

HoGent

BEDRIJF
EN
ORGANISATIE

Webapplicaties 1

H6 – Responsive

Responsive webdesign

- ▶ Wat verstaan we onder Responsive Webdesign?
- ▶ Separate Mobile Sites.
- ▶ Design for one web.
 - Responsive web design
 - Mobile first web design
- ▶ Belangrijke elementen:
 - viewport + initial scale.
 - CSS media queries – verschillende stijlen voor verschillende schermgroottes.
 - flexible grids.
 - Optimalisatie van afbeeldingen
- ▶ Appendix
 - Optimalisatie van URL's
 - Screen density

Responsive Webdesign

Responsive Web Design



Responsive Web Design

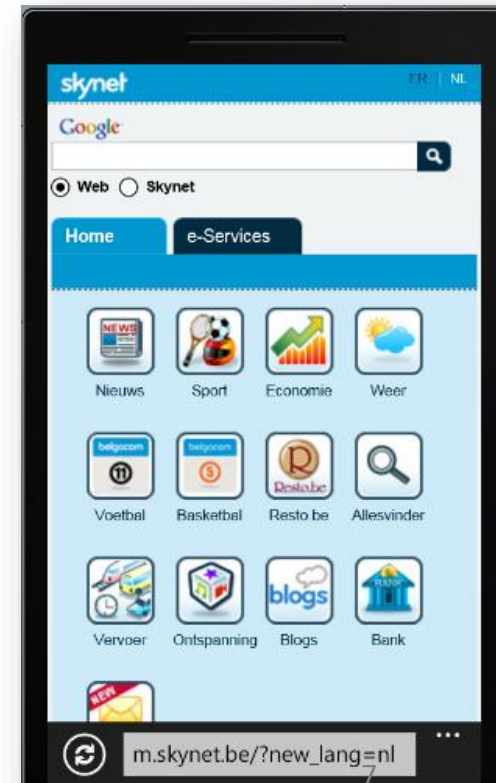
- ▶ Voor het commerciële succes van smartphones en tablets, werden webpagina's enkel weergegeven op schermen van desktops of portable computers (van 760 pixels wide tot widescreen displays).
- ▶ Nu moeten webpagina's kunnen weergegeven worden op een hele grote diversiteit schermen.

Responsive Web Design

- ▶ Twee mogelijke benaderingen om dit te realiseren:
 - Seperate Mobile Site: verschillende websites (html-css), voor desktop en mobiel.
 - Desktop: <http://www.facebook.com>
 - Mobiel: <http://m.facebook.com>
 - Eén website, die een andere layout heeft, naargelang de dimensies van het scherm: **Responsive Web Design**.
<http://www.vrtnws.be>

Seperate Mobile Site

- ▶ Dedicated sites (mdot strategie)
 - een subdomein:
m.bedrijf.be, mobile.bedrijf.be,
mobi.bedrijf.be, ...
 - Andere versie, andere inhoud
 - een DotMobi TLD: www.bedrijf.mobi
 - Weinig gebruikt, onbegrepen
 - een virtual directory: www.bedrijf.be/mobile
- ▶ Voorbeelden
 - m.mobistar.be
 - m.skynet.be



Seperate Mobile Site

▶ Pro

- design gemakkelijker te onderhouden want specifiek voor mobile
- subdomein = korter te typen
- auto redirection

▶ Contra

- dubbel onderhoud: content en design
- mobi target voor domain squatters:
is www.hogent.mobi ingenomen?

▶ Wat heb je nodig: device-detection scripts.

Design for one web

▶ 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

Design for one web

► Pro

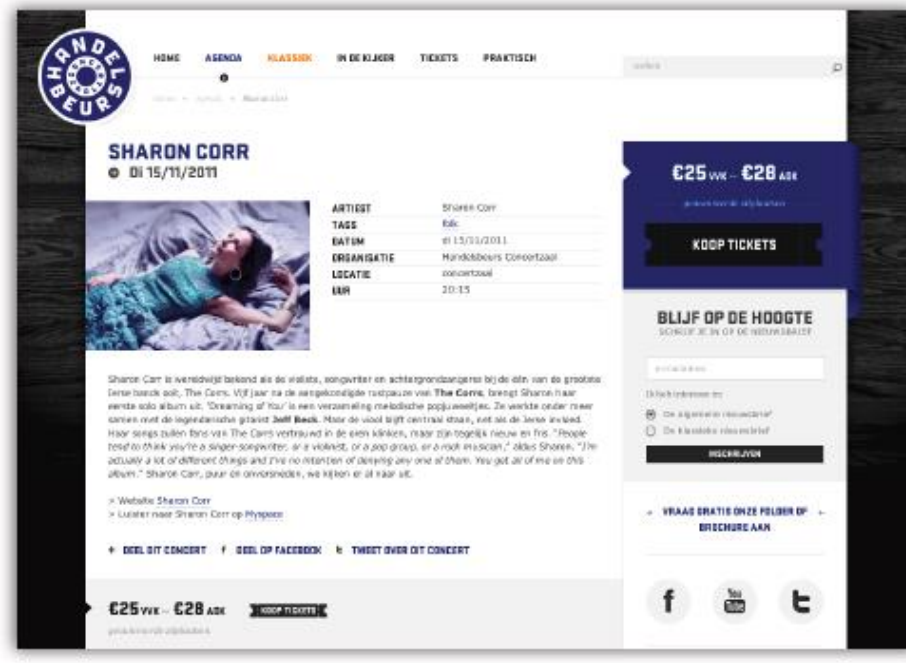
- SEO: alle content op de zelfde site, links verwijzen naar de dezelfde site.
- aanpasbaarheid: het aantal nieuwe devices blijft toenemen: je kan niet blijven specifieke dedicated sites maken.
- minder onderhoud want maar 1 site, anders inhoud van meerdere sites updaten.
- layout past zich aan aan de schermen.

