

# Afbeeldingen

Web Frontend Basics

<b>6</b>	<b>WERKEN MET AFBEELDINGEN</b>	<b>3</b>
<b>6.1</b>	<b>Figuren voor het Web</b>	<b>3</b>
6.1.1	Formaat	3
6.1.2	Kleur	3
6.1.3	Afmetingen en resolutie	4
6.1.4	Snelheid	4
6.1.5	Transparantie en animatie	4
<b>6.2</b>	<b>Afbeeldingen vinden en/of bewerken</b>	<b>5</b>
<b>6.3</b>	<b>Het Image element</b>	<b>6</b>
6.3.1	Figuren uitlijnen	6
1.1.1.1	alignen	6
1.1.1.2	Floaten	7
<b>6.4</b>	<b>Figuren als Hyperlinks</b>	<b>8</b>
<b>6.5</b>	<b>Achtergrondfiguren</b>	<b>8</b>
<b>6.6</b>	<b>Een Favicon toevoegen</b>	<b>9</b>

### 6.1 FIGUREN VOOR HET WEB

Figuren maken voor het web verschilt van figuren maken die afgedrukt moeten worden. Hoewel de karakteristieken dezelfde blijven, moeten we met meer zaken rekening houden; de figuur moet in elke browser weergegeven kunnen worden en ze mag ook niet te groot zijn.

Er zijn zes factoren waarmee we rekening moeten houden: formaat, kleur, resolutie, snelheid, transparantie en animatie.

#### 6.1.1 FORMAAT

Wanneer je figuren maakt die bedoeld zijn voor het web is het belangrijk om te weten of de figuur eigenlijk wel weergegeven kan worden voor de bezoekers. Dagelijks bevinden er zich miljoenen Windows PC's, Macs, Linux, en andere computers op het web en elke systeem heeft zijn eigen grilletjes. De taak als webdesigner is *(alweer)* om de grootste gemene deler te zoeken zodat de figuren steeds op dezelfde manier weergegeven kunnen worden.

De meest ondersteunde formaten zijn:

- **GIF** (.gif / Graphics Interchange Format)
  - Gebruikt voor figuren die **weinig kleuren** hebben, bijvoorbeeld logo's of computer getekende figuren.
  - Transparantie mogelijk.
  - Animatie mogelijk.
- **JPEG** (.jpg of .jpeg / Joint Photographic Experts Group)
  - Gebruikt voor figuren die **veel kleuren** hebben, bijvoorbeeld foto's.
  - Transparantie niet mogelijk
  - Animaties niet mogelijk
- **PNG** (.png / Portable Network Graphics)
  - Patentvrij formaat ter vervanging van GIF.
  - Meestal betere compressie dan GIF met dezelfde kwaliteit (kleiner)
  - Zeer precieze transparantie mogelijk (24-bit PNGs)
  - Animaties niet mogelijk (wel met de zgn. "APNG" die weinig ondersteund is)

#### 6.1.2 KLEUR

Tegenwoordig zijn 8-bit monitors nagenoeg uitgestorven en zijn we redelijk zeker dat de monitor van onze gebruikers de meeste kleuren op dezelfde manier weergeeft. Het zogenaamde Browser-Safe kleurpallet is zo langzamerhand in onbruik geraakt *(hoewel er nog redenen zijn om deze te gebruiken)*.

Voor figuren kunnen we stellen dat we **GIF** of **PNG** gebruiken voor figuren met 256 kleuren of minder. Voor figuren met meer kleuren, zoals een foto, gebruiken we het **JPEG** formaat voor maximale compressie en kwaliteitsbehoud.

### 6.1.3 AFMETINGEN EN RESOLUTIE

Digitale figuren worden gemeten in **pixels**. Dit zegt in principe nog niets over hoe groot deze overkomt voor de lezer omdat dit afhankelijk is van de gebruikte resolutie. Wanneer we een foto van 1600 x 1200 (breedte x hoogte) afdrukken met een printer die een **resolutie** van 200ppi (pixels per inch) dan meten we 8 bij 6 inch ( $1600/200=8$  en  $1200/200=6$ ) of 20,32 cm bij 15,24 cm. Monitors hebben meestal een lagere resolutie, rond de 86ppi (ergens tussen 72 tot 100ppi) en zullen de figuur veel te groot weergeven (18 x 14 inch of 46 x 36cm) om ze op het scherm te laten passen.

