

Autore: Gabriel Rovesti

Indice

1. [Selettori](#)
2. [Posizionamento](#)
3. [Layout e Box Model](#)
4. [Flexbox](#)
5. [Responsive Design](#)
6. [Bootstrap](#)
7. [Progetti Pratici](#)

Selettori

Selettori Base

- **Tag:** `h1 { color: red; }`
- **Classe:** `.red-header { color: red; }`
- **ID:** `#about-us-header { color: red; }`
- **Universale:** `* { color: red; }`

Selettori Combinatori

- **Gruppo:** `h1, h2, h3 { color: red; }`
- **Discendente:** `div h1 { color: red; }` → tutti gli h1 dentro div
- **Figlio diretto:** `div > h1 { color: red; }` → solo h1 figli diretti di div
- **Adiacente:** `div + h1 { color: red; }` → primo h1 successivo a div
- **Successivo:** `h1 ~ p { color: red; }` → tutti i p successivi a h1

Pseudo-classi (Stati)

- **Link:**
 - `a:link { color: #FF0000; }` → link da visitare
 - `a:visited { color: #00FF00; }` → link visitati
 - `a:active { color: #0000FF; }` → link attivo (cliccato)
- **Selezione:** `div:hover { background-color: blue; }`
- **Posizione:**
 - `li:first-child { color: #FF0000; }`
 - `li:last-child { color: #FF0000; }`
 - `li:nth-child(2) { color: #FF0000; }`

- `li:nth-last-child()`

Specificità

Ordine di priorità (dal più al meno specifico):

1. Proprietà con `!important`
2. CSS inline (`<div style="color: red">`)
3. Selettori ID (`#my-div`)
4. Selettori classe e pseudo-classe (`.my-div` , `:hover`)
5. Selettori tag (`div`)

Priorità dei fogli di stile: inline > incorporato > esterno

Posizionamento

Tipi di Tag

- **Block-level:** occupa tutta la larghezza, inizia su nuova riga (es. `<p>` , `<h1>` , `<div>`)
- **Inline-level:** occupa solo lo spazio necessario (es. `<a>` , `` , ``)

Proprietà Display

- `display: inline;` → inline (default per elementi inline)
- `display: block;` → blocco
- `display: none;` → nascosto
- `display: inline-block;` → inline con caratteristiche di blocco (può avere width, height, margin)

Float

- `float: left;` → sposta elemento a sinistra
- `float: right;` → sposta elemento a destra
- `clear: left/right/both;` → interrompe flusso attorno agli elementi fluttuanti

Position

- `position: static;` → posizionamento normale (default)
- `position: relative;` → posizionamento relativo alla posizione originale
- `position: absolute;` → posizionamento rispetto al primo contenitore con position diverso da static
- `position: fixed;` → posizionamento rispetto al viewport
- `position: sticky;` → ibrido tra relative e fixed, diventa fixed al raggiungimento di una posizione

Proprietà di Posizionamento

- `top`, `bottom`, `left`, `right` → scostamento dai bordi
- `z-index` → ordine di sovrapposizione (asse z)
- `opacity` → trasparenza (0-1)

Layout e Box Model

Centratura

- **Testo:**
 - `text-align: center;`
 - `line-height: [altezza-contenitore];`
- **Immagini:** `background-position: center;`
- **Elementi di blocco:**
 - Con position absolute: `top: 50%; left: 50%; margin: -[metà-altezza] 0 0 - [metà-larghezza];`
 - Con margin auto: `margin: [valore-verticale] auto;`

Overflow

- `overflow: visible;` → default, contenuto visibile anche fuori dal contenitore
- `overflow: hidden;` → nasconde contenuto fuori dal contenitore
- `overflow: scroll;` → aggiunge barre di scorrimento
- `overflow: auto;` → aggiunge barre di scorrimento solo se necessario

Text Overflow

- `text-overflow: clip;` → tronca il testo (default)
- `text-overflow: ellipsis;` → aggiunge "..." al testo troncato

Flexbox

Container Flex

- `display: flex;` → attiva il modello flexbox
- `display: inline-flex;` → versione inline

Proprietà Direzione

- `flex-direction: row;` → orizzontale (default)
- `flex-direction: column;` → verticale
- `flex-direction: row-reverse;` → orizzontale invertito

- `flex-direction: column-reverse;` → verticale invertito

Proprietà Wrap

- `flex-wrap: nowrap;` → singola linea (default)
- `flex-wrap: wrap;` → multiple linee
- `flex-wrap: wrap-reverse;` → multiple linee invertite
- `flex-flow: row wrap;` → shorthand per direction e wrap

Allineamento Orizzontale (justify-content)

- `justify-content: flex-start;` → inizio (default)
- `justify-content: flex-end;` → fine
- `justify-content: center;` → centro
- `justify-content: space-between;` → spazio equo tra elementi
- `justify-content: space-around;` → spazio equo intorno agli elementi
- `justify-content: space-evenly;` → spazio uniformemente distribuito

Allineamento Verticale (align-items)