► Contra

- bandbreedte/snelheid: een dedicated site bevat minder inhoud en aangepaste beelden, tenzij een responsive site daarop kan voorzien.
- bestandsgrootte: beelden, libraries.
- aantal HTTP requests: stylesheets, javascripts, media.

Design for one web

- ▶ Voorbeelden
 - www.standaard.be
 - www.handelsbeurs.be



Design for one web

- ▶ Wat verandert er doorgaans wanneer we kiezen voor een ander medium (smartphone / tablet / desktop) ?
 - aantal kolommen: op smartphone is er vaak maar 1 kolom, terwijl er op desktop bijvoorbeeld 3 kolommen zijn
 - flexibele breedtes
 - witruimte: vaak minder witruimte op smartphone dan op desktop
 - font size: vaak is de tekst iets kleiner op smartphone

Design for one web

- ▶ Wat verandert er doorgaans wanneer we kiezen voor een ander medium (smartphone / tablet / desktop) ?
 - navigatie menu: wordt dikwijls een knop op smartphone
 - <http://bradfrostweb.com/blog/web/responsive-nav-patterns/>
- minder belangrijke inhoud verbergen op mobiele toestellen
- background images: op desktop heb je een grote foto als background, terwijl deze foto niet getoond wordt op smartphone.



Design for one web

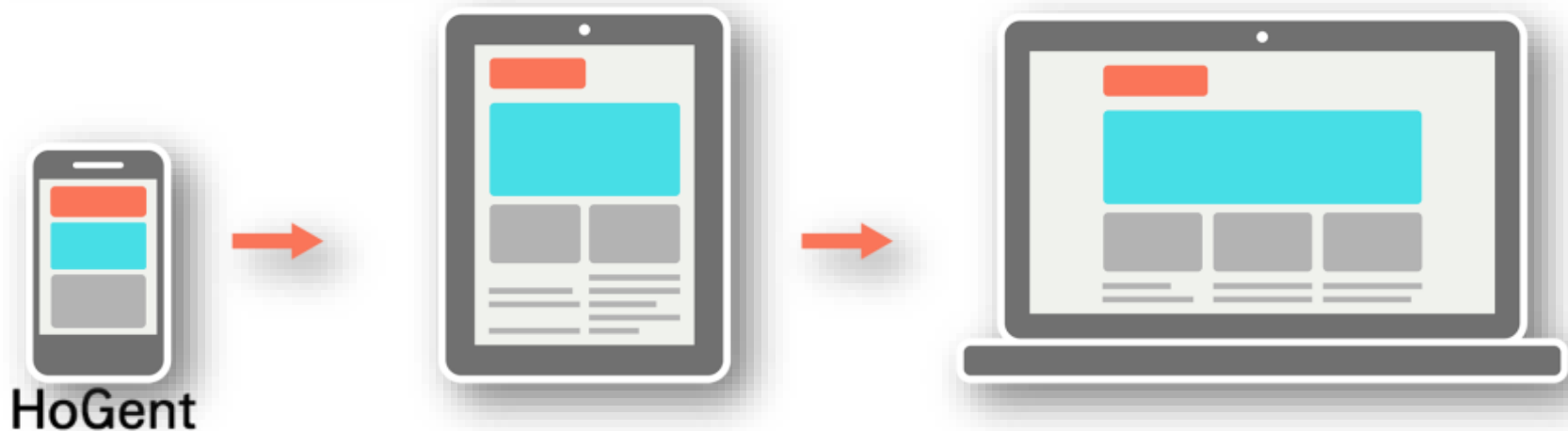
- ▶ De waarden (vaak in pixels) waarop de stijl wijzigt noemt men **breakpoints**.
- ▶ Bijvoorbeeld:
 - indien het scherm kleiner is dan 480 pixels, pas dan de 'smartphone' stijl toe.
 - indien het scherm groter is dan 480 pixels en kleiner dan 769 pixels, pas dan de 'tablet' stijl toe.
 - indien het scherm groter is dan 769 pixels, pas dan de 'desktop' stijl toe.
- ▶ Testen:
 - door op een desktop de grootte van je browser te wijzigen
 - extensie(s) van je browser
 - Chrome ontwikkeltools

RWD vs Mobile First Web Design



Responsive Web Design

Mobile First Web Design



RWD vs Mobile First Web Design

▶ Responsive Web Design

- **Graceful degradation**
- Ethan Marcotte in mei 2010
- “Een desktop-website past zich automatisch aan aan andere **context** (schermgrootte, oriëntatie, resoluties)”



▶ Mobile First Web Design

- **Progressive enhancement**
- Luke WROBLEWSKI
- “Sites must be *device agnostic*. What matters is *content*”
 1. start general + mobile styles
 2. extend for tablet
 3. extend for desktop



RWD vs Mobile First Web Design

- ▶ Wij werken steeds Mobile First!!

Mogelijke patronen

Mostly fluid

Column drop

Layout shifter

Tiny tweaks

Off canvas

Be **Mobile** Ready, First!



Meer informatie over deze patronen vind je op:

- <https://developers.google.com/web/fundamentals/design-and-ui/responsive/patterns/?hl=en>
- <http://bradfrost.github.io/this-is-responsive/patterns.html>