Vroeger hadden de meeste monitors een resolutie van 640px bij 480px. Tegenwoordig is dit vaak Full HD (1920 x 1080) of zelfs meer. Het is dus belangrijk om er voor te zorgen dat de tekst en/of figuren niet verdwijnen aan de rechterkant van de pagina en rekening te houden met de resoluties die we tegenwoordig mogen verwachten van de gemiddelde monitor. Hou ook rekening met mobiele apparaten!

### 6.1.4 SNELHEID

Een krant zou weinig succes hebben indien lezers zouden moeten wachten op de figuren die erbij horen. In het geval van webpagina's is er steeds een wachttijd om alle data over te brengen, en van al deze data gebruiken de figuren meestal de meeste bandbreedte. Het is dus immens belangrijk om figuren zo klein mogelijk te houden met minimaal kwaliteitsverlies.

De gemakkelijkste manier om een figuur zo klein mogelijk te houden qua grootte is deze te herschalen. We zullen in elk geval onze foto van 1600 x 1200 moeten herschalen naar een werkbaar formaat, bijvoorbeeld 533 x 400, wat al veel minder bandbreedte zou verbruiken. De figuur kan tevens volledig weergegeven worden in een browserscherm zonder dat we moeten scrollen.

Via een verhoogde compressie kunnen we de bestandsgrootte van de figuur verkleinen met behoud van de afmetingen. Compressie is sterk afhankelijk van het formaat: GIF en PNG doet dit beter voor illustraties en figuren met weinig kleuren, JPEG compressie is specifiek ontwikkeld voor kleurrijke foto's. De meeste programma's voor figuurbewerking (GIMP, Photoshop, ...) hebben opties om de compressie in te stellen voor een optimale grootte/kwaliteitsverhouding.

### 6.1.5 TRANSPARANTIE EN ANIMATIE

Alle figuren die we weergeven zijn rechthoekig in afmeting. Met transparante figuren kunnen we de layout van onze webpagina echter complexer maken (ronde of grillige randen, de ene figuur boven een andere plaatsen, enz.). Onze pagina kan hierdoor veel aantrekkelijker worden. Enkel GIF en PNG ondersteunen transparantie.

Geanimeerde figuren worden [best niet in overmaat gebruikt](#). Het kan werkelijk irritant zijn voor de bezoeker wanneer bijna elke afbeelding beweegt of de aandacht probeert te trekken. Meer gebruikt zijn de "loading" animaties die we meer en meer tegenkomen op sites die javascript of AJAX gebruiken om inhoud in te laden. Figuuranimatie voor webpagina's is enkel mogelijk met het GIF formaat.

