

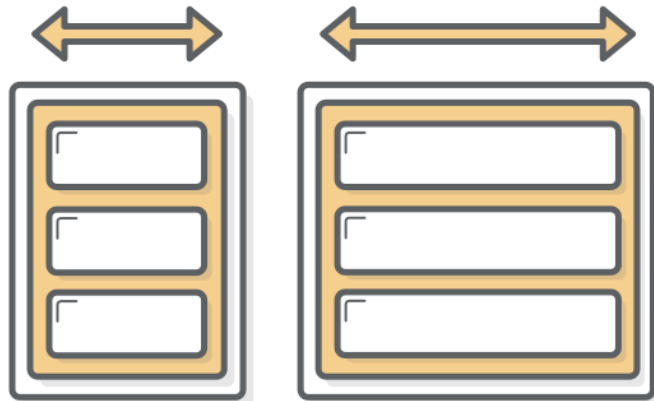
LAYOUT DELLE PAGINE

Layout delle pagine

Nel web design, il layout determina come i contenuti si distribuiscono nella pagina.

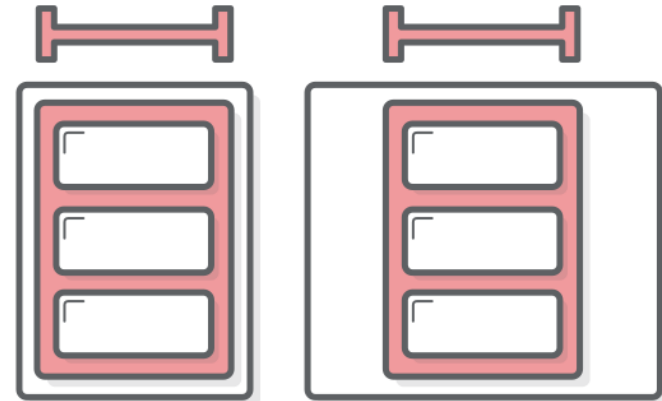
- **Fluid**

- proporzionale alla larghezza del browser

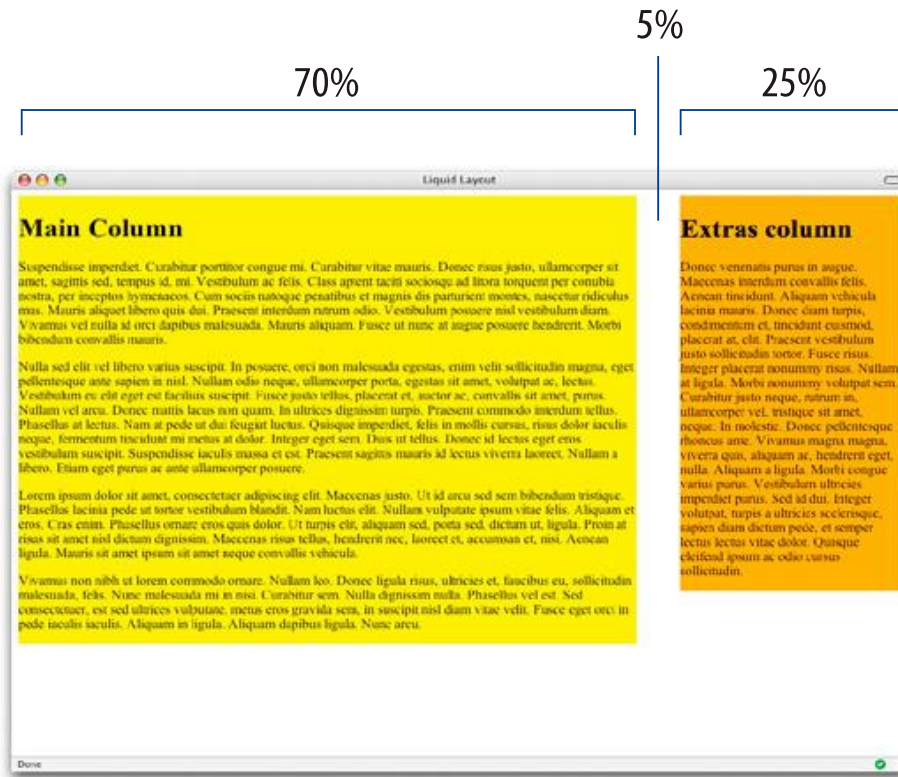


- **Fixed**

- larghezza indipendente dalla finestra del browser



Fluid layout



```
.wrapper {
  display: flex;
  width: 100%;
}

.main {
  width: 70%;
  background-color: yellow;
  margin-right: 5%;
}

.extras {
  width: 25%;
  background-color: orange;
}
```