Belangrijke elementen

Responsive Web Design

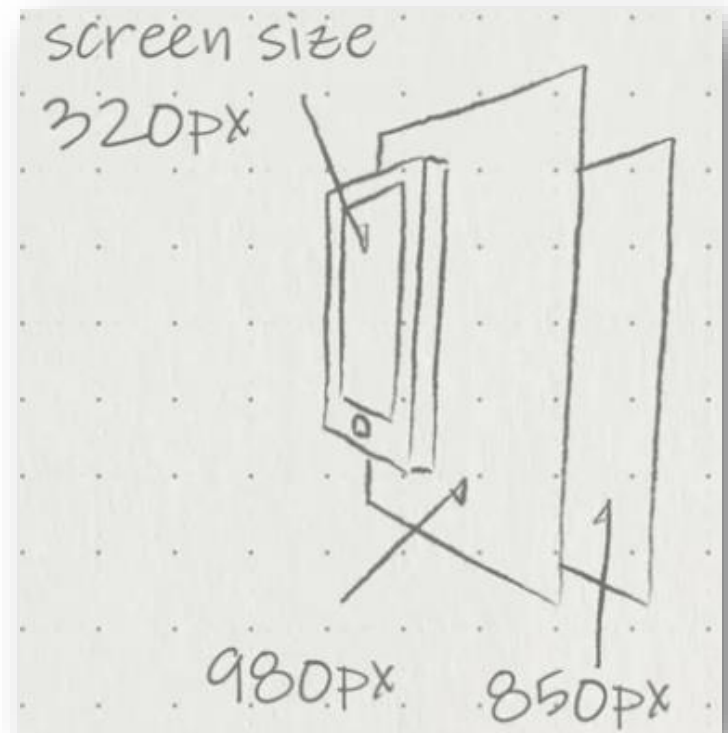
- ▶ In Responsive Web Design zijn volgende elementen belangrijk:
 - Viewport + Initial scale.
 - CSS media queries – verschillende stijlen voor verschillende schermgroottes.
 - Flexible grids.
 - Afbeeldingen.

Viewport

- ▶ Wat is de 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. Op mobiele toestellen is het niet mogelijk om het browservenster te wijzigen. De viewport is dan gelijk aan de grootte van het venster.

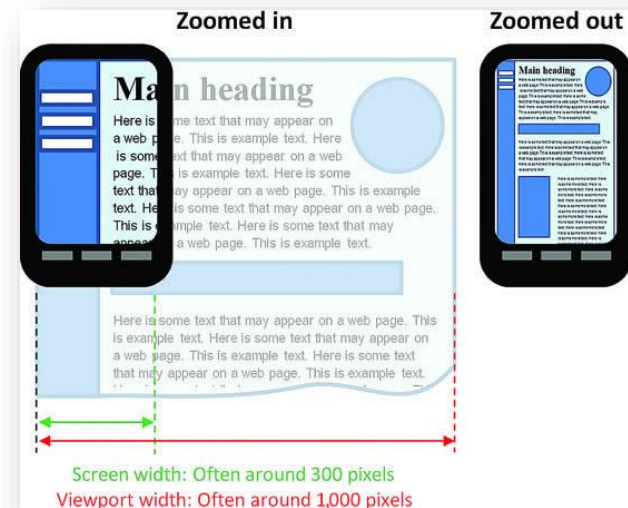
Viewport

- ▶ Viewports op mobiele toestellen zijn zo ontworpen dat ze groter zijn dan de resolutie van het scherm.
 - Mobile Safari heeft een viewport van 980px breed
 - Mobile Opera heeft een viewport van ca. 850px breed
- ↕
- Samsung Galaxy S4 Mini heeft een breedte van 540px
 - Apple iPhone 6 heeft een breedte van 750px



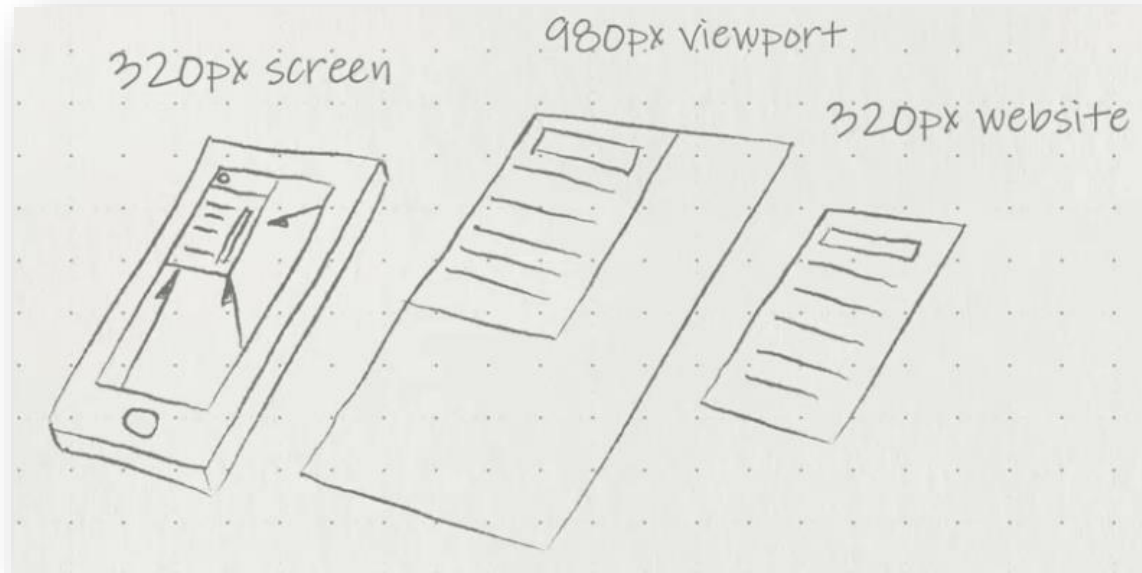
Viewport

- ▶ 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 je maar een klein gedeelte van de webpagina zien. We noemen dat keyhole browsing – te vergelijken met wanneer je door een sleutelgat naar een volledige ruimte kijkt.
- ▶ De webpagina wordt dus nu wel in zijn geheel getoond op het scherm, maar verkleind.
- ▶ Voorbeeld voor iPhone:
<http://www.paintcodeapp.com/news/ultimate-guide-to-iphone-resolutions>



Viewport

- ▶ Een eerste naïeve oplossing voor dit probleem (het verkleinen van de website naar de de dimensie van de screen size) zou kunnen zijn dat je bijvoorbeeld een website ontwerpt met een breedte van 320px.
- ▶ Maar als de viewport van de browser op het mobiele toestel bijvoorbeeld 980px is, zal je site ongeveer een derde van de ruimte innemen
- ▶ Dit werkt dus niet!



