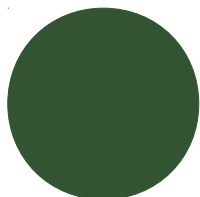




Mijn-
eigen-
website
.nl

Wat je moet weten over...

Foto's op je website

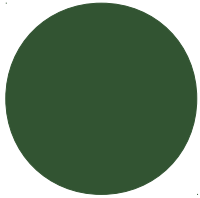


Inhoudsopgave

Voorwoord

1. Over de oorsprong van foto's
Hoe kom ik aan foto's of afbeeldingen?
2. Over jpg's, gif's, png's en svg's
Welk bestandsformaat moeten mijn foto's hebben?
3. Over resolutie en bestandsgrootte
Wat moet de resolutie en bestandsgrootte van mijn foto's zijn?
4. Over bestandsnamen
Welke namen moet ik mijn foto's geven?
5. Over de html-afbeeldingscode
Hoe neem ik een foto op in mijn pagina?
6. Over de positionering van je foto's
Hoe positioneer ik een foto?
7. Over bijschriften
Hoe zet ik een bijschrift bij mijn foto?
8. Over achtergrondafbeeldingen
Hoe geef ik mijn webpagina een achtergrondfoto?

Tot slot

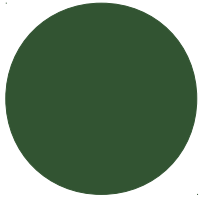


Voorwoord

Een foto (of andere afbeelding) op je website zetten is niet zo moeilijk, maar er zijn wel een paar dingen waar je rekening mee moet houden. In dit e-book lees je:

1. hoe je aan foto's voor je website komt
2. welk bestandsformaat deze foto's moeten hebben
3. wat de resolutie en bestandsgrootte van je foto's moet zijn
4. waar je op moet letten als je je fotobestanden een naam geeft
5. hoe je ervoor zorgt dat een foto op je webpagina zichtbaar wordt
6. hoe je de positie van een foto op je webpagina bepaalt
7. hoe je een bijschrift bij je foto('s) zet
8. hoe je een foto als achtergrond gebruikt

In de meeste hoofdstukken staan voorbeelden en voorbeeldcodes, zodat je direct aan de slag kunt.



1. Over de oorsprong van foto's

Er zijn meerdere manieren waarop je aan foto's voor je website kunt komen.

a. foto's maken met je eigen camera of smartphone

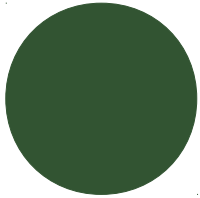
Foto's die je op je website wilt laten weergeven hoeven niet zo goed van kwaliteit te zijn als foto's die je wilt laten afdrucken. Daardoor kun je ook foto's die met een gewone camera of smartphone zijn gemaakt best gebruiken voor je website.

b. foto's downloaden van het web

Er zijn veel plekken op internet waar je (al dan niet gratis) foto's vandaan kunt halen. Wil je ze gebruiken voor een commerciële website, let er dan op dat je daarvoor wel toestemming hebt. In het artikel [Waar kan ik gratis afbeeldingen downloaden?](#) lees je waar je geschikte foto's voor je website vandaan haalt.

c. foto's bewerken met speciale software

Om bestaande foto's te bewerken of om fotocollages te kunnen maken heb je speciale software nodig. Dat hoeft niets te kosten: er zijn genoeg [gratis programma's voor fotobewerking](#). Het kost misschien even tijd om te leren omgaan met zo'n programma, maar het is een vaardigheid die je als website-beheerder goed van pas zal komen.



2. Over jpg's, gif's, png's en svg's

Om een foto of andere afbeelding te laten weergeven op je webpagina, moet deze een van de volgende extensies hebben: `jpg`, `gif`, `png` of `svg`. Welk formaat je het beste kunt kiezen, hangt helemaal af van de soort afbeelding die je wilt laten zien. Hier een paar vuistregels:

Afbeeldingen met verlopende kleuren

Met 'verlopende kleuren' bedoel ik dat de verschillende kleuren niet scherp van elkaar gescheiden zijn, maar dat ze langzaam in elkaar overlopen. Bij foto's is dat altijd het geval. Daarom moet een foto altijd in een 24-bits formaat worden opgeslagen. Want in 24-bits afbeeldingen kunnen er meer dan 16 miljoen kleuren worden gebruikt voor die afbeelding.



Ter vergelijking: een afbeelding die in een 8-bits formaat is opgeslagen wordt in niet meer dan 256 kleuren weergegeven, en dat geeft natuurlijk een heel ander resultaat!

Er zijn twee bestandsformaten geschikt voor het opslaan van afbeeldingen met verlopende kleuren: **jpg** en **24-bits png**. Een van de verschillen tussen die twee is dat je afbeeldingen in `jpg`-formaat beter kunnen worden gecomprimeerd, waardoor het uiteindelijke bestand kleiner zal zijn. En: hoe kleiner het bestand, hoe sneller de browsers de afbeelding kunnen laden, dus `jpg` is het beste formaat om een foto of afbeelding met verlopende kleuren op te slaan.

Afbeeldingen in vlakke kleuren

Een afbeelding in vlakke kleuren – dus met scherpe afscheidingen tussen de verschillende kleuren, zoals in de afbeelding hiernaast – kun je het beste opslaan in **gif**- of in **8-bits png**-formaat. Je krijgt zo een relatief klein bestand van goede kwaliteit.



