

1 Basis HTML5 en CSS3

1.1 Inleiding

HTML is de *lingua franca* van het internet. Het is de programmeertaal die de technologieën achter het internet heeft verenigd en bevordert alle communicatie binnen het internet. Alle communicatie loopt via webpagina's en alle webpagina's zijn in HTML geschreven.

Evolutie van HTML

HTML of HyperText Markup Language is in 1991 ontwikkeld door sir Tim Berners-Lee om wetenschappelijke documenten van het CERN, een Europese organisatie die fundamenteel onderzoek doet naar elementaire deeltjes, toegankelijk te maken. Al snel werden de mogelijkheden van HTML onderkend en in de loop der jaren verschenen steeds betere browsers. Het World Wide Web Consortium (W3C) nam in 1996 de ontwikkeling van HTML over. Versie HTML 3.2 kwam in januari 1997 tot stand. In december 1997 volgde een eerste versie van HTML 4, gevolgd door de eerste versie van HTML5 in januari 2008. Al in 2007 zei Steve Jobs dat HTML5 de software-Flash overbodig zou maken. Hij herhaalde zijn woorden bij de introductie van de iPad in 2010.

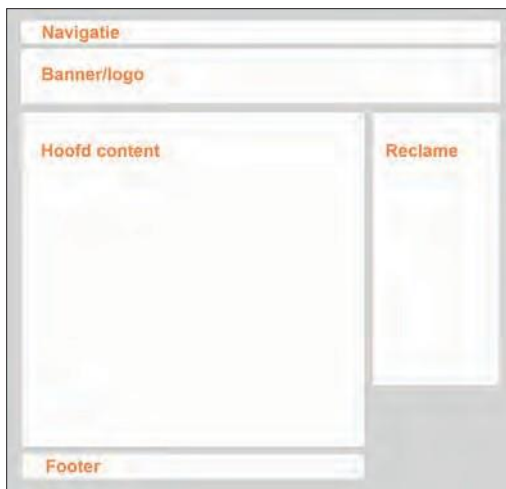
De belangrijkste aanpassingen in HTML5 zijn het uniform afhandelen van fouten, de mogelijkheid om een document logischer op te bouwen en nieuwe invoertypes. HTML5 wordt ondersteund door alle nieuwe browsers, waaronder Firefox, Google Chrome en Safari.

Lay-out en design van een webpagina

Uit onderzoek blijkt dat de meeste webdesigners in de loop der jaren zijn gaan ontwerpen volgens dezelfde patronen. Bij de meeste moderne websites komen we het volgende ontwerp tegen:

- een webpagina met een banner en navigatie bovenaan;
- een of meer artikelen die de content vormen;
- een kolom aan de rechterkant voor reclame;
- een voettekst met bedrijfsinformatie onderaan.

Een voorspelbare website is makkelijker om te navigeren en de informatie is eenvoudiger te vinden.



Figuur 1.1 Lay-out van een webpagina

HTML-editors

Om de opgaven en lab-opdrachten uit dit boek te kunnen maken heb je een HTML-editor nodig. Een HTML-editor is een applicatie voor het schrijven van HTML-teksten. Deze teksten noemen we webpagina's. Hier zijn twee websites waar je gratis HTML-editors kunt downloaden:

Lijst van HTML-editors

- Notepad++
- Microsoft visual studio html editor

In deze paragraaf maken we de code van de indexpagina. We maken kennis met de volgende HTML-elementen:

- comments
- DOCTYPE
- geneste elementen
- meta-elementen
- semantische elementen
- ID en classes

HTML-elementen

Webpagina's bestaan uit platte tekst met *markeringstekens*, ook wel aangeduid als *tags*. Hierin wordt de lay-out aangegeven, oftewel hoe de tekst gestructureerd wordt. De *tags* worden door de webbrowser geïnterpreteerd en de gebruiker krijgt

de opgemaakte tekst in de vorm van een webpagina te zien. Een markering tussen haakjes, bijvoorbeeld `<html>` of `<form>`, wordt een *tag* genoemd. De syntax van tags is als volgt:

`<open tag>` content van de tag `</sluiten tag>`

Als je een *begintag* hebt geopend, bijvoorbeeld `<html>`, eindig je *altijd* met een *eindtag*, bijvoorbeeld `</html>`. Je ziet dat de *eindtag* altijd een `/` bevat.

Element `<!-- commentaren -->`

We begonnen de code met het commentaar:

```
<!-- website -->
```

Zulk soort commentaar kunnen we overal op de pagina plaatsen en ze zijn bedoeld om de structuur en code van de pagina toe te lichten. Een commentaar heeft geen effect op de weergave van de pagina.

Element `<!DOCTYPE html>`

Browsers kunnen meerdere soorten documenten interpreteren, zoals webpagina's en PDF's. Het is daarom belangrijk om ons HTML-document te identificeren met de tag `<!DOCTYPE html>`. Zo kan de browser onze code vertalen in een webpagina.

Element `<html lang="nl">`

De tweede regel in Webbouw 1 is het element `<html>`. We coderen de namen van alle tags in kleine letters. De enige tag met hoofdletters is het element `<!DOCTYPE>`. Sommige elementen kunnen attributen bevatten. Alle attributen krijgen een naam en een waarde tussen dubbele aanhalingstekens. In dit geval is het attribuut `lang="nl"`. Zo geef je aan dat de content van deze pagina in het Nederlands wordt geschreven.

Element `</html>` sluiten

Zoals te zien is sluiten we alle elementen met het slash-teken (`/`). Alle elementen moeten gesloten worden. Als je een element niet goed sluit, leidt dit tot fouten in het weergeven van de pagina.

Geneste elementen

Het eerste wat opvalt in Webbouw 1 zijn de geneste elementen. Dat zijn elementen binnen andere elementen. In dit geval bevat `<html>` de elementen `<head>` en `<body>`. Het `<head>`-element bevat op zijn beurt de `<meta>`- en `<title>`-elementen. Een ge- nest element begin je met de Tab-toets. Bijvoorbeeld:

```

<html lang="nl">
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
<title>Titel</title>
    </head>
    <body>
        . . .
    </body>
</html>

```

We gebruiken Tab's om de code leesbaarder te maken. Zo zien we in één oogopslag welke elementen andere geneste elementen bevatten. Het gebruiken van Tab's heeft geen effect op het weergeven van de webpagina.

Geneste elementen moeten in de juiste volgorde worden gesloten. Een genest element moet als eerste gesloten worden. Daarna sluit je het element van het niveau hoger. De volgende figuur laat zien hoe je geneste elementen goed sluit.








Figuur 1.4 Geneste elementen

Element <head>

We hebben de webpagina ingedeeld in <head>(hoofd) en <body>(lichaam). We gebruiken het <head>-element voor meta-informatie en voor andere elementen zoals <title>, en het <body>-element voor de content.

Browsers

Om de resultaten van de opgaven en de lab-opdrachten te kunnen zien moeten we eerst de volgende browsers downloaden en installeren.

Browser	Website
	https://www.google.nl/chrome/browser/desktop/
	https://support.mozilla.org/nl
	http://www.opera.com/
	http://safari.nl.softonic.com/
	http://windows.microsoft.com/nl-nl/internet-explorer/download-ie

Na het downloaden en installeren van deze browsers openen we het bestand **in-dex.html** door er met de muis op te klikken. De browser presenteert de code van Webbouw 1 zoals te zien is in de volgende figuur.



