



## Les 8

Lay-out:  
Responsive design – positioning  
– floats – multi-column

# Content

- Responsive design
- Positionering: position - float - clear
- Multi-column

# Responsive design

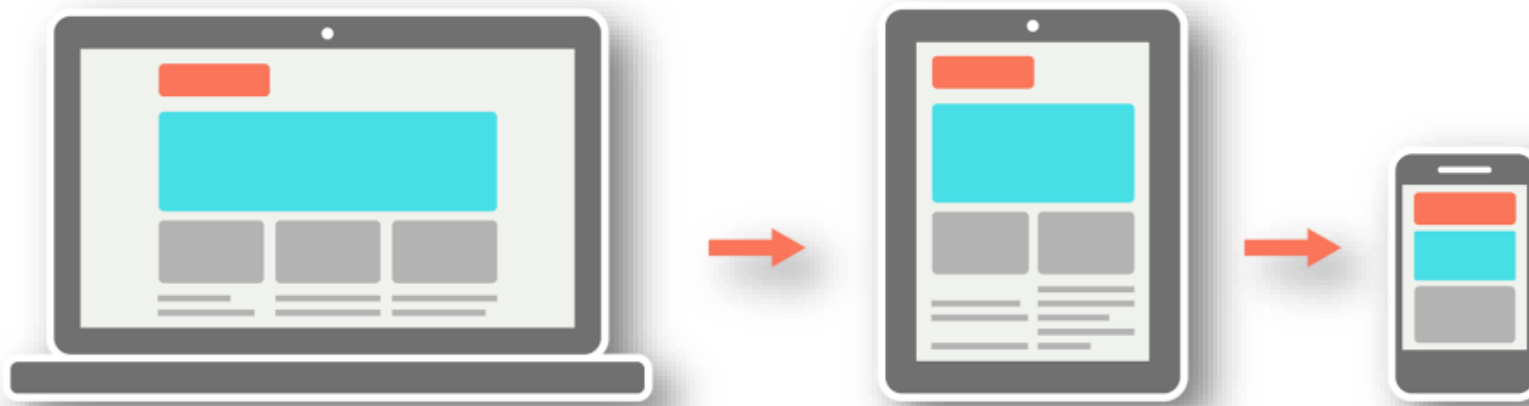
# Responsive design



# Responsive design

- Responsive webdesign
  - Het doel is om webpagina's leesbaar te maken op verschillende devices (= > verschillende schermgroottes).
  - Een pagina met meerdere kolommen is gemakkelijk te lezen op een groot scherm, maar niet op een klein scherm.
  - Responsive Web Design maakt gebruik van HTML en CSS om de pagina's geschikt te maken voor alle soorten schermen.
- Concreet
  - De layout van de webpagina is afhankelijk van het scherm
  - Sommige elementen worden verborgen of getoond naargelang de schermgrootte
  - Buttons, menus, ... worden anders weergegeven naargelang de schermgrootte