Transparante afbeeldingen

Soms wil je dat de achtergrond van je webpagina achter de afbeelding zichtbaar blijft. Een gedeelte van de afbeelding die je op die achtergrond wilt neerzetten, moet dus transparant zijn. Dat is alleen mogelijk bij een afbeelding die is opgeslagen in **gif**- of **png**-formaat.

Als voorbeeld heb ik de afbeelding van het hart hier nog eens neergezet, maar heb ik tegelijkertijd de achtergrond van deze alinea lichtgrijs gemaakt. De afbeelding in zijn geheel is rechthoekig (150 pixels breed bij 127 pixels hoog), maar omdat het gedeelte zonder kleuren transparant is, schijnt de achtergrond van deze paragraaf erdoorheen en lijkt het plaatje er bovenop te liggen.



Plaatjes met animaties

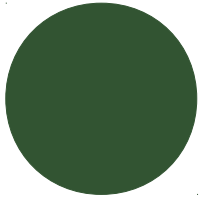
Is je afbeelding een ‘bewegend’ plaatje, bestaande uit twee of meer afbeeldingen die elkaar in snel tempo afwisselen, dan kun je dat effect alleen behouden als je het opslaat als **(animated) gif**.

Tekeningen, logo's, grafieken e.d.

Foto's en afbeeldingen die zijn opgeslagen in jpg-, gif- of png-formaat zijn 'gerasterd', dat wil zeggen dat ze zijn omgezet in een verzameling puntjes. Als je sterk op zo'n afbeelding inzoomt, zie je die puntjes vanzelf verschijnen: de afbeelding wordt een stuk onscherper. Voor tekeningen, logo's en grafieken kun je daarom beter het **svg**-formaat gebruiken.



De afkorting 'svg' staat voor *scalable vector graphics*. Het betekent dat de afbeelding zo groot kan worden weergegeven als je maar wilt terwijl ze toch scherp blijft. De afbeelding is namelijk niet gerasterd, maar de instructies voor het samenstellen ervan zijn vastgelegd in een gewoon tekstbestand. Svg-afbeeldingen kun je zelf maken met een tekenprogramma als Adobe Illustrator of (het gratis programma) Inkscape. De afbeelding bij deze alinea is afkomstig van openclipart.org, waar je clipart kunt downloaden in png- of svg-formaat. Ook pixabay.com is een goede bron voor deze zogenaamde 'vectorafbeeldingen' (kies bij 'image-type' voor: Vektorgrafik).



3. Over resolutie en bestandsgrootte

Zoals in het vorige hoofdstuk te lezen is, is een afbeelding die is opgeslagen in het jpg-, gif- of png-formaat 'gerasterd'. Deze is omgezet in een verzameling puntjes. Je krijgt in dat geval te maken met het begrip **resolutie**: daarmee wordt de dichtheid van die puntjes aangegeven.

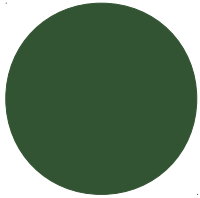
Deze resolutie wordt uitgedrukt in het aantal puntjes per inch (ppi). Nu moet je een 'inch' (ca. 2,5 cm) in dit geval niet al te letterlijk nemen: een afbeelding van 72 pixels (dus een inch) breed, kan op het ene beeldscherm best wat breder of smaller uitvallen dan op het andere. Maar de verhoudingen blijven natuurlijk wel intact: een afbeelding van 100 pixels breed zal twee keer zo breed worden weergegeven als een afbeelding van 50 pixels breed.

Lange tijd kwam 1 afbeeldingspixel overeen met 1 beeldschermpixel. Maar tegenwoordig gaat die regel niet meer op. De resolutie van beeldschermen is flink toegenomen en als 1 afbeeldingspixel nog steeds 'vertaald' zou worden naar 1 beeldschermpixel, dan bleef er van je afbeeldingen weinig meer over. Hoe je afbeeldingspixels zich dan wel verhouden tot de beeldschermpixels is per beeldscherm verschillend. Het is dan ook onmogelijk om te voorspellen wat de exacte afmetingen van de afbeeldingen op je webpagina zullen zijn.

Toch is een resolutie van 72 pixels per inch nog steeds een goed uitgangspunt als je je foto's geschikt wilt maken voor een website. Voor sommige foto's, waarvan het belangrijk is dat ze altijd scherp worden weergegeven, kun je een hogere resolutie instellen (bijvoorbeeld 144 ppi). Maar dat moet je zeker niet voor al je foto's doen, want je afbeeldingsbestanden worden er wel een stuk groter van (waardoor een browser meer tijd nodig heeft om ze in te laden).

Wat deze **bestandsgrootte** betreft, probeer die onder de 150kB per afbeelding te houden. Als dat een enkele keer niet lukt is dat natuurlijk geen probleem, maar neem geen foto's in je webpagina's op die vele malen groter zijn. Dan is het risico veel te groot dat je bezoeker je pagina wegklikt omdat het veel te lang duurt voordat deze geladen is.

Om je foto's of afbeeldingen te bewerken (dus bij te snijden of te roteren bijvoorbeeld), heb je een foto-bewerkingsprogramma nodig. Zie weer deze [gratis software voor fotobewerking](#) en [Inkscape](#) voor vector-afbeeldingen.