Discussione Layout Fluido

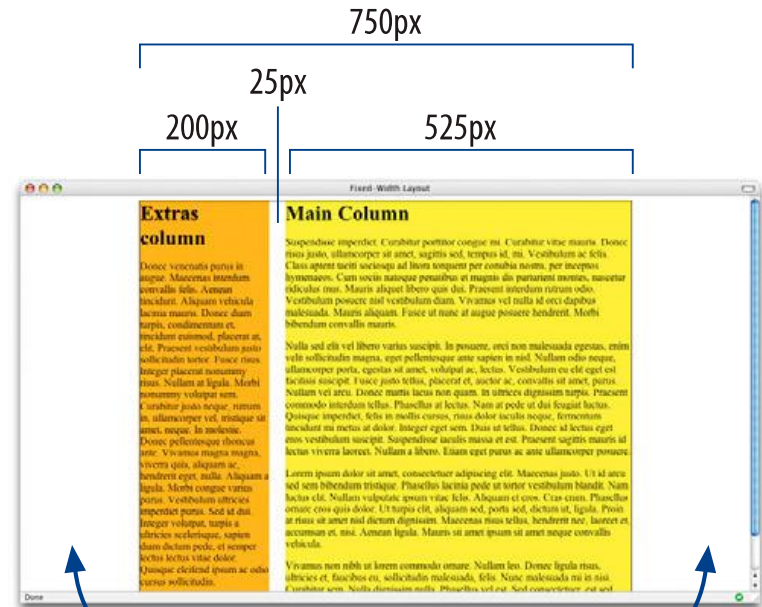
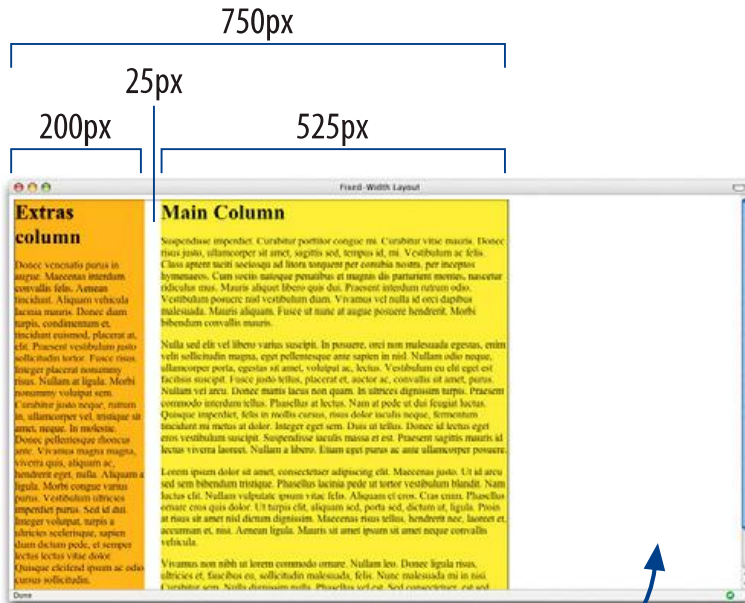
- **Vantaggi**

- Sembra più vicino alla natura dei browser
- Elimina spazi potenzialmente vuoti
- L'utente può controllare la pagina
- Non ho scrollbar orizzontali

- **Svantaggi**

- Se il monitor è grande ho righe troppo lunghe
 - limitare la dimensione
- Non è facile predire la posizione degli elementi e l'effetto grafico finale
- Problemi nei browser piccoli
- I conti sono un poco più complicati

Fixed layouts



Extra space split on left and right sides

```
.wrapper {
  width: 750px;
  display: flex;
  justify-content: flex-start;
  margin-left: auto;
}

.extras {
  width: 200px;
  margin-right: 25px;
  background-color: orange;
}

.main {
  width: 525px;
  background-color: yellow;
}
```

```
.wrapper {
  width: 750px;
  display: flex;
  justify-content: flex-start;
  margin: 0 auto;
}

.extras {
  width: 200px;
  margin-right: 25px;
  background-color: orange;
}

.main {
  width: 525px;
  background-color: yellow;
}
```