Viewport

- ▶ Hoe kunnen we er met andere woorden voor zorgen dat een webpagina die speciaal ontworpen is voor kleinere toestellen, niet meer geschaald wordt door toedoen van de viewport.
- ▶ Dit kunnen we bereiken door de afmetingen van de viewport gelijk te stellen aan de afmetingen van het toestel.



viewport werd
ingesteld

viewport werd
niet ingesteld

Hoe kunnen we de viewport instellen?

- ▶ via de viewport meta tag → ondersteund door alle browsers.

```
<html>                                     .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>
```

- ▶ via de @viewport CSS regel
(doorgaans de eerste regel in de CSS stylesheet) → niet ondersteund door alle browsers.

```
                                           .CSS
@viewport {
  width: device-width;
  zoom: 1;
}
```

Initial scale

- ▶ Initial scale bepaalt de zoom factor van de inhoud op het ogenblik dat de webpagina de eerste keer getoond wordt.
- ▶ Eens de pagina werd getoond, staat het de gebruiker vrij om in – en uit te zoomen op de webpagina.
- ▶ Als je de viewport instelt op device-width, dan stelt de browser automatisch de initial-scale in op 100%
- ▶ Je kan ook een minimum en een maximum instellen voor het schalen. Beide waarden gaan van 0.25 tot 10.

.html

```
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
      minimum-scale=1.0, maximum-scale=2.0, user-scalable=yes">
    <title> ... </title>
    <body> ... </body>
  </head>
</html>
```

Viewport – initial scale: voorbeeld.

- ▶ Stel in de HTML je viewport en initial scale in.
- ▶ Link naar de reset stylesheet en de responsive stylesheet (hier gebeurt de magic).

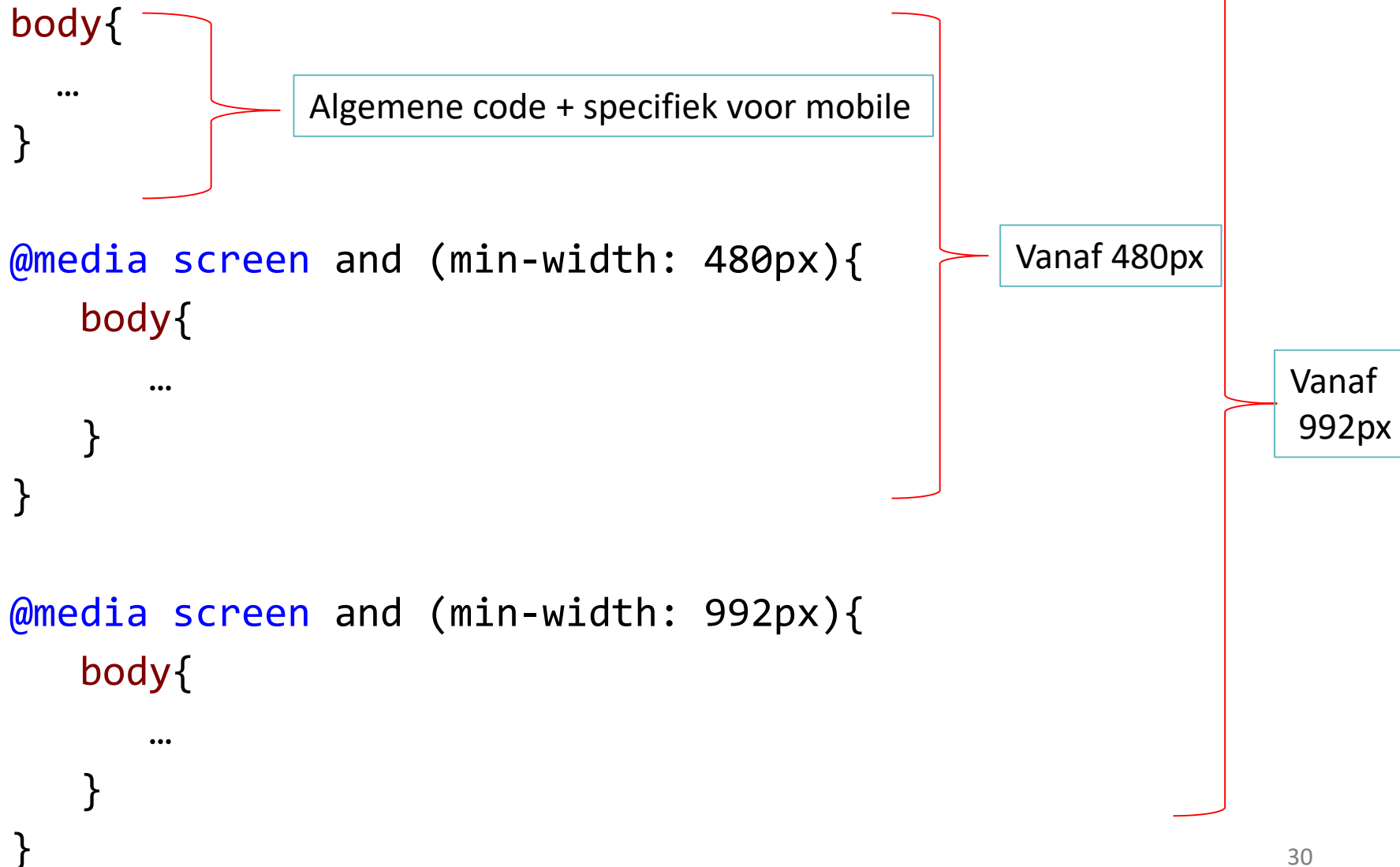
```
<head>
  <title>Mobile First Responsive</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=yes">
  <link href="css/reset_browser_styles.css" rel="stylesheet">
  <link href="css/responsive.css" rel="stylesheet">
</head>
```