Figuur 1.5 Het resultaat van Webbouw 1

LET OP

Gebruik dubbele aanhalingstekens (") in plaats van twee enkele aanhalingstekens (' '). Bijvoorbeeld:

`<html lang="nl">` is niet correct.

`<html lang="nl">` is correct.

We krijgen alleen maar de puntjes binnen het `<body>`-element te zien. In ons

geval hebben we Mozilla gebruikt. Het is dus niet nodig om een eigen domeinnaam bij een provider aan te schaffen om de resultaten van de opgaven te kunnen zien. Met de browsers kunnen we zowel internetten als onze webpagina's bekijken. Als je later je eigen website wilt uploaden moet je wel een eigen domeinnaam aanschaffen.

Meta-elementen

We gebruiken <meta>-elementen voor het toevoegen van meta-informatie. Dit is informatie over de informatie van de pagina. Deze informatie is relevant voor

browsers en zoekmachines. Google en Yahoo maken gebruik van deze informatie in hun zoekalgoritmes. We kunnen meta-elementen ook gebruiken voor andere doelen, zoals het automatisch laden van een andere pagina. Ieder meta-element bevat de attributen:

name content

Bijvoorbeeld, om de naam van de auteur van de pagina aan te duiden codeer je het volgende:

```
name="author" content="naam van auteur"
```

In de volgende opgave coderen we belangrijke meta-informatie over de te bouwen website.

Een meta-tag hoeft je niet te sluiten met een / (het slash-teken).

LET OP

We mogen een lange HTML-tag alleen afbreken na een spatie. Dit doen we bij-

voorbeeld om de code te verduidelijken.

<meta charset>

Charset is de set tekens ontworpen voor computers. In Webbouw 2 hebben we het volgende attribuut gespecificeerd:

```
charset="UTF-8"
```

Dit is de karakterset voor alle Europese talen. Maar er zijn andere karaktersets, bijvoorbeeld:

charset	taal
charset="ISO-8859-6"	Grieks
charset="ISO-8859-7"	Arabisch
charset="UTF-8"	Europese talen

<meta name="robots">

Gebruik deze tag als je gevonden wilt worden door zoekmachines:

```
<meta name="robots" content="all">
```

<meta name="language">

Om de taal van de content specifiek aan te geven gebruiken we dit attribuut:

```
<meta name="language" content="Dutch">
```

<meta name="author">

Om de naam van de auteur van de pagina aan te geven gebruiken we:

```
<meta name="author" content="je eigen naam">
```

<meta name="description">

Voor een korte beschrijving van de website gebruiken we het attribuut description:

```
<meta name="description"  
content="RadioGaGa marketing">
```

<meta name="keywords">

Voor een lijst met sleutelwoorden over de website gebruiken we het attribuut keywords:

```
<meta name="keywords"  
content="RadioGaGa, muziek, playlist">
```


<meta http-equiv="refresh">

Het attribuut `http-equiv="refresh"` kunnen we gebruiken om het aantal seconden voor het *refreshen* van deze pagina te specificeren. Het kan ook de naam van een pagina aangeven waar we na een aantal seconden automatisch naar willen doorlinken, bijvoorbeeld een inleidingspagina:

```
<meta http-equiv="refresh"content="10;url=http://website.com">
```

Semantische elementen

De volgende HTML5-elementen hebben een semantische betekenis die de positie en het doel van het element in een webpagina beschrijven.

<head>	hoofd
<body>	lichaam
<nav>	navigatie
<main>	hoofd-content
<section>	sectie
<div>	divisie
<p>	paragraaf
<footer>	voet

Element <body>

Na het definiëren van het <head>-element met alle <meta>- en <title>-elementen kunnen we verder met het coderen van de structuur van de pagina. Dit doen we binnen het <body>-element.

Element <nav>

Binnen het <nav>-element coderen we de elementen die nodig zijn voor de navigatie van de website.

Element <main>

In het <main>-element coderen we de hoofd-content van de pagina, bijvoorbeeld een groep afbeeldingen en teksten die het thema van de pagina definiëren.

Element <section>

Het <main>-element mag één of meer secties of kolommen hebben. Dit doen we met het <section>-element. Hier kunnen we verschillende gerelateerde content met een thema combineren. Bijvoorbeeld, een <section>-element met het thema sport mag een sectie sport en een sectie sportnieuws hebben.

Element <div>

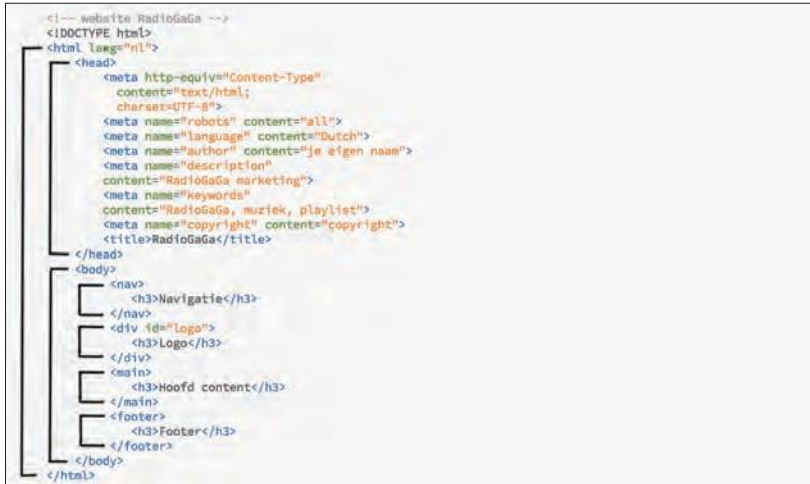
Binnen het element <div> plaatsen we generieke (diverse) content, bijvoorbeeld algemene plaatjes en teksten. Een <section> mag meerdere <div>'s hebben.

Element <footer>

Het <footer>-element is bedoeld voor informatie over het bedrijf, de webdesigner en de copyrights. Ook kunnen we hier belangrijke links vinden.

Hiërarchie van elementen

Parent-elementen bevatten child-elementen en dit zijn geneste elementen. Het `<html>`-element is het parent-element van alle andere elementen. Het `<body>`-element is een child-element van het `<html>`-element maar ook een parent-element van de elementen `<nav>`, `<main>` en `<footer>`.



Figuur 1.6

Elementen `<h1>` tot `<h6>`

Met de elementen `<h1>` tot `<h6>` definiëren we de lettergrootte van teksten binnen de elementen. Bijvoorbeeld:

```

<h1>Kopje 1</h1>
<h2>Kopje 2</h2>
<h3>Kopje 3</h3>
<h4>Kopje 4</h4>
<h5>Kopje 5</h5>
<h6>Kopje 6 </h6>

```

id of class?

