# **Responsive Web Design**

## Wat is Responsive Web Design?

Responsive webdesign zorgt ervoor dat uw webpagina er goed uitziet op alle apparaten.

Responsive webdesign maakt alleen gebruik van HTML en CSS.

Responsive webdesign is geen programma of JavaScript.

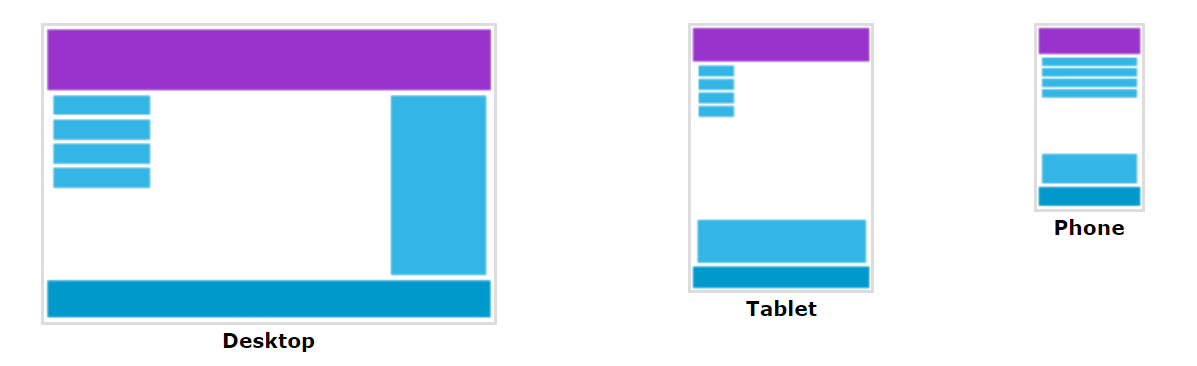
## Ontwerpen in functie van user experience.

<https://www.w3schools.com/css/css_rwd_intro.asp>

Webpagina's kunnen worden bekeken met behulp van veel verschillende apparaten: desktops, tablets en telefoons.   
Uw webpagina moet er goed uitzien en gebruiksvriendelijk zijn, ongeacht het apparaat.

Webpagina's mogen geen informatie weglaten om op kleinere apparaten te passen, maar moeten de inhoud aanpassen om op elk apparaat te passen.

Men spreekt van responsief webdesign wanneer u CSS en HTML gebruikt om de inhoud te verkleinen, te verbergen, te verkleinen, te vergroten of te verplaatsen zodat deze er op elk scherm goed uitziet.



## Viewport.

De viewport, vertaald, het kijkvenster is het zichtbare gedeelte van een webpagina voor de gebruiker.

De viewport is afhankelijk van het apparaat en zal op een mobiele telefoon kleiner zijn dan op een computerscherm.

Voorheen werden webpagina's alleen ontworpen voor computerschermen, en het was gebruikelijk dat webpagina's een statisch ontwerp en een vast formaat hadden.

Toen we vervolgens op het internet begonnen te surfen met behulp van tablets en mobiele telefoons, waren webpagina's met een vast formaat te groot om in de viewport te passen. Om dit te verhelpen, hebben browsers op die apparaten de hele webpagina verkleind om op het scherm te passen.

Dit was niet perfect! Maar een snelle oplossing.

## Setting the viewport.

HTML5 introduceerde een methode om webdesigners de controle te laten nemen over de viewport, via de <meta>-tag.

U zou het volgende <meta> viewport element in al uw webpagina's moeten opnemen.  
<meta name="viewport" content="width=device-width, initial-scale=1.0">

Een <meta> viewport element geeft de browser instructies over de afmetingen en de schaling van de pagina.

Het width=device-width deel stelt de breedte van de pagina in om de schermbreedte van het apparaat te volgen (die varieert afhankelijk van het apparaat).