Media Queries

- ▶ Het doel van media queries is de mogelijkheid om stijlregels toe te kennen, afhankelijk van de breedte en hoogte van het browservenster.
- ▶ Zo is het mogelijk om aangepaste stijlen te maken naargelang het type toestel: smartphone – tablet – desktop
- ▶ We starten met alle algemene stijlregels, en deze specifiek voor mobile
- ▶ Stap voor stap, breakpoint per breakpoint breiden we uit voor 'grotere' toestellen

Media Queries - Declaratie

- ▶ Embedded in 1 stylesheet: 1 request naar de server nodig



Media Queries – Breakpoints declaratie

▶ Aparte css per breakpoint

◦ HTML

```
<link href="css/base.css" rel="stylesheet">  
<link href="css/medium.css" rel="stylesheet" media="(min-width:480px)">  
<link href="css/large.css" rel="stylesheet" media="(min-width:992px)">
```

◦ CSS

- Maak een style sheet aan met deze drie directives in.
- Koppel deze aan je html pagina.

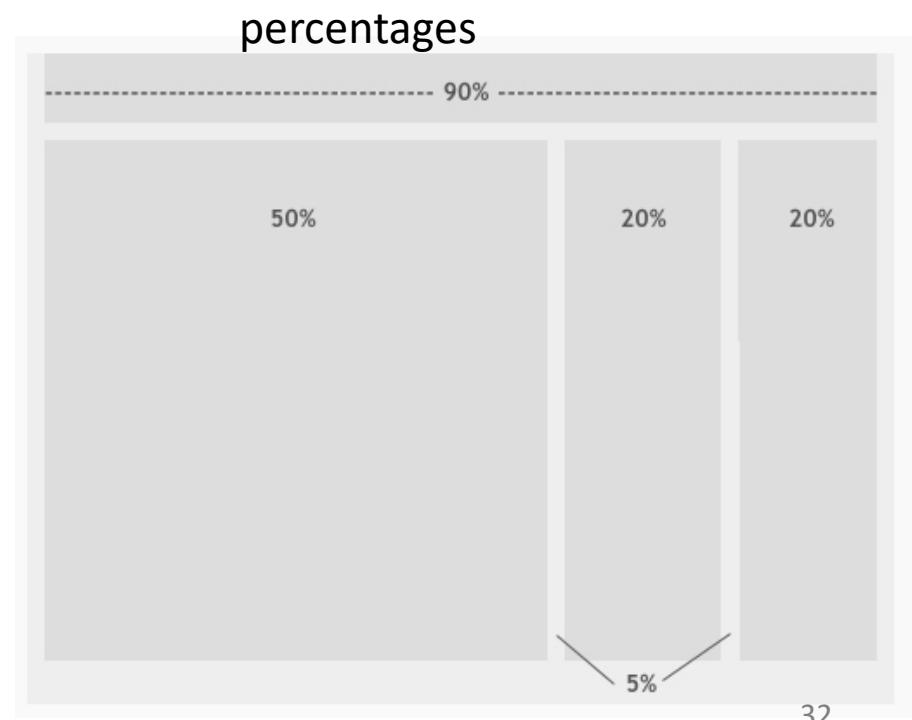
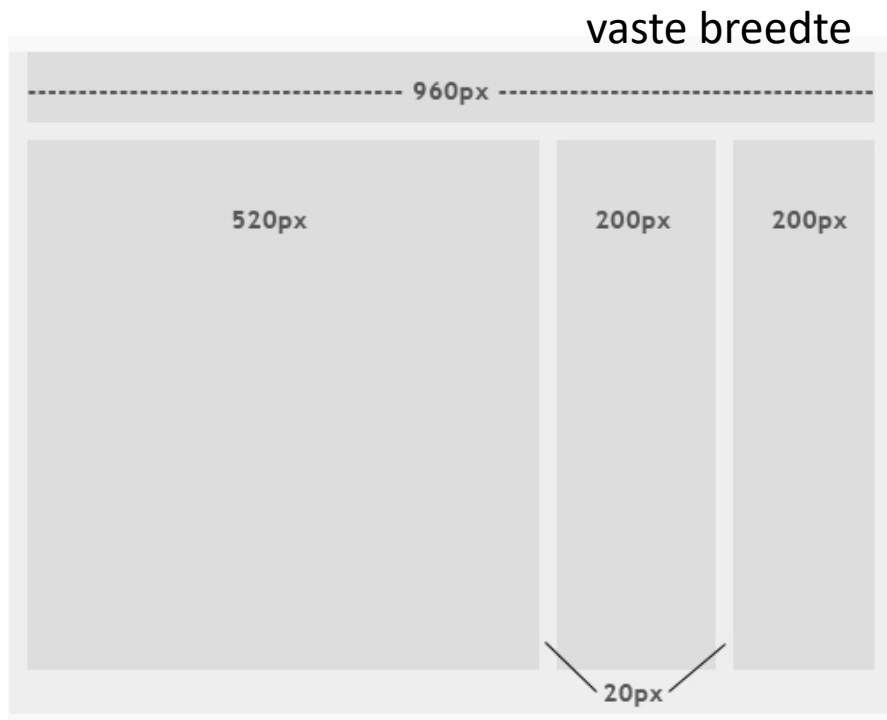
```
@import url(css/base.css); /* no media query, applies to all */
```

```
@import url(css/medium.css) (min-width:481px);
```

```
@import url(css/large.css) (min-width: 769px);
```

Liquid layout

- ▶ Maak geen gebruik van vaste breedtes, maar van percentages
- ▶ Voor desktop kan je eventueel een maximale breedte instellen in pixels met max-width.



Liquid layout

- ▶ De volgorde van de blokken in je HTML – pagina is van belang.
 - Bij Responsive Design zal men float eigenschappen vaak enkel toepassen op de tablet en desktop layout en niet op smartphones omdat die meestal maar 1 kolom hebben.
 - Zijdelingse kolommen zoals menu's of ads, zullen vaak op een mobile device onderaan staan. Dus de main content moet bovenaan staan in de HTML – pagina.

