

Lay-out controleren

Content boxes centreren

Een van de basisvereisten bij het weergeven van documentcomponenten is de mogelijkheid om inhoudsblokken horizontaal te centreren.

In CSS wordt de mogelijkheid om elementen op blokniveau horizontaal te centreren geleverd door de eigenschap **margin** en het sleutelwoord **auto**. Wanneer het sleutelwoord **auto** is opgegeven voor de eigenschap **margin** van een element, berekent de browser eerst de afstand links en rechts van dat element, tot aan de grenzen van het element dat het bevat, en verdeelt vervolgens het totaal in twee om de waarde van elke zijmarge te berekenen. Als bijvoorbeeld een **margin : auto** toepast op een "**div**" - element met een breedte van 80px, dat zich binnen een buiten element van 200px breed bevindt, deelt de browser het totale verschil van 12 px in de helft en past vervolgens 60px brede marges toe aan elke zijde van het div-element - zodat het centraal wordt geplaatst binnen de element.

Merk op dat **margin : auto** niet verticaal centreert, maar alleen de bovenste en onderste marges op nul zet, dus er zijn geen margegebieden boven of onder het elementblok.

Het is belangrijk om op te herkennen dat **margin : auto** inhoudsvakken op blokniveau centreert en niet moet worden verward met **text-align : center** dat kan worden gebruikt om inhoud te centreren in inline inhoudvakken.

- 1 Start een teksteditor en maak een HTML5-document met een "div" -element dat nog twee elementen bevat. Tik de code in:

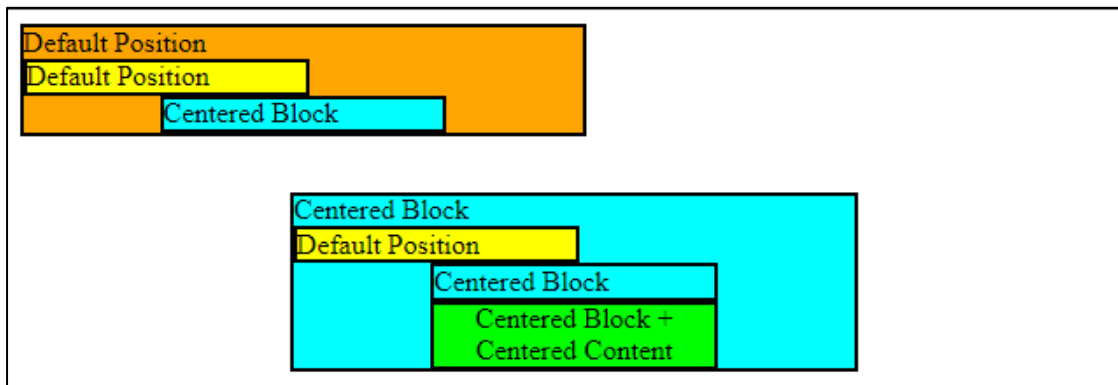
```
<!DOCTYPE HTML >
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Centering Content</title>
    <link rel="stylesheet" type="text/css" href="center.css">
  </head>
  <body>
    <div>Default Position
      <div class="inner">Default Position</div>
      <div class="inner center">Centered Block</div>
    </div>
    <div class="center">Centered Block
      <div class="inner">Default Position</div>
      <div class="center inner">Centered Block</div>
      <div class="center inner center-text">Centered Block +
        Centered Content</div>
    </div>
  </body>
</html>
```

Lay-out controleren

- 2 Sla het document op als "**center.html**".
- 3 Start een nieuw CSS document. Tik de onderstaande code in:

```
/* Validated by W3C CSS Validation Service */
div { margin: 0 0 2em 0 ; width : 20em ; border : 0.1em solid black ;
background : orange }
div.inner { margin : 0 ; width : 10em ; background : yellow }
div.center { margin : auto ; background : aqua }
div.center-text { text-align : center ; background : lime }
```

- 4 Sla het document op als "**center.css**".
- 5 Open het document **center.html** in een browser en bekijk het resultaat. Analyseer de code.