4. Over bestandsnamen

Een foto of afbeelding kun je elke naam geven die je maar wilt, maar je moet daarbij wel op een aantal zaken letten:

- ◆ **Zet geen spaties in je bestandsnaam**

Spaties worden door browsers vaak omgezet in de tekencombinatie %20, en als je afbeeldingsnaam een spatie bevat en je verwijst er later in je afbeeldingscode naar, dan klopt die verwijzing niet meer omdat er in de url nu '%20' staat in plaats van een spatie.

- ◆ **Gebruik alleen gewone, gangbare letters of cijfers**

Vermijd in je bestandsnamen tekens als een apostrof, vraag- of uitroepteken, aanhalingsteken, komma, punt of puntkomma. Houd het bij de letters a tot en met z en de cijfers 0 tot en met 9 en er kan eigenlijk niet veel meer misgaan.

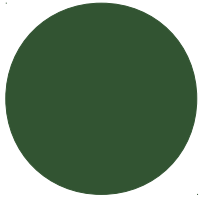
- ◆ **Gebruik altijd kleine letters**

Er is op zich niets mis mee om hoofdletters te gebruiken, alleen maak je het jezelf gemakkelijker als je jezelf aanwent alleen kleine letters te gebruiken. Wil je in je html-code naar je afbeeldingen verwijzen, dan hoef je niet steeds op te zoeken hoe je de naam van die afbeelding ook alweer had gespeld.

- ◆ **Schrijf afbeeldingsextensies altijd met een kleine letter**

(om dezelfde reden als bij het vorige punt). Zo hoef je niet elke keer als je naar een afbeelding wilt verwijzen op te zoeken of deze nu `foto.jpg` heet of `foto.JPG`. Een foute verwijzing kan tot gevolg hebben dat je foto niet zichtbaar is.

[Kun je in Windows Verkenner niet zien wat de extensie van je afbeeldingsbestanden is, ga dan naar Start – Configuratiescherm – Mapopties – tabblad Weergave – en haal het vinkje weg voor *Extensies voor bekende bestandstypen verbergen*. In Mac Finder ga je naar Voorkeuren – Geavanceerd – en zet je een vinkje voor: Toon alle bestandsnaamextensies.]



5. Over de html-afbeeldingscode

De html-code voor een foto of afbeelding is deze:

```
<img>
```

Wat daaraan direct opvalt, is dat het een leeg element is: er is geen afzonderlijke begin- en eind-tag waartussen de inhoud komt te staan.

src

Het `image`-element mag dan leeg zijn, het is natuurlijk wel de bedoeling dat er op de plaats waar deze `tag` staat, een foto of afbeelding wordt ingeladen. Daarvoor moet er aan de `image-tag` iets worden toegevoegd, namelijk de plaats waar die foto of afbeelding te vinden is. Dat gebeurt op deze manier:

```

```

Het is verplicht om een `source` (`src`) in het `image`-element te zetten, anders zal er niets worden weergegeven. Wat er tussen de dubbele aanhalingstekens achter `src` komt te staan, is de plaats waar de foto of afbeelding te vinden is. Dat gebeurt op dezelfde manier als bij een link. Je kunt deze locatie helemaal uitschrijven, zoals in het voorbeeld hierboven is gedaan. Dat is een **absolute verwijzing**.

Maar stel dat ik vanaf mijn homepage (<http://www.mijn-eigen-website.nl/index.php>) wil verwijzen naar de plaats waar deze afbeelding staat, dan hoef ik de naam van mijn website niet meer te noemen. Ik kan dan

volstaan met een **relatieve verwijzing**:

```

```

(De afbeelding uit dit voorbeeld staat in een aparte map, dus ik moet wel de naam van die map vermelden natuurlijk, anders wordt de afbeelding niet gevonden.)

alt

Er is nog iets wat verplicht aan een `image`-element moet worden toegevoegd, en dat is een `alt`-tekst. De afkorting 'alt' staat voor *alternate* (oftewel: alternatieve) tekst en het is een korte beschrijving van de afbeelding of foto. Als de afbeelding om de een of andere reden niet vertoond kan worden (bijvoorbeeld omdat hij niet meer op de server staat, omdat de verwijzing naar de locatie niet goed is, of omdat de surfer zijn browser zo heeft ingesteld dat plaatjes niet worden vertoond), dan wordt deze `alt`-tekst weergegeven. De afbeeldingscode van hierboven komt er met `alt`-tekst zo uit te zien:



```

```

Er hoeft niet per se iets tussen de dubbele aanhalingstekens achter `alt` te staan. Als de afbeelding die je wilt laten zien bijvoorbeeld alleen een plaatje is van een pijltje waarmee je kunt terugklicken naar een voorgaande

pagina, dan hoef je geen beschrijvende tekst op te nemen in je afbeeldingscode. Maar je moet de `alt`-toevoeging wel laten staan:

```

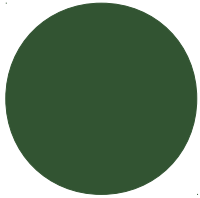
```



title

Je kunt nog veel meer informatie aan het `image`-element toevoegen, maar alleen de `src` en de `alt` zijn verplicht. Een voorbeeld van iets wat je nog kunt toevoegen, is een `title`. Tussen de dubbele aanhalingstekens daarachter zet je ook een beschrijving van het plaatje, net zoals bij de `alt`-toevoeging. De `title` fungeert dan als een soort 'tooltip': als een surfer met zijn muis over het plaatje heen beweegt, dan komt de inhoud van de `title` in beeld.