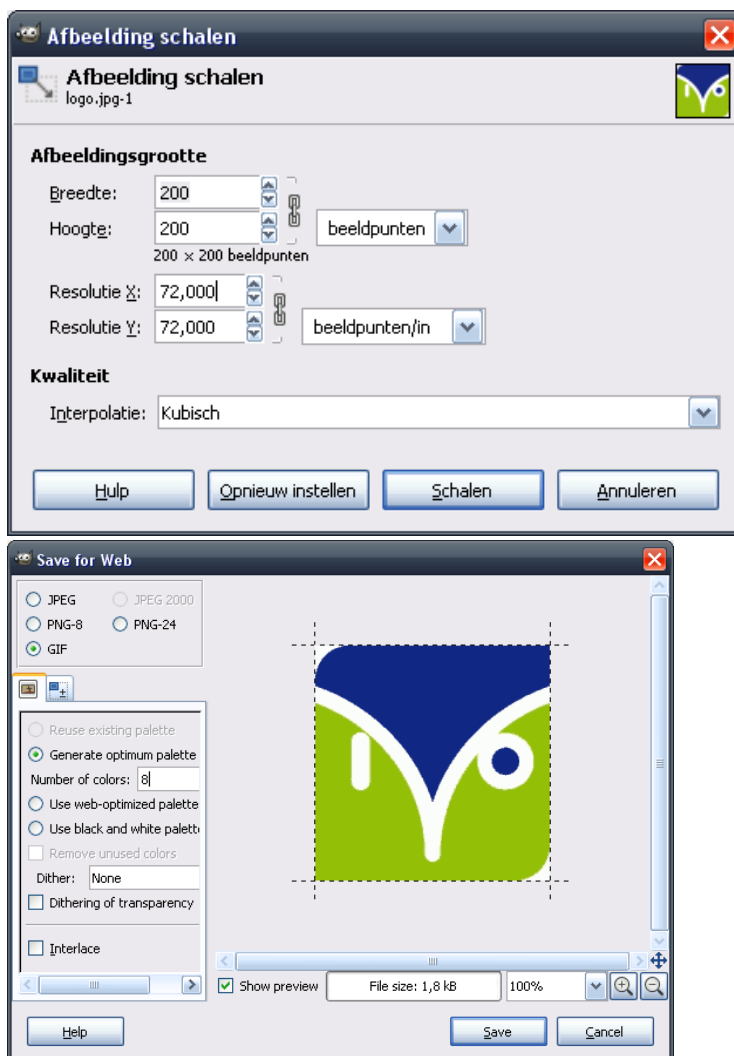
## 6.2 AFBEELDINGEN VINDEN EN/OF BEWERKEN

Creatievelingen kunnen zelf hun afbeeldingen en icoontjes die afkomstig zijn van een scanner, digitale camera, illustraties met software of pen, enz... maken en bewerken. Wanneer een nieuwe site ontworpen wordt, zal deze meestal getekend worden met een programma (*Photoshop, GIMP, Illustrator,...*) en vervolgens in verschillende afbeeldingen gesneden voor gebruik in de HTML code.

We kunnen ook veel figuren vinden op het web:

- Google (*opgelet voor copyrights voor die figuren!*)
- Gratis figuren downloaden van bepaalde sites (*zgn. "royalty free"*)
- Kwaliteit "stock" illustraties/foto's kopen en downloaden (*vanaf \$1 per foto*)

Meestal zullen we deze figuren verder willen bewerken vooraleer we ze gebruiken, of tenminste de grootte aanpassen zodat ze snel gedownload worden naar de browser. Er bestaan talloze programma's om figuren te bewerken en op te slaan in een zuinig formaat voor webpagina's. GIMP is een gratis alternatief voor Photoshop met gelijkaardige functionaliteiten.



GIMP Opties voor herschalen van het IVO logo en een "Save for Web" plugin met preview.

Wanneer je zelf figuren bewerkt, is het belangrijk om vertrouwd te raken met je gekozen programma. De meeste bevatten basisopties om de compressie in te stellen en om de figuur te herschalen.

## 6.3 HET IMAGE ELEMENT

We zijn het **img** element reeds tegengekomen in de loop van de cursus. Dit hoofdstuk besteedt hieraan bijzondere aandacht en beschrijft alle mogelijkheden ervan.

```

```

*Voorbeeld van een img element met passend alt attribuut*

Het **src** attribuut bevat de locatie van de figuur (*het pad*). Wanneer de figuur in een map op onze eigen website staat, gebruiken we best een **relatieve** verwijzing, bij een externe figuur kunnen we enkel **absolute** verwijzingen gebruiken (*zie hoofdstuk over relatieve en absolute verwijzing*).

Het gebruik van het **alt** attribuut is **verplicht** in HTML waarin we een beschrijvend stukje tekst plaatsen over de figuur. Dit is noodzakelijk indien de figuur niet beschikbaar is in de browser of voor personen (*bvb slechtzienden*) die speciale tekst naar audio apparatuur gebruiken om met de PC te werken.

