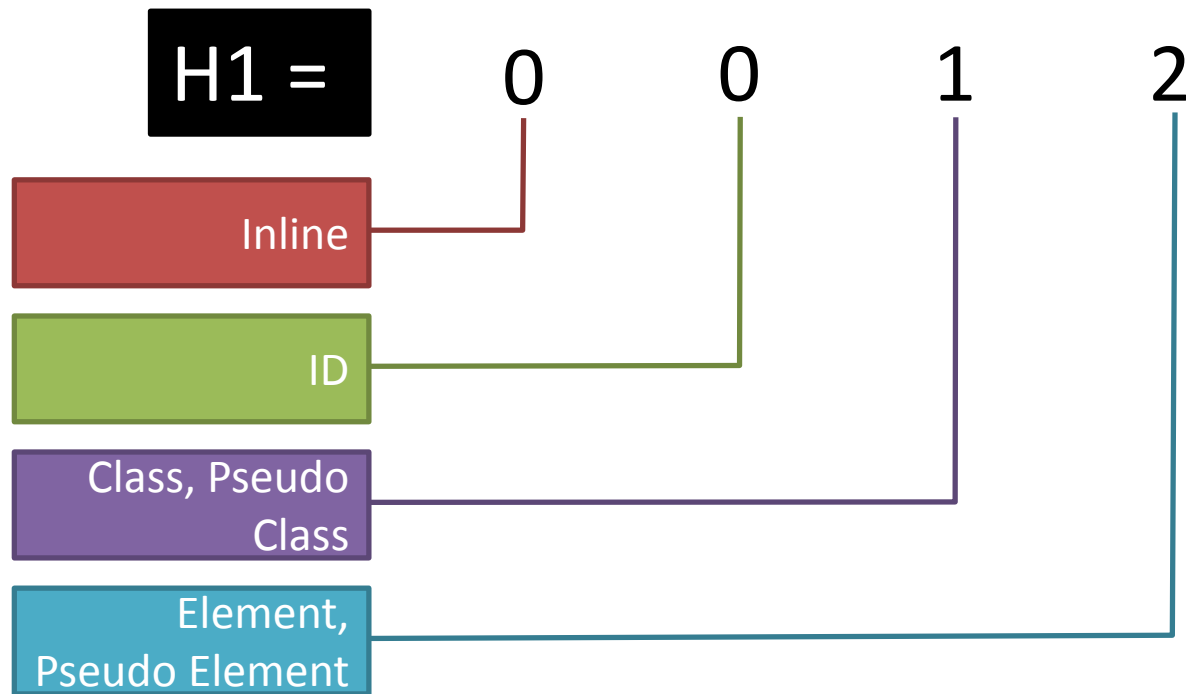
```
.content #sidebar {  
  width: 30%;  
}
```



```
.content #sidebar {  
  width: 30%;  
}
```



```
.post p:first-letter {  
  font-size: 16px;  
}
```



# Sommige tools kunnen een goede hulp zijn



1 ☐ .post p:first-letter {  
2     Selector specificity: 0, 1, 2 ; 16px;  
3 }

The image shows a code editor with three lines of CSS code. A yellow vertical bar highlights the first three lines. A tooltip box is positioned over the second line, displaying the text 'Selector specificity: 0, 1, 2 ; 16px;'. The code is as follows:

```
1 ☐ .post p:first-letter {  
2     Selector specificity: 0, 1, 2 ; 16px;  
3 }
```



important!

# important

Important overschrijft een  
bepaalde selector

Dit maakt debuggen zeer  
moeilijk



# Universele selector




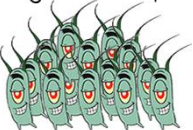



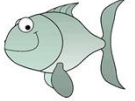
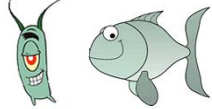






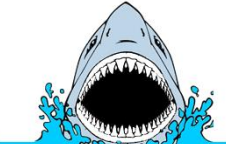
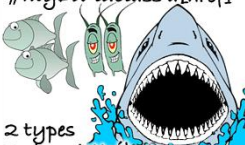





Selector om een rule aan alle elementen toe te wijzen

Heeft geen gewicht, wordt gemakkelijk overschreven

# CSS Specificity

with Plankton, Fish and Sharks

 universal selector 0-0-0	 1 element 0-0-1	 2 elements 0-0-2	 14 elements 0-0-14
 1 class 0-1-0	 1 universal selector 1 class 0-1-0	 1 universal selector 1 attribute selector 0-1-0	 1 pseudo-class 0-1-0
 1 element 1 class 0-1-1	 1 element 1 attribute 0-1-1	 2 elements 1 pseudo-class 0-1-2	 2 elements 1 attribute 0-1-2
 1 element 1 class 1 pseudo-class 0-2-1	 1 element 1 class 1 attribute 0-2-1	 10 class/attribute/ pseudo-classes 0-10-0	 1 ID Selector 1-0-0
 2 types 2 class/attribute 1 ID Selector 1-2-2	 2 ID Selectors 1 type selector 2-0-1	 inline style 1-0-0-0	 !important 1-0-0-0-0