Het initial scale=1.0 deel stelt het initiële zoomniveau in wanneer de pagina voor het eerst door de browser wordt geladen.

## Inhoud aanpassen aan de viewport

Gebruikers scrollen verticaal door websites, zowel op desktop- als mobiele apparaten - maar niet horizontaal!

Dus, als de gebruiker gedwongen wordt om horizontaal te scrollen, of uit te zoomen, om de hele webpagina te zien, resulteert dit in een slechte gebruikerservaring.

Enkele extra regels om te volgen:

1. Gebruik GEEN grote elementen met een vaste breedte.   
Als een afbeelding bijvoorbeeld breder wordt weergegeven dan de viewport, kan dit ertoe leiden dat de gebruiker horizontaal moet scrollen.  
Vergeet niet deze inhoud aan te passen aan de breedte van de viewport.

2. Laat de inhoud NIET afhankelijk zijn van een bepaalde viewportbreedte om goed te renderen.   
Aangezien de schermafmetingen en -breedte in CSS-pixels sterk verschillen tussen apparaten, moet de inhoud niet afhankelijk zijn van een bepaalde viewportbreedte om goed te renderen.

3. Gebruik CSS-media-query's om verschillende stijlen toe te passen voor kleine en grote schermen.   
Door grote absolute CSS-breedtes in te stellen voor pagina-elementen wordt het element te breed voor de viewport op een kleiner apparaat. Overweeg in plaats daarvan het gebruik van **relatieve breedtewaarden**, zoals **breedte: 100%.**Wees ook voorzichtig met het gebruik van grote absolute positioneringswaarden.   
Op kleine apparaten kan het element buiten de kijkvenster vallen.

<https://www.w3schools.com/css/css_rwd_viewport.asp>

## Responsive Images

Gebruik maken van de max-width eigenschap

Als de maximale breedte-eigenschap is ingesteld op 100%, zal de afbeelding worden verkleind als dat nodig is, maar nooit worden opgeschaald om groter te zijn dan de oorspronkelijke grootte.

**Codevoorbeeld : 1responsive**

Bekijk met F12 op verschillende devices.  
Verander de code met de image img\_girl.

## Verschillende afbeeldingen tonen afhankelijk van de breedte van de browser

Met het HTML <picture>-element kunt u voor elke venstergrootte een andere afbeelding tonen.

**Codevoorbeeld : 2responsive**

## Responsive Text Size

De tekstgrootte kan worden ingesteld met een "vw"-eenheid, dat wil zeggen de "viewport-breedte".  
Op die manier volgt de tekstgrootte de grootte van het browservenster, viewport is immers de grootte van het browservenster.

1vw = 1% van de viewportbreedte.   
Als de viewport 50 cm breed is, is 1vw 0,5 cm.

Vergroot het browservenster om te zien hoe de schalen voor de tekstgrootte zijn.

Gebruik de "vw" eenheid bij het schalen van de tekst.   
10vw stelt de grootte in op 10% van de breedte van het kijkvenster.

**Codevoorbeeld : 3responsive**

## Media Queries

Naast het aanpassen van tekst en afbeeldingen, is het ook gebruikelijk om media queries te gebruiken in responsieve webpagina's.

