

Tecnico progettazione e sviluppo Web

Sviluppo ed implementazione

Daniel Rampanelli

<https://bit.ly/3YF6Sq7>



Link

- HTML - **Hyper** Text Markup Language
- Navigazione verso ancora, pagina "interna" oppure indirizzo "esterno"
- Percorsi relativi oppure assoluti
- Applicabile a testo oppure altri elementi (immagini, etc.)
- Effetto "hover" per esplicitare all'utente che si tratta di un elemento cliccabile

```
<nav>  
  <a href="/home.html"></a>  
  <a href="/azienda.html">L'azienda</a>  
  <a href="/catalogo.html">Catalogo</a>  
  <a href="/contatti.html">Contatti</a>  
</nav>
```

Link

- Tag: <a>
- Attribuito **id**: nome/identificativo univoco
- Attributo **style**: stili in linea
- Attributo **title**: informazioni aggiuntive
- Attributo **href**: percorso oppure indirizzo/url, con eventuale ancora
- Attributo **target**: modalità di apertura del link (_self, _blank)

<https://codepen.io/danielrampanelli/pen/RwXBRLy>

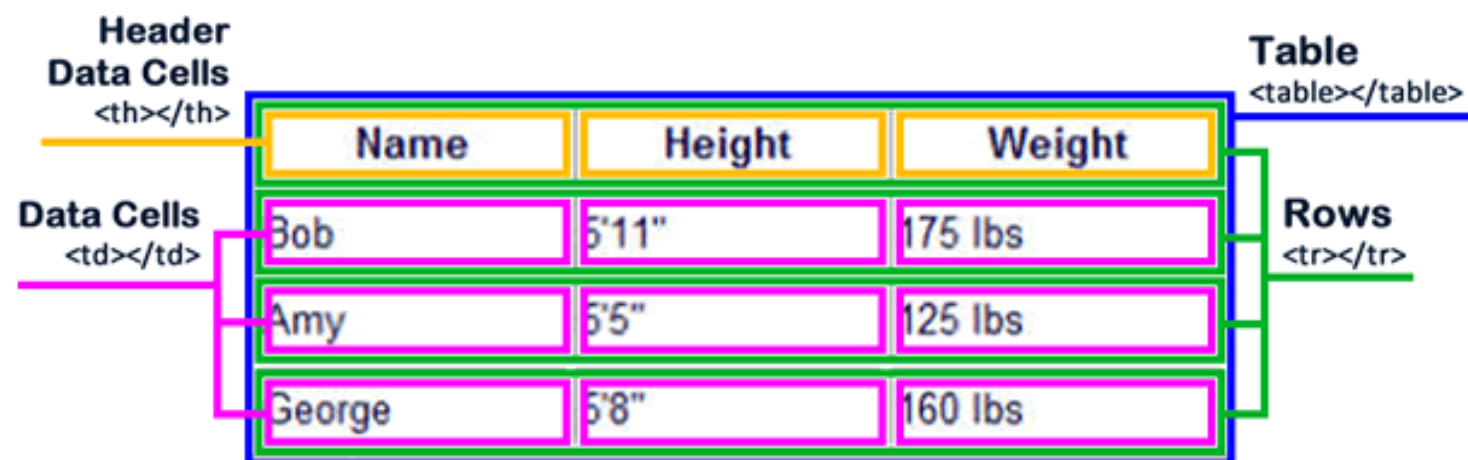
Ancore

- Definizione dell'elemento "ancorabile" tramite attributo "id"
- Link usando il formato "#idelemento"
- Applicabile sia per pagine interne ed esterne

<https://codepen.io/danielrampanelli/pen/RwXBGRd>

Tabelle

- Struttura a griglia per elementi di tutti i tipi (testi, immagini, etc.)
- Componenti: righe, colonne, celle, header, footer e didascalia
- Allineamento e avvolgimento del contenuto delle celle
- Celle isolate dalle altre
- Elemento non flessibile e non controllabile da CSS



Tabelle

- Tag: `<table>`, `<thead>`, `<tbody>`, `<tfoot>`, `<tr>`, `<th>`, `<td>`, `<caption>`
- Attributo **id**: nome/identificativo univoco
- Attributo **style**: stili in linea
- Attributo **title**: informazioni aggiuntive

<https://codepen.io/danielrampanelli/pen/oNKMLVq>

Grid

- Struttura flessibile per arrangiamento su righe e colonne
- Applicabile a tutti gli elementi
- Chiara separazione tra contenuti/celle (HTML) e disposizione (CSS)
- Definizione di griglia in maniera esplicita o impliciti
- Controllo degli spazi tra righe e colonne

<https://codepen.io/danielrampanelli/pen/PoMBGEw>

Grid

- `grid-template-columns`: lista di istruzioni per colonne
- `grid-template-rows`: lista di istruzioni per righe
- Unità assolute oppure relative (100px, 20vw, 1fr)
- Funzione `repeat(n, d)` per facilitare la definizione di strutture complesse
- Funzione `minmax(m, n)` permette di definire il range di righe o colonne
- Utilizzo della parola chiave **auto** per struttura di base senza dimensione

Grid

- Righe/colonne implicite quando non ci sono informazioni sulla struttura
- `grid-template-areas`: definizione logica della struttura usando etichette/label
- `grid-area`: assegnazione di una cella ad un area
- Utilizzo del segnaposto "." per una cella vuota

<https://codepen.io/danielrampanelli/pen/poMZNbV>

Grid

- Definizione di spaziature tra righe e colonne
- `gap`: spazio tra righe/colonne
- `row-gap`: spazio tra righe
- `column-gap`: spazio tra colonne

Grid