Als je een element identificeert met het attribuut `id=`, dan moet dit een unieke naam zijn. We mogen geen twee elementen met dezelfde `id` benoemen. In Web- bouw 3 hebben we het element:

```
<div id="logo">
```

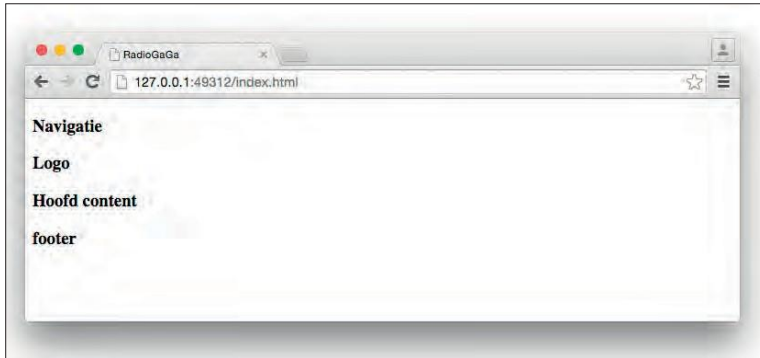
Wanneer je meerdere elementen dezelfde visuele attributen wilt geven, krijgen ze allemaal dezelfde `class`. Bijvoorbeeld:

```

<div id="logo" class="blauw">
<div id="footer" class="blauw">

```

Zo identificeren we elementen om ze later visuele attributen te kunnen geven. Dat doen we in de volgende paragraaf wanneer we beginnen met het coderen van CSS- stylesheets.



Figuur 1.7

LET OP

Als je geen resultaat krijgt dan moet je je code verbeteren. De meest voorkomende beginnersfouten zijn het niet goed afsluiten van de tags.

Ook spellingsfouten bij namen van de elementen zijn een probleem.

Bijvoorbeeld

`<budy>` in plaats van `<body>`.

Of `<htm1>` met het getal 1 in plaats van `<html>` met de letter l.

Het resultaat van deze fouten kan een lege pagina in de browser zijn.

- **Lab 01**

In deze lab-opdracht maak je een nieuw script en sla je het op als **labs.html**. In dit script maak je vier div's met de volgende hiërarchie:

div1

div2

div 3

div 4

1.2 CSS3

In deze paragraaf beginnen we met de vormgeving van de indexpagina. We maken kennis met de volgende CSS-attributen:

- width
- height
- commentaren
- background
- overflow
- text-align
- margin
- border

- font
- box-shadow

CSS (Cascading Style Sheets) is ontstaan uit onvrede onder webdesigners over het gebruik van HTML-tags voor het opmaken van webpagina's. De CSS Working Group van de W3C heeft versie 1 van CSS al eind 1996 ontwikkeld. Het gaf designers voor het eerst de mogelijkheid om de opmaak van lettertypen en de positie van de elementen van een webpagina te bepalen.

Inleiding

CSS3 biedt meer geavanceerde mogelijkheden en dringt het aantal Javascript-elementen terug, waardoor de totale omvang van een pagina kleiner wordt, wat ten goede komt aan de ranking bij zoekmachines. Belangrijk zijn daarnaast de grafische mogelijkheden, zoals drop shadows, ronde hoeken, meerdere achtergronden, animaties en transparanties.

CSS-syntaxis

Met CSS-commando's kunnen we één of meer HTML-elementen selecteren om vervolgens stijlattributen toe te wijzen. Zo kunnen we de geselecteerde elementen visuele eigenschappen toeschrijven. Elk commando heeft de volgende syntaxis:

```
selector {attribuut: waarde;}
```

Het commando bestaat uit twee delen:

```
{ ; } selector attribuut:waarde
```

- 1 Selector is de naam van het HTML-element dat we willen selecteren.
- 2 Het paar attribuut:waarde beschrijft het attribuut en de attribuutwaarde die we willen toekennen aan het geselecteerde HTML-element.

We schrijven de naam van het HTML-element zonder de haakjes < >. Tussen de accolades {} schrijven we het paar attribuut:waarde. We eindigen met een punt-komma. We kunnen de geselecteerde elementen een of meer attributen toekennen, bijvoorbeeld:

```
h1 {
  attribuut1: waarde1; attribuut2:
  waarde2;
}
```

Elementen selecteren

Hier zien we een voorbeeld van het selecteren van meerdere HTML-elementen om vervolgens visuele attributen toe te wijzen.

We passen de syntaxis toe:

```
h1 {  
  color: black;  
  text-align: center;  
}
```

color en text-align

In het vorige voorbeeld hebben we alle `<h1>`-elementen geselecteerd om vervolgens het attribuut `color` (tekstkleur) toe te wijzen. Het attribuut `color` heeft de waarde `black` (zwart) gekregen. Het resultaat is dat de content van alle `<h1>`-elementen zwart wordt weergegeven. Het attribuut `text-align` (tekst uitlijnen) krijgt de waarde `center`(centreren). De attributen `color` en `text-align` zijn verwisselbaar en worden herkend door alle browsers.

Een #id selecteren

Om een uniek element te selecteren gebruiken we de `id` van het element.

Bijvoorbeeld, in **index.html** is er het element `<div id="logo">`. We kunnen door middel van de `id` dit specifieke element selecteren.

Hier passen we de syntaxis toe:

```
#logo {  
  height: 200px;  
}
```

Omdat geen twee elementen dezelfde `id` mogen hebben, hebben we hier het specifieke element met `id="logo"` geselecteerd en het attribuut `height`(hoogte) toegekend met de waarde `200px`, met andere woorden: 200 pixels hoog.

Een .class selecteren

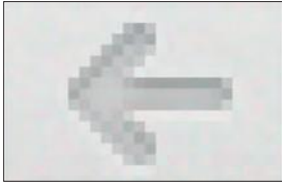
Om alle elementen met dezelfde class-naam te selecteren gebruiken we een punt gevolgd door de naam van de class. We kunnen door middel van de class-naam visuele attributen voor meerdere elementen definiëren.

Hier passen we de syntaxis toe:

```
.blauw {  
  height: 200px;  
}
```


Wat zijn pixels?

Een pixel is een vierkante punt op het beeldscherm van je PC. De resolutie van beeldschermen wordt weergegeven in aantallen pixels. Schermen van smart-tele-foons hebben bijvoorbeeld 600 pixels in de breedte en schermen van pc's hebben tot 1200 pixels in de breedte. Pixels zijn erg klein. Om de pixels van een beeld te zien moeten we de afbeelding vergroten. In de volgende figuur zien we de pixels van de afbeelding.



Figuur 1.9

width en height

Met de attributen `width` (breedte) en `height` (hoogte) definiëren we de dimensies van de elementen in pixels en percentages, bijvoorbeeld:

```
#logo {  
width: 100%; height:  
200px;  
}
```

color

Om te werken met tekst in kleur gebruiken we het attribuut `color` als volgt:

```
color: black;
```

backgroundColor

De attributen `background`, `background-color` en `backgroundColor` zijn verwisselbaar en herkend bij alle browsers.

Nieuwe HTML-properties, zoals `backgroundColor`, worden geschreven in camel-Case. Dat wil zeggen, zonder verbindingstreepje en de eerste letter van het tweede woord krijgt een hoofdletter, dus `backgroundColor`.