Discussione Layout Fisso

- **Vantaggi**

- Controllo del numero delle righe
- è semplice da realizzare
- è una visualizzazione comune per i desktop

- **Svantaggi**

- se la finestra è piccola la parte destra viene coperta
- se lo schermo è grande ho spazi a dx ed sx da gestire
- se ho carattere grandi ho problemi di righe corte
- la pagina non è controllabile dall'utente

Confronto fixed vs fluid

Caratteristica

Adattabilità



Controllo grafico



Usabilità mobile



Complessità tecnica



SITI RESPONSIVE

Responsive Web Design

- Fornire layout diversi per schermi diversi
 - Il layout si adatta in automatico alla dimensione dello schermo
 - Stessi contenuti, ma presentazione ottimizzata per ogni dispositivo
 - Un solo file HTML, ma con CSS variabili grazie alle media query



Come fare un sito repsonsive

1. Controllare il viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

2. Gestire Layout con media queries

```
@media (max-width: 768px) {  
  .sidebar { display: none; }  
}
```

3. Usare dei Media “fluidi”

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

<META> E VIEWPORT

Viewport e dispositivi mobili

- Finestra virtuale in cui viene disegnato il sito
- I dispositivi mobili **barano**
 - dichiarano 980px
 - poi adattano ai veri pixel dello schermo

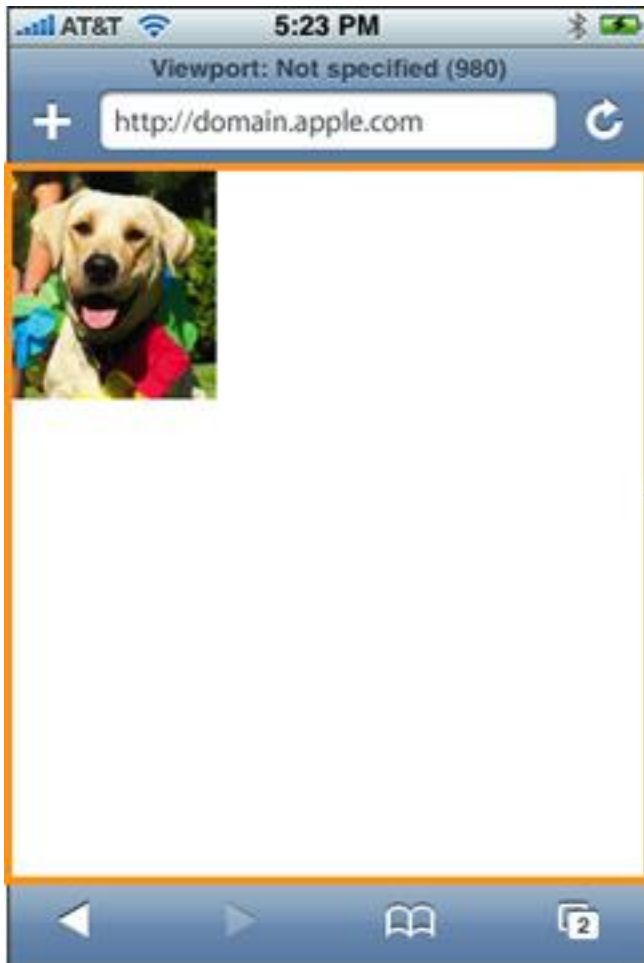


Dimensioni reali e dichiarate



Ridimensionamento dei contenuti

- Immagine 320 x 356 px



Viewp
default



Viewport
width = 320 pixels
scale = 1.0

Controllo viewport

- Tag **meta** da inserire nell'head html

<meta name="viewport" content="width=device-width, initial-scale=1">

- Opzioni

- width, height -> device-width o device-height oppure px
- user-scalable -> no, yes
- initial-scale -> 1
- minimum-scale -> 1 non scala
- maximum-scale -> 1 non scala

- <https://developer.apple.com/library/ios/documentation/AppleApplications/Reference/SafariWebContent/UsingtheViewport/UsingtheViewport.html>