Een img element is steeds **zelfsluitend**.

### 6.3.1 FIGUREN UITLIJNEN

Image elementen zijn **inline** elementen. Alle tekst die er voor of na geplaatst wordt zal op dezelfde lijn weergegeven worden, maar we kunnen dit manipuleren met behulp van CSS.

#### 1.1.1.1 ALIGNEN

```
<p>
  Een afbeelding is steeds


  <strong>inline</strong> met onze tekst
</p>
<p>
  De tekst


  is top-gealigneerd
</p>
<p>
  De tekst


  is bottom-gealigneerd
</p>
<p>
  Onze tekst is middle-gealigneerd


</p>
```

*HTML syntax voorbeeld voor uitlijnen*

```

/*Element aan de bovenzijde alignen*/
.topAlign {
    vertical-align:top;
}

/*Element aan de onderzijde alignen*/
.bottomAlign {
    vertical-align:bottom;
}

/*Element aan in het midden alignen*/
.middleAlign{
    vertical-align:middle;
}

```

CSS syntax voorbeeld voor uitlijnen

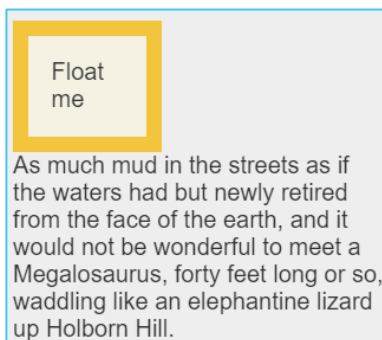
### 1.1.1.2 FLOATEN

Een andere manier om een afbeelding te positioneren naast je tekst zelf is door gebruik te maken van het floaten.

Float is een CSS property die je op eender welk element kan toepassen. Het vaakst komt deze property voor in toepassingen met afbeeldingen. Float heeft als mogelijke waarden **left**, **right** en **none** (de *default* waarde). Door gebruik te maken van deze property 'drijft' het element als het ware aan de zijde die je specificeert.

```
float: none;
```

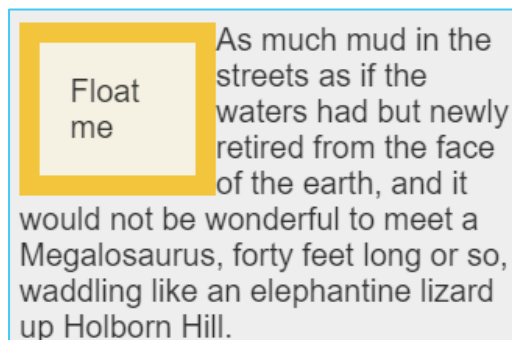
CSS syntax voor float none



MDN Voorbeeld van de float property met value none

```
float: left;
```

CSS syntax voor float left

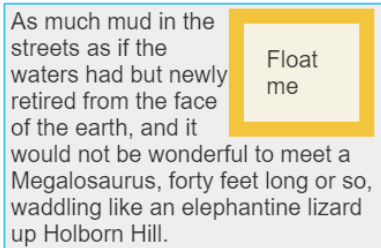


MDN Voorbeeld van de float property met value left

```
float: right;
```

We kunnen bepalen waar de tekst stopt met langs de zijkant van de figuur te lopen (*wrappen*) door gebruik te maken van de css property **clear** op eenzelfde manier zoals we **float** aanroepen. Clear heeft als mogelijke waarden **none**, **left**, **right** en **both**.