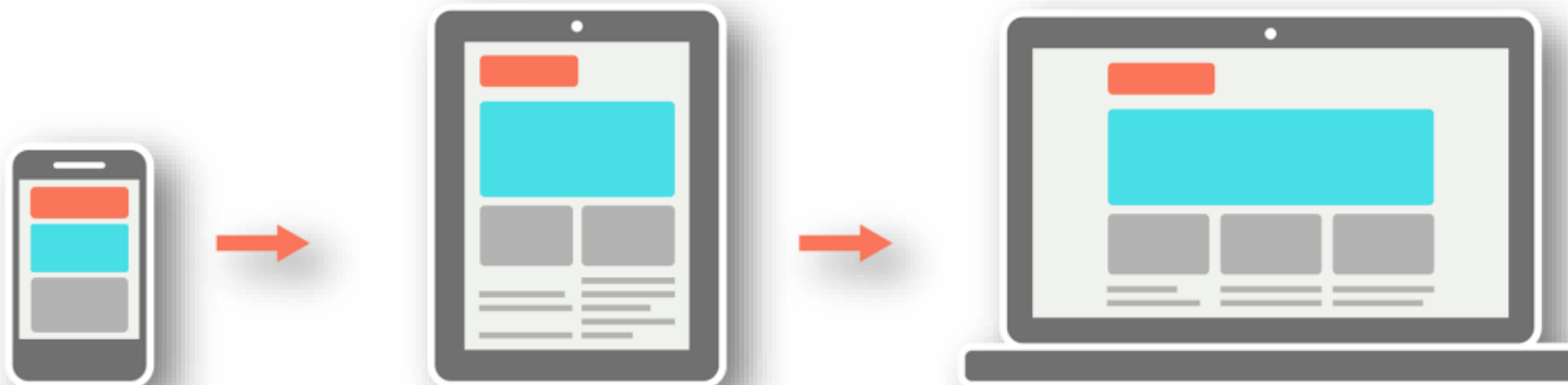
# Responsive design



Responsive Web Design

---

Mobile First Web Design



# Responsive design: media queries

- doel:  
opmaak koppelen aan eigenschappen van een device.
- media queries bestaan uit:
  - apparaat type (screen, print, speech).
  - media kenmerken (breedte, orientatie):  
**breakpoints.**
  - een blok { } van stijlregels die geldig zijn voor het gedeclareerde apparaat type en de media kenmerken.

```
@media apparaat and (media feature) {  
    selector {  
        declaraties;  
    }  
}
```

# Responsive design: media queries

Voorbeeld:

Indien het scherm minimaal 1024px breed is zal het main element opgedeeld worden in twee kolommen van 66% en 34% van de breedte van het scherm (grid layout: zie later).

```
@media screen and (min-width: 1024px) {  
  main {  
    display: grid;  
    grid-template-columns: 66% 34%;  
  }  
}
```



# Responsive webdesign: viewport

- Het browservenster is de viewport.
- Op een desktop zijn we ermee vertrouwd dat de viewport onafhankelijk is van het scherm.  
We zijn ermee vertrouwd dat we het browservenster kunnen vergroten en verkleinen al naargelang onze wensen.
- Op mobiele toestellen is de viewport volledig hetzelfde: het is de ruimte die wordt gebruikt om de webpagina te tonen, ongeacht de grootte van het scherm.

# Responsive webdesign: viewport

- Op mobiele toestellen is het niet mogelijk om het browservenster te wijzigen.  
De viewport is dan gelijk aan de grootte van het venster.
- Viewports op mobiele toestellen zijn zo ontworpen dat ze groter zijn dan de resolutie van het scherm.
- Men onderscheidt:
  - een zichtbare viewport = scherm
  - een lay-out viewport = pixel size, groter dan de zichtbare viewport (enkele voorbeelden op volgende slide).
- css eigenschappen worden gedeclareerd voor de lay-out viewport

# Responsive webdesign: viewport

## Apple Products

	Pixel Size	Viewport
<b>iPhone</b>		
iPhone XR	828 x 1792	414 x 896
iPhone XS	1125 x 2436	375 x 812
iPhone XS Max	1242 x 2688	414 x 896
iPhone X	1125 x 2436	375 x 812
iPhone 8 Plus	1080 x 1920	414 x 736
iPhone 8	750 x 1334	375 x 667
iPhone 7 Plus	1080 x 1920	414 x 736
iPhone 7	750 x 1334	375 x 667
iPhone 6 Plus/6S Plus	1080 x 1920	414 x 736
iPhone 6/6S	750 x 1334	375 x 667
iPhone 5	640 x 1136	320 x 568
<b>iPod</b>		
iPod Touch	640 x 1136	320 x 568
<b>iPad</b>		
iPad Pro	2048 x 2732	1024 x 1366
iPad Third & Fourth Generation	1536 x 2048	768 x 1024
iPad Air 1 & 2	1536 x 2048	768 x 1024
iPad Mini 2 & 3	1536 x 2048	768 x 1024
iPad Mini	768 x 1024	768 x 1024

