

**Responsive**

Il **responsive web design** (RWD) è l'approccio volto a rendere le pagine Web ben visualizzate su schermi di tutte le dimensioni e risoluzioni, garantendo al tempo stesso una buona usabilità. È il modo che si utilizza per progettare un web multi-dispositivo.

Dobbiamo offrire agli utenti mobili un'esperienza adeguata, poiché ci sono ancora vincoli come la durata della batteria e la larghezza di banda.

Il termine “**responsive design**”, coniato da Ethan Marcotte nel 2010, descrive l'utilizzo di griglie fluide, immagini fluide e query multimediali per creare contenuti reattivi.

I moderni metodi di layout CSS sono intrinsecamente reattivi ed esistono una moltitudine di funzionalità integrate nella piattaforma web per rendere più semplice la progettazione di siti reattivi.

## Desktop



@media screen and  
(min-width: 1024px)  
{...}

## Tablet



@media screen and  
(min-width: 768px) and  
(max-width: 1023px)  
{...}

## Smartphone



@media screen and  
(max-width: 767px)  
{...}

# Viewport meta tag

---

Questo meta tag indica ai browser che devono impostare la larghezza del viewport sulla larghezza del dispositivo e ridimensionare il documento al 100% della dimensione prevista.

Impostando width=device-width stai sovrascrivendo l'impostazione predefinita di un dispositivo mobile (ad esempio per Apple 980px) con la larghezza effettiva del dispositivo.

Quindi dovresti sempre includere il meta tag viewport nell'intestazione dei tuoi documenti.

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

# Layout fluidi

---

Un layout fluido si basa su valori dinamici come una percentuale della larghezza del viewport.

Questa tecnica aumenterà o diminuirà dinamicamente le diverse dimensioni degli elementi del contenitore in base alle dimensioni dello schermo.



width: 63%

width: 32%

# Layout Flexbox e Grid

---

Flexbox e Grid sono progettati per essere più efficienti nel disporre più elementi, anche quando la dimensione del contenuto all'interno dei contenitori è sconosciuta.

Un contenitore flessibile espande gli elementi in modo che riempiano lo spazio libero disponibile o li restringe per evitare che trabordino. Questi contenitori flessibili hanno una serie di proprietà uniche, come justify-content, che non è possibile modificare con un normale elemento HTML.

Grid ci dà la possibilità di creare griglie differenti a seconda dei breakpoints scelti e creare un'unico codice che si modellerà attraverso le griglie costruite all'inizio.

# Immagini reattive

---

L'iterazione più elementare delle immagini reattive segue lo stesso concetto di layout fluido, utilizzando un'unità dinamica per controllare la larghezza o l'altezza.

L'unità % si avvicina ad una singola percentuale della larghezza o dell'altezza del viewport e fa in modo che l'immagine rimanga proporzionata allo schermo. Il problema di questo approccio è che ogni utente deve scaricare l'immagine a grandezza naturale, anche su cellulare.

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

# Tipografia Reattiva

---

Per un design veramente reattivo, si dovrebbero regolare in modo appropriato anche le dimensioni dei caratteri, in modo che si adattino alle dimensioni dello schermo.

Un metodo è impostare un valore statico per la dimensione del font, come 22 px, e adattarlo in ogni media query.

Un'alternativa sono le unità viewport vw che possono essere utilizzate anche per abilitare la tipografia reattiva, senza la necessità di impostare punti di interruzione con le query.

1vw è uguale all'1% della larghezza del viewport, il che significa che se imposti la dimensione del carattere utilizzando vw, si riferirà sempre alla dimensione del viewport.

```
@media (min-width:
1200px) {
  h1 {
    font-size: 22px;
  }
}
```

```
h1 {
  font-size: 6vw;
}
```



Il problema con le unità viewport è che l'utente perde la possibilità di ingrandire qualsiasi testo impostato utilizzando l'unità vw, poiché quel testo è sempre correlato alla dimensione del viewport. Pertanto non dovresti mai impostare il testo utilizzando solo queste unità.

Esiste una soluzione e implica l'utilizzo di calc(). Se aggiungi l'unità vw a un set di valori utilizzando una dimensione fissa come ems o rems, il testo sarà comunque zoomabile.

```
h1 {  
  font-size: calc(1.5rem + 3vw);  
}
```

Il carattere aumenta quindi gradualmente man mano che si aumenta la dimensione del viewport.

# Media Queries

---

La seguente query verifica se la pagina Web corrente viene visualizzata come supporto su schermo (quindi non come documento stampato) e il viewport è compreso tra 768px e 1023px (dimensioni del tablet). Il CSS per il selettore `.container` verrà applicato solo se queste due cose sono vere.

```
@media screen and (min-width: 768px)
and (max-width: 1023px){
  .container {
    margin: 10px;
  }
}
```

Le media query ci consentono di applicare i CSS in modo selettivo per modellare la pagina in modo appropriato alle esigenze dell'utente.

<https://css-tricks.com/a-complete-guide-to-css-media-queries/>

# Breakpoints

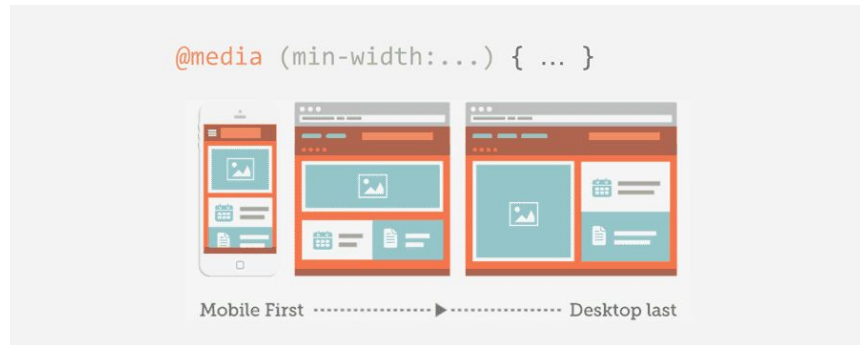
Per lavorare con le media query, è necessario decidere i “**breakpoint responsive**” o i breakpoint delle dimensioni dello schermo. Un breakpoint è la larghezza dello schermo in cui si utilizza una media query per implementare nuovi stili CSS.

Dimensioni comuni dello schermo

- Mobile: 360 x 640
- Mobile: 375 x 667
- Mobile: 360 x 720
- iPhone X: 375 x 812
- Pixel 2: 411 x 731
- Tablet: 768 x 1024
- Laptop: 1366 x 768
- Laptop o desktop ad alta risoluzione: 1920 x 1080

# Mobile first

Se si sceglie un approccio di design **mobile-first**, con una singola colonna e font di dimensioni più piccole come base, non è necessario includere i breakpoint mobili – a meno che non si voglia ottimizzare il design per modelli specifici. In questo modo è possibile creare un design reattivo di base con due soli breakpoint, uno per tablet e uno per computer portatili e desktop.



# Esercizio 5

## Responsive

- Crea una pagina HTML e un foglio CSS collegato.
- Nell'elemento Header inserisci un H1 con un'istanza media query CSS per cambiare il colore del testo del testo in verde quando la larghezza della finestra del browser è inferiore a 600 pixel.
- Dentro all'elemento Main inserisci due immagini (impostate con width: 100%) e applica un'istanza di media query CSS per ridurre le dimensioni dell'immagine a 49% della larghezza originale quando la larghezza della finestra del browser è inferiore a 800 pixel.