With the viewport meta tag

Programmazione WEB

MEDIA QUERY

Media query

- Definisco degli stili per determinati media e device

```
@media not|only mediatype and (media feature) {  
    CSS-Code;  
}
```

```
<link rel="stylesheet" media="mediatype and|not|only (media feature)" href="mystylesheet.css">
```

Media Type

– all, print, screen, speech

```
@media screen {  
  p {  
    font-family: verdana, sans-serif;  
    font-size: 17px;  
  }  
}
```

```
@media print {  
  p {  
    font-family: georgia, serif;  
    font-size: 14px;  
    color: blue;  
  }  
}
```

```
@media screen, print { ... }
```

Media Features

```
@media (orientation: landscape) { ... }
```

```
@media (max-width: 12450px) { ... }
```

```
@media (color) { ... }
```

Media features

Feature	Description
width	The width of the display area (viewport).
height	The height of the display area (viewport).
device-width	The width of the devices rendering surface (the whole screen).
device-height	The height of the devices rendering surface (the whole screen).
orientation	Whether the device is in portrait or landscape orientation. (Does not accept min-/max- prefixes.)
aspect-ratio	Ratio of the viewport's width divided by height (width/height).
device-aspect-ratio	Ratio of the whole screen's (rendering surface) width to height.
color	The bit depth of the display; for example, color: 8 tests for whether the device has at least 8-bit color.
color-index	The number of colors in the color lookup table.
monochrome	The number of bits per pixel in a monochrome device.
resolution	The density of pixels in the device. This is increasingly relevant for detecting high-resolution displays.
scan	Whether a tv media type uses progressive or interlace scanning. (Does not accept min-/max- prefixes.)
grid	Whether the device uses a grid-based display, such as a fixed-width font. (Does not accept min-/max- prefixes.)

https://www.w3schools.com/cssref/css3_pr_mediaquery.asp

Media Query complesse

- Combinare le features

- @media (min-width: 30em) **and** (orientation: landscape) { ... }
- @media screen **and** (min-width: 30em) **and** (orientation: landscape) { ... }
- @media print **and** (min-resolution: 300dpi) { ...styles here... }

- Combinare le query

- @media (min-height: 680px), screen and (orientation: portrait) { ... }
-

Esempi MQ

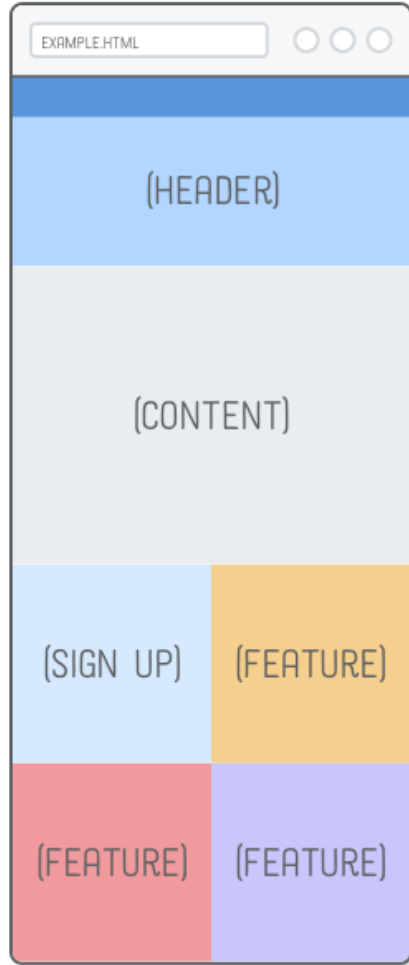
- https://www.w3schools.com/csS/css3_mediaqueries.asp
- https://www.w3schools.com/csS/css3_mediaqueries_ex.asp
- https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_media2