NB: Op een smartphone of tablet werkt dit niet, omdat je dan geen muis hebt. Wil je daar toch een tooltip laten zien, dan moet je gebruikmaken van [een andere methode om zo'n tooltip op te roepen](#).



6. Over de positionering van je foto's

Om je foto's te kunnen positioneren op je pagina, moet je stijlfenomenen toevoegen aan je afbeeldingscode. Dat doe je door het `style`-attribuut in de afbeeldingscode te zetten. Tussen de dubbele aanhalingstekens kun je allerlei instructies opnemen over de afmetingen van je afbeelding en de positie ervan op je pagina.

```
style=""
```

breedte en hoogte

```
width: 300px; max-width: 30%;
```

Het eerste wat vaak tussen de dubbele aanhalingstekens achter `style` wordt neergezet, zijn waarden voor de breedte van de foto (de hoogte wordt automatisch berekend aan de hand van de breedte die je opgeeft). Geef je geen breedte op, dan wordt je foto vertoond op de breedte (in pixels) van het originele fotobestand.

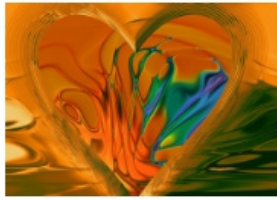
Maar stel dat je originele foto (zoals je hem hebt geüpload naar je server) 600 pixels breed is en 400 pixels hoog, dan kun je deze foto ook met een breedte van 300 pixels laten vertonen (en hoogte 200 pixels) door een waarde in te stellen voor de `width`. De foto zal er wat scherper door worden.

Het is verstandig om ook altijd een `max-width` voor je foto in te stellen, omdat je natuurlijk nooit weet of er wel genoeg ruimte is om je foto weer te geven. Je weet namelijk niet of de pagina met je foto erin wordt bekeken op een desktop, laptop, tablet of smartphone.

Achter `max-width` kun je waarden in verschillende eenheden neerzetten, bijvoorbeeld in percentages, in vw (*viewport width* of beeldscherm breedte), en ook weer in pixels. Met `max-width: 30%` bepaal je dat je foto nooit meer ruimte in beslag neemt dan 30% van het vak waarbinnen deze zich bevindt (bijvoorbeeld binnen een balk aan een van de zijkanten van je site). Met `max-width: 30vw` zorg je ervoor dat je foto nooit groter wordt dan 30% van de breedte van de *viewport* (dus het hele beeldscherm), en met `max-width: 100px` wordt je foto nooit groter weergegeven dan 100 pixels breed (die waarde kan ook een smartphone nog wel aan).

links, rechts of in het midden

Nadat je de breedte van de foto hebt vastgelegd, bepaal je hoe hij vertoond moet worden in vergelijking met alles wat eromheen staat. Doe je niks, dan komt de foto aan de linkerkant te staan. Staat er direct achter je afbeeldingscode een paragraaf (een stuk tekst tussen paragraaf-*tags*), dan komt die tekst er niet naast te staan, maar eronder. Dat ziet er dan zo uit:

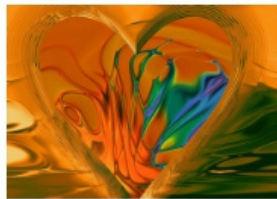


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur bibendum orci sit amet odio varius pulvinar. Quisque malesuada iaculis nibh, a vulputate orci rutrum sed. In vitae neque nibh. Maecenas semper dolor nec orci vestibulum cursus. Phasellus vel eros urna, at laoreet lorem. Nam diam libero, vulputate non aliquet et, tempus consectetur magna. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Morbi consequat ornare arcu at convallis. Quisque porta ultrices ligula et vulputate. Nulla et lacus elit, id dapibus ante. Ut vulputate aliquam consequat. In nulla quam, consectetur in consequat quis, facilisis non nisi.

Wil je dat je foto aan de linkerkant moet komen, en de tekst er rechts naast, dan voeg je aan de stijkenmerken van de afbeelding toe:

```
float: left;
```

Het resultaat is dan:



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur bibendum orci sit amet odio varius pulvinar. Quisque malesuada iaculis nibh, a vulputate orci rutrum sed. In vitae neque nibh. Maecenas semper dolor nec orci vestibulum cursus. Phasellus vel eros urna, at laoreet lorem. Nam diam libero, vulputate non aliquet et, tempus consectetur magna. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Morbi consequat ornare arcu at convallis. Quisque porta ultrices ligula et vulputate. Nulla et lacus elit, id dapibus ante. Ut vulputate aliquam consequat. In nulla quam, consectetur in consequat quis, facilisis non nisi.

