

Bootstrap

- Framework CSS (+ JavaScript) per Responsive Web Design
 - Progetto interno di Twitter, 2011
 - La dimensione del viewport viene categorizzata in breakpoint
 - $\text{xxl} \geq 1400\text{px}$, $\text{xl} \geq 1200\text{px}$, $\text{lg} \geq 992\text{px}$, $\text{md} \geq 768$, $\text{sm} \geq 576$ (ed extra small)
- Download da <https://getbootstrap.com/>
 - Info: <https://getbootstrap.com/docs/versions/>
- CDN da <https://www.bootstrapcdn.com/>
- Progetto di riferimento
 - <https://github.com/egalli64/nesp> (*modulo 4*)

Setup

- Assicurarsi che il browser interpreti la pagina come HTML5
 - `<!DOCTYPE html>`
- Head
 - Inserire i seguenti **meta**
 - `<meta charset="utf-8">`
 - `<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">`
 - Inserire il **link** a Bootstrap, ad es.:
 - <https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css>
 - Per le componenti che richiedono JavaScript (come il carousel):
 - <https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js>
 - Bundle che include Popper.js e il plugin di Bootstrap

Container

- Classe base, si occupa di stilare il contenuto
 - Il grid di bootstrap deve essere in un container
- Due classi container
 - **container**, lunghezza fissa per ogni breakpoint
 - **container-fluid**, è sempre il 100% del viewport

```
<div class="container">  
  <h1>Hello from Bootstrap</h1>  
</div>
```

```
<div class="container-fluid">  
  <h1>Hello from Bootstrap</h1>  
</div>
```

Grid

- Deve essere all'interno di un container
- Elementi organizzati in righe e colonne
 - Un blocco di classe **row** per ogni riga
 - Un blocco di classe **col** per ogni cella
 - Implicitamente tutte della stessa dimensione
 - Possono avere una dimensione in **dodicesimi**
- Classi **col[-breakpoint]-n**
 - Breakpoint in sm ... xxl, n in 1 .. 12
- Classi **offset[-breakpoint]-n**
 - Abbinare a “col”, lasciano un “buco” a sinistra
- Strutture complesse: una row può avere sub-row

```
<div class="container">
  <div class="row">
    <div class="col">...</div>
    <div class="col">...</div>
    <!-- ...-->
  </div>
  <div class="row">
    <div class="col">...</div>
    <div class="col">...</div>
    <!-- ...-->
  </div>
  <div class="row">
    <div class="col-9 col-lg-3">...</div>
    <div class="col-1 col-lg-3">...</div>
  </div>
  <div class="row">
    <div class="col-4 offset-8">2/2</div>
  </div>
</div>
```

Table

- Per stilare un elemento table
 - Lo si mette in un container
 - Gli si applica la classe bootstrap table
 - Una o più classi per tabella come
 - table-striped
 - table-bordered
 - table-hover
 - ...
 - Altre classi bootstrap per tabelle ed elementi contenuti
 - table-dark, table-light
 - table-success, table-danger, table-info, table-warning, ...

```
<table class="table table-dark">
  <thead class="table-light">
    <tr>
      <th scope="row">\</th>
      <th scope="col">Left</th>
      <th scope="col">Right</th>
    </tr>
  </thead>
  <tr>
    <th scope="row">Top</th>
    <td>X</td>
    <td>Y</td>
  </tr>
  <tr>
    <th scope="row">Low</th>
    <td class="table-danger">1</td>
    <td>2</td>
  </tr>
</table>
```

Carousel

- Richiede anche il plugin JavaScript di Bootstrap e Popper.js
 - <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- Un DIV con classi *carousel* e *slide* e attributo *data-ride carousel*
 - (partenza automatica)
- Contiene un DIV *carousel-inner*
 - che contiene un DIV *carousel-item* per ogni slide del carousel
- Il *carousel-item active* sarà il primo a partire
- È spesso una buona idea dare alle slide classe *w-100*
- Personalizzazione del carousel via JavaScript:
 - `new bootstrap.Carousel(document.querySelector('#myCarousel'), { interval: 1000 });`