LAYOUT RESPONSIVE

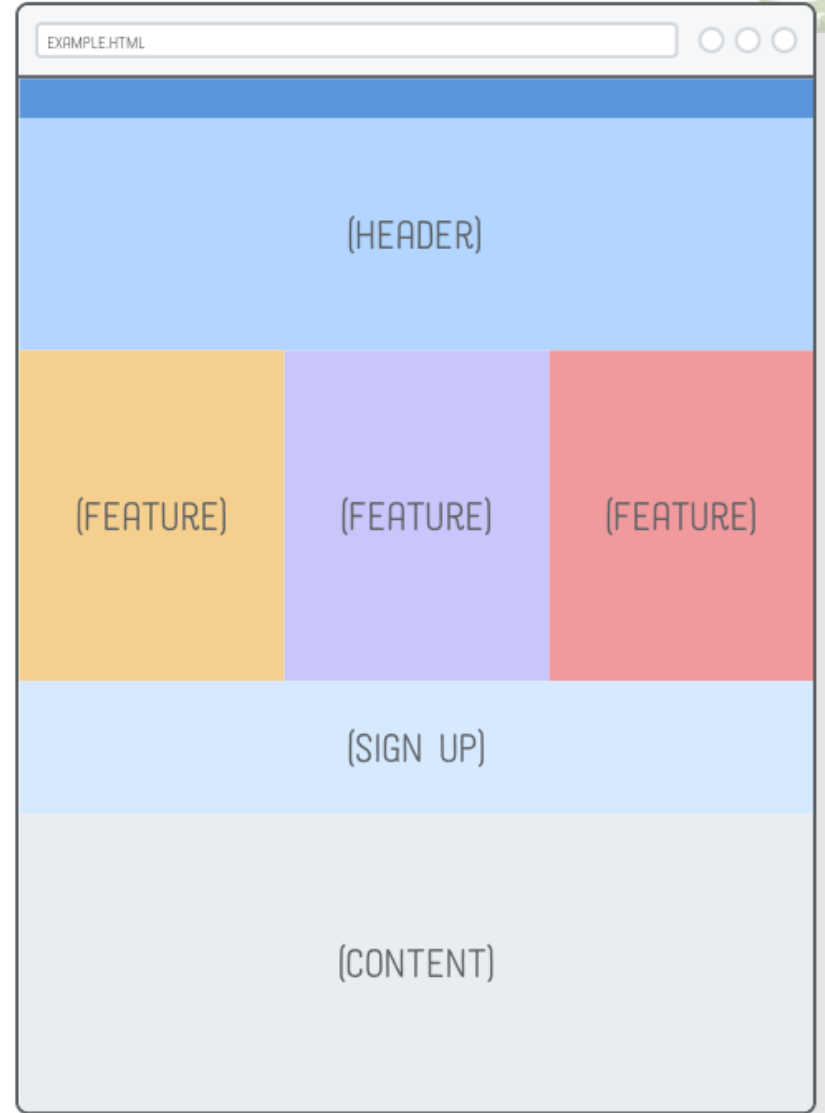
MOBILE



TABLET



DESKTOP



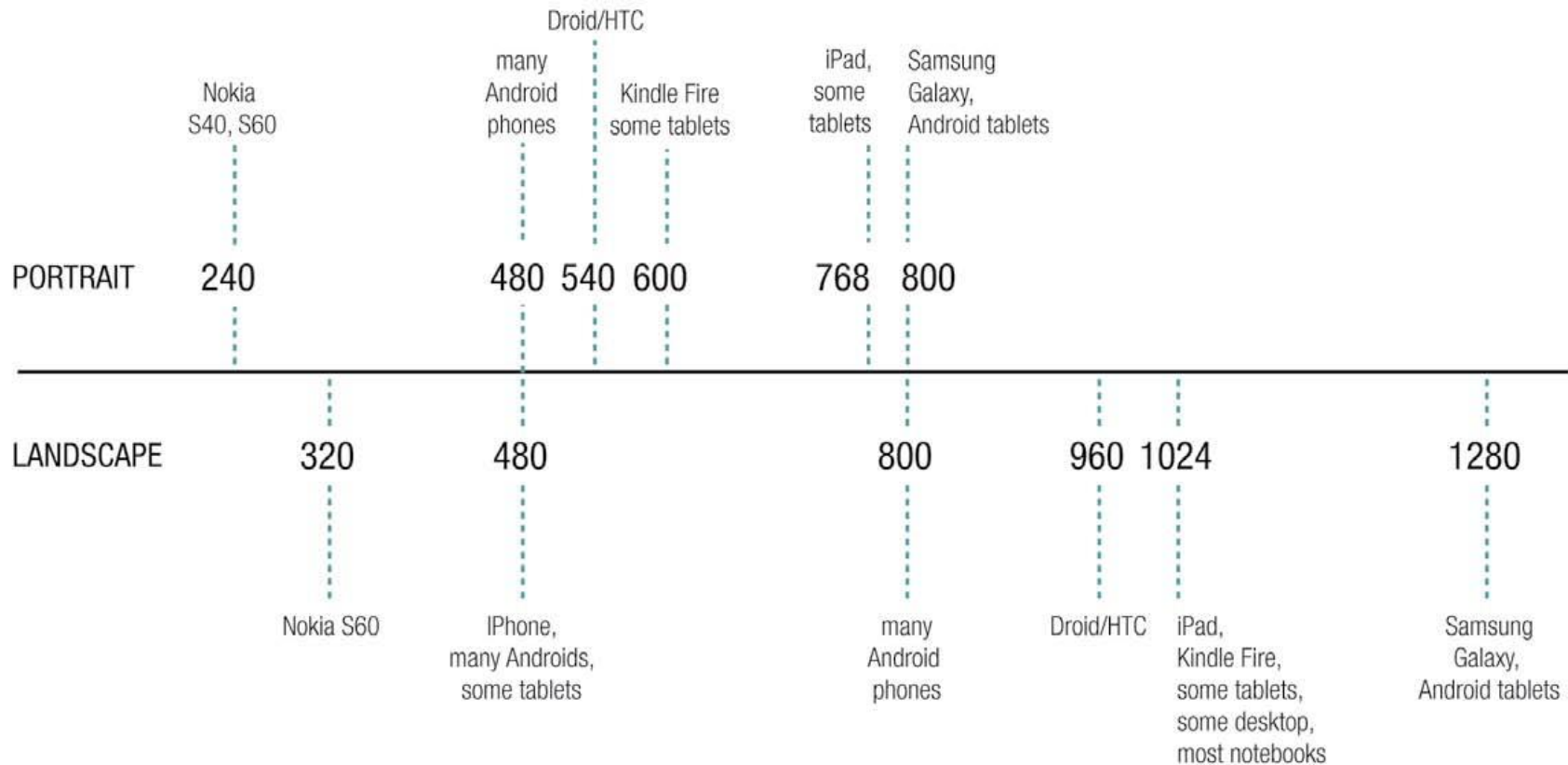
Breakpoints

- I breakpoint sono punti su una scala ideale di larghezza del viewport in cui si verifica una qualche modifica al layout della pagina
- I breakpoint si definiscono con valori numerici nelle media query

```
@media only screen and (max-width: 600px) {  
  body{  
    background-color: ■ red;  
  }  
}
```

```
@media only screen and (min-width: 601px) and (max-width: 900px) {  
  body{  
    background-color: ■ skyblue;  
  }  
}
```