Wil je dat je foto rechts van de tekst komt te staan, dan gebruik je natuurlijk:

```
float: right;
```

En wil je dat hij in het midden komt te staan, zonder tekst ernaast, dan zet je de `img` binnen een *division* (`div`) neer, waarvan je de tekst laat centreren:

```
<div style="text-align: center;">
  
</div>
```

Dat laatste lijkt misschien een beetje raar, maar dat is het niet. Het `image`-element is namelijk een zogenaamd *inline* element, wat betekent dat het op dezelfde manier behandeld wordt als een letter in een regel. Een letter of een stuk tekst kun je centreren met de opdracht `text-align: center;` en afbeeldingen op deze manier dus ook.

marges

Waarschijnlijk wil je niet dat je afbeelding strak tegen de paragraaftitels, de tekst (of wat er dan ook maar naast staat) komt te staan, maar wil je wat afstand houden. Dat doe je met:

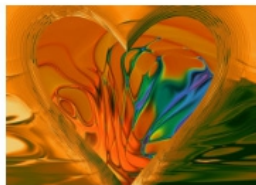
```
margin: 0 0 0 0;
```

De eerste waarde achter `margin` bepaalt de ruimte aan de bovenkant, de tweede aan de rechterkant, de derde aan de onderkant, en de vierde aan de linkerkant. Zijn de waardes voor boven/onder en links/rechts dezelfde, dan hoef je maar twee waardes op te geven. (En staat er maar één waarde, dan heeft die betrekking op alle vier de zijden.)

Staat er dus rechts van je afbeelding een stuk tekst, dan wil je waarschijnlijk wat ruimte hebben aan de bovenkant, rechterkant en onderkant van deze afbeelding. De linkerkant staat tegen de kantlijn aan te staan, dus die blijft op 0:

```
margin: 6px 6px 6px 0;
```

Nog even uitgaande van hetzelfde voorbeeld als hierboven, dan wordt dit het resultaat:



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur bibendum orci sit amet odio varius pulvinar. Quisque malesuada iaculis nibh, a vulputate orci rutrum sed. In vitae neque nibh. Maecenas semper dolor nec orci vestibulum cursus. Phasellus vel eros urna, at laoreet lorem. Nam diam libero, vulputate non aliquet et, tempus consectetur magna. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Morbi consequat ornare arcu at convallis. Quisque porta ultrices ligula et vulputate. Nulla et lacus elit, id dapibus ante. Ut vulputate aliquam consequat. In nulla quam, consectetur in consequat quis, facilisis non nisi.

De waardes voor `margin` kun je op meerdere manieren aangeven. Ik heb in dit voorbeeld de marges in pixels aangegeven, maar er zijn ook andere mogelijkheden. Meer informatie over de maatvoering op webpagina's lees je in het artikel [Welke maateenheden gebruik je in responsive design?](#)

borders

Een randje rond je afbeeldingen zetten, doe je met:

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

Eerst wordt de dikte van de lijn gegeven (1px), dan het type lijn (dit kan bijvoorbeeld zijn: `solid` (ononderbroken), `dashed` (streepjes), `dotted` (stippeltjes), `double` (dubbele lijnen)), en daarna de kleur (`dimgray`).

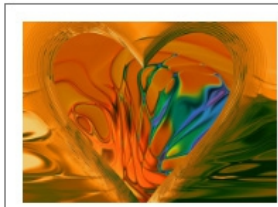
padding

Het lijntje van de rand wordt standaard strak rond je foto's gezet. Wil je wat ruimte tussen je foto en de rand eromheen, dan gebruik je daarvoor de eigenschap `padding`:

```
padding: 12px;
```

Met deze instelling wordt er rondom 12px ruimte rond je foto vrij gehouden. Je kunt ook verschillende waarden voor de ruimte boven, rechts, onder en links van je foto instellen, dan komen er gewoon vier waarden achter `padding` te staan (op dezelfde manier als bij de `margin`).

Het eindresultaat zie je op de volgende pagina: een afbeelding die links van de omringende tekst is geplaatst en daarbij 6 pixels afstand houdt van zijn directe omgeving, en die verder voorzien is van een lichtgrijze, ononderbroken rand, die er niet strak omheen staat, maar op een afstand van 12 pixels.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur bibendum orci sit amet odio varius pulvinar. Quisque malesuada iaculis nibh, a vulputate orci rutrum sed. In vitae neque nibh. Maecenas semper dolor nec orci vestibulum cursus. Phasellus vel eros urna, at laoreet lorem. Nam diam libero, vulputate non aliquet et, tempus consectetur magna. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Morbi consequat ornare arcu at convallis. Quisque porta ultrices ligula et vulputate. Nulla et lacus elit, id dapibus ante. Ut

vulputate aliquam consequat. In nulla quam, consectetur in consequat quis, facilisis non nisi.

De complete afbeeldingscode hiervoor is:

```

```

Meerdere afbeeldingen naast en onder elkaar

Wil je meerdere afbeeldingen naast en onder elkaar zetten (zonder tekst ernaast), dan hoef je tussen de `style`-kenmerken alleen instructies neer te zetten voor de `margin` (en eventueel voor de `border` en `padding` als je dat wilt).

Want een `image`-element gedraagt zich standaard net zoals een letter in een regel, dus als je er meerdere naast elkaar zet, komen ze gewoon netjes naast elkaar te staan. Is de 'regel' vol, dan komt de afbeelding waarvoor geen ruimte meer is, vanzelf op de volgende 'regel' terecht.



Hierboven heb ik vier afbeeldingen naast elkaar neergezet. De code daarvoor is heel eenvoudig:

```
<div class="fotogroep">
  
  
  
  
</div>
```

Zoals je ziet, staan er geen `style`-kenmerken meer in de codes voor de foto's. Omdat ik alle foto's dezelfde opmaak wil geven, is het gemakkelijker als ik de vier foto's groepeer door ze binnen een *division* (`div`) te zetten, die ik een `class`-naam geef. In mijn *stylesheet* kan ik dan in één keer de `style`-kenmerken van de foto's binnen deze `class` opnemen:

```
div.fotogroep img {  
  width: 200px;  
  max-width: 95%;  
  margin: 2px;  
}
```

Voor de `style`-kenmerken van de foto's heb ik alleen de breedte, maximum-breedte en marge vastgelegd (als er maar één waarde achter `margin` staat, geldt die voor alle vier de kanten, dus boven, rechts, onder en links).

