



Responsive design & Bootstrap

Materiale didattico – CSS: competenze avanzate

Anno scolastico 2024/2025

Responsive design

Principi

Il **responsive design** è un approccio che permette ai siti web di adattarsi a diverse dimensioni di schermo e dispositivi, offrendo un'esperienza utente ottimale ovunque (es. tablet, smartphone, desktop).

- Fluidità: utilizzo di unità di misura relative.
- Proporzionalità dei media: mantenimento delle proporzioni dei media.
- Layout flessibili: uso di tecnologie CSS per natura adattive (es. Flexbox e Grid) oppure di apposite librerie CSS come Bootstrap.
- Media Queries: stili diversi in base alle dimensioni dello schermo.

Responsive design

Viewport

Il meta tag `viewport` nel `<head>` ci dice come adattare la larghezza della pagina e il livello di zoom al dispositivo dell'utente. È una condizione fondamentale per implementare il responsive design.

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

- `device-width`: imposta la larghezza del viewport uguale a quella dello schermo del dispositivo.
- `initial-scale`: definisce il livello di zoom iniziale (fattore di scala).

Responsive design

Viewport

senza meta
viewport



con meta
viewport



Visualizzazione di un sito
desktop su smartphone

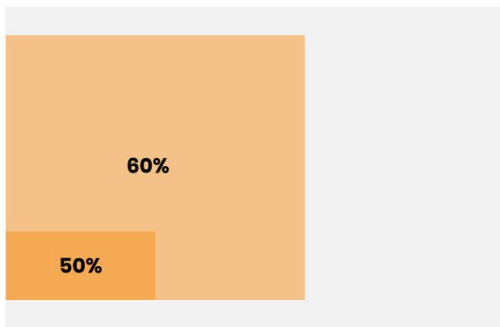
Responsive design

Misure relative

A differenza delle misure fisse (es. px), le misure relative si basano su proporzioni rispetto a un riferimento, il che garantisce una maggiore fluidità.

Le più utilizzate sono:

- **em/rem**
- **%**
- **vw/vh**



Responsive design

Media



È importante che un media (ad esempio immagini, video, o iframe) non perda le proporzioni originali su dispositivi di dimensioni diverse.

Ecco alcune strategie comuni per ottenere questo risultato:

- Non specificare contemporaneamente un valore numerico per le proprietà `width` e `height`. Utilizza misure relative (es. percentuale).
- Applica `height: auto` per preservare il rapporto tra le dimensioni.
- Assicurati che le immagini abbiano una qualità sufficiente ad essere visualizzate su schermi grandi o utilizza la grafica vettoriale (es. SVG).

Responsive design

Flexbox

Flexbox è un potente strumento CSS per creare layout flessibili e responsivi, grazie alla sua capacità di adattare automaticamente gli elementi al contenitore e di ridisporli in base allo spazio disponibile.

In particolare la proprietà `flex-wrap: wrap` permette agli elementi di andare a capo quando lo spazio è insufficiente. Le proprietà `flex-grow` e `flex-shrink` gestiscono l'ingrandimento e il rimpicciolimento degli elementi.

Infine, la proprietà `order`, in combinazione con le media query, permette di ordinare gli elementi diversamente in base alla dimensione dello schermo.

Responsive design

Media Queries

Le **media query** sono una funzionalità che consente di applicare regole di stile specifiche in base a determinate caratteristiche del dispositivo o della finestra del browser, come larghezza, altezza, orientamento, risoluzione e altro.

Utilizza la regola `@media` seguita da una condizione e da un blocco di CSS:

```
@media (condizione) {  
    /* Regole CSS applicate se la condizione è vera */  
}
```


Responsive design

Media Queries

- `max-width`: applica lo stile se la larghezza dello schermo è inferiore o uguale al valore specificato.

```
@media (max-width: 768px) { ... }
```

- `min-width`: applica lo stile se la larghezza dello schermo è maggiore o uguale al valore specificato.

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

Responsive design

Media Queries

- `orientation`: adatta lo stile in base all'orientamento del dispositivo.

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

- `aspect-ratio`: rapporto tra larghezza e altezza dello schermo.

```
@media (aspect-ratio: 16/9) { ... }
```

Responsive design

Media Queries

operatore **and**

```
@media (min-width: 768px) and (max-width: 1024px) {  
    /* Stili applicati a schermi medi */  
}
```

```
@media (max-width: 600px), (orientation: portrait) {  
    /* Stili per schermi piccoli o in orientamento verticale */  
}
```

operatore **or**

operatore **not**

```
@media not (max-width: 600px) {  
    /* Stili per schermi con larghezza maggiore di 600px */  
}
```