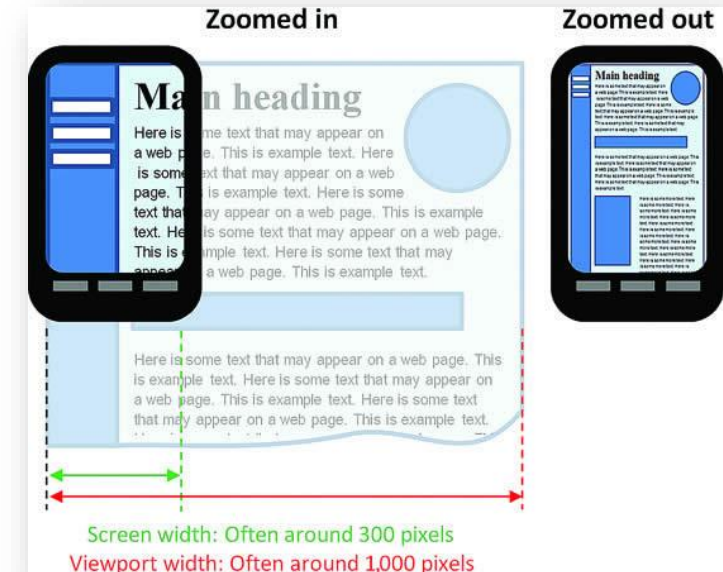
## Android Devices

	Pixel Size	Viewport
<b>Phones</b>		
Nexus 6P	1440 x 2560	412 x 732
Nexus 5X	1080 x 1920	412 x 732
Google Pixel 4 XL	1440 x 869	412 x 869
Google Pixel 4	1080 x 2280	412 x 869
Google Pixel 3a XL	1080 x 2160	412 x 824
Google Pixel 3a	1080 x 2220	412 x 846
Google Pixel 3 XL	1440 x 2960	412 x 847
Google Pixel 3	1080 x 2160	412 x 824
Google Pixel 2 XL	1440 x 2560	412 x 732
Google Pixel XL	1440 x 2560	412 x 732
Google Pixel	1080 x 1920	412 x 732
Samsung Galaxy Note 10+	1440 x 3040	412 x 869
Samsung Galaxy Note 10	1080 x 2280	412 x 869
Samsung Galaxy Note 9	1440 x 2960	360 x 740
Samsung Galaxy Note 5	1440 x 2560	480 x 853
LG G5	1440 x 2560	480 x 853
One Plus 3	1080 x 1920	480 x 853
Samsung Galaxy S9+	1440 x 2960	360 x 740
Samsung Galaxy S9	1440 x 2960	360 x 740

Samsung Galaxy S8+	1440 x 2960	360 x 740
Samsung Galaxy S8	1440 x 2960	360 x 740
Samsung Galaxy S7 Edge	1440 x 2560	360 x 640
Samsung Galaxy S7	1440 x 2560	360 x 640
<b>Tablets</b>		
Nexus 9	1536 x 2048	768 x 1024
Nexus 7 (2013)	1200 x 1920	600 x 960
Pixel C	1800 x 2560	900 x 1280
Samsung Galaxy Tab 10	800 x 1280	800 x 1280
Chromebook Pixel	2560 x 1700	1280 x 850

# Responsive webdesign: viewport

- De mobiele browser zal de webpagina in zijn totaliteit laten zien in de viewport. Daardoor krimpt de pagina naar de breedte van het scherm. Als dat niet het geval zou zijn, zou er maar een klein gedeelte van de webpagina te zien zijn. We noemen dat keyhole browsing – te vergelijken met wanneer je door een sleutelgat naar een volledige ruimte kijkt.
- Dit is het probleem: de pagina wordt verkleind zodat er niet meer horizontaal moet gescrolld worden.



# Responsive webdesign: viewport

- Hoe lossen we dit probleem op?
- Door de viewport width gelijk te stellen aan de device width. Dit gebeurt in de head van je html document met de viewport meta tag.



```
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0, user-scalable=yes">
  <title> ... </title>
  <body> ... </body>
</head>
</html>
```

# Lay-out: positioning

# Positionering

- Omvattende blok – containing blok (container).
- Elk blok krijgt een positie toegewezen in het omvattende blok.
- Schema's:
  - normale flow – plaatsing afhankelijk van display waarde
  - floats: vlottend/zwevend (links/rechts), uit normale flow gehaald
  - absolute positionering: krijgt absolute positie ten opzichte van de eerste niet static omvattende blok.

# Eigenschap position en eigenschappen voor de offset

- eigenschap position: kan volgende waarden hebben:
  - static (default)
  - relative
  - absolute
  - fixed
- offset:
  - top
  - right
  - bottom
  - left





# Positionering: eigenschap float en clear

- Door **float**(left/right) te gebruiken haal je het blok uit de normale flow en kan de inhoud van de onderliggende blokken erom heen lopen.
- Dat laatste kan door de eigenschap **clear**(left/right/both) worden voorkomen.
- Vroeger werd float ook gebruikt voor volledig pagina lay-outs. Dit gebeurt nu minder of niet omdat daarvoor **flex** en **grid** meer geschikt zijn.