Absoluut positioneren van boxes

Bij het indelen van de element inhoudsvakken van een webpagina heeft de CSS **position** eigenschap een standaardwaarde van **static** - die het normale **flow positioning** schema vertegenwoordigt. Door een andere waarde toe te wijzen aan de **position** eigenschap van een element, kan dat element uit de normale stroom (*flow*) worden verwijderd zodat het onafhankelijk kan worden gepositioneerd.

Alternatieven voor de standaard **static** waarde kunnen worden opgegeven met behulp van de trefwoorden **absolute**, **relative** en **fixed** om een alternatief positioneringsschema op te geven tegenover de normale *flow layout*.

De **absolute**, **relative** en **fixed** positioneringsschema's gebruiken elk een of meer van de CSS "*offset*" -eigenschappen **top**, **right**, **bottom** en **left**, om hun positie te definiëren.

Lay-out controleren

Wanneer de **position** eigenschap als **absolute** wordt opgegeven, plaatst het positioneringsschema het element op de opgegeven offset-afstand van de grenzen van het bevattende element. Een "div"-element met waarden voor de **top** en **left** met een waarde van 100px wordt bijvoorbeeld 100px onder en rechts van de grenzen van de container geplaatst.

- 1 Start een teksteditor en maak een HTML5-document met vier "div"-elementen die elk een binnenste div-element bevatten. Tik de code in:

```
<!DOCTYPE HTML >
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Absolute Positioning</title>
    <link rel="stylesheet" type="text/css" href="absolute.css">
  </head>
  <body>
    <div class="top-left large">Top:0 Left:0
      <div class="btm-right small">Bottom:0 Right:0</div>
    </div>

    <div class="top-right large">Top:0 Right:0
      <div class="btm-left small">Bottom:0 Left:0</div>
    </div>

    <div class="btm-left large"><br><br><br>Bottom:0 Left:0
      <div class="top-right small">Top:0 Right:0</div>
    </div>

    <div class="btm-right large"><br><br><br>Bottom:0 Right:0
      <div class="top-left small">Top:0 Left:0</div>
    </div>
  </body>
</html>
```

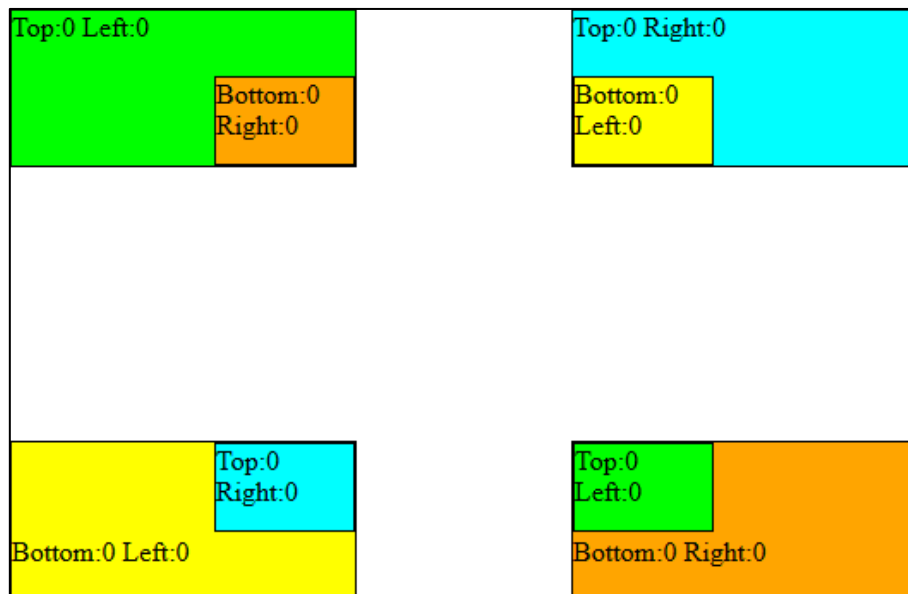
- 2 Sla het document op als "**absolute.html**".
- 3 Start een nieuw CSS document. Tik de onderstaande code in:

```
div.large { width : 200px ; height : 90px ; border : 1px solid black }
div.small { width : 80px ; height : 50px ; border : 1px solid black }
div.top-left { position : absolute ; top : 0px ; left : 0px ; background : lime }
```