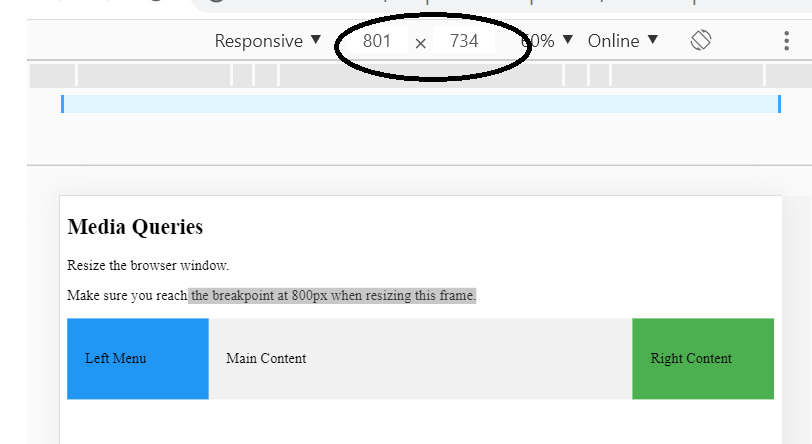
Met media queries kunt u volledig verschillende stijlen definiëren voor verschillende browsermaten.  
Je moet hiervoor zelf breakpoints bepalen. Een breakpoint bepaalt de minimale breedte van een device.   
**Nog beter is het om te werken met maximale breedte omdat men tegenwoordig tracht om mobile first te werken.**

Voorbeeld:   
Wijzig de grootte van het browservenster om te zien dat de drie div-elementen hieronder horizontaal worden weergegeven op grote schermen en verticaal worden gestapeld op kleine schermen.

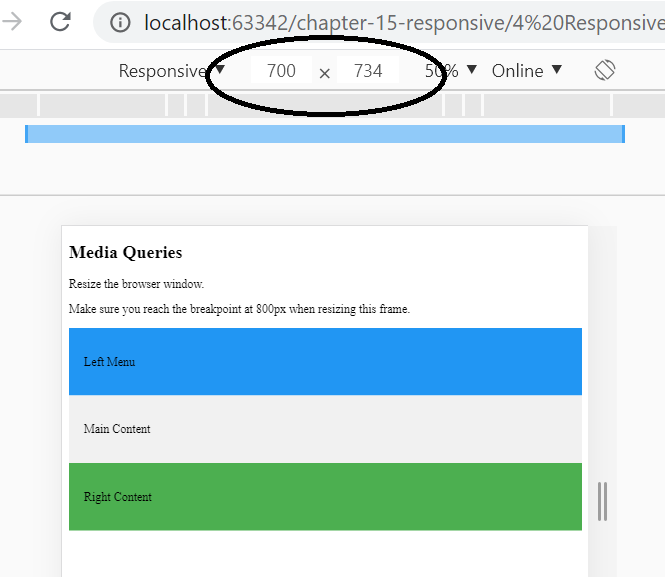
**@media screen and** (**min-width**: 800**px**) {  
 .**left**, .**right** {  
 **width**: 20%;  
 }  
 .**main**{  
 **width**: 60%;  
 }  
}

Wanneer het scherm een **minimale breedte** heeft van 800px, wordt bovenstaande CSS gebruikt.

**Codevoorbeeld : 4responsive**

****

Stijl die wordt toegepast bij een viewport van minimaal 800 of breder

****

Stijl die wordt toegepast bij een viewport die smaller is dan 800.

**TO DO :**   
Pas de code aan zodat de elementen onder elkaar worden getoond indien de viewport kleiner is dan 800px.

## Altijd ontwerpen voor Mobile First

Mobile First betekent eerst ontwerpen voor mobiele apparaten alvorens te ontwerpen voor desktop of laptop (dit zal de pagina sneller weergeven op kleinere apparaten).

**Codevoorbeeld : 5responsive**

**TO DO :**   
Pas de code aan zodat je mobile first werkt.

## Breedte viewports

<https://docs.adobe.com/content/help/en/target/using/experiences/vec/mobile-viewports.html>

@media (min-width: 30em) and (max-width: 50em) { ... }

@media (30em <= width <= 50em ) { ... }

/\* Extra small devices (phones, 600px and down) \*/  
@media only screen and (max-width: 600px) {...}  
  
/\* Small devices (portrait tablets and large phones, 600px and up) \*/  
@media only screen and (min-width: 600px) {...}  
  
/\* Medium devices (landscape tablets, 768px and up) \*/  
@media only screen and (min-width: 768px) {...}  
  
/\* Large devices (laptops/desktops, 992px and up) \*/  
@media only screen and (min-width: 992px) {...}  
  
/\* Extra large devices (large laptops and desktops, 1200px and up) \*/  
@media only screen and (min-width: 1200px) {...}

## Oefeningen

## Werk steeds mobile first!

**Oefening 1**

Vertrekcode : Oefening1.html

Pas de code aan zodat de kolommen onder elkaar worden getoond bij een device kleiner dan 620px. Voorzie in dat geval een marge van 2,5%.