- `align-items: stretch;` → estesi (default)
- `align-items: flex-start;` → inizio
- `align-items: flex-end;` → fine
- `align-items: center;` → centro
- `align-items: baseline;` → allineamento alla linea di base del testo

Allineamento Multi-linea (align-content)

- `align-content: stretch;` → estende (default)
- `align-content: flex-start;` → inizio
- `align-content: flex-end;` → fine
- `align-content: center;` → centro
- `align-content: space-between;` → spazio equo tra linee
- `align-content: space-around;` → spazio equo intorno alle linee
- `align-content: space-evenly;` → spazio uniformemente distribuito

Proprietà degli Item

- `order: [numero];` → ordine di visualizzazione (default: 0)
- `flex-grow: [numero];` → fattore di crescita (default: 0)
- `flex-shrink: [numero];` → fattore di riduzione (default: 1)
- `flex-basis: [valore];` → dimensione iniziale (default: auto)
- `flex: [grow] [shrink] [basis];` → shorthand

- `align-self: [valore];` → allineamento individuale

Responsive Design

Principi

- **Fluidità**: unità di misura relative
- **Proporzionalità dei media**: mantenimento proporzioni
- **Layout flessibili**: tecnologie adattive (Flexbox, Grid)
- **Media Queries**: stili diversi in base alle dimensioni

Viewport

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

Misure Relative

- `em / rem` : relative al font
- `%` : percentuale rispetto al contenitore
- `vw / vh` : percentuale viewport

Media Queries

```
@media (max-width: 768px) { /* Stili per schermi fino a 768px */ }  
@media (min-width: 1024px) { /* Stili per schermi da 1024px in su */ }  
@media (orientation: portrait) { /* Stili per orientamento verticale */ }  
@media (min-width: 601px) and (max-width: 1024px) { /* Stili per tablet */ }
```

Mobile-first

Progettazione che parte dalle dimensioni mobili, aggiungendo complessità per schermi più grandi

Bootstrap

Introduzione

Framework front-end open-source per creare interfacce responsive.

Versione attuale: 5.3.5

Inclusione

```
<link  
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.cs
```

```
s" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstrap.bundle.m
in.js"></script>
```

Breakpoints

- `xs` : <576px (extra small)
- `sm` : ≥576px (small)
- `md` : ≥768px (medium)
- `lg` : ≥992px (large)
- `xl` : ≥1200px (extra large)
- `xxl` : ≥1400px (extra extra large)

Container

- `.container` : larghezza fissa per ogni breakpoint
- `.container-fluid` : larghezza 100%
- `.container-{breakpoint}` : larghezza 100% fino al breakpoint

Griglia

Sistema a 12 colonne:

```
<div class="row">
  <div class="col-5">5/12</div>
  <div class="col-7">7/12</div>
</div>
<div class="row">
  <div class="col-sm-6 col-lg-3">Responsive</div>
  <div class="col-sm-6 col-lg-3">Responsive</div>
  <div class="col-sm-6 col-lg-3">Responsive</div>
  <div class="col-sm-6 col-lg-3">Responsive</div>
</div>
```

Utilità

- **Colori**: `.bg-primary` , `.text-danger`
- **Testo**: `.text-start` , `.text-center` , `.text-md-end`
- **Spaziatura**: `.p-3` (padding), `.m-2` (margin), `.pt-4` (padding-top)
- **Dimensioni**: `.w-50` (width), `.h-25` (height)
- **Flexbox**: `.d-flex` , `.justify-content-center` , `.align-items-end`

Componenti

- **Bottoni:** `.btn`, `.btn-primary`, `.btn-danger`
- **Form:** vari stili per input, checkbox, radio, ecc.
- **Tabelle:** `.table`, `.table-striped`, `.table-bordered`

Progetti Pratici

Esercizio 1: Selettori

Formattare la pagina `selettori.html` come da mockup sfruttando i selettori CSS.

- Non modificare il codice HTML
- Usare box-sizing: border-box
- Per il menu usare display appropriato
- La voce "Il teatro" deve essere sottolineata al passaggio del mouse

Esercizio 2: Posizionamento

Formattare la pagina `position.html` come da mockup.

- Non modificare il codice HTML
- L'immagine "approfondimento", l'icona "torna su" e l'intestazione tabella devono avere comportamenti specifici durante lo scorrimento
- L'icona "torna su" deve reagire al passaggio del mouse
- Tabella con colore `#ffcc00` e border-collapse: collapse

Esercizio 3: Blocchi

Formattare la pagina `blocchi.html` come da mockup.

- Non modificare il codice HTML
- Dimensioni specifiche: menu 25%, banner 800px, box immagini 30%, immagini 75%
- Usare inline-block per allineare elementi
- Trovare margine percentuale adeguato per spaziatura
- Menu deve reagire al passaggio del mouse
- Testare il ridimensionamento della pagina

Esercizio 4: Flexbox

Formattare la pagina `flexbox.html` come da mockup.

- Non modificare il codice HTML
- Immagini con larghezza 100px

- Usare flexbox per allineamento menu, prodotti e contenuti