Lay-out controleren

```
div.top-right { position : absolute ; top : 0px ; right : 0px ; background : aqua }
div.btm-left { position : absolute ; bottom : 0px ; left : 0px ; background : yellow }
div.btm-right { position : absolute ; bottom : 0px ; right : 0px ; background : orange }
```

- 4 Sla het document op als "**absolute.css**".
- 5 Open het document **absolute.html** in een browser en bekijk het resultaat. Analyseer de code.

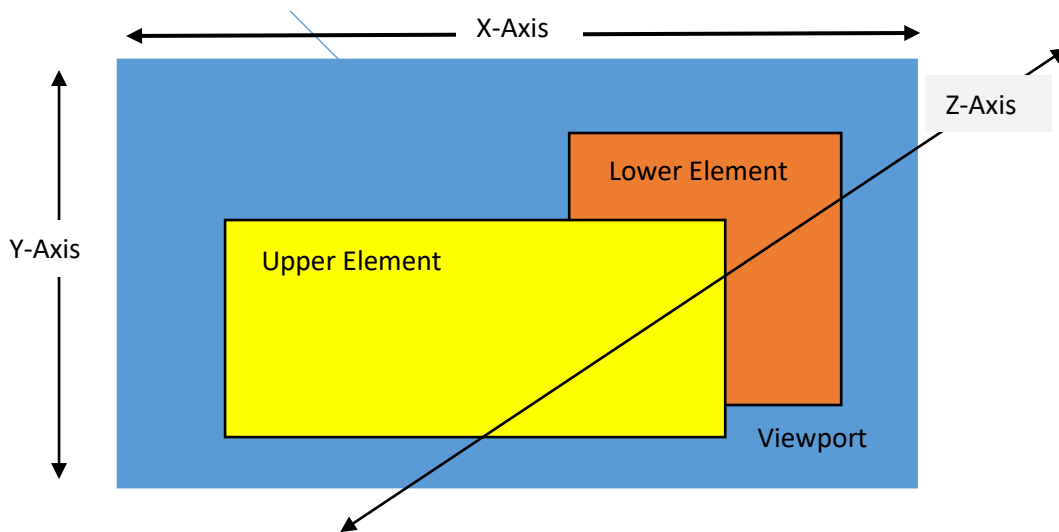


Inhoudvakken stapelen

Door van het **static** standaard positieschema te veranderen, door de **absolute** waarde toe te wijzen aan de eigenschap **position**, kunnen elementen elkaar overlappen - ze worden boven elkaar gestapeld in dezelfde volgorde als dat ze in de HTML-code voorkomen.

De stapelvolgorde kan echter expliciet worden opgegeven in CSS door een geheel getal toe te wijzen aan de eigenschap **z-index** van elk element. Het element met de hoogste waarde verschijnt bovenaan, daaronder verschijnt het element met de volgende hoogste waarde enzovoort.

Het **absolute** positioneringsschema maakt het dus mogelijk om de elementpositie nauwkeurig te regelen in drie dimensies met behulp van XYZ-coördinaten - langs de X-as met **left** en **right** *offset* eigenschappen, langs de Y-as met **top** en **bottom** *offset* eigenschappen en langs de Z-as met de **z-index** stapelvolgorde eigenschap.