X-0-0: The number of ID selectors

0-Y-0: The number of class selectors, attributes selectors, and pseudo-classes

0-0-Z: The number of type selectors and pseudo-elements

\*, +, >, ~: The universal selector has no value and combinators do not increase specificity

:not(x): The negation selector has no value, but the argument passed increases specificity

Estelle Weyl \* @estellew \* www.standardista.com \* 2012



# Onthoud

ID's hebben teveel gewicht in vergelijking met classes

- Vermijd id's zoveel mogelijk

selectoren met hetzelfde gewicht, telt de laatste

- Principe van cascading

\* heeft geen gewicht

!important overschrijft alles

- Vermijd important



**NEW MEDIA &  
COMMUNICATION  
TECHNOLOGY**

# **CSS Richtlijnen**



Je hoeft deze niet te volgen,  
maar we raden ze wel sterk aan!



# SHOUT

When you don't agree

# CSS files

ALTIJD External files en bovenaan de file

Tijdens development  
meerdere files

Deployed

Zo weinig mogelijk files en geminified

browsers renderen elementen anders.

Met een reset zet je alles terug naar default waarden

use reset.css of normalize.css



# LINT!



You can pick and choose which errors and warnings to test for below.

## Errors

- ☒ Beware of broken box sizing
- ☒ Require properties appropriate for `display`
- ☒ Disallow duplicate properties
- ☒ Disallow empty rules
- ☒ Require use of known properties

## Compatibility

- ☒ Disallow adjoining classes
- ☒ Disallow `box-sizing`
- ☒ Require compatible vendor prefixes
- ☒ Require all gradient definitions
- ☒ Disallow negative `text-indent`
- ☒ Require standard property with vendor prefix
- ☒ Require fallback colors
- ☒ Disallow star hack NEW
- ☒ Disallow underscore hack NEW

## Performance

- ☒ Don't use too many web fonts
- ☒ Disallow `@import`
- ☒ Disallow duplicate background images
- ☒ Disallow selectors that look like regexs
- ☒ Disallow universal selector
- ☒ Disallow unqualified attribute selectors
- ☒ Disallow units for 0 values
- ☒ Disallow overqualified elements
- ☒ Require shorthand properties

## Maintainability & Duplication

- ☒ Disallow too many floats
- ☒ Don't use too many font sizes
- ☒ Disallow IDs in selectors
- ☒ Disallow `!important`

## Accessibility

- ☒ Disallow `outline:none`

## OOCSS

- ☒ Disallow qualified headings
- ☒ Heading should only be defined once

[Learn More About the Rules](#)

# Classes ipv ID

CSS bestaat er in dat je UI stuff gaat herhalen voor meerdere elementen

Wij hebben niet gezegd dat je geen ID mag gebruiken, we raden het alleen niet aan (eerder af)

er zijn uitzonderingen



# Altijd verkorte syntax gebruiken

## Good Practice:

```
margin: 20px;  
padding: 0 5px 1px 0;
```

## Bad Practice:

```
padding-top: 5px;  
padding-bottom: 5px;  
padding-left: 5px;  
padding-right: 5px;
```

# Geen presentatie linken in class

Gebruik namen die vertellen wat het moet doen zonder rekening houden met kleuren, of UI, ...

Slecht voorbeeld:

```
.red-link {  
  
}  
  
.small-button{  
  
}
```

# Gebruik child selector ipv descendant selector

## Good Practice:

```
ul > li > a {  
  
}
```

## Bad Practice:

```
ul li a {  
  
}
```

# Vermijd images als het kan opgelost worden met CSS

## Good Practice:

```
nav > ul > li + li {  
    border-left: 2px solid black;  
  
    -webkit-border-radius: 20px;  
    -moz-border-radius: 20px;  
    border-radius: 20px;  
}
```

## Bad Practice:

```
background: url(images/hr.gif) no-repeat left bottom;
```

# OO CSS

C -> Cascading

Hergebruik elementen  
overerving

Ga na wat allemaal hetzelfde is, plaats dat in een element

Wanneer iets wijzigt, voeg je een class toe die het voorgaande overschrijft

# Zelfstudie

<http://oocss.org/>

Why best practices are killing us



**NEW MEDIA &  
COMMUNICATION  
TECHNOLOGY**

**CSS3**

# CSS3 === Modules

CSS3 is geen single specification

CSS3 is opgebouwd uit modules

Elke module:

- Eigen specificatie
- Eigen authors

<http://www.w3.org/Style/CSS/current-work>



Borders

Backgrounds

Colors

Text-effects

Box-sizing

Selectors

Transitions

Media Queries

Multi-column  
layout

Border-radius

Box-shadow

...