Responsive design

Media Queries

```
@media (min-width: 1025px) {  
    /* Stili per desktop */  
}
```



```
@media (min-width: 601px) and (max-width: 1024px) {  
    /* Stili per tablet */  
}
```

```
@media (max-width: 600px) {  
    /* Stili per smartphone */  
}
```



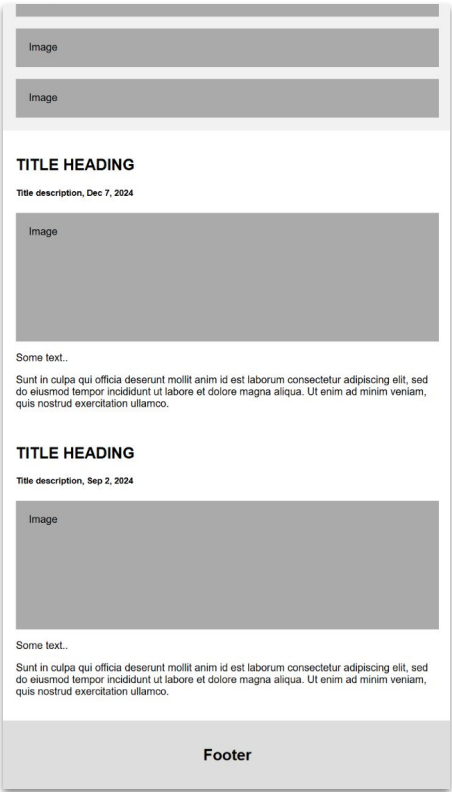
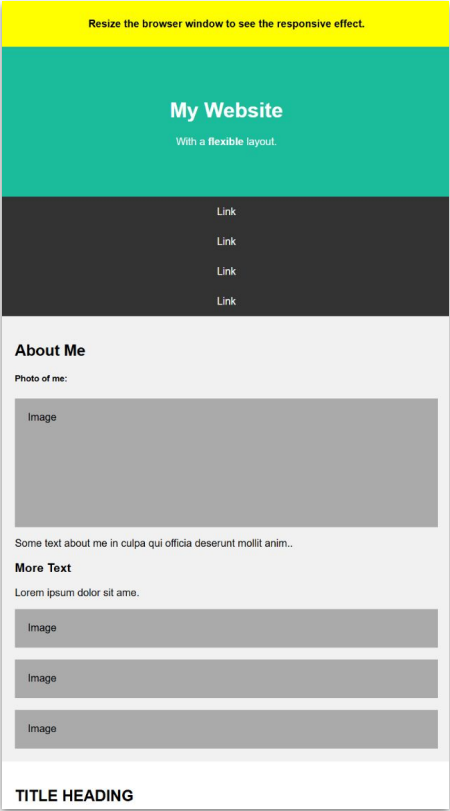
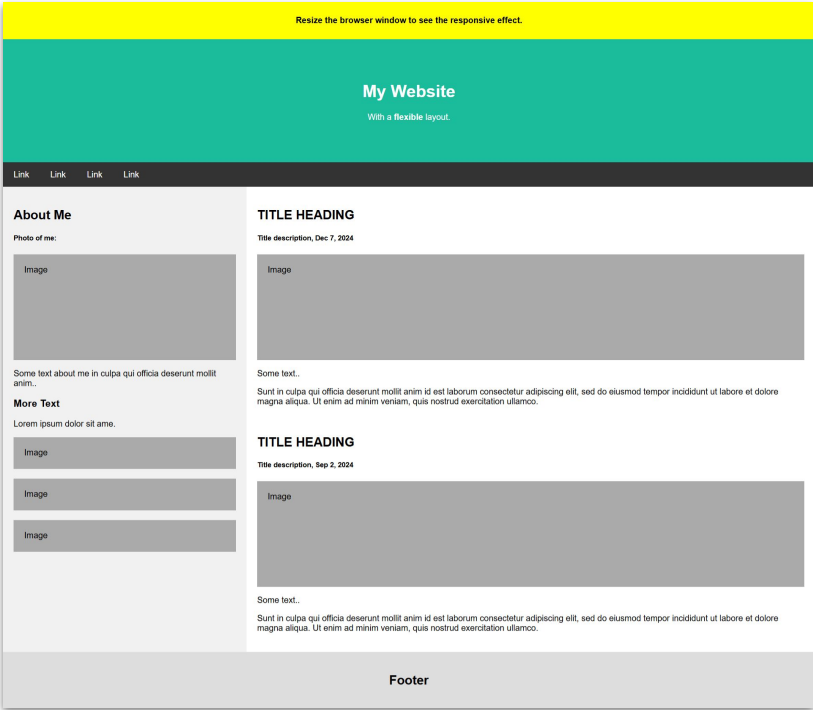
Responsive design

Mobile-first

L'approccio **mobile-first** è una metodologia di progettazione e sviluppo web che pone il focus iniziale sui dispositivi mobili, garantendo che i contenuti e le funzionalità principali di un sito siano ottimizzati per schermi più piccoli. Successivamente, si espande il design per includere dispositivi con schermi più grandi, come tablet e desktop.