Om een achtergrondkleur toe te passen gebruiken we het attribuut `backgroundColor` als volgt:

```
backgroundColor: red;
```

Hier volgt een lijst met de meest voorkomende webkleuren – waarbij je ook het gebruik van camelCase ziet:

aqua

aquaMarine

black

blue blueViolet

brown

cadetBlue

chocolate coral

crimson darkBlue

gray green

greenYellow

indigo

orangeRed red

border

Borders zijn de randen van de elementen. Met het attribuut borderbepalen we de stijl, kleur en dikte van de borders. Er zijn de volgende stijlen:

border: solid	ononderbroken
border: dotted	puntjes
border: dashed	streepjes

Bijvoorbeeld de volgende borderstijlen:

```
border:2pxdottedblack;
```

```
backgroundColor: orange;
```

geven een zwarte rand van puntjes van 2 pixels dik met een oranje achtergrond, zoals hieronder:



Stijlen coderen

Er zijn drie manieren om stijlen te declareren:

1. Binnen het <head>-element (voor korte opdrachten)
2. In line (voor het direct toepassen van stijlen)
3. In een externe style sheet (voor grotere opdrachten)

<head>-stijlen

Stijlen binnen de <head> coderen we met de tags <style> en </style>, bijvoorbeeld:

```
<head>
  <style>
    body{
width: 40%;
font-size: 22px;
    }
  </style>
</head>
```

Hier hebben we stijlen voor de <body> gecodeerd.

In-line-stijlen

In-line-stijlen coderen we direct binnen een element met de eigenschap style=, bijvoorbeeld:

```
<h1 style="color: green;">
```

Hier zeggen we dat de tekstkleur in element <h1> groen is.

In de volgende opgave passen we twee methodes toe bij het coderen van stijlen.

Hier volgt een aantal extra oefeningen om de CSS-attributen goed te doorgronden

- **Oefening 1: backgroundColor**

Open een nieuw bestand en sla het op als **oefening.html**. Voeg de volgende code toe:

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta http-equiv="Content-Type"
        content="text/html;
        charset=UTF-8">
  <title>Oefening</title>
  <style>
    body{
      border: 1px solid black;
      width: 40%;
      font-size: 80px;
    }
    #A{
      width: 150px;
      height: 150px;
      backgroundColor: green;
    }
    #B{
      width: 150px;
      height: 150px;
      backgroundColor: red;
    }
    #C{
      width: 150px;
      height: 150px;
      backgroundColor: blue;
    }
  </style>
</head>
```

```
<body>  
<div id="A"> A  
  </div>  
<div id="B"> B  
  </div>  
<div id="C"> C  
  </div>  
</body>
```



```
</html>
```

Figuur 1.10 Resultaat van oefening 1

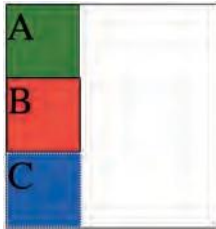
• Oefening 2: Borders

In deze oefening coderen we in-line-stijlen, bijvoorbeeld:

```
<divstyle="border:1pxsolidblack;">
```

voor de borders van elementen A, B en C:

- A met een ononderbroken zwarte rand met een dikte van 1 pixel.
- B met een ononderbroken zwarte rand met een dikte van 2 pixels.
- C met een ononderboken en gestippelde rand met een dikte van 3 pixels.



Figuur 1.11 Resultaat met borders

• Oefening 3: Marges

In deze oefening gaan we meer stijlen toevoegen binnen de <head>.

```
<head>  
  <style>  
    ...  
  </style>  
</head>
```

A krijgt de volgende marges:

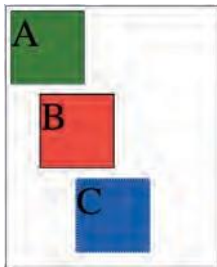
- Een bovenmarge van 10 pixels.
- Een linkermarge van 10 pixels.

B krijgt de volgende marges:

- Een bovenmarge van 20 pixels.
- Een linkermarge van 70 pixels.
- Een ondermarge van 20 pixels.

C krijgt de volgende marges:

- Een auto linkermarge.
- Een auto rechtermarge.
- Een ondermarge van 30 pixels.



Figuur 1.12 Resultaat met marges

• Oefening 4: Padding

De padding (invulling) verwijst naar de ruimte tussen de rand en de content van een element. Een paddinggeven we aan in percentages of pixels. Bijvoorbeeld:

```
padding: 10px 5px 0% 30px;
```

is hetzelfde als:

```
padding-top:10px;
```

```
padding-right: 5px;
```

```
padding-bottom: 0%;
```

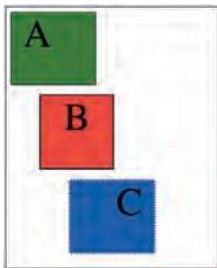
```
padding-left: 30px;
```

Wanneer alle paddingsgelijk zijn zeggen we gewoon:

```
padding: 5%;
```

Codeer in de <head>de volgende stijlen voor A en C.

- A met een linker-padding van 25 pixels.
- C met een rechter-padding van 25 pixels.



Figuur 1.13 Resultaat met paddings

In deze figuur zien we dat elementen A en C breder dan B zijn geworden. Dat komt door de toegevoegde paddings. Het resultaat is dat A en C nu 175 pixels breed zijn.

• Oefening 5: Box-shadow

Voor schaduwen van elementen gebruiken we box-shadow. Bijvoorbeeld:

```
box-shadow: 5px 5px 10px black;
```

Hier coderen we een zwarte schaduw van:

- 5 pixels rechts
- 5 pixels onder
- 10 pixels straal

Met negatieve getallen kunnen we experimenteren met negatieve schaduwen, bij- voorbeeld:

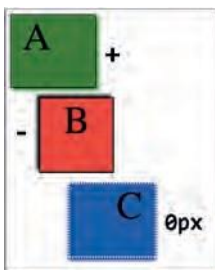
```
box-shadow:-5px -5px 10px black;
```

Dit is een schaduw met:

- 5 pixels links
- 5 pixels boven

Codeer in de <head>schaduwstijlen voor A, B en C.

- A met een groene schaduw van 5px bij 5px en met een straal van 10px.
- B met een zwarte schaduw van -5px bij -5px (negatieve) en met een straal van 10px.
- C met een blauwe schaduw van 0px bij 5px y (onder) en met een straal van 10px.



Figuur 1.14 Resultaat met schaduwen

• Oefening 6: Tekstschaduw

We kunnen ook schaduwen coderen voor teksten. Bijvoorbeeld:

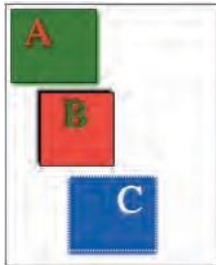
```
text-shadow: 5px 5px 10px black;
```

Dit is een positieve schaduw voor teksten. Het lijkt alsof het licht van boven links komt. Met negatieve getallen kunnen we met negatieve schaduwen experimenteren. Bijvoorbeeld:

```
text-shadow:-5px -5px 10px black;
```

Codeer in de <head>schaduwen voor de teksten A, B en C.

- A met rode tekst en een rodeschaduw
- B met groene tekst en een negatieve groene schaduw
- C met witte tekst en een witte schaduw onder



Figuur 1.15 Resultaat met tekstschaduw

Hier merken we op dat de schaduw van de letters dezelfde richting uitgaan, net als de schaduwen van de borders.

De stylesheet

In de volgende opgave maken we onze eerste stylesheet met de naam **styles** en met de extensie **.css**. Alle stylesheets krijgen de extensie **.css** dat staat voor Cascading Style Sheet.

De <link>-tag

Om onze webpagina te verbinden met onze stylesheet gebruiken we de tag <link>. Zo kunnen we onze stijlen linken met een of meerdere webpagina's. In de volgende opgave linken we **index.html** met het bestand **stijlen.css**.

De tag <link> bevat de volgende attributen:

rel="stylesheet" Stelt de relatie vast met
stylesheet href="styles.css" Verwijst naar het bestand
styles.css

/* commentaar */

Met commentaar kunnen we onze stijlen beschrijven. CSS-commentaar plaatsen we tussen /* en */ in.

overflow

Als een element te klein is voor de tekstinhoud dan zeggen we dat de tekst over- loopt. In deze gevallen kunnen we kiezen voor het tonen van een scrollbar (schuif- balk).

overflow: auto;

text-align

Je ziet in het resultaat van Webbouw 5 en 6 dat alle teksten gecentreerd zijn weer- gegeven. Het attribuut text-align (tekst uitlijnen) kent vier opties:

text-align: left text-

align: center text-align:

right text-align: justify

links uitlijnen
centreren
rechts uitlijnen
de tekst uitvullen over de hele regel

Een tekst centreren is niet hetzelfde als het centreren van een element. In ons geval hebben we het <body>-element gecentreerd binnen het <html>-element. Dit hebben we gedaan met het attribuut margin.

margin

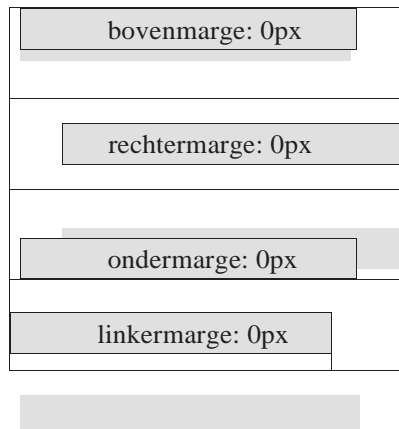
Marges zijn de ruimtes tussen alle elementen. We kunnen individuele marges specifie- ceren:

margin-top: 0px

margin-right: 0px

margin-bottom: 0px

margin-left: 0px



Hierboven zien we de marges van de grijze elementen ten opzichte van de elementen waar ze zich in bevinden.

We kunnen ook alle marges in één keer vaststellen. Dit doen we met de klok mee: boven, rechts, onder, links:

```
margin: 0 0 0 0;
```

Als alle vier de marges hetzelfde zijn doe je dit zo:

```
margin: 0; of margin: 5%;
```

Elementen centreren

Ons `<body>`-element heeft in Webbrowser 4 een bovenmarge van 0% en een ondermarge van 5% gekregen. Om het element te kunnen centreren moeten we de linker- en de rechtermarges als auto opgeven. Bijvoorbeeld:

```
body{
margin-left:auto; margin-
right: auto;
}
```

Border

Het `<body>`-element heeft een border (rand) van 1 pixel gekregen, als volgt:

```
border: 1px solid;
```

font

We gebruiken het attribuut `font` om het lettertype van de teksten binnen een element te definiëren. Hier volgt een lijst met lettertypes die geschikt zijn voor alle webbrowsers:

Georgia, serif
Palatino, serif
Times New Roman
Arial, Helvetica, sans-serif
Arial Black, Gadget, sans-serif
"Comic Sans MS", cursive, sans-serif
Impact, Charcoal, sans-serif
"Lucida Sans Unicode", "Lucida Grande", sans-serif
Tahoma, Geneva, sans-serif
"Trebuchet MS", Helvetica, sans-serif
Verdana, Geneva, sans-serif
"Courier New", Courier, monospace
"Lucida Console", Monaco, monospace

Figuur 1.18

font-size

Het attribuut font-size(tekengrootte) definieert de lettergrootte van je teksten.

Dit attribuut kunnen we waarden geven in pixels, punten, percentages of relatief ten opzichte van het font in de body, bijvoorbeeld 1.5 de font-size van het <body>- element. Hieronder zien we hoe het font serif gedefinieerd kan worden in pixels, punten, percentages of relatief ten opzichte van de basis:

```
cerif12px      pixels
cerif24pt      punten
cerif48%
               percentage cerif 1.5rem
               relatief
```

De meest exacte lettergrootte geven we aan in pixels (px).

Voor een betere kwaliteit afdruk van je teksten gebruik je punten (pt).

Percentages en rem zijn gerelateerd aan de lettergrootte van het <body>- element, bijvoorbeeld:

body{font-size:16px;}	basis
body h1{font-size:2.0rem;}	relatief: $2.0 \times 16 = 32\text{px}$
bodyh2{font-size:50%;}	percentage: 50% van $16 = 8\text{px}$

Subelementen

In onze webpagina zien we dat de body een nav-element heeft en dat het nav-element een h3-element heeft. Deze elementen noemen we subelementen.

We kunnen heel specifiek een subelement selecteren door het pad van het element aan te geven. Bijvoorbeeld:

```
body nav {color: black}
```

of

```
body nav h3 {color: black}
```

font-style

We kunnen ons lettertype verder vormgeven met font-style. Hiervoor zijn er vier opties:

normal	normaal
bold	vet
bolder	vetter
lighter	lichter

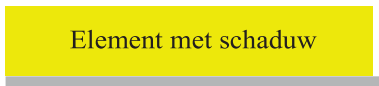
box-shadow

Om de dimensies van onze elementen beter te zien hebben we in Webbouw 5 schaduwen gebruikt. In ons geval hebben we de volgende schaduw gedefinieerd:

```
box-shadow: 4px 4px 10px rgba(0,0,0,0.5);
```

Dit betekent:

- 4 pixels aan de rechterkant
- 4 pixels onderaan
- 10 pixels vervaging
 - De kleur van de schaduw is zwart (0 rood, 0 groen, 0 blauw, dat is geen kleur, dus zwart) en is 50% transparant. Kleurencodes en de functie `rgba` bespreken we in de volgende paragraaf.



• Lab 02

In deze lab-opdracht codeer je stijlen voor de vier div-elementen zoals in de volgende figuur te zien is. Codeer een stylesheet met de volgende stijlen en sla die op als **labs.css**.

Div 1 Border-shadow 80%
breed Hoogte 80px Font-size
20px Auto marge Marge
boven 3% Marge onder 3%

Div 2 80% breed
Hoogte 220px Font-size
15px Linkermarge 5% Marge
onder 3%

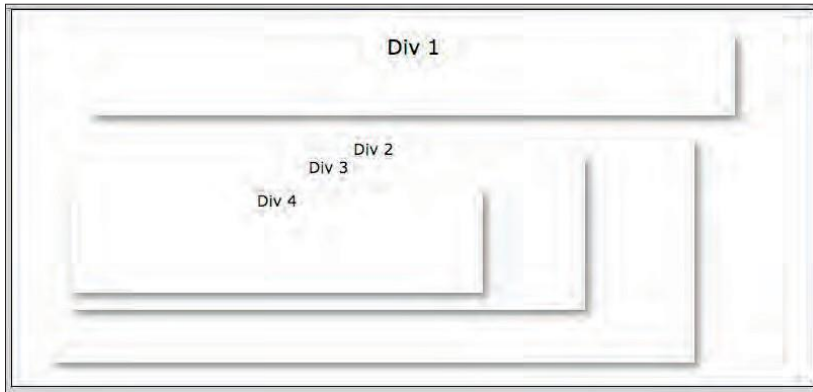
Div 3 80% breed
Hoogte 150px Linkermarge 3%

Div 4 80% breed
Hoogte 100px Linkermarge 0%

Open **labs.html** en voeg de volgende link eraan toe:

```
<link rel="stylesheet" href="labs.css">
```


Het resultaat moet er als volgt uitzien:



Figuur Lab 02

1.3 Webkleuren

In deze paragraaf maken we kennis met de volgende CSS-attributen:

- rgb
- rgba

Inleiding

Webkleuren zijn gebaseerd op de drie primaire kleuren: rood, groen en blauw. Door deze drie basiskleuren te combineren kunnen we een spectrum van kleuren creëren. In de volgende figuur zie je primaire, secundaire en tertiaire webkleuren.



Figuur 1.19 Kleurenschijf

Er zijn drie manieren om kleuren te produceren in CSS:

- met decimale getallen;
- met hexadecimale getallen;
- met percentages.

Omdat de browsers constant in ontwikkeling zijn en omdat de ene browser een betere manier vindt om bijvoorbeeld kleuren te coderen, ontstaan hierdoor meerdere manieren om kleuren te coderen.

De meest voorkomende manier om kleuren te coderen voor een webpagina is met hexadecimale getallen. Hexadecimaal is een numeriek systeem gebaseerd op tien getallen en zes letters (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). We gaan niet dieper in op het hexadecimale systeem, maar het is belangrijk om te weten hoe het systeem wordt gebruikt om kleuren te combineren. Dit doen we met het attribuut `rgb`.

`rgb`

Met het attribuut `rgb` kunnen we de drie primaire kleuren, rood, groen en blauw mengen met als resultaat een nieuwe kleur. De syntaxis is als volgt:

```
rgb(red,green,blue)
```

De mogelijke waarden voor iedere kleur zijn de volgende:

- hexadecimaal 0 t/m F;
- decimaal 0 t/m 255;
- percentage 0 t/m 100.

Bijvoorbeeld, de kleur blauw coderen we als volgt:

```
rgb(0,0,F)
```

Dit betekent: nul rood, nul groen en maximaal blauw.

We kunnen ook groen toevoegen, als volgt:

```
rgb(0,6,F)
```

Dit geeft als resultaat een blauwgroene kleur. Dezelfde kleur kunnen we op een verkorte manier coderen:

```
#06F
```

Nog een andere manier is met zes cijfers. Dit doen we door elk cijfer te verdubbelen. Hierdoor heb je veel meer kleurenmogelijkheden. Bijvoorbeeld:

```
#06F is hetzelfde als #0066FF
```

• **Oefening 7: BackgroundColor**

Open een nieuw script en sla dit op als **oefening7.html**. Maak de volgende vijf div's met de aangegeven achtergrondkleuren. Gebruik het attribuut `backgroundColor`:

`backgroundColor: #fff;`

Het resultaat moet er als volgt uitzien:

kleur	percentage	decimaal	hexadecimaal	kort
zwart	rgb(0%,0%,0%)	rgb(0,0,0)	#000000	#000
rood	rgb(100%,0%,0%)	rgb(255,0,0)	#FF0000	#f00
groen	rgb(0%,0%,100%)	rgb(0,255,0)	#00FF00	#0f0
blauw	rgb(0%,0%,100%)	rgb(0,0,255)	#0000FF	#00f
wit	rgb(100%,100%,100%)	rgb(255,255,255)	#FFFFFF	#fff

Figuur 1.20

Op internet vind je talloze webpagina's met tabellen met kleurcodes zoals in de volgende figuur:

FFF	CCC	999	666	333	000	FFC	FF9	FF6	FF3										
FFF	CCC	999	666	333	000	C00	900	600	300										
99C						CC9	FFC	FFC	FF9	FF6	CC3							CC0	
C00						900	C33	C66	966	633	300							033	
CCF	CCF	333	666	999	CCC	FFF	CC9	CC6	330	660	990	CC0	FF0	FF3	FF0				
F00	F33	300	600	900	C00	F00	933	633	000	000	000	000	000	366	033				
99F	CCF	99C	666	999	CCC	FFF	996	993	663	993	CC3	FF3	CC3	FF6	FF0				
F00	F66	C33	633	933	C33	F33	600	300	333	333	333	366	699	066					
66F	99F	66C	669	999	CCC	FFF	996	663	996	CC6	FF6	990	CC3	FF6	FF0				
F00	F66	C33	900	966	C66	F66	633	300	666	666	666	033	399	6CC	099				
33F	66F	339	66C	99F	CCC	FFF	CC9	CC6	CC9	FF9	FF3	CC0	990	FF3	FF0				
F00	F33	900	C00	F33	C99	F99	966	600	999	999	399	066	066	3CC	0CC				
00C	33C	336	669	99C	CCF	FFF	FFC	FF9	FFC	FF9	CC6	993	660	CC0	330				
C00	C00	600	933	C66	F99	FCC	C99	933	CCC	9CC	699	366	033	099	033				
33C	66C	00F	33F	66F	99F	CCF				CC9	996	993	990	663	660				
C33	C66	F00	F33	F66	F99	FCC				9CC	699	399	099	366	066				
006	336	009	339	669	99C					FFC	FF9	FF6	FF3	FF0	CC6	CC3			
600	633	900	933	966	C99					CFF	9FF	6FF	3FF	0FF	6CC	3CC			
003	00C	006	339	66C	99F	CCF	339	99C	CCC	CC9	996	663	330	990	CC0				
300	C33	633	966	C99	FCC	FFF	9FF	CFF	CFF	9FF	6CC	399	066	0CC	0CC				
00F	33F	009	00C	33F	99F	99C	006	669	999	999	993	660	660	CC3	CC0				
F33	F66	933	C66	F99	FFF	CCC	6CC	9CC	9FF	9CC	3FF	0CC	099	3FF	0FF				
00F	66F	33C	009	66F	66C	669	003	336	666	666	666	330	993	CC6	990				
F66	F99	C66	966	FFF	CCC	999	366	699	6FF	6CC	699	099	3CC	6FF	0FF				
00F	66F	33C	33F	33C	339	336	006	003	333	333	333	333	663	996	660				
F99	FCC	C99	FFF	CCC	999	666	699	399	3FF	3CC	399	366	3CC	6FF	0FF				
00F	33F	00F	00C	009	006	003	339	336	000	000	000	000	000	663	330				
FCC	FCC	FFF	CCC	999	666	333	9CC	6CC	0FF	0CC	099	066	033	3FF	0FF				
00C							009	33C	66C	669	336	003			330				
C99							9CC	CFF	CFF	9FF	6FF	3CC			0CC				
							00C	009	006	003									
							CFF	9FF	6FF	3FF									

Figuur 1.21

Met de decimale methode kunnen we dezelfde groenblauwe kleur met de decimale waarden 0 t/m 255 als volgt creëren:

```
rgb(0,56,255)
```

Met percentages kunnen we dezelfde kleur als volgt creëren:

```
rgb(0%, 22%, 100%)
```

De afwezigheid van kleuren geeft als resultaat de kleur zwart:

```
rgb(0, 0, 0)
```

Hier zie je dat spaties na de komma optioneel zijn.

rgba

We gebruiken het attribuut `rgba` om transparantie toe te voegen, waarbij de transparantie aangeeft en een waarde tussen 0.0 en 1.0 krijgt. Bijvoorbeeld:

```
rgba(0, 255, 0, 0.3)
```

Hier krijgen we een groene kleur met een transparantie van 30%. In de vorige opga- ve hebben we dit attribuut gebruikt om een zwarte schaduw met een transparantie van 50% te creëren:

```
box-shadow: 4px 4px 10px rgba(0,0,0,0.5);
```

• Oefening 8: rgba

Open een nieuw script en sla dit op als **oefening8.html**. Maak de volgende div's met de aangegeven transparanties in rood.



Figuur 1.22

- **Oefening 9: #kleurcodes**

Open een nieuw script en sla het op als **oefening9.html**. Maak de volgende div's met de aangegeven kleurencodes:



Figuur 1.23

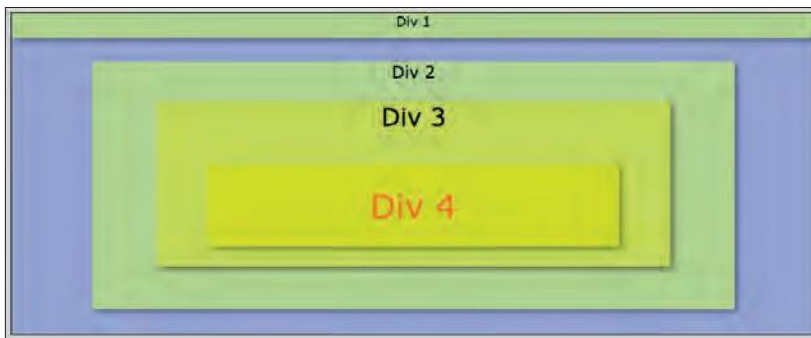
- **Lab 03**

Open **labs.css** en geef iedere div een eigen achtergrondkleur, hoogte en breedte. Bij div 4 centreer je verticaal de tekst als volgt:

```
#div4 p{
position: relative; top: 50%;
transform: translateY(-50%);
}
```

Position wordt behandeld in paragraaf 1.5 en transform in paragraaf 1.6.

Het resultaat moet er als volgt uitzien:



Figuur Lab 03

1.4 Navigatie

In deze paragraaf maken we de navigatie van de website en we maken kennis met een aantal HTML-elementen en een aantal CSS-attributen.

Een goed gestructureerde navigatie bepaalt of je website logisch en intuïtief is. De navigatie moet zodanig gemaakt worden dat de gebruiker de gezochte informatie gemakkelijk kan vinden. We beginnen met de HTML-code en eindigen met de CSS-code.

Codering van de navigatie

De navigatie moet logisch opgebouwd worden zodat gezochte informatie nooit meer dan drie klikken verwijderd is van de homepage. Het moet mogelijk zijn om vanuit iedere webpagina terug te keren naar de homepage. HTML5 heeft de nieu- we tag `<nav>` met als betekenis 'navigatie'.

Element `<nav>`

Meestal plaatsen we de navigatie bovenaan of aan de rechterkant van de pagina zodat we door de hele website makkelijk kunnen navigeren. Omdat de navigatie in de vorm van een lijst wordt gecodeerd kijken we nu naar HTML-lijsten.

Genummerde lijsten ``

Soms wil je een bepaald stuk tekst als lijst weergeven. Er bestaan twee soorten lijsten: genummerde lijsten en opsommingslijsten. We kijken eerst naar genummerde lijsten.

```
<ol>
<li>Webdesign</li>
<li>Webhosting</li>
<li>Webprogrammering</li>
</ol>
```

Dit genereert het volgende resultaat: een genummerde lijst `` met drie items ``:

1. Webdesign
2. Webhosting
3. Webprogrammering

Items ``

Alle lijsten moeten minimaal een item `` hebben. Deze genummerde items worden automatisch genummerd volgens hun positie in de lijst.

Opsommingslijsten ``

Een opsommingslijst is een lijst met bullets. Bijvoorbeeld:

```
<ul>
<li>Duurzaam</li>
<li>Ecologisch</li>
<li>Groene voetafdruk</li>
</ul>
```

genereert de volgende bullets-lijst:

- Duurzaam
- Ecologisch
- Groene voetafdruk

Element <a>

Het element <a> is een klikbaar element voor het linken van twee documenten. Deze elementen noemen we hyperlinks. Een hyperlink coderen we als volgt:

```
<a href="index.html">Home</a>
```

Het <a>-element bestaat uit drie delen:

- 1 , met het attribuut href met de naam van het te linken document. In dit geval index.html.
- 2 De tekst waar we moeten klikken om te linken. In dit geval Home.
- 3 , om de hyperlink te sluiten.

HTML bestaat voor Hyper Text Markup Language. De hypertext is de content of de tekst tussen de tags in. In een hypertext is het zo dat in de browser een, twee of meer spaties als één spatie worden gezien. Als we een extra spatie willen toevoegen gebruiken we de code . Bijvoorbeeld
<h1> tekst</h1>.

In de volgende opgave vervangen we het <h3>-element binnen de navigatie <nav> door een lijst met hyperlinks.

```
<nav>
<h3>Navigatie</h3>
</nav>
```

list-style-type

Dit attribuut is voor de opmaak van lijsten. We willen onze hyperlinks hebben in een lijst zonder bullets. Dus zeggen we het volgende:

```
li{
list-style-type: none;
}
```

Naast de optie none kunnen we de volgende opties gebruiken:

disc	disc	?
circle	cirkel	?
square	vierkant	?
decimal	decimaal	1
lower-latin	kleine letter Latijn	i
upper-latin	hoofdletter Latijn	I

Het <body>-element bevat het <nav>-element en dat krijgt als breedte 100% van het <body>-element. In de vorige opgave hebben we een navigatie gemaakt met vier hyperlinks, dus krijgt iedere link 25% van de breedte van de navigatie.

```
nav ul li{
width: 25%;
}
```

display: block

De volgende HTML-elementen krijgen automatisch het attribuut display: block.

- <h1>/<h6>
- <p>
-
- <div>
- <main>
- <footer>
- <section>

Onze -elementen binnen de navigatie krijgen ieder 25% van de breedte van het nav-element en worden als blokken onder elkaar weergegeven.

 Home 	
 Playlist 	
 Formulier 	
 Contact 	

float

Om onze hyperlinks horizontaal te groeperen gebruiken we het attribuut float. Elementen met dit attribuut hebben de mogelijkheid om naar links of rechts te ‘zweven’.

- float: left naar links zweven
- float: right naar rechts zweven

Neem als voorbeeld de letters in een tekst: alle letters hebben de neiging om naar links te ‘zweven’ naar de vorige letter. Bijvoorbeeld:

ABCD

Maar als het scherm te klein is gaat de letter onderaan naar de volgende regel, bij- voorbeeld:

ABCD
E

Dat gebeurt ook met HTML-elementen met het attribuut float. Als we bijvoorbeeld onze -elementen een breedte van 30% geven dan ‘zweeft’ het vierde element onderaan naar de volgende beschikbare regel.

Home	Playlist	Formulier	
Contact			



clear

Met het attribuut clear kunnen we het ‘zweven’ ongedaan maken.

clear: left naar links zweven ongedaan maken

clear: right naar rechts zweven ongedaan maken

clear: both naar links en rechts zweven ongedaan maken

gradient (kleurverzadiging of kleurverloop)

Als laatste hebben we in de vorige opgave het attribuut linearkleurverloop gecodeerd. Met dit attribuut vullen we een achtergrond in met een kleurverloop. Er zijn twee soorten gradients: lineairen radial. De syntaxis is als volgt:

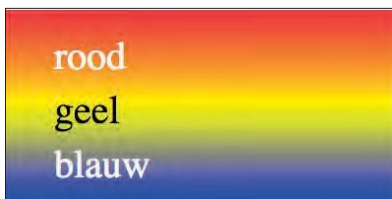
```
linear-gradient(kleur-1, kleur-2, . . .)
```

Bijvoorbeeld:

```
linear-gradient(red, blue);
```

```
linear-gradient(red, yellow, blue);
```

De volgende figuur illustreert het resultaat van de tweede regel: een lineair kleur- verloop van rood naar geel naarblauw.



Figuur 1.26

We kunnen ook een diagonaal kleurverloop coderen:

```
linear-gradient(45deg, red, yellow, blue); linear-
```

```
gradient(tolefttop,red,yellow,blue); linear-
```

```
gradient(tobottom,red,yellow,blue);
```

text-decoration

Met het attribuut text-decoration kunnen we de volgende opties toepassen: door- strepen, onderstrepen, bovenstrepen.

none geen
underline
 onderstrepen overline
 lijn erboven line-trog
 ~~doorhalen~~

Alle hyperlinks krijgen automatisch een onderstreping. Onderstrepen schakelen we als volgt uit:

text-decoration: none;

opacity

Het attribuut opacity bepaalt de transparantie van een element. In de vorige op- gave kregen alle hyperlinks een transparantie van 50%.

opacity: 0.5;

pseudoklassen

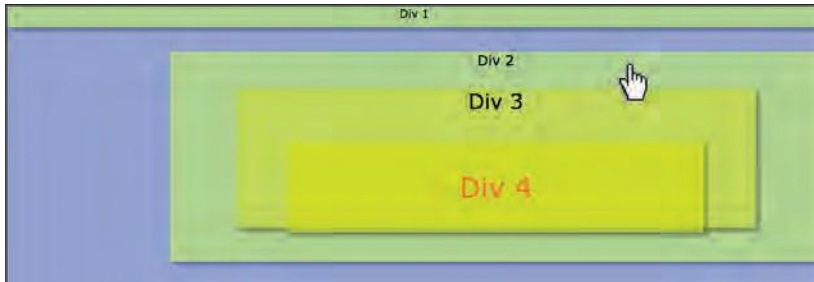
Het beeldscherm is de interface van de browser. Een interface is het punt van inter- actie tussen twee systemen of objecten. Browsers hebben listeners (luisteraars) die permanent aan het luisteren zijn naar events (gebeurtenissen) op het beeldscherm. Bijvoorbeeld wanneer de muis over een tekst zweeft. Dit noemen we muis-events. Muis-events kunnen we programmeren met behulp van pseudoklassen. Hyperlinks hebben de volgende pseudoklassen:

a:hover wanneer de muis over een hyperlink zweeft
a:focus wanneer de muis op een hyperlink focust
a:visited wanneer de muis een hyperlink heeft gactiveeerd
a:active wanneer de muis een hyperlink activeert

De pseudoklassen a:visited en a:active veranderen de kleur van de bezochte en de actieve hyperlinks. Zo kunnen we zien welke links we bezocht hebben. In de laatste opgave hebben we met behulp van de pseudoklasse a: hover een hover-effect gecodeerd zodat de kleur van de tekst van alle hyperlinks verandert wanneer de muis over de link zweeft.

• Lab 04

Open **labs.css**. In deze lab-opdracht codeer je een hover-effect zodat div 2 naar rechts zweeft met een muis-over-event. Het resultaat moet er als volgt uitzien:



Figuur Lab 04

Element img

Hier hebben we een afbeelding toegevoegd met het element ``. Dit element heeft drie attributen:

`width` de afbeelding wordt weergegeven met deze breedte

`src` de bron of URL van de afbeelding

`alt` alternatieve tekst wanneer de afbeelding niet gevonden is

Het attribuut `src` bevat de bron van de afbeelding. Dit kan een URL zijn of een pad naar waar de afbeelding geplaatst is. In ons geval is dit de map **images** gevolgd door `/` en de naam van de afbeelding:

```
src="images/logo.png"
```

Wat is een URL?

URL is het acroniem voor *Uniform Resource Locator*, het is de manier waarop we een document of resource lokaliseren op het internet. De algemene vorm van een URL is:

```
protocol: //domain/map/document
```

protocol is een set normen voor de transmissie van informatie tussen twee computers. In ons geval gebruiken we het http-protocol voor de transmissie van webpagina's over het internet.

domain is een gereserveerde naam die de computer of server in het internet identificeert, bijvoorbeeld:

`http://google.com`

`map/document` geeft de locatie en naam van het gezochte document.

Design van het logo

In de vorige webbouw hebben we het logo in een ``-element geplaatst. Een andere manier om ons logo weer te geven is met behulp van de `background-property`. CSS biedt ons meer mogelijkheden.

background

Alle elementen hebben een achtergrond. Aan het attribuut `background` kun je afbeeldingen, kleuren en arcering toekennen. Dit attribuut heeft de volgende opties:

<code>background-image</code>	achtergrond van de afbeelding of kleurverloop
<code>backgroundColor</code>	achtergrondkleur
<code>background-position</code>	positie van de achtergrond van de afbeelding
<code>background-repeat</code>	herhaling van de afbeelding in de achtergrond
<code>background-url</code>	url van de afbeelding voor deze achtergrond

background-url

In ons geval hebben we een url met een afbeelding gecodeerd:

```
background-image: url(images/disc.png);
```

background-position

Het attribuut position bepaalt de positie van de afbeelding in de achtergrond, het heeft de volgende opties:

top boven

center midden

bottom onder

topleft boven links

topcenter boven midden

topright boven rechts

bottomleft onder links

bottomcenter onder

midden bottomright onder rechts

In ons geval hebben we gekozen voor:

```
background-position: center;
```

De optie repeat is voor het herhalen van de afbeelding in de achtergrond. We hebben gekozen voor:

```
background-repeat: no-repeat;
```

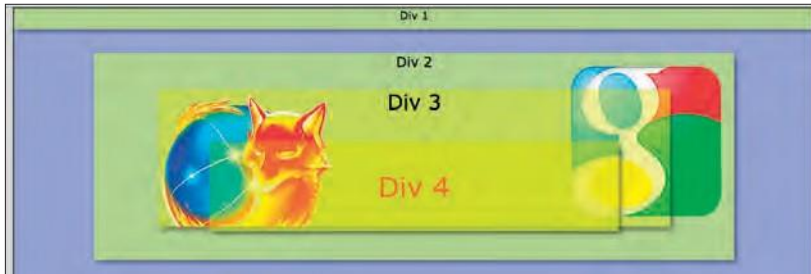
- **Lab 05**

Open **labs.css** en plaats de volgende twee background-url's als afbeeldingen binnen div 2 en div 3.

<http://icons.iconarchive.com/icons/yootheme/social-bookmark/256/social-google-box-icon.png>

<http://icons.iconseeker.com/ico/mozilla/firefox-2.ico>

Het resultaat is het volgende:



Figuur Lab 05