Completed	Current	Upcoming	Notes	i
CSS Snapshot 2010	NOTE	–	Latest stable CSS	i
CSS Snapshot 2007	NOTE	–		i
CSS Color Level 3	REC	REC	See Errata	i
CSS Namespaces	REC	REC		i
Selectors Level 3	REC	REC		i
CSS Level 2 Revision 1	REC	REC	See Errata	i
CSS Level 1	REC	–	Unmaintained, see Snapshot	i
CSS Print Profile	NOTE	–		i
Media Queries	REC	REC		i

Stable	Current	Upcoming	Notes	i
CSS Style Attributes	CR	PR		i

Testing	Current	Upcoming	Notes	i
CSS Backgrounds and Borders Level 3	CR	LC		i
CSS Conditional Rules Level 3	CR	CR		i
CSS Image Values and Replaced Content Level 3	CR	PR		i
CSS Marquee	CR	PR		i
CSS Multi-column Layout	CR	CR		i
CSS Speech	CR	PR		i
CSS Values and Units Level 3	CR	PR		i
CSS Flexible Box Layout	CR	PR		i
CSS Text Decoration Module Level 3	CR	PR		i
CSS Mobile Profile 2.0	CR	PR	Status unknown	i
CSS TV Profile 1.0	CR	PR	Status unknown	i

<http://www.w3.org/Style/CSS/current-work.en.html>

Refining	Current	Upcoming	Notes	i
----------	---------	----------	-------	---

# Cascading Style Sheets (CSS 3)

## BACKGROUND

<b>background</b>	<i>background-image background-position background-size background-repeat background-attachment background-origin background-clip background-color</i>
<b>background-attachment</b>	scroll   fixed
<b>background-break</b>	bounding-box   each-box   continuous
<b>background-clip</b>	<i>length %</i> border-box   padding-box   content-box   no-clip
<b>background-color</b>	<i>color</i> transparent
<b>background-image</b>	<i>url</i> none
<b>background-origin</b>	border-box   padding-box   content-box
<b>background-position</b>	top left   top center   top right   center left   center center   center right   bottom left   bottom center   bottom right <i>x-% y-% x-pos y-pos</i>
<b>background-repeat</b>	repeat   repeat-x   repeat-y   no-repeat
<b>background-size</b>	<i>length %</i> auto   cover   contain

## BORDER

<b>border</b>	<i>border-width border-style border-color</i>
<b>border-break</b>	<i>border-width border-style color</i>

## BORDER

<b>border-top</b>	<i>border-top-width border-style border-color</i>
<b>border-top-color</b>	<i>border-color</i>
<b>border-top-style</b>	<i>border-style</i>
<b>border-top-width</b>	thin   medium   thick <i>length</i>
<b>border-width</b>	thin   medium   thick <i>length</i>
<b>border-radius</b>	<i>border-top-right-radius border-bottom-right-radius border-bottom-left-radius border-top-left-radius</i>
<b>border-top-right-radius</b>	<i>length</i>
<b>border-bottom-right-radius</b>	<i>length</i>
<b>border-bottom-left-radius</b>	<i>length</i>
<b>border-top-left-radius</b>	<i>length</i>
<b>box-shadow</b>	inset    [ <i>length, length, length, length</i>    <color> ] none
<b>border-style</b>	none   hidden   dotted   dashed   solid   double   groove   ridge   inset   outset

## FONT

<b>font</b>	<i>font-style font-variant font-weight font-size/line-height font-family caption   icon   menu   message-box   small-caption   status-bar</i>
<b>font-family</b>	<i>family-name generic-family inherit</i>
<b>font-size</b>	xx-small   x-small   small

## BOX MODEL

<b>float</b>	left   right   none
<b>height</b>	auto <i>length %</i>
<b>max-height</b>	none <i>length %</i>
<b>max-width</b>	none <i>length %</i>
<b>min-height</b>	none   inherit <i>length %</i>
<b>min-width</b>	none   inherit <i>length %</i>
<b>width</b>	auto <i>% length</i>
<b>margin</b>	<i>margin-top margin-right margin-bottom margin-left</i>
<b>margin-bottom</b>	auto <i>length %</i>
<b>margin-left</b>	auto <i>length %</i>
<b>margin-right</b>	auto <i>length %</i>
<b>margin-top</b>	auto <i>length %</i>
<b>padding</b>	<i>padding-top padding-right padding-bottom padding-left</i>

# Process

1. Working Draft
2. Last Call
3. Candidate Recommendation
4. Proposed Recommendation
5. Recommendation

# **Graceful degradation vs progressive enhancement**



Dus er zijn zaken die niet zullen werken in alle browsers



Hoe ga je daar mee om?

## 2 manieren om compatibiliteit aan te pakken


### Graceful degradation

- Begin met de meest recente browsers
- Start met de nieuwste features te implementeren
- Voorzie de functionaliteit in de oudere browsers

### Progressive Enhancement

- Maak de web app met features die alle browsers ondersteunen
- Voeg nieuwe functionaliteit toe





Moeten web applicaties  
er in elke browser  
dezelfde look hebben?