Device e Breakpoint



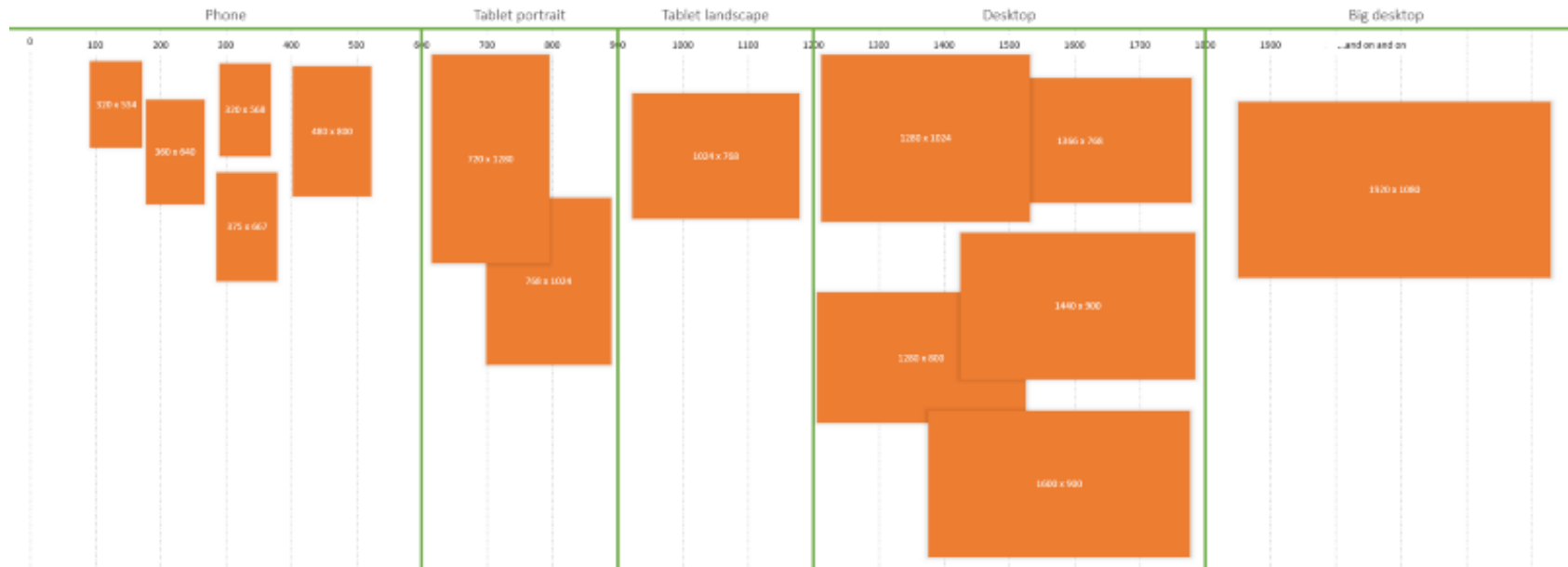
Range con MQ

Min-Width

Applies *at or above* a certain width

Max-Width

Applies *at or below* a certain width



Strategie per le MQ

- Exclusive

- Ogni range di larghezza usa delle regole esclusive

- Override

- Parto da regole per un range e poi le riscrivo per gli altri range
 - Mobile First
 - Parto dagli schermi piccoli (**min-width**)
 - Desktop First
 - Parto dagli schermi grandi (**max-width**)