Liquid layout

- ▶ Door het toekennen van padding of border aan een element, kan het element groter worden dan het opgegeven, toebedeelde percentage => het element staat niet op de verwachte plaats (want het heeft eigenlijk te weinig ruimte). Maak daarom gebruik van `box-sizing: border-box` om dit te vermijden. Bij box-sizing wordt de padding en de border inbegrepen bij de width en de height.

```
opsomming html elementen{  
    box-sizing: border-box;  
}
```

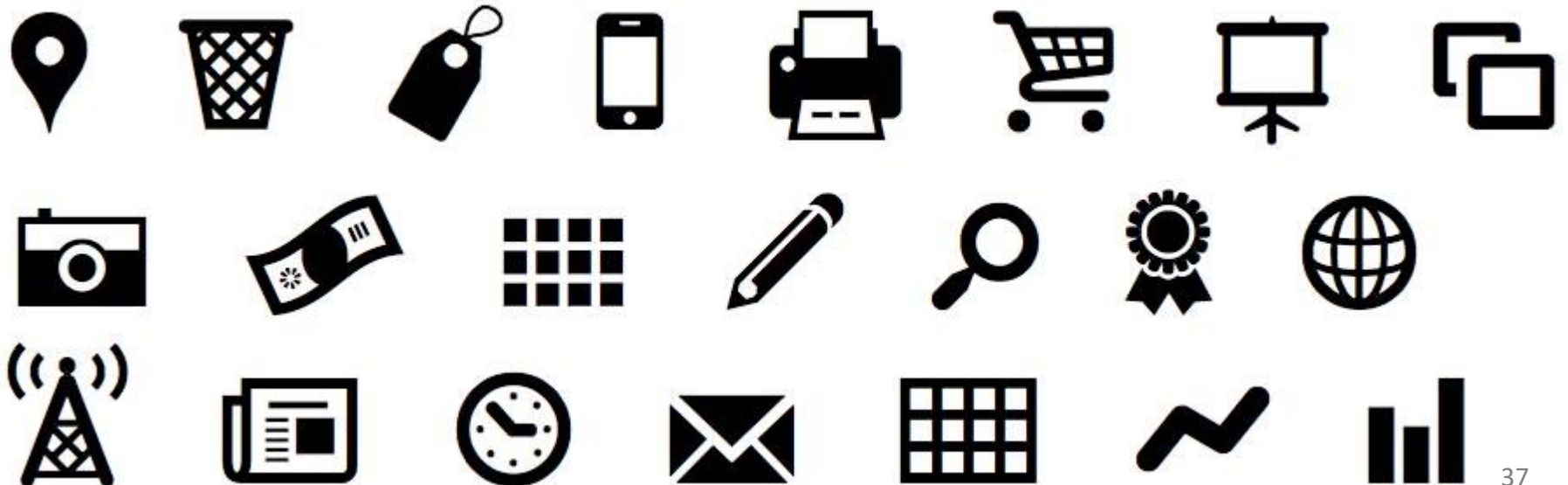
Optimalisatie afbeeldingen

Afbeeldingen: optimalisatie.

- ▶ Afbeeldingen nemen een groot deel van de bandbreedte in.
- ▶ Tracht het aantal HTTP requests (voor elk beeld één) en de grootte van de afbeeldingen te reduceren.
 1. Maak gebruik van HTML **special characters** voor kleine iconen, vb ★ voor ★
 2. Maak gebruik van **Sprites** ipv aparte icoontjes (1 beeldbestand dat volzit met icoontjes) – zie hoofdstuk 6.
 3. Gebruik **CSS ipv afbeeldingen**
 4. Verschillende afbeeldingen voor verschillende scherm breedtes – picture tag.

Optimalisatie: icon fonts

- ▶ Je kan gebruik maken van Icon fonts (bijvoorbeeld Modern Pictograms op fontsquirl.com). Deze icons schalen mee met de tekst. Het nadeel is wel dat op de plaats waar je een afbeelding wil zetten, er nu karakters moeten toegevoegd worden aan de html - code



Optimalisatie: CSS ipv afbeeldingen

- ▶ Dit werd ook in het gedeelte CSS-advanced besproken.

- Een voorbeeld van een afbeelding in een HTML bestand

.css

```
div#noImage {  
  color: white;  
  border-radius: 5px;  
  box-shadow: 5px 5px 4px 0 rgba(9,130,154,0.2);  
  background: linear-gradient(rgba(9, 130, 154, 1), rgba(9, 130, 154, 0.5)); }
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque sit amet augue eu magna scelerisque porta ut ut dolor. Nullam placerat egestas nisl sed sollicitudin. Fusce placerat, ipsum ac vestibulum porta, purus dolor mollis nunc, pharetra vehicula nulla nunc quis elit. Duis ornare fringilla dui non vehicula. In hac habitasse platea dictumst. Donec ipsum lectus, hendrerit malesuada sapien eget, venenatis tempus purus.

Afbeeldingen en media-queries.

- De picture tag

```
<picture>
  <source
    media="(min-width: 650px)" —————> media-query: als de min-width 650px is
    srcset="images/kitten-stretching.png">
  <source
    media="(min-width: 465px)" —————> media-query: als de min-width 465px is
    srcset="images/kitten-sitting.png">
   fallback, als picture niet wordt
    alt="a cute kitten">
</picture>
```

Appendix: optimalisatie Data URL's

Optimalisatie: Data URL's

- ▶ Data URL's of URI's is een schema dat je toelaat gegevens inline te gebruiken net alsof ze uit een extern bestand kwamen. Die gegevens zitten in een base-64-encoded string.
 - Voordelen :
 - er is geen extra Request voor een externe bron, dus sneller
 - je kan ervan gebruik maken in HTML en CSS
 - Nadelen:
 - herhaling van data voor gebruik van dezelfde images (icoontjes)
 - geen caching mogelijk in HTML, wel in CSS want stylesheets worden altijd gecached
 - aanmaak van de code vergt tijd en moet opnieuw gedaan worden bij elke wijziging
 - base64 code is ongeveer 33% groter dan oorspronkelijk data. Er is ook een grootte beperking afhankelijk van de browser

Optimalisatie: Data URL's

- ▶ Er zijn een aantal websites waar je een bestand kan converteren naar base64: dataurl.net/#dataurlmaker
- ▶ Data URI's worden ondersteund door de meeste browsers.

De syntax `data:[mimetype][;base64],data`

Optimalisatie: Data URL's

 **DATAURL.NET**
* Your one-stop HTML5 spot for all things Data URL *

AboutData URL MakerCSS OptimizerDownloads

Data URL Maker

Select or drag a file to get the Data URL: Geen bestand gekozen

realisatie.jpg

