

MEDIA QUERY

Cosa sono le Media Queries?

Le Media queries sono quella cosa che rende i siti responsive, cioè adattabili al device con cui si visita il sito.

Fino ad ora abbiamo visto regole CSS che si applicano indipendentemente dalle dimensioni dello schermo.

Le Media queries permettono di applicare diversi stili a seconda del device utilizzato, solitamente in base alla larghezza del viewport: organizza il sito in due colonne, ma se lo schermo è più largo di x allora usa tre colonne.

Una storia abbastanza recente

Anche se CSS2 supportava i media type (come screen o print), è solo con CSS3 che si è in grado di fare query basate sulle caratteristiche del device, come la larghezza del viewport. Questo perché il Responsive Design inizia a diffondersi nel 2010!

Come applicare le Media query in un CSS?

```
body { background-color: green; }  
@media only screen and (min-width: 40em) {  
  body { background-color: blue; }  
}
```

La prima riga imposta il colore del background a verde, la seconda invece dice che SE il viewport è almeno largo 40 em allora il body deve avere un background blu.

Alternativa: applicare un CSS specifico in base alla larghezza

<head>

<link rel="stylesheet" href="styles/mainstyles.css">

**<link rel="stylesheet" href="styles/widerscreen.css"
media="only screen and (min-width: 40em)">**

</head>

Oppure inline

Proprio come abbiamo visto per i CSS

Scenari d'uso comuni

- 1) Colonne: una con device piccoli, poi due crescendo fino a tre o più se c'è spazio sufficiente**
- 2) Navigazione: mostra un drop-down menu su schermi piccoli e menu completo con tutte le voci su device più larghi**
- 3) Tipografia: cambia la dimensione del carattere o di una colonna per migliorare la leggibilità**
- 4) Immagini: mostra differenti dimensioni o ritagli dell'immagine a seconda dello spazio a disposizione**

Compatibilità con i vari browser

Tendenzialmente supportato da tutti i browser moderni.

Se è richiesta la compatibilità con versioni precedenti a Internet Explorer 9 ci sono due possibilità:

- 1) usare polyfill: aggiunge codice javascript per "implementare" le funzionalità non presenti nel browser**
 - 2) aggiungere commenti condizionali includendo altro css**
- ```
<!--[if (lt IE 9)]> <link rel="stylesheet" href="ie-styles.css"> <![endif]-- >
```

## Approccio per il Responsive Design

- 1) Progressive enhancement: si parte dalle basi e si aggiunge fino a che il browser/device lo permette, in quest'ordine: HTML, CSS, Javascript. Si parte dalle basi per supportare tutti**
- 2) Griglie: creare ordine dal disordine**
- 3) Colonne: solitamente 12, i vari box usano 1, 2, 3, 4, 6 colonne**
- 4) Mobile First: prima si fa la versione mobile (aiuta a porre l'attenzione ai contenuti importanti), poi è più facile allargare ed aggiungere stili per abbellire gli stili in dispositivi larghi**



## **Rendiamo la pagina del profilo personale responsive con un layout a doppia colonna**

- 1) per evitare testi troppo "larghi" e dunque difficilmente leggibili, proviamo ad aggiungere la seconda colonna quando troviamo circa 75 caratteri in una riga, corrispondenti a circa 36em per il profilo che abbiamo creato.**
- 2) ricordiamoci che bisogna usare float e clear per stabilire inizio e fine della doppia colonna!**
- 3) useremo una grid per definire le dimensioni delle colonne**

## Finalmente il codice!

```
@media only screen and (min-width: 36em) {
 article { width: 67%; float: left; margin-right: 0; }
 aside { width: 21%; float: right; margin-left: 0; }
 footer { clear: both; }
}
```

**ma non siamo ancora soddisfatti al 100%...**

**Con schermi veramente larghi, la lettura è ancora difficoltosa**

**Aggiungiamo una dimensione massima alla larghezza:**

**1) per semplicità, nell'html inseriamo un `<div id="fullpage">`**

**dentro al tag body che racchiude tutte le sezioni della pagina**

**2) impostiamo una regola css che fissa la lunghezza massima a**

**63em e centra la pagina:**

**`#fullpage { max-width: 63em; margin: auto; }`**

**Pagina di esempio da usare come base per personalizzazioni**

**[https://github.com/ldalzovo/corso\\_analista\\_programmatore\\_mobile/tree/master/o3\\_media\\_query](https://github.com/ldalzovo/corso_analista_programmatore_mobile/tree/master/o3_media_query)**

## **Caratteristiche**

- abbiamo sovrascritto alcune regole css per una diversa risoluzione, è molto semplice capire quale regola si applica**
- è possibile studiare diverse risoluzioni per adattarsi per esempio a: smartphone, tablet, desktop**

## Esercizio

**Provare a creare un nuovo stile che:**

- gestisce tre colonne**
- ha tre diversi layout in base alla risoluzione del viewport**

## Salviamo le modifiche sul cloud

- applicate lo stile modificato alla vostra pagina**
- salvate pagina e stile online**

## Riepilogo

- **Cosa sono le Media Query**
- **Come rendere Responsive una pagina HTML**
- **Come organizzare i contenuti in base allo schermo**

## Prossimo passo

- **Aggiungiamo le immagini!**

**GRAZIE**  
**per l'attenzione!**