Wordt de ruimte te smal om deze vier afbeeldingen naast elkaar weer te geven, dan worden ze vanzelf onder elkaar neergezet:



Het is natuurlijk het mooiste als alle foto's dezelfde hoogte/breedte-verhouding hebben, zoals in dit voorbeeld. Soms zul je daarvoor de oorspronkelijke foto's in een fotobewerkingsprogramma moeten bijsnijden, zodat iedere foto dezelfde hoogte/breedte-verhouding heeft.

7. Over bijschriften

Soms wil je een bijschrift naast of onder je foto of afbeelding laten weergeven. Daarvoor zet je je foto of afbeelding (`img`) binnen een `figure`-element neer. En binnen dat `figure`-element komt dan een `figcaption` (bijschrift) te staan. Ik heb de verschillende onderdelen elk even zijn eigen kleur gegeven:

```
<figure style="margin: 0; padding: 0; display: flex; align-items: flex-start;">
  
  <figcaption style="width: 250px; max-width: 100%; box-sizing: border-box;
  padding: 6px;">Krokussen: een van de eerste lentebloemen</figcaption>
</figure>
```

Je ziet dat voor de `figure` de `margin` en de `padding` beide op 0 zijn gezet. Dat is gedaan omdat sommige browsers uit zichzelf wat ruimte om het `figure`-element zetten. En hoeveel ruimte dat moet zijn, kun je beter zelf bepalen. Voorlopig staan deze waarden daarom op 0.

Verder staan er in de `style`-kenmerken van de `figure` de volgende twee instructies:

```
display: flex;
align-items: flex-start;
```

Met die twee instellingen kun je de positie van de onderdelen binnen de `figure` bepalen. Met `display: flex` worden de onderdelen binnen de `figure` als blokjes naast elkaar gezet en met `align-items: flex-start` worden die onderdelen aan de bovenrand met elkaar uitgelijnd.

De foto (`img`) en het bijschrift (`figcaption`) zijn beide 250 pixels breed, mogen nooit breder zijn dan 100% van het vak waarbinnen ze staan, en het bijschrift heeft 6 pixels ruimte rondom (vanwege de instelling `box-sizing: border-box` wordt die 100% breedte berekend inclusief `padding` en eventueel `border`).

In je browser ziet dat er zó uit:



Krokussen: een van de eerste
lentebloemen

Positie bijschrift

De foto en het bijschrift zijn dus uitgelijnd aan de bovenkanten met behulp van de waarde `flex-start` voor de eigenschap `align-items`, maar ze kunnen ook aan de onderkanten zijn uitgelijnd, of zijn gecentreerd. Dat doe je door `flex-start` te veranderen in `flex-end` of `center`:



Krokussen: een van de eerste
lentebloemen

```
align-items: flex-end;
```



Krokussen: een van de eerste
lentebloemen

```
align-items: center;
```

Tekst eronder

Om de tekst onder de foto te laten weergeven, hoef je aan de `style`-kenmerken van het `figure`-element maar één instructie toe te voegen:

```
flex-direction: column;
```

Daarmee worden de onderdelen binnen het `figure`-element niet naast elkaar weergegeven, zoals standaard gebeurt, maar ónder elkaar. De regel voor de `align-items` haal je weg, want die is niet meer nodig. En om het bijschrift te laten centreren onder de foto voeg je de regel toe: `text-align: center;` Je krijgt dan dit:



Krokussen: een van de eerste
lentebloemen

De complete code van deze afbeelding met bijschrift eronder is:

```
<figure style="margin: 0; padding: 0; display: flex; flex-direction: column;">
  
  <figcaption style="width: 250px; max-width: 100%; box-sizing: border-box;
  padding: 6px; text-align: center;">Krokussen: een van de eerste
  lentebloemen</figcaption>
</figure>
```

Tekst over de afbeelding heen

Wat je ook nog kunt doen, is het bijschrift over de afbeelding heen laten zetten. Daarvoor moet je eerst aan de stijkenmerken van het `figure`-element de volgende regel toevoegen:

```
position: relative;
```

Met die instelling zorg je ervoor dat je de onderdelen binnen dit `figure`-element (de foto zelf en het bijschrift) nauwkeurig kunt positioneren.

Daarna voeg je aan de stijkenmerken van het `figcaption`-element (het bijschrift) deze regels toe:

```
position: absolute;  
bottom: 0;
```

Daarmee wordt het `figcaption`-element tegen de onderkant van het `figure`-element geplaatst, en dus óver de afbeelding heen.