Ja en Neen 😊

Soms heb je in moderne browsers ronde hoeken, in de oudere behoud je rechte hoeken

# Vendor Prefixes

Go back In horror





Een van de properties die de CSSWG  
voorstelde was de **clip** property



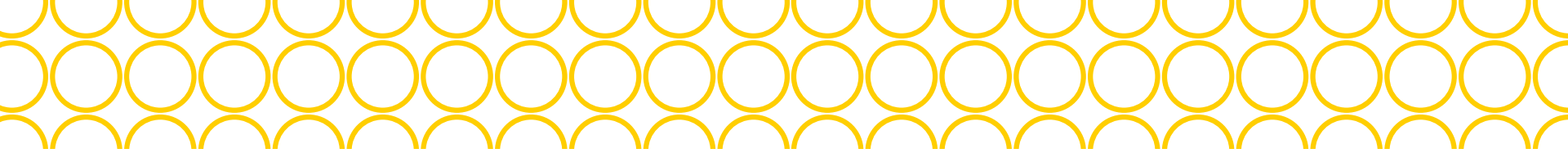
Browsers implementeerden deze property  
zoals de specs het voorschreef



Wanneer deze browsers gereleased werden, veranderde de  
CSSWG de property volledig







# Specification <> browsers



Oplossing:

Vendor prefix

```
.element {  
    -webkit-border-radius: 12px;  
    -moz-border-radius: 12px;  
    border-radius: 12px;  
}
```

**Box-Shadow**

# Box-Shadow

Werkt alleen in IE9+, Firefox 4, Chrome, Opera en Safari 5.1.1

Gracefull degradation nodig

Voegt schaduw toe aan een element

Voor tekst kan je Text-Shadow gebruiken

H-shadow

Blur

Color

box-shadow: 10px 10px 20px 1px #793b1c;

V-shadow

Spread

# Graceful degradation



IE8

```
-ms-filter: "progid:DXImageTransform.Microsoft.Shadow(  
    Strength=4, Direction=135, Color='#793b1c')";
```

```
filter:progid:DXImageTransform.Microsoft.Shadow(  
    Strength=4, Direction=135, Color='#793b1c');
```



IE7



# Backgrounds

## The Cicada Project Gallery

Upload your own

Another Brick



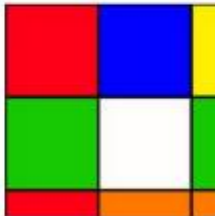
Mosaic



Camellia



Rubiks Cube



Concrete Mix



Parchment



Worn Chessboard



Organics



Cicada Gumballs



## What is this?

45

Tweet

The **cicada principle** is a new web design technique for creating efficient tiling backgrounds that don't appear to repeat.

This gallery is a chance for you to deconstruct and understand what you are seeing in working examples.

Learn how it works here!

Upload your own now!

Subscribe



961



1502



# Eigenschaften

## Available properties

- background-color
- background-image
- background-repeat
- background-attachment
- background-position
- Background-size



Je kan ook meerdere backgrounds gaan instellen

Our goal was to simplify the voter registration process during the Obama campaign. Silverback was clearly the easiest way to gain true insights into user behavior, and resulted in a successful, more informed site at launch. There is nothing silver about Silverback, all gold in my opinion!

so says Scott Thomas of Obama for America | [Next testimonial »](#)



Clearleft presents

# Silverback 2.0

Guerrilla usability testing software for designers and developers

- 🍌 Capture screen activity
- 🍌 Add chapter markers on-the-fly
- 🍌 Video the tester's face
- 🍌 Control recording with the remote
- 🍌 Record the tester's voice
- 🍌 Export to Quicktime

**NEW  
FEATURES**

## Features in 2.0 include

### Preview

Watch sessions within Silverback

### Tasks & Highlights

Set tasks and mark noteworthy moments within a session

### Batch Export

Save selected sessions, tasks, highlights or projects in one go

### Performance

Faster export, better usability

**Download** 

FREE FOR 30 DAYS

**Buy NOW**

**\$69.95** FREE upgrade for existing users



Silverback requires  
Mac OS X (10.4-10.6)