- 1 Start een teksteditor en maak een HTML5-document met vier "div"-elementen die overlappend moeten worden geplaatst. Tik de code in:

```
<!DOCTYPE HTML >
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Absolute Positioning</title>
    <link rel="stylesheet" type="text/css" href="stack.css">
  </head>
  <body>
    <div id="pale-gray"> </div>
    <div id="lite-gray"> </div>
    <div id="dark-gray"> </div>
    <div id="text-box">
      Text content here on the top of the element stack - lower
      offset elements create a drop shadow effect
    </div>
  </body>
</html>
```

- 2 Sla het document op als "**absolute.html**".
- 3 Start een nieuw CSS document. Tik de onderstaande code in:

Lay-out controleren

```
div { width : 400px ; height : 50px }
```

```
div#text-box { position : absolute ; top : 20px ; left : 20px ; background : yellow ; z-index : 4 }
```

```
div#dark-gray { position : absolute ; top : 24px ; left : 24px ; background : #666 ; z-index : 3 }
```

```
div#lite-gray { position : absolute ; top : 28px ; left : 28px ; background : #999 ; z-index : 2 }
```

```
div#pale-gray { position : absolute ; top : 32px ; left : 32px ; background : #CCC ; z-index : 1 }
```

- 4 Sla het document op als "**absolute.css**".
- 5 Open het document **absolute.html** in een browser en bekijk het resultaat. Analyseer de code.

Text content here on the top of the element stack - lower offset elements create a drop shadow effect

Inhoudsvakken relatief plaatsen

In tegenstelling tot een **absolute** positioneringsschema, dat een element volledig uit de normale stroomindeling verwijdert, past een **relative** positioneringsschema de positie van een element aan ten opzichte van de positie die het oorspronkelijk zou innemen in een normale *flow layout*.

Door van het standaard **static** positioneringsschema te veranderen, door de **relative** waarde toe te wijzen aan de CSS **position**-eigenschap, kunnen elementen vanuit hun normale stroom opmaak (flow layout) positie worden verplaatst met behulp van de **offset** eigenschappen **top**, **right**, **bottom** en **left**.

Als bijvoorbeeld een **top** waarde van -18px wordt opgegeven, wordt het geselecteerde element naar boven verplaatst en als een waarde voor **left** een waarde van 100px wordt opgegeven, wordt het naar rechts verplaatst - maar cruciaal is dat de ruimte die wordt ingenomen door de oorspronkelijke lay-outpositie bewaard blijft. Het toepassen van deze

a repositioned span element
 There is on this line - it pays no heed to other paragraph content
 for it is an inline content box

Lay-out controleren

relatieve positiewaarden op een span-element binnen een alinea heeft dit effect:

Merk op hoe de originele inhoud wordt verplaatst van de normale positie van de stroomindeling naar een nieuw gecreëerde inhoudvak positie op de opgegeven afstand ten opzichte van de oorspronkelijke positie. Deze relatieve positie zal behouden blijven, zelfs wanneer de positie van het buitenste inhoudselement wordt veranderd.

Dus hoewel **absolute** positionering typisch de positie van het buitenste element kan regelen, is het **relative** positioneringsschema vaak nuttig om de positie van geneste elementen te regelen.

- 1 Start een teksteditor en maak een HTML5-document met vier geneste "div" -elementen die overlappend moeten worden geplaatst. Tik de code in:

```
<!DOCTYPE HTML >
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Absolute Positioning</title>
    <link rel="stylesheet" type="text/css" href="relative.css">
  </head>
  <body>
    <div id="pale-gray">
      <div id="lite-gray">
        <div id="dark-gray">
          <div id="text-box">
            Text content here on the innermost element
            - outer offset elements create a drop shadow
            effect
          </div>
        </div>
      </div>
    </div>
    <p>
      There is <span>a repositioned span element</span> on this
      line - it pays no heed to other paragraph content for it is an
      inline content box
    </p>
  </body>
</html>
```

- 2 Sla het document op als "**relative.html**".