# Positionering: eigenschap float en clear

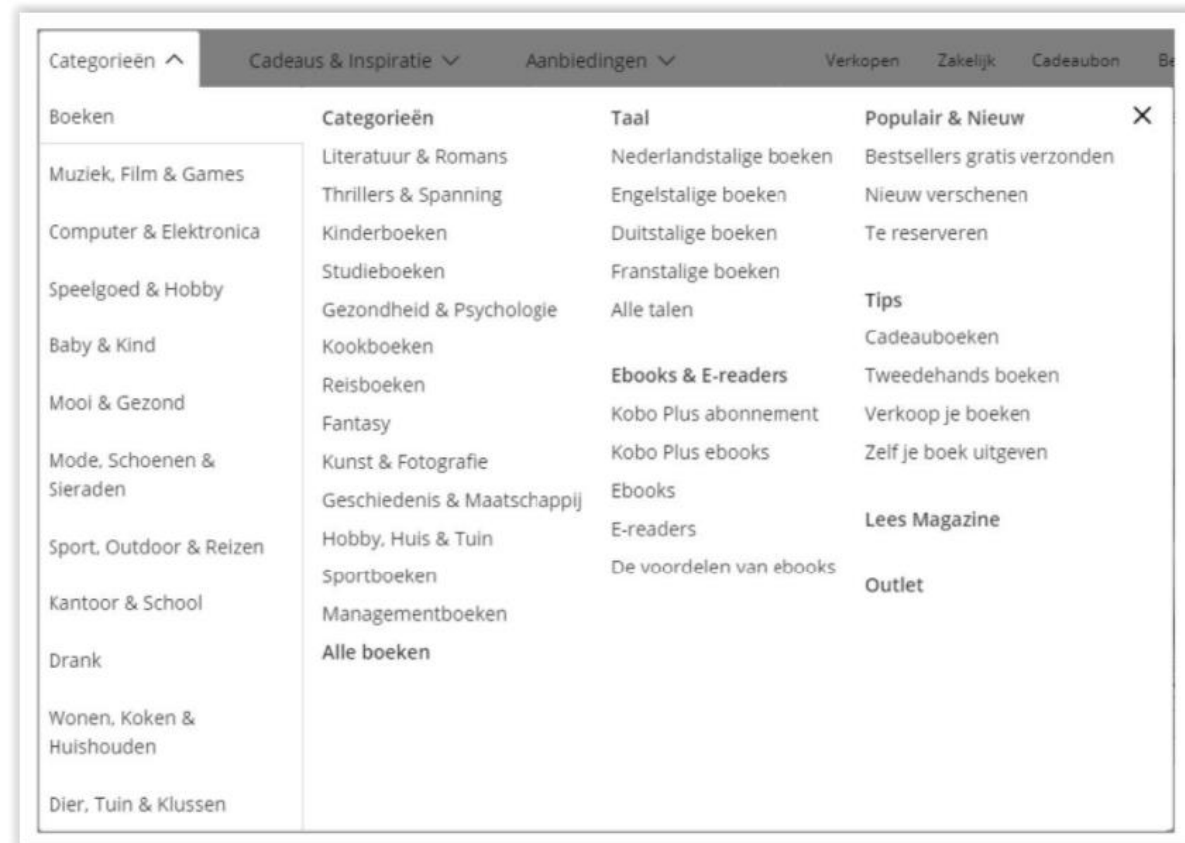
Vaak gebruikt bij tekst rond afbeeldingen.

Float: afbeelding naar links	Float: afbeelding naar rechts
 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ad alias animi aperiam debitis earum, eos et, laborum magni maiores molestiae nemo quia quibusdam reiciendis ut voluptate. Consequatur eaque et sint. Dolorem, nostrum, vero!</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ad alias animi aperiam debitis earum, eos et, laborum magni maiores molestiae nemo quia quibusdam reiciendis ut voluptate. Consequatur eaque et sint. Dolorem, nostrum, vero!</p> 

# Lay-out - multicolumn


# Multi-column

Het is ook mogelijk om een container of een deel van je pagina op te delen in kolommen. Zie voorbeeld bol.com



# Multi-column

Eigenschappen:

- [column-count](#): gewenste aantal kolommen
  - [column-width](#): gewenste breedte
  - [column-span](#): overspannen van kolommen: none/all
  - [column-gap](#): breedte van ruimte tussen kolommen
  - [column-rule](#): lijn tussen kolommen (color/style/width)
  - [column-fill](#): opvullen van de kolommen - gebalanceerd of niet.
- 
- A diagram consisting of a right-facing curly bracket grouping the first two list items, [column-count](#) and [column-width](#). To the right of the bracket is a rectangular box with an orange border containing the text 'columns'.