Omdat het bijschrift een standaardtekstkleur heeft (meestal zwart), en de tekst van het bijschrift daarom onleesbaar kan worden als de afbeelding van zichzelf ook vrij donker is, is het handig nog meer eigenschappen in te stellen voor de `figcaption`, bijvoorbeeld:

```
background-color: rgba(0,0,0,0.5);  
color: white;
```

Hiermee geef je het bijschrift een donkere, enigszins transparante achtergrondkleur en een witte tekstkleur.
Het geheel ziet er dan zó uit:



En dit is de complete code daarvoor:

```
<figure style="margin: 0; padding: 0; display: flex; flex-direction: column;
position: relative">
  
  <figcaption style="width: 250px; max-width: 100%; box-sizing: border-box;
padding: 4px; text-align: center; position: absolute; bottom: 0;
background-color: rgba(0,0,0,0.5); color: white;">Krokussen: een van de
  eerste lentebloemen</figcaption>
</figure>
```

Meerdere foto's met bijschriften naast of onder elkaar

Als je meerdere foto's met bijschriften naast of onder elkaar wilt zetten, dan is het weer het gemakkelijkste als je alles eerst groepeert in een `div`:

```
<div class="fotogroep">
<figure>
  
  <figcaption>Krokussen: een van de eerste lentebloemen</figcaption>
</figure>
<figure>
  
  <figcaption>Rode tulpen zorgen voor een kleurig accent</figcaption>
</figure>
<figure>
  
  <figcaption>Narcissen zijn er in allerlei geeltinten</figcaption>
</figure>
<figure>
  
  <figcaption>Klein maar fijn: pittige blauwe druifjes</figcaption>
</figure>
</div>
```

De `style`-kenmerken voor dat `div` en de onderdelen daarbinnen zet je vervolgens in je *stylesheet* neer, zodat je ze niet elke keer hoeft te herhalen.

Dit `div`-element (met de `class` *fotogroep* in het voorbeeld) krijgt in de eerste plaats de eigenschap `display: flex`, om invloed te krijgen op de positionering van de verschillende onderdelen die erbinnen staan.

Om ervoor te zorgen dat de verschillende `figure`-elementen onder elkaar komen te staan in plaats van naast elkaar als het beeldscherm niet breed genoeg is, krijgt dit element ook nog de eigenschap `flex-wrap: wrap`. De eigenschap `align-items: flex-start` ten slotte, voeg je toe om de `figure`-elementen netjes aan de bovenrand uit te lijnen.

De code in je *stylesheet* voor dit `div` wordt dus:

```
div.fotogroep {
  display: flex;
  flex-wrap: wrap;
  align-items: flex-start;
}
```

Gecombineerd met de eigenschappen zoals we die al hadden voor de afzonderlijke `figure`-elementen, krijg je bijvoorbeeld dit:



Krokussen: een van de eerste
lentebloemen



Rode tulpen zorgen voor een kleurig
accent



Narcissen zijn er in allerlei geeltinten



Klein maar fijn: pittige blauwe druifjes

Of, als het scherm te smal is om de foto's naast elkaar weer te geven:



Krokussen: een van de eerste
lentebloemen



Rode tulpen zorgen voor een kleurig
accent



Narcissen zijn er in allerlei geeltinten



Klein maar fijn: pittige blauwe druifjes

Je krijgt natuurlijk weer het beste resultaat als je uitgaat van foto's die dezelfde breedte/hoogte-verhouding hebben.

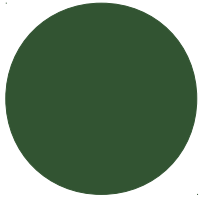
De volledige *stylesheet*-code voor de foto's op de vorige pagina is:

```
div.fotogroep {
  display: flex;
  flex-wrap: wrap;
  align-items: flex-start;
}

div.fotogroep figure {
  margin: 0;
  padding: 0;
  display: flex;
  flex-direction: column;
}

div.fotogroep figure img {
  width: 250px;
  max-width: 100%;
  padding: 4px;
  box-sizing: border-box;
}
```

```
div.fotogroep figure figcaption {  
  width: 250px;  
  max-width: 100%;  
  padding: 4px;  
  box-sizing: border-box;  
  text-align: center;  
}
```



8. Over achtergrondafbeeldingen

Er is nog een andere manier om foto's of afbeeldingen in je website op te nemen, en dat is door een achtergrond aan je webpagina (of aan een gedeelte daarvan) te koppelen. Dat doe je met:

```
background-image: url();
```

Tussen de set ronde haken komt de locatie van je afbeelding te staan, net zoals je die bij een gewone afbeelding zou neerzetten tussen de dubbele aanhalingstekens bij de *source* (`src`).

Het verschil is alleen dat `background-image` een *style*-eigenschap is. Zet deze regel dus in je stylesheet neer, bij de kenmerken van het element dat je een achtergrond wilt geven. Is dat je webpagina, dan zou dat dus het `body`-element zijn.

Maar om in een later stadium problemen op bepaalde smartphones te voorkomen, is het beter de achtergrondafbeelding niet instellen voor het `body`-element zelf, maar voor een kunstmatig element dat je eraan vooraf laat gaan:

```
body:before {  
    background-image: url(https://www.mijn-eigen-website.nl/images/heuvels.jpg);  
}
```

(Let er natuurlijk wel weer op dat de bestandsgrootte van de afbeelding die je op de achtergrond wilt laten zien binnen de perken blijft, om de laadtijd van je pagina's niet te lang te laten worden.)