# Meerdere backgrounds

Door een comma-separated list

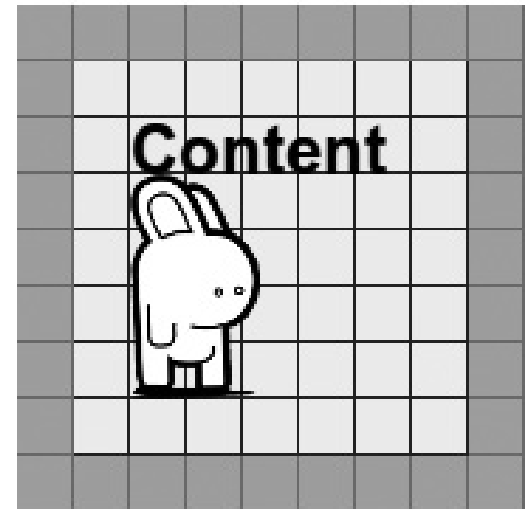
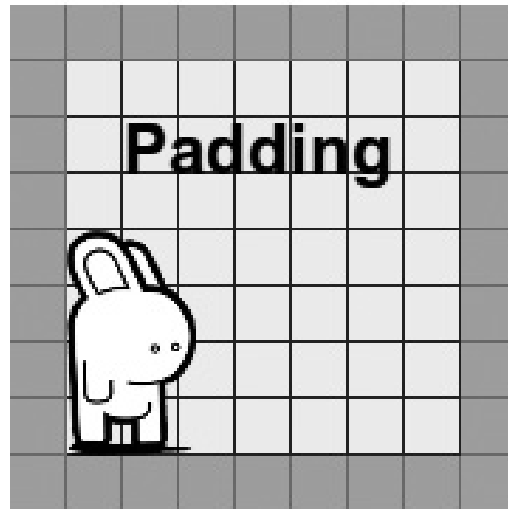
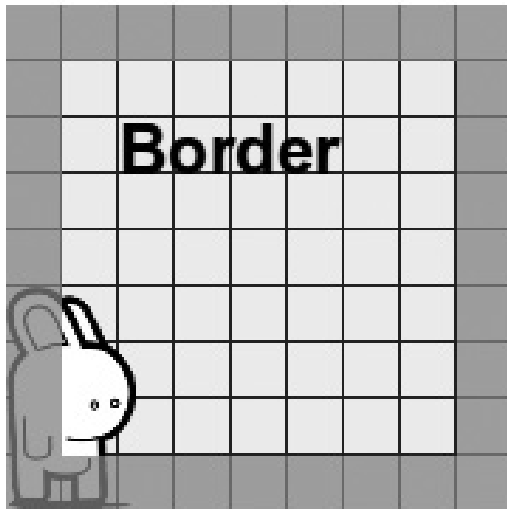
```
.element {  
    background-image: url('howest.png'), url('nmct.jpg');  
    background-position: 95% 85%, 50% 50%;  
    background-repeat: no-repeat;  
}
```

# Background-clip

Waar moeten de borders getoond worden tov. het parent element.



# Background-origin

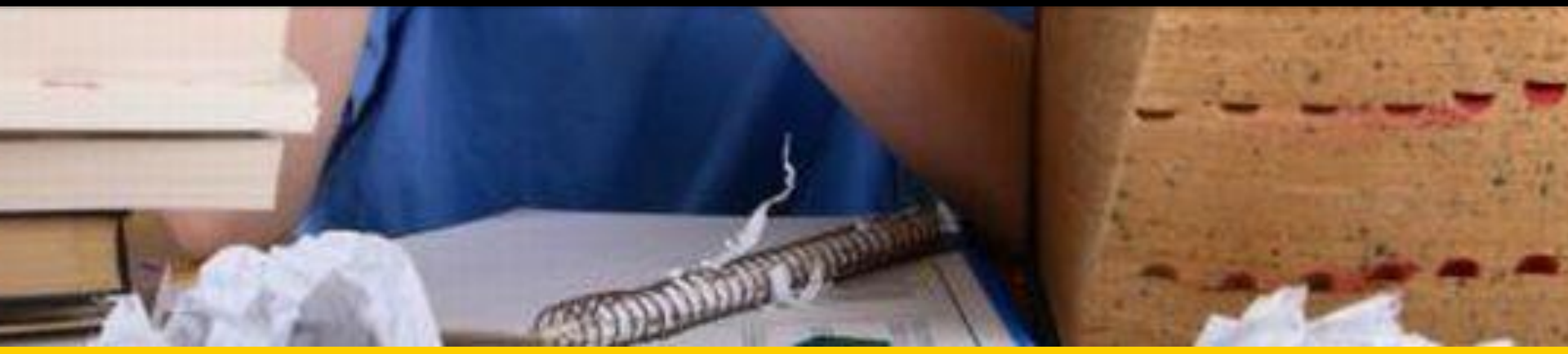




# Border-images



Hoe kan ik een image als border gebruiken





## Contents

How does it work?  
Documentation  
Browser support  
Making your own  
modules  
FAQ  
Download



# chainvas

*Chaining for everyone*

Only  
1-2KB  
minified

Click to download ->

**chain•vas** |chānvəs| *n.* A tiny, modular library that can add chaining to any API that isn't naturally chainable, like the Canvas API, the DOM and more.

## How does it work?

**chainvas** goes through all the methods in a prototype and wraps a function around them that returns the returned value or, if that is **undefined**, the function's context (**this**). Since the function context is our original object, we can keep calling methods without having to write the variable name over and over again, like this:

```
obj.foo().bar().baz();
```

instead of this:

```
obj.foo();  
obj.bar();  
obj.baz();
```

This practice is usually called **chaining** and was popularized by [jQuery](#) back in 2006. If you find the first code example harder to read than the second, keep in mind that you can format it like this:

```
.element
{
    border-image: source slice / width repeat;
}
```

- Source:** URL naar de image
- Slice:** lengte van het frame dat getoond wordt
- Repeat:** Herhaalt image verticaal of horizontaal

# Gradients



Gradient is een transitie tussen 2 or meerdere color stops

# Gradients

Linear gradient

gradient volgens een rechte lijn

Radial gradient

gradient volgens cirkels

# Linear gradient

```
background-image: linear-gradient(  
    point or angle  
    , from-stop  
    , color-stop  
    , to-stop);
```



# Meerdere linear gradients

```
background-image: -linear-gradient(bottom  
    , #0392CE  
    , #FD5308 20px  
    , #0392CE);
```