“Mobile first” media queries

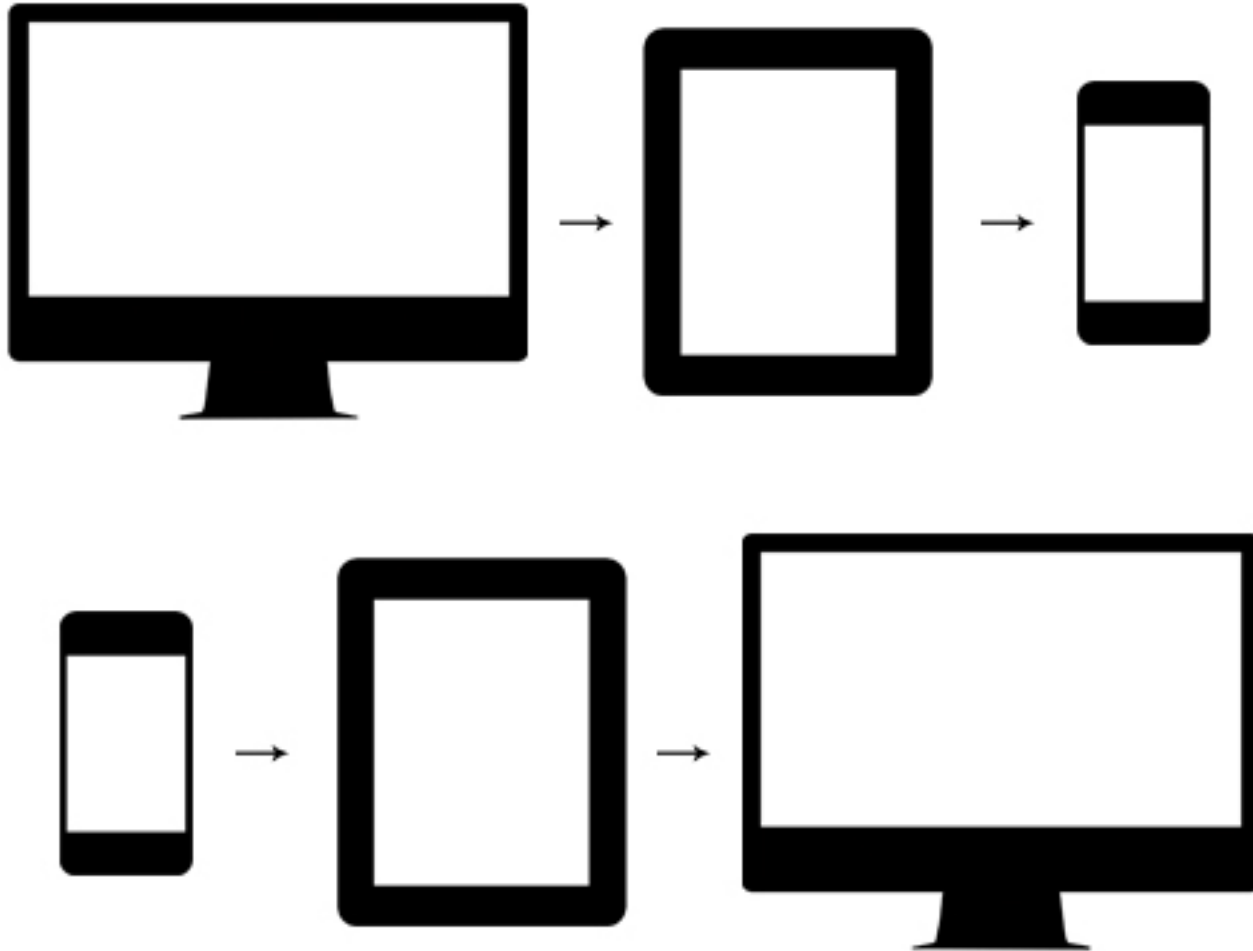
- Si parte definendo prima gli stili per i device mobili
 - si usa **min-width**
- Si aggiungono stili e/o sovrascrivono proprietà per quelli più larghi
- La maggior parte dei framework sono mobile first

```
@media (min-width: 768px) {
  .container {
    width: 750px;
  }
}
```

```
@media (min-width: 992px) {
  .container {
    width: 970px;
  }
}
```

```
@media (min-width: 1200px) {
  .container {
    width: 1170px;
  }
}
```