Je ziet dat de locatie in beide voorbeelden is weergegeven als een **absolute verwijzing**: dat wil zeggen dat de locatie van de afbeelding helemaal is 'uitgeschreven'. Dat hoeft niet per se, je kunt ook een **relatieve verwijzing** gebruiken. 'Denk' daarbij vanuit je stylesheet. Staat die in een aparte map, dan moet je dus eerst als het ware een map 'omhoog', en daarna weer naar 'beneden' naar de map waarin je afbeeldingen staan. In mijn geval zou een relatieve verwijzing vanuit mijn stylesheet (dat in een aparte map staat) de volgende zijn:

```
background-image: url(../images/heuvels.jpg);
```

background-repeat

Het handigste is als je voor je achtergrondafbeelding een foto in landscape-formaat kiest, want beeldschermen zijn vaker breder dan dat ze hoog zijn.

Maar ook al kies je een *landscape*-afbeelding, deze zal nooit exact dezelfde afmetingen hebben als het beeldscherm waarop je website wordt bekeken. Ze wordt daarom in horizontale en verticale richting zo vaak herhaald als nodig is om het hele scherm te vullen. Omdat je waarschijnlijk wilt dat je achtergrondafbeelding maar één keer wordt vertoond, voeg je de volgende regel aan je stylesheet toe:

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

Daarmee wordt voorkomen dat je achtergrondafbeelding óf in horizontale óf in verticale richting wordt herhaald om het scherm te vullen.

background-size

Maar nu je hebt voorkomen dat je afbeelding wordt herhaald om het scherm te vullen, kan het zijn dat ze te klein is om als achtergrond te dienen. Daarom voeg je vervolgens toe:

```
background-size: cover;
```

Daarmee zorg je ervoor dat het hele beeldscherm bedekt wordt. De foto wordt naar beide zijden zoveel uitgerekt als nodig is om het hele scherm te vullen. Dat kan natuurlijk wel betekenen dat een gedeelte van de foto is verdwenen. Is je pagina bijvoorbeeld erg lang, dan wordt de foto zo veel ‘opgerekt’ dat hij tegelijkertijd veel breder is geworden dan nodig is. Aan de zijkanten wordt je afbeelding dan als het ware ‘afgesneden’.

background-position

Je foto wordt nu helemaal vertoond — althans, voor zover mogelijk. De linkerbovenhoek van de afbeelding wordt standaard in de linkerbovenhoek van het beeldscherm geplaatst. Maar bestaat het bovenste gedeelte van je foto bijvoorbeeld uit lucht, dan zie je misschien liever dat de rechteronderhoek van je foto in de rechteronderhoek van het beeldscherm wordt neergezet. Dat kun je doen met de regel:

```
background-position: right bottom;
```

Je kunt de woorden: `right`, `left`, `top`, `bottom` en `center` gebruiken om je foto te positioneren. De eerste waarde die je opgeeft heeft altijd betrekking op de positie langs de horizontale as, de tweede op de positie langs de verticale as.

background-color

Vaak wordt aan het body-element ook nog een achtergrondkleur gegeven (met behulp van `background-color`). Dat lijkt onzinnig, omdat een achtergrondafbeelding altijd bovenop de achtergrondkleur wordt vertoond, maar als deze achtergrondafbeelding om de een of andere reden niet geladen kan worden, dan wordt in ieder geval de achtergrondkleur nog vertoond, en ziet je website er niet al te kleurloos uit.

De instructie daarvoor in je stylesheet ziet er zó uit:

```
background-color: lightblue;
```

Eigenschappen `body:before`-element

De eigenschappen van de achtergrondafbeelding zijn daarmee vastgelegd. Maar je moet ook nog de eigenschappen vastleggen voor het kunstmatige element dat je had aangemaakt (`body:before`). Omdat dit element behalve deze achtergrondafbeelding verder geen inhoud heeft, heeft het uit zichzelf geen breedte of hoogte en zul je het dus ook niet zien. Daarom moet je het nog de volgende eigenschappen meegeven:

```
content: "";  
display: block;  
width: 100%;  
height: 100%;  
position: fixed;  
left: 0;
```

```
top: 0;
z-index: -10;
```

Dit element heeft dus geen inhoud (`content: ""`), maar moet wel het hele scherm omvatten (`display: block; width: 100%; height: 100%`).

De positie van dit blok is vast, de linkerbovenhoek ervan bevindt zich altijd links bovenaan het beeldscherm (`position: fixed; left: 0; top: 0`).

En met de eigenschap `z-index` ten slotte stel je in hoe de verschillende lagen van een webpagina over elkaar heen geplaatst moeten worden. Dit kunstmatige element komt dus vóór het `body`-element, en het wordt helemaal onderop geplaatst (er zijn nergens anders waarden voor de `z-index` opgegeven, dus een waarde van `-10` is genoeg om deze laag helemaal onderaan je webpagina te laten weergeven).

Daarmee is de volledige code voor de achtergrondafbeelding:

```
body:before {
  content: "";
  display: block;
  width: 100%;
  height: 100%;
  position: fixed;
  left: 0;
  top: 0;
```



```
z-index: -10;
background-image: url(https://www.mijn-eigen-website.nl/images/heuvels.jpg);
background-repeat: no-repeat;
background-position: left top;
background-size: cover;
background-color: lightblue;
}
```

Op mijn website kun je [het resultaat van deze code bekijken](#).

Tot slot

Met al deze voorbeelden en codes heb je genoeg in handen om met succes foto's of andere afbeeldingen in je webpagina's op te nemen. Natuurlijk kun je nog veel meer met je foto's doen, zoals ze laten vertonen in een popup-venster, slideshow of fotogalerij. Meer suggesties daarvoor staan op de pagina [Leuke manieren om foto's op je website te zetten](#).