# Gradients herhalen

```
div
{
    background-image: repeating-
        linear-gradient(white,black 25%);
}
```

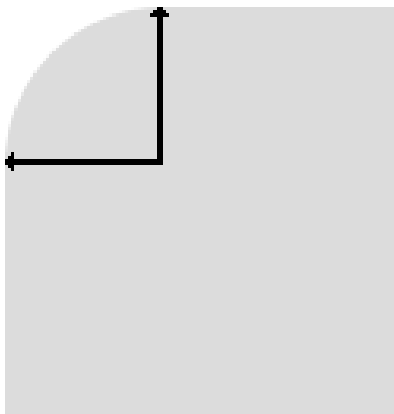
# Radial Gradients

```
background-image: radial-gradient(  
    position or angle  
    , shape or size  
    , from-stop  
    , color-stop  
    , to-stop);
```

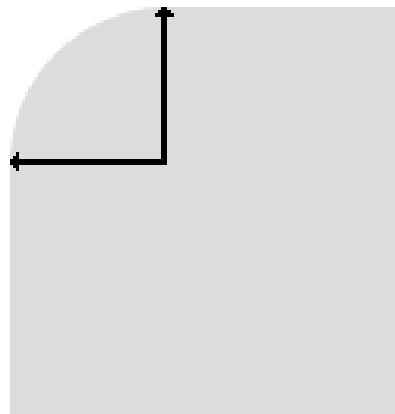
**Border-Radius**

# Border-Radius

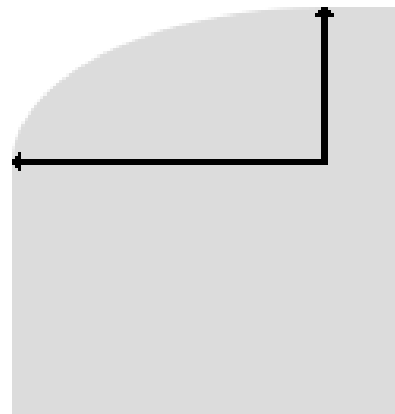
Waarschijnlijk een van de meeste gevraagde CSS features



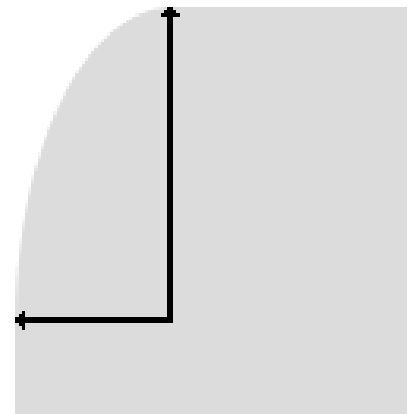
```
border-top-left-radius:  
50px;
```



```
border-top-left-radius:  
50px 50px;
```



```
border-top-left-radius:  
100px 50px;
```



```
border-top-left-radius:  
50px 100px;
```

**border-radius: *1-4 length | % / 1-4 length | %;***

- Volgorde: top-left, top-right, bottom-right, bottom-left
- Length: grootte van de hoeken
- %: grootte van de hoeken in %



## LATEST PROJECT

I'm currently working on developing a free Wordpress theme called *Liquorice*.

★ TO PORTFOLIO ★

About

Services

Portfolio

Downloads

Contact

Journal

in Linked in

e Twitter

⊕ Dribbble

as Last.fm

# Powerful Flash CMS

Create your Flash websites with the most flexible, rebrandable and feature-rich Flash CMS with no limits in design and animation.

**FREE TRIAL**  
NO CREDIT CARD REQUIRED

**TAKE A TOUR**  
ALL FEATURES REVEALED



## What is it?

Moto CMS is an advanced Flash content management system that allows you to create attractive, functional and SEO-friendly Flash websites.

## Who is it for?

Moto CMS is created for Flash developers, design studios and freelancers who make money on creating and maintaining Flash websites.

## Why use it?

With Moto CMS you can create Flash websites more either by creating new Flash sites or by maintaining existing ones.