```
data:image/jpeg;base64,/9j/4AAQSkZJRgAB
BAQEAYABgAAD/4QEARXhpZgAATU0AKgAAAAB
QEaAAUAAAAAABAAASgEbAAUAAAAAABAAAUgEoAAM
AAAAABAAIAAAExAAIAAAASAAAAWodpAAQAAAABA
AAAbAAAAAABgAAAAQAAAGAAAAABUGFpbnQ
uTkVUIHYzLjUuMTAAAGShgAAAAAegAAAH4AA
AAAVU5JQ09ERQAAQwBSAEUAQQBUAE8AUgA6ACA
AZwBkAC0AagBwAGUAZwAgAHYAMQAUADAIAAoA
HUAcwBpAG4AZwAgAEkASgBHACAASgBQAEUARwA
gAHYANGAyACKALAAgAHEAdQBhAGwAAQ80AHkAI
AA9ACAAOAAwAAR/2wBDAAEBAQEBAQEBAQEBAQE
BAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBA
QEBAQEBAQEBAQEBAQEBAQEBAQH/2wBDAQE
BAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBA
QEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQE
BAQEBAQH/wAARCABzAHMDASIAAhEBAxEB/8QAH
```

Data URL Size: 13023 bytes
Original size: 9748 bytes

Image preview



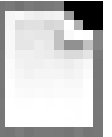
Optimalisatie: Data URL's

► Afbeeldingen: Data URI's

- Een voorbeeld van een afbeelding in een HTML bestand

```
<html>
  <head> ... </head>
  <body>
    
  </body>
</html>
```


.html



- Een voorbeeld van een afbeelding in een CSS bestand