In pratica, si progetta partendo da una base minima (mobile) e si aggiungono progressivamente elementi e funzionalità utilizzando media query per migliorare l'esperienza su schermi più grandi.

Responsive design



Bootstrap

Introduzione

Bootstrap è un framework front-end open-source sviluppato da Twitter.

È progettato per semplificare lo sviluppo web, offrendo una vasta gamma di strumenti e componenti predefiniti per la creazione di interfacce utente responsive e moderne.

La versione attuale è la 5.3.5.



<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

Bootstrap

Caratteristiche

Le principali caratteristiche offerte da Bootstrap sono:

- una **griglia flessibile** per creare pagine dalla struttura responsive;
- **componenti pronti** all'uso e dal design responsive;
- supporto a **Sass** (*Syntactically awesome style sheets*);
- temi personalizzabili;
- stili per migliorare l'accessibilità;
- intercompatibilità con i browser;
- ottima documentazione e supporto dalla community.

Bootstrap

Includere il framework

In alternativa è possibile scaricare i file .css e .js per utilizzarli in locale anche offline.

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container">
  <h1>La mia prima pagina Bootstrap</h1>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Bootstrap

Breakpoints

I **breakpoint** sono i mattoni fondamentali del design responsive e sfruttano le media query per rendere responsive i componenti della pagina.

L'obiettivo è il design **mobile-first**: Bootstrap mira ad applicare il minimo indispensabile di stili per far funzionare un layout al breakpoint più piccolo, e quindi aggiunge stili per adattare quel design ai dispositivi più grandi. Questo ottimizza i CSS, migliora i tempi di rendering e offre un'ottima esperienza agli utenti finali.

Bootstrap

Infissi

| Breakpoint | Class infix | Dimensions |
|-------------------|-------------|------------|
| Extra small | <i>None</i> | <576px |
| Small | <i>sm</i> | ≥576px |
| Medium | <i>md</i> | ≥768px |
| Large | <i>lg</i> | ≥992px |
| Extra large | <i>xl</i> | ≥1200px |
| Extra extra large | <i>xxl</i> | ≥1400px |

Bootstrap

Container

Il **container** è l'elemento di layout più basilare in Bootstrap, necessario quando si utilizza il sistema di griglia predefinito.

I container vengono utilizzati per contenere, spaziare e (a volte) centrare il contenuto al loro interno.

```
<div class="container">  
  <h1>La mia prima pagina Bootstrap</h1>  
</div>
```

Bootstrap

Container

| | Extra small <576px | Small ≥576px | Medium ≥768px | Large ≥992px | X-Large ≥1200px | XX-Large ≥1400px |
|-------------------------------|------------------------------|------------------------|-------------------------|------------------------|---------------------------|----------------------------|
| <code>.container</code> | 100% | 540px | 720px | 960px | 1140px | 1320px |
| <code>.container-sm</code> | 100% | 540px | 720px | 960px | 1140px | 1320px |
| <code>.container-md</code> | 100% | 100% | 720px | 960px | 1140px | 1320px |
| <code>.container-lg</code> | 100% | 100% | 100% | 960px | 1140px | 1320px |
| <code>.container-xl</code> | 100% | 100% | 100% | 100% | 1140px | 1320px |
| <code>.container-xxl</code> | 100% | 100% | 100% | 100% | 100% | 1320px |
| <code>.container-fluid</code> | 100% | 100% | 100% | 100% | 100% | 100% |

Bootstrap

Griglia

Bootstrap offre una **griglia flessibile** basata su un sistema a **12 colonne** che consente di creare layout responsivi per adattarsi a diverse dimensioni di schermo, da dispositivi mobili a desktop.

Consiste nell'organizzare il contenuto della pagina in righe (**rows**) e colonne (**columns**). Il numero massimo di colonne per riga è 12.

La dimensione delle colonne cambia a seconda della larghezza del viewport.