**HSL(A) – RGBA (A)**

# HSLA

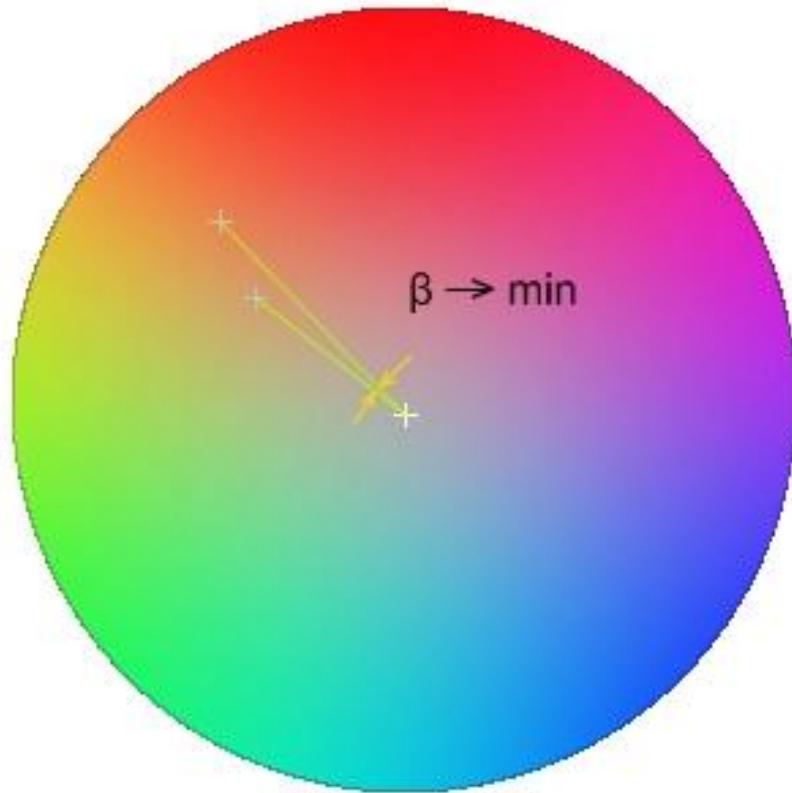
## Hue-Saturation-Lightness-Alpha

- Hue: aantal graden voor kleur
- Saturation: 100% is het volledige kleur
- Lightness: 0% is donker, 100% is licht

## Alternatief voor RGB

### Nadelen van RGB

- Hardware-oriented: komende van CRT's
- RGB is niet intuïtief



Hue	hsl(H,S,L)	Color	Color Name
0	hsl(0, 100%, 50%)		red
60	hsl(60, 100%, 50%)		
120	hsl(120, 100%, 50%)		green
180	hsl(180, 100%, 50%)		
240	hsl(240, 100%, 50%)		blue
300	hsl(300, 100%, 50%)		
360	hsl(360, 100%, 50%)		red

# Saturatie en lightness

		Saturation				
		100%	75%	50%	25%	0%
Lightness	100%					
	88%					
	75%					
	63%					
	50%					
	38%					
	25%					
	13%					
	0%					

# Transformations

# Transformaties

4 soorten transformaties

- Rotate
- Translate
- Skew
- Scale

Je kan deze natuurlijk met elkaar gaan combineren

# Transform origin

Het punt van waaruit alle transformaties starten

Dit punt wijzigen kan rare maar ook gewenste effecten opleveren

2 mogelijkheden

- Instellen met pixels
- Plaats instellen (links, rechts, boven, onder)

```
.element
{
    transform-origin: 0 0;
}

.element
{
    transform-origin: left top;
}
```



# Rotate

```
-moz-transform: rotate(-25deg);  
-ms-transform: rotate(-25deg);  
-o-transform: rotate(-25deg);  
-webkit-transform: rotate(-25deg);
```

# Translate

```
.element
{
    transform: translateX(value);
    transform: translateY(value);
}
.element
{
    transform: translate(translateX,translateY);
}
```

# Skew

```
.element
{
    transform: skewX(value);
    transform: skewY(value);
}
.element
{
    transform: skew(skewX,skewY);
}
```

# Scale

```
.element
{
    transform: scaleX(value);
    transform: scaleY(value);
}

.element
{
    transform: scale(scaleX,scaleY);
}
```