**Oefening 2**

Vertrekcode : oefening2.html

Indien device kleiner dan 780px : toon dan elke kolom onder elkaar en definieer een marge van 1,5%.  
Indien device tussen 780px en 1279px : toon de kolommen per twee.  
Indien device groter dan 1279px: toon de kolommen per vier.

**Tip : werk steeds in rijen.   
Gebruik de klasse clearfix (module 1 user interface) om elke rij te definiëren.**

.**clearfix**::**before**, .**clearfix**::**after** {  
 **content**: **" "**;  
 **display**: **table**;  
}  
  
.**clearfix**::**after** {  
 **clear**: **both**;  
}

**Oefening 3**

Vertrekcode : Oefening3.html

Indien device minimaal 620px is, toon dan telkens een rij van drie gelijke kolommen. In het andere geval toon je alle paragrafen onder elkaar. Gebruik telkens een marge van 1%.

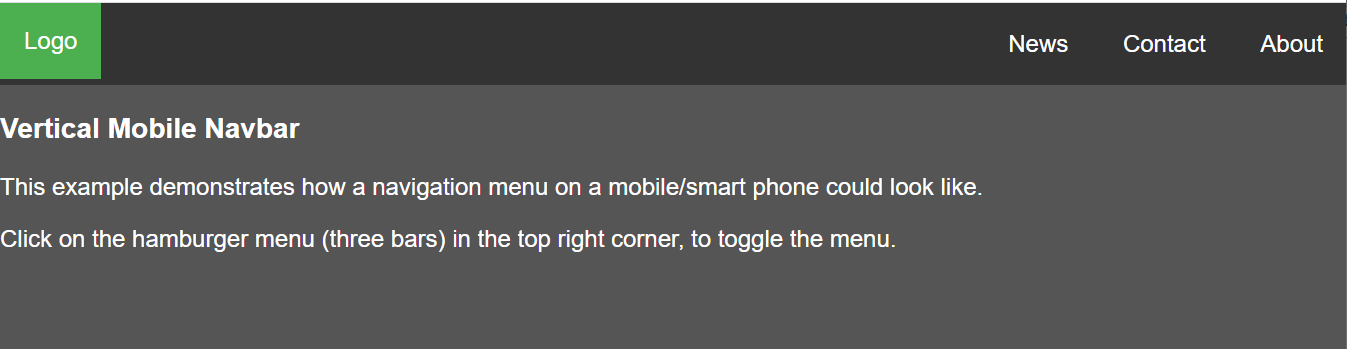
**Oefening 4**

Vertrekcode : Oefening4.html

Bestudeer grondig de vertrekcode, test uit. **Deze code werkt alleen goed voor een device kleiner dan 761px. Dat ga jij aanpassen.**  
Zorg dat je pagina de favicon kent.  
<**link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"**>

Bekijk de Javascript in je HTML. Je hoeft dit nog niet helemaal te begrijpen. Je leert dit zelf te schrijven.

**Pas aan.**Indien device groter dan 760px is, toon dan het volledig uitgeklapt menu zoals in afbeelding hieronder.



**Oefening 5**

Vertrekcode : oefening5.html

Indien device kleiner dan 620px, toon dan de paragrafen onder elkaar en niet met drie naast elkaar.  
Vervang menu door uitklapbaar menu.

Bekijk in vorige oefening goed wat je allemaal moet kopiëren.