Graceful Degradation vs. Progressive Enhancement



Esempi

Override

```
@media (min-width: 400px) {  
  html { background: red; }  
}  
  
@media (min-width: 600px) {  
  html { background: green; }  
}  
  
@media (min-width: 800px) {  
  html { background: blue; }  
}
```

Exclusive

```
@media (max-width: 400px) {  
  html { background: red; }  
}  
  
@media (min-width: 401px) and (max-width: 800px) {  
  html { background: green; }  
}  
  
@media (min-width: 801px) {  
  html { background: blue; }  
}
```


Esempi

Desktop first
max-width

```
html { background: red; }  
  
@media (max-width: 600px) {  
    html { background: green; }  
}
```

Mobile First
min-width

```
html { background: red; }  
  
@media (min-width: 600px) {  
    html { background: green; }  
}
```

ALTRE TECNICHE RESPONSIVE

https://www.w3schools.com/css/css_rwd_images.asp

https://www.w3schools.com/css/css_rwd_videos.asp

Media flessibili

- **Controllo della larghezza**

- width: L'immagine si espande al 100%
- max-width: Se il container è più grande della dimensione in pixel dell'immagine si ferma il resize
- non posso impostare width ed height

```
img {  
    max-width: 100%;  
}
```

- **La stessa tecnica si può usare per video, object o altri elementi**

- **Problema:**

- immagini grandi occupano molta banda
- il sito risulta lento

https://www.w3schools.com/css/css_rwd_images.asp

Background-size

- contain



- cover



https://www.w3schools.com/css/css_rwd_images.asp

Programmazione WEB
https://www.w3schools.com/css/tryit.asp?file=name=tryresponsive_image_background3

Ottimizzazione banda

```
/* For devices smaller than 400px: */  
body {  
    background-image: url('img_smallflower.jpg');  
}
```



```
/* For devices 400px and larger: */  
@media only screen and (min-device-width: 400px) {  
    body {  
        background-image: url('img_flowers.jpg');  
    }  
}
```



<source> srcset Attribute

- Posso specificare immagini diverse a seconda della MQ

```
<picture>  
  <source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">  
  <source media="(min-width: 465px)" srcset="img_white_flower.jpg">  
    
</picture>
```

https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_picture