Lay-out controleren

3 Start een nieuw CSS document. Tik de onderstaande code in:

```
div#pale-gray { width : 400px ; height : 50px ; background : #CCC ;
margin : 20px }
div#lite-gray { position : relative ; left : -4px ; top : -4px ; height :
100% ; background : #999 }
div#dark-gray { position : relative ; left : -4px ; top : -4px ; height :
100% ; background : #666 }
div#text-box { position : relative ; left : -4px ; top : -4px ; height :
100% ; background : lime }
span { position : relative ; top : -18px ; left : 100px ; background : lime
; border : 1px solid black }
p { background : aqua ; border : 1px solid black }
```

4 Sla het document op als "**relative.css**".

5 Open het document **relative.html** in een browser en bekijk het resultaat. Analyseer de code.

Text content here on the innermost element - outer offset elements create a drop shadow effect

Constate posities vastleggen

Een **fixed** (*vaste*) positioneringsschema, zoals het **absolute** positioneringsschema, verwijderd het inhoudvak van het geselecteerde element volledig uit de normale (*flow layout*) stroomindeling. Maar in tegenstelling tot **absolute** positionering, waar *offset* waarden betrekking hebben op de grenzen van het bevattende element, hebben de *offset* waarden in **fixed** positionering betrekking op het **viewport** (*kijkvenster*) - de positie is relatief ten opzichte van het browservenster en niet tot enig deel van het document.

Element posities worden kunnen worden vastgesteld om een frame-achtige interface te emuleren waarbij sommige frames op een constante positie blijven staan, ongeacht hoe de pagina wordt gescrold. Een bannerkop frame kan bijvoorbeeld aan de bovenkant van de pagina worden bevestigd, zodat het constant zichtbaar blijft, zelfs wanneer de pagina wordt gescrold.

Lay-out controleren

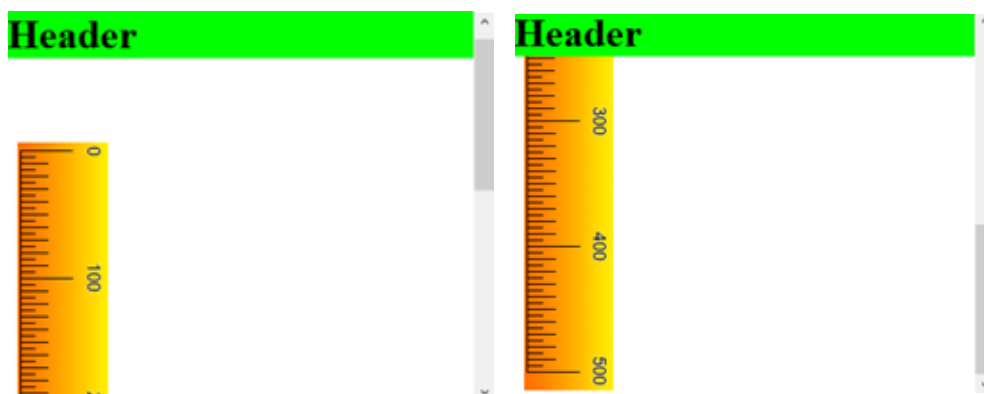
- 1 Start een teksteditor en maak een HTML5-document met een kopelement en een alinea met een lange afbeelding om te scrollen. Tik de code in:

```
<!DOCTYPE HTML >
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Absolute Positioning</title>
    <link rel="stylesheet" type="text/css" href="fixed.css">
  </head>
  <body>
    <h1>Header</h1>
    <p>
      
    </p>
  </body>
</html>
```

- 2 Sla het document op als "**fixed.html**".
- 3 Start een nieuw CSS document. Tik de onderstaande code in:

```
h1 { position : fixed ; top : 0 ; left : 0 ; margin : 0 ;background : lime ;
width : 100% ; z-index : 2 }
p { position : absolute ; top : 100px ; z-index : 1 }
```

- 4 Sla het document op als "**fixed.css**".
- 5 Download en plaats de afbeelding **ruler.png** op dezelfde plek als **fixed.html** en **fixed.css**.
- 6 Open het document **fixed.html** in een browser en bekijk het resultaat. Analyseer de code.