Bootstrap

Griglia

```
<div class="row">
  <div class="col-5">.....</div>
  <div class="col-7">.....</div>
</div>
<div class="row">
  <div class="col-md-4">.....</div>
  <div class="col-md-2">.....</div>
  <div class="col-md-6">.....</div>
</div>
<div class="row">
  <div class="col-sm-6 col-lg-3">.....</div>
  <div class="col-sm-6 col-lg-3">.....</div>
  <div class="col-sm-6 col-lg-3">.....</div>
  <div class="col-sm-6 col-lg-3">.....</div>
</div>
```

Bootstrap

Sfondo

<https://getbootstrap.com/docs/5.3/utilities/background/>

.bg-primary

.bg-primary-subtle

.bg-secondary

.bg-secondary-subtle

.bg-success

.bg-success-subtle

.bg-danger

Bootstrap

Colore

<https://getbootstrap.com/docs/5.3/utilities/colors/>

`.text-primary`

`.text-primary-emphasis`

`.text-secondary`

`.text-secondary-emphasis`

`.text-success`

`.text-success-emphasis`

`.text-danger`

`.text-danger-emphasis`

`.text-warning`

`.text-warning-emphasis`

`.text-info`

`.text-info-emphasis`

`.text-light`

Bootstrap

Testo

<https://getbootstrap.com/docs/5.3/utilities/text/>

Start aligned text on all viewport sizes.

Center aligned text on all viewport sizes.

End aligned text on all viewport sizes.

End aligned text on viewports sized SM (small) or wider.

End aligned text on viewports sized MD (medium) or wider.

End aligned text on viewports sized LG (large) or wider.

End aligned text on viewports sized XL (extra large) or wider.

End aligned text on viewports sized XXL (extra extra large) or wider.

```
<p class="text-start">Start aligned text on all viewport sizes. </p>
```

```
<p class="text-center">Center aligned text on all viewport sizes. </p>
```

```
<p class="text-end">End aligned text on all viewport sizes. </p>
```

```
<p class="text-sm-end">End aligned text on viewports sized SM (small) or wider. </p>
```

```
<p class="text-md-end">End aligned text on viewports sized MD (medium) or wider. </p>
```

```
<p class="text-lg-end">End aligned text on viewports sized LG (large) or wider. </p>
```

```
<p class="text-xl-end">End aligned text on viewports sized XL (extra large) or wider. </p>
```

```
<p class="text-xxl-end">End aligned text on viewports sized XXL (extra extra large) or wider. </p>
```

Bootstrap

Spaziatura

<https://getbootstrap.com/docs/5.3/utilities/spacing/>

```
<div class="pt-4 bg-warning">I only have a top padding (1.5rem = 24px)</div>  
<div class="p-5 bg-success">I have a padding on all sides (3rem = 48px)</div>  
<div class="m-5 pb-5 bg-info">I have a margin on all sides (3rem = 48px) and a bottom padding (3rem = 48px)</div>
```

`{property}{sides}-{breakpoint}-{size}`

property: m, p

sides: t, b, s, e, x, y, blank

breakpoint: sm, md, lg, xl, xxl, blank

size: 0, 1, 2, 3, 4, 5, auto

I only have a top padding (1.5rem = 24px)

I have a padding on all sides (3rem = 48px)

I have a margin on all sides (3rem = 48px) and a bottom padding (3rem = 48px)

Bootstrap

Dimensioni

<https://getbootstrap.com/docs/5.3/utilities/sizing/>

```
<div class="w-25 bg-warning">Width 25%</div>  
<div class="w-50 bg-warning">Width 50%</div>  
<div class="w-75 bg-warning">Width 75%</div>  
<div class="w-100 bg-warning">Width 100%</div>  
<div class="mw-100 bg-warning">Max Width 100%</div>
```

Width 25%

Width 50%

Width 75%

Width 100%

Max Width 100%

Bootstrap

Flexbox

<https://getbootstrap.com/docs/5.3/utilities/flex/>

```
<div class="d-flex">...</div>
<div class="d-inline-flex">...</div>
<div class="d-flex flex-nowrap">...</div>
<div class="d-lg-flex">...</div>
<div class="d-flex flex-row">...</div>
<div class="d-flex flex-row flex-md-column-reverse">...</div>
<div class="d-flex justify-content-center justify-content-lg-between">...</div>
<div class="d-flex align-items-center align-items-md-start">...</div>
```

Bootstrap

Bottoni

<https://getbootstrap.com/docs/5.3/components/buttons/>

Basic

Primary

Secondary

Success

Info

Warning

Danger

Dark

Light

[Link](#)

```
<button type="button" class="btn">Basic</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-link">Link</button>
```

Bootstrap

Form

<https://getbootstrap.com/docs/5.3/forms/overview/>

Email

Password

Address

1234 Main St

Address 2

Apartment, studio, or floor

City

State

Choose... ▼

Zip

☐ Check me out

Sign in

Bootstrap

Tabelle

<https://getbootstrap.com/docs/5.3/content/tables/>

| Class | Heading | Heading |
|-----------|---------|---------|
| Default | Cell | Cell |
| Primary | Cell | Cell |
| Secondary | Cell | Cell |
| Success | Cell | Cell |
| Danger | Cell | Cell |
| Warning | Cell | Cell |
| Info | Cell | Cell |
| Light | Cell | Cell |
| Dark | Cell | Cell |