```
.gestreept {
  height: 100px;
  border: 1px solid gray;
  padding: 10px;
  background-image:
url("data:image/gif;base64,R0lGODlhAwADAIAAAP///8zMzCH5BAAAAAAAAALAAAAADAAMAAAIEBHIJ
BQA7");
}
```

.css



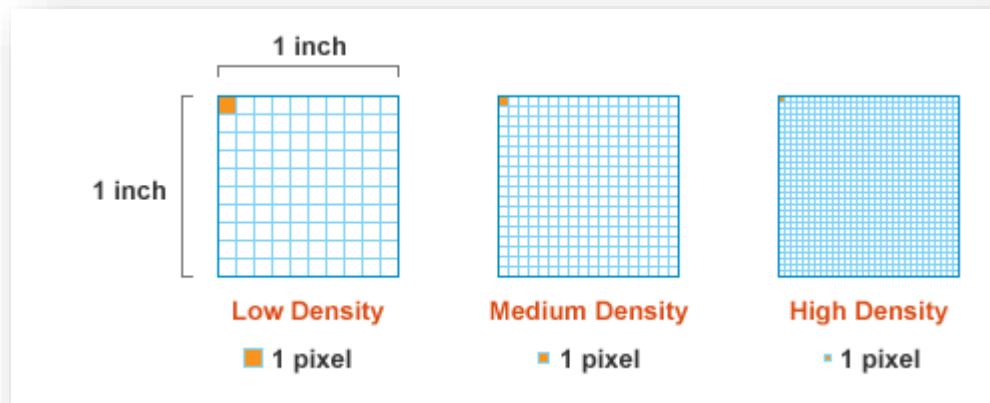
Appendix: Screen density

Wat is screen density?

- ▶ Bekijk de specificaties van de iPad2 en de iPad3 hiernaast

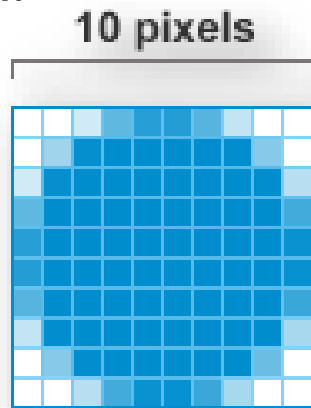
iPad 2	iOS 5.0 – 8.0	Mobile Safari	9.7" (25 cm)	1024 × 768 px	132ppi
iPad 3	iOS 5.1 – 8.0	Mobile Safari	9.7" (25 cm)	2048 × 1536 px	264ppi

- ▶ De iPad2 en de iPad3 hebben dezelfde schermgrootte (9.7"), maar de resolutie is verschillend. De iPad2 heeft een lagere **screen density** (= dichtheid van de pixels) dan de iPad3.

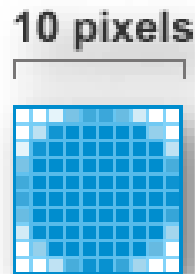


Hardware pixels

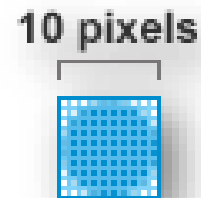
- ▶ Een hardware pixel is het kleinste punt dat een scherm kan laten zien. We spreken over DPI = dots per inch
- ▶ Als we zouden gebruik maken van hardware pixels, dan zou de afbeelding er op een scherm met low density veel groter uitzien dan op een scherm met high density:
Verschillende weergave van een afbeelding van 10px * 10px.



Low Density



Medium Density



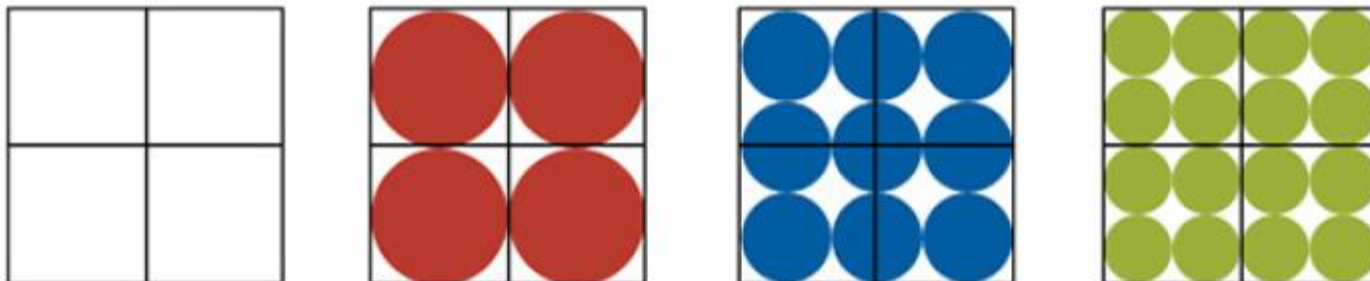
High Density

Hardware pixels

- ▶ Om dit te voorkomen werden CSS pixels (ook Density-Independent Pixels (DIP) genoemd) geïntroduceerd.
- ▶ Er wordt dus een schaafactor toegepast op de elementen. We noemen dit de **device-pixel-ratio**, dit is de verhouding tussen de CSS pixels en de hardware pixels.

Reference pixels = CSS pixels

- ▶ Zolang de DPI exact overeenkomen met de CSS pixels is er geen probleem bij het voorstellen van de afbeeldingen
- ▶ Afbeelding van $10\text{px} * 10\text{px}$ is
 - In het tweede van de onderstaande gevallen is de verhouding van de fysische pixels tot de CSS pixels gelijk aan 1:1
 - In het derde van de onderstaande gevallen is de verhouding van de fysische pixels tot de CSS pixels gelijk aan 1.5:1 => Er zullen 1.5^2 zoveel pixels gebruikt worden (bvb Android scherm)
 - In het derde van de onderstaande gevallen is de verhouding van de fysische pixels tot de CSS pixels gelijk aan 2:1 => Er zullen 2^2 zoveel pixels gebruikt worden (bvb Retina (Apple) scherm)



Hoe rekening houden met screen density?

- ▶ Je wil een afbeelding toevoegen aan je webpagina met width = height = 100px.
- ▶ Voor een scherm met high density en device-pixel-ratio = 2, maak je een hoge resolutie afbeelding van 200px * 200px die je vervolgens resized (in CSS) naar 100px * 100px. De extra pixels zullen op die manier gemapt worden naar hardware pixels.
- ▶ Het gebruik van deze afbeelding op low density schermen kan, maar de bestanden zijn dan wel te groot.
- ▶ Omgekeerd als je de kleine afbeelding gebruikt is de kwaliteit slecht op high density devices.
- ▶ <https://bjango.com/articles/min-device-pixel-ratio/> voor overzicht van DPR.

Hoe rekening houden met screen density?

- ▶ Momenteel zijn er een aantal oplossingen voor het probleem, maar worden nog niet door alle browsers ondersteund.
- ▶ Voor background-images kan je gebruik maken van media queries om voor verschillende device-pixel-ratio's de geschikte background-image te gebruiken (zie hiervoor).
- ▶ Voor HTML kan je gebruik maken van srcset attribuut of picture tag (zie verder).
- ▶ Meer op <https://developers.google.com/web/fundamentals/media/images/>
.css

Hoe rekening houden met screen density?

- ▶ Art direction in responsive images – retina displays
 - Gebruik meerdere beelden: **picture** element



The result of using only one image that is scaled up or down based on the viewport width.



Use several different images to more appropriately fill the browser viewport.

Hoe rekening houden met screen density?

- ▶ Art direction in responsive images en retina display
 - De picture tag

```
<picture>
  <source
    media="(min-width: 650px)" —————> media-query: als de min-width 650px is
    srcset="images/kitten-stretching.png,
            images/kitten-stretching-1.5x.png 1.5x, ———— DPR=1.5 : 1.5 fysische pixels =
            images/kitten-stretching-2x.png 2x">          1 css pixel
  <source
    media="(min-width: 465px)" —————> media-query: als de min-width 465px is
    srcset="images/kitten-sitting.png,
            images/kitten-sitting-1.5x.png 1.5x
            images/kitten-sitting-2x.png 2x"> —————> DPR=2 : 2 fysische pixels = 1 css
  <img —————> fallback, als picture niet wordt ondersteund pixel
    src="images/kitten-curled.png"
    srcset="images/kitten-curled-1.5x.png 1.5x,
            images/kitten-curled-2x.png 2x"
    alt="a cute kitten">
</picture>
```

Hoe rekening houden met screen density?

► Javascript replacement

- Met JavaScript kan je de device-pixel-ratio opvragen en vervolgens de afbeelding met hoge resolutie opvragen in plaats van de afbeelding met een lagere resolutie
- We noemen dit **adaptive images**. Het voordeel is dat je enkel het juiste beeld bestand laadt , maar de download start pas als pagina geladen is

.js

```
function getDevicePixelRatio() {  
    if (window.devicePixelRatio === undefined) return 1;  
    return window.devicePixelRatio;  
}
```

Device Mode & Mobile Emulation

- ▶ Maak gebruik van Google Chrome Device Mode & Mobile Emulation
- ▶ <https://developer.chrome.com/devtools/docs/device-mode#gc-pagecontent>
 - Open Developer Tools
 - Klik op  (links van Elements)
 - Je kan bovenaan zelf het gewenste toestel instellen