Drijvende inhoudsvakken

Met de CSS **float** eigenschap kan een inhoudsvak worden geplaatst aan de zijgrens van het bijbehorende element - met behulp van de **left** of **right** sleutelwoorden om de gewenste zijde op te geven. Op typerende wijze wordt deze functie gebruikt om tekst rond afbeeldingen te laten vloeien die naast een alinea-element zijn geplaatst.

Je kunt ook expliciet voorkomen dat tekst langs een vak met zwevende inhoud stroomt met behulp van de CSS eigenschap **clear** - door **left**, **right** of **bottom** sleutelwoorden op te geven om te bepalen welke zijde vrij moet zijn, zodat tekst begint onder het vak met zwevende inhoud.

- 1 Start een teksteditor en maak een HTML5-document met drie alinea's en twee afbeeldingen. Tik de code in:

```
<!DOCTYPE HTML >
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Floating Content Boxes</title>
    <link rel="stylesheet" type="text/css" href="float.css">
  </head>
  <body>
    <p>
      Massive acceleration - the forbidden fruit! It's easy to avoid
      such unlawful  activities in a normal vehicle.
      But there is an evil serpent; a Viper that tempts you to take a
      bite out of the asphalt. With a tasty 500-hp V10 powering a
      mere 3,300-lb roadster, the Dodge Viper SRT-10 tricks you
      into playing music with the loud pedal.
    </p>

    <p>
      This car is too excessive, too epic for most people to use on a
      daily basis. But for otherwise nice couples
      who need only two seats this is the car that will shame those
      who come up against them.
    </p>

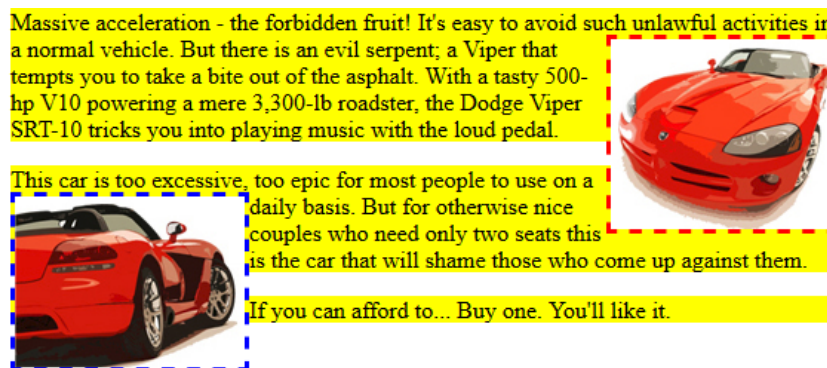
    <p class="clear">If you can afford to... Buy one. You'll like it.</p>
  </body>
</html>
```

Lay-out controleren

- 2 Sla het document op als "**float.html**".
- 3 Start een nieuw CSS document. Tik de onderstaande code in:

```
p { background : yellow }
img[src="viper-1.png"] { float : right ; border : 3px dashed red }
img[src="viper-2.png"] { float : left ; border : 3px dashed blue }
p.clear { clear : left }
```

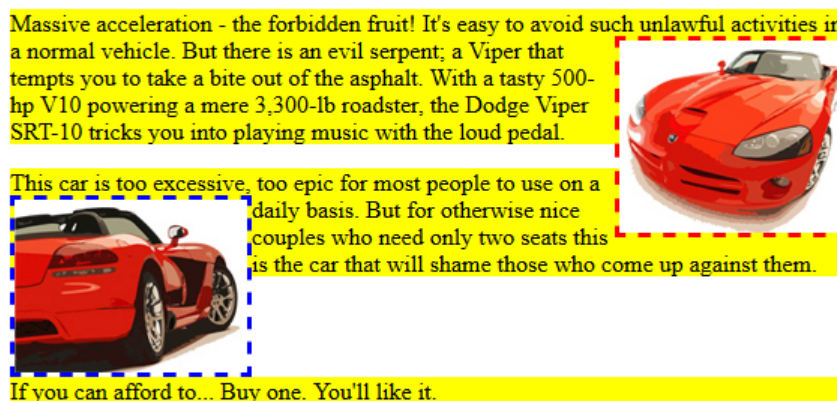
- 4 Sla het document op als "**float.css**".
- 5 Download en plaats de afbeeldingen **viper-1.png** en **viper-2.png** op dezelfde plek als **float.html** en **float.css**.
- 6 Open het document **float.html** in een browser en bekijk het resultaat. Analyseer de code.



- 7 Pas het stijlblad (float.css) aan om een regel toe te voegen die voorkomt dat de laatste alinea langs de tweede zwevende afbeelding stroomt.

```
p.clear { clear : left }
```

- 8 Sla het stijlblad nogmaals op en ververs (*refresh*) het browservenster om de laatste alinea nu onder de tweede zwevende afbeelding te zien.



Kolommen construeren

Traditioneel hebben webpagina-auteurs HTML-tabellen gebruikt om te bepalen hoe pagina-inhoud is ingedeeld door een grenzeloos cellenraster te maken waarin componenten van de pagina worden geplaatst. Deze methode werkt goed genoeg voor een algemene lay-out, maar mist de precieze controle die CSS biedt.

Met het **absolute** positionerings-schema kunnen componenten bijvoorbeeld eenvoudig worden gepositioneerd met uiterste nauwkeurigheid, maar dit kan niet eenvoudig worden bereikt met behulp van tabellen.

CSS biedt dus een superieure lay-out methode, zodat auteurs van webpagina's nu worden afgeraden HTML-tabellen voor dit doel te gebruiken.

De overgang naar pagina lay-out met CSS, roept vragen op over hoe de pagina-inhoud het beste kan worden beheerd.

In het volgende voorbeeld wordt een manier voorgesteld om een algemene pagina lay-out te maken met een kop- en voettekst plus drie inhoudskolommen.

- 1 Start een teksteditor en maak een HTML5-document met "div" - elementen voor kop- en inhoudspanelen, allemaal verpakt in een buitenste containerelement dat wordt gevolgd door een voettekst-element. Tik de onderstaande code in:

```
<!DOCTYPE HTML >
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Columns</title>
    <link rel="stylesheet" type="text/css" href="columns.css">
  </head>
  <body>
    <div id="wrapper">
      <div id="hdr">
        Header Panel<br>
      </div>
      <div id="nav">
        Navigation Panel<br>&bull;Link<br>&bull;Link
      </div>
    </div>
  </body>
</html>
```

Lay-out controleren

```

</div>
<div id="ads">
  Supplement Panel<br>
</div>
<div id="txt">
  Content Panel<br>
</div>
</div>
<div id="ftr">Footer Panel</div>
</body>
</html>

```

2 Sla het document op als **columns.html**.

3 Start een nieuw CSS document. Tik de onderstaande code in:

```

/* Validated by W3C CSS Validation Service */
* { margin: 0; padding : 0 ; text-align : center }
html, body { height: 100%; }
#wrapper
{
  min-height: 100%;
  height: auto !important;
  height: 100%;
  margin: 0 auto -30px; /* the bottom margin is the negative value
                        of the footer's height */
}
#hdr { height: 60px; background : aqua }
#ftr { height: 30px; background : aqua }
#nav { float : left ; width : 100px ; background : yellow }
#ads { float : right ; width : 100px ; background : fuchsia }
#txt { margin : 0 100px ; background : lime }

```

4 Sla het document op als **columns.css**.

5 Download en plaats de afbeelding **tile.png** op dezelfde plek als **columns.html** en **columns.css**.

Lay-out controleren

- 6 Open het document **columns.html** in een browser en bekijk het resultaat. Analyseer de code.