CSS syntax voor float right



MDN Voorbeeld van de float property met value right

## 6.4 FIGUREN ALS HYPERLINKS

In grafische omgevingen zijn we het gewoon om op figuren te klikken om bepaalde acties uit te voeren. In HTML kunnen we een figuur zonder probleem ook interactief maken door deze bijvoorbeeld in een hyperlink (*a element*) te plaatsen. Op deze manier creëren we een navigatie “knop” zodat de gebruiker naar de betreffende URL of pagina wordt doorverwezen (zie hoofdstuk *Hyperlinks*).

```
<p>U bekijkt de Da vinci pagina <em>2</em> van 3</p>
<a href="pagina1van3.html" title="Naar vorige pagina (1/3)">

</a>
<a href="pagina3van3.html" title="Naar volgende pagina (3/3)">

</a>
```

Voorbeeld van het omsluiten van een *img* element met een *a* element om een hyperlink te maken

De figuur gedraagt zich nu als een hyperlink. Via het **title** attribuut van de hyperlink kunnen we een tooltip instellen om verwarring te vermijden. Vergeet echter het **alt** attribuut van je figuur niet!

Elke figuur die zich in een **a** element bevindt, wordt standaard voorzien van een border. We kunnen de pixelbreedte van deze rand instellen met behulp van het CSS indien de border ongewenst is.

```
border:none;
```

CSS syntax voor het verwijderen van de default border

## 6.5 ACHTERGRONDFIGUREN

In de plaats van een element enkel een achtergrondkleur te geven, kunnen we ook figuren instellen als achtergrond. Dit stelt ons in staat om mooie ontwerpen te maken voor webpagina's zonder het gebruik van een **img** element (*voorgond-figuur!*).

Deze achtergrondfiguren kunnen herhaald worden langs de x of y as, uitgelijnd volgens verschillende instellingen. Een achtergrondfiguur is als een laag boven de originele achtergrondkleur (*indien deze is ingesteld*).



CSS Property	Beschrijving
<b>background-attachment</b>	Bepaalt of de achtergrondfiguur mee volgt wanneer er gescrold wordt ( <i>we kunnen soms scrollen wanneer de inhoud van een blokelement te lang is, bij de <b>body</b> is dit automatisch zo</i> )
<b>background-image</b>	Specificeert de url van de weer te geven figuur. We doen dit met behulp van de css notatie: een online figuur. Bvb.: <code>url(<a href="http://www.mijn.figuur.com/figuur.jpg">http://www.mijn.figuur.com/figuur.jpg</a>);</code> een figuur die lokaal opgeslagen is in een map images bvb: <code>url(img/mijnfiguur.jpg);</code>
<b>background-position</b>	Positioneert de figuur links, rechts, midden, boven, onder... We kunnen ook X Y Coördinaten opgeven in pixels of %.
<b>background-repeat</b>	Specificeert of de figuur herhaald zal worden langs de X of Y as, beide assen of geen enkele.

## 6.6 EEN FAVICON TOEVOEGEN

Op vele sites vinden we icoontjes in de adresbalk, en wanneer we een site toevoegen aan de favorieten krijgt de snelkoppeling hetzelfde icoontje. Deze icoontjes zijn steeds 16 x 16 figuren, meestal van het type **.ico** (sommige browsers ondersteunen ook **.gif**). In de nieuwste browsers vinden we ook vaak **tabbladen**. Het is belangrijk om het icoon een transparante achtergrond te geven voor maximale compatibiliteit met de vensterkleur van de browser.

Om een favoriet icoon toe te voegen moeten we een **link** element toevoegen in de **head** sectie van onze HTML pagina.

```
<head>
...
<link rel="icon" href="img/html5.ico" type="image/x-icon" />
<link rel="shortcut icon" href="img/html5.ico" type="image/x-icon" />
<link rel="icon" type="image/png" href="img/html5_16.png" sizes="16x16">
<link rel="icon" type="image/png" href="img/html5_32.png" sizes="32x32">
<link rel="icon" type="image/png" href="img/html5_96.png" sizes="96x96">
...
</head>
```

*Voorbeelden van de mogelijkheden om een favicon toe te voegen*

We gebruiken het **rel** attribuut om mee te delen dat het hier over het website icoon gaat. Gebruik beide bovenstaande notaties om compatibiliteit tussen browsers te garanderen. Het **href** attribuut is het pad naar het eigenlijke icoon. **Type** is in dit geval van het type **image/x-icon**. Het is belangrijk het juiste formaat mee te delen. Indien je een gif gebruik als icoon dan is dit: **image/gif**