# Transities

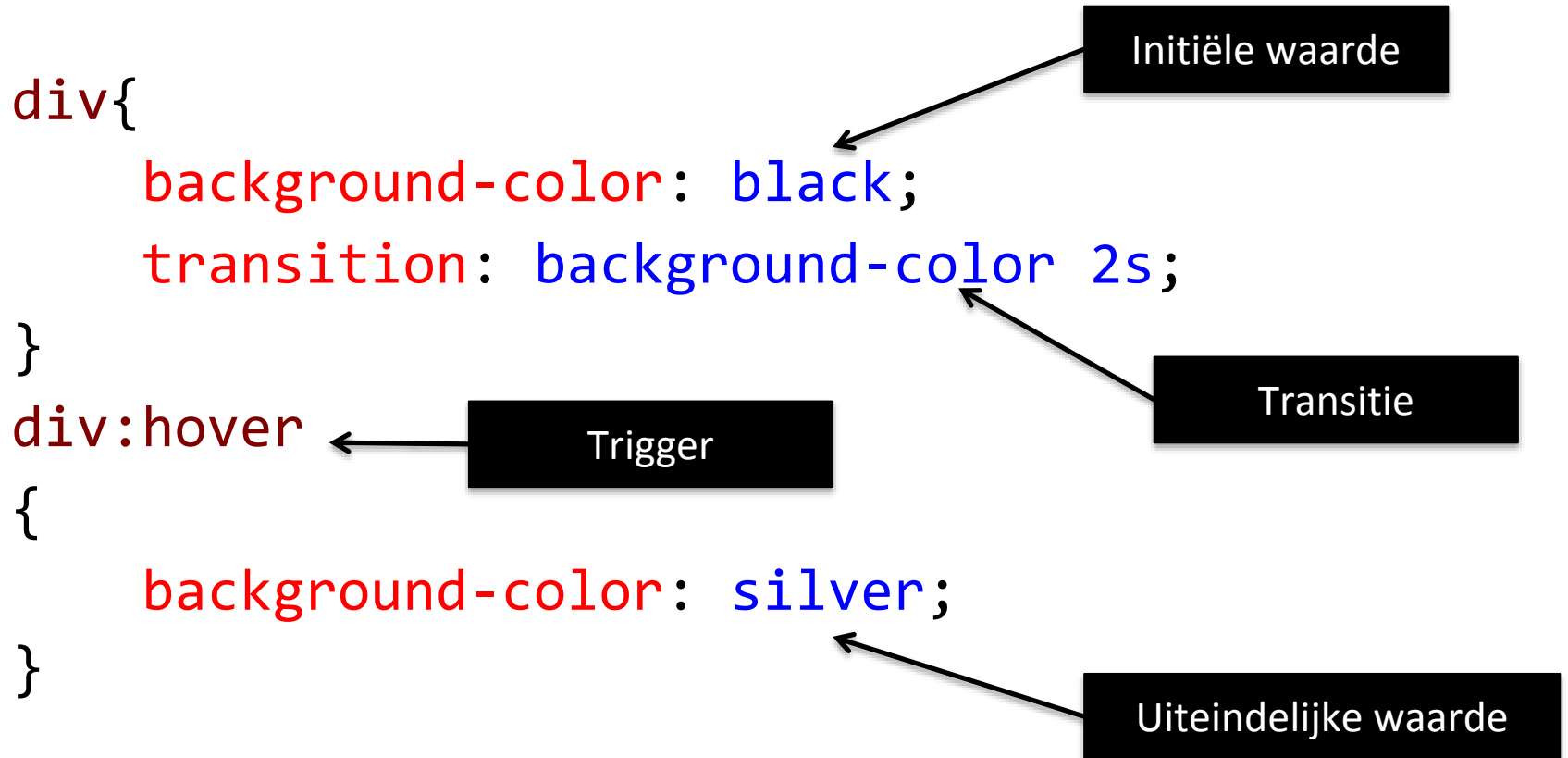
# Wat

Een animatie waarbij men van een waarde van een property naar de andere gaat

- Over tijd

4 voorwaarden dienen voldaan te zijn

- Initiele waarde
- Uiteindelijke waarde
- De transitie zelf
- Een trigger



# Transition-property

Welke CSS eigenschap wil je wijzigen

Niet alle eigenschappen kunnen gewijzigd worden

```
.element
{
    font-size: 150%;
    transition-property: font-size;
}
```



# Transition-duration

Tijd nodig om de transitie uit te voeren

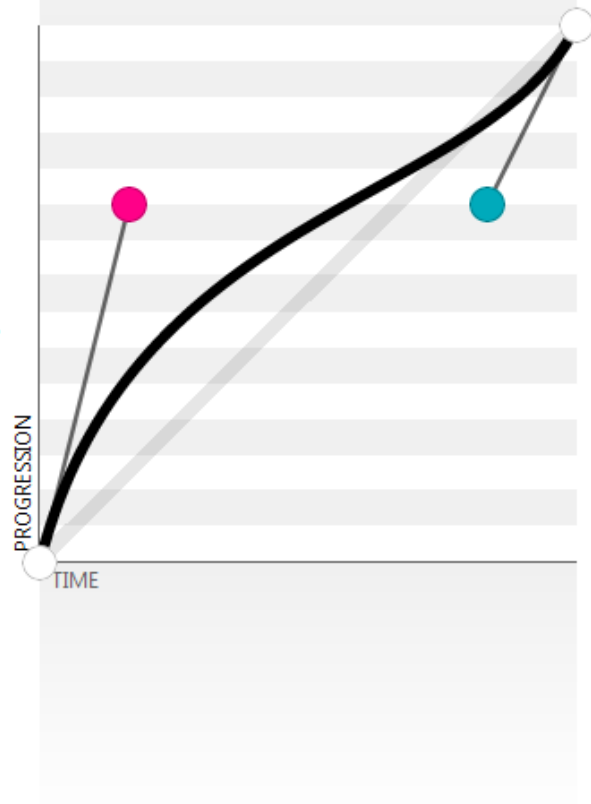
```
.element
{
    font-size: 150%;
    transition-property: font-size;
    transition-duration: 2s;
}
```

# Timing function

Animaties voelen natuurlijker aan wanneer je werkt met versnellingen en vertragingen

- Ease
- Linear
- Ease-in
- Ease-out
- Ease-in-out

```
.element
{
    font-size: 150%;
    transition-property: font-size;
    transition-duration: 2s;
    transition-timing-function: ease-in;
}
```



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

SAVE

Preview & compare

GO!

Duration:  1 second



Library

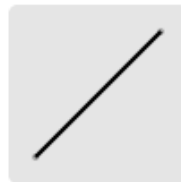
IMPORT

EXPORT

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



ease



linear



ease-in



ease-out



ease-in-out

**Tip:** Right click on any library curve and select "Copy Link Address" to get a permalink to it which you can share with others

# Transition-delay

```
.element
{
    font-size: 150%;
    transition-property: font-size;
    transition-duration: 2s;
    transition-delay: 250ms;
}
```



<http://www.the-art-of-web.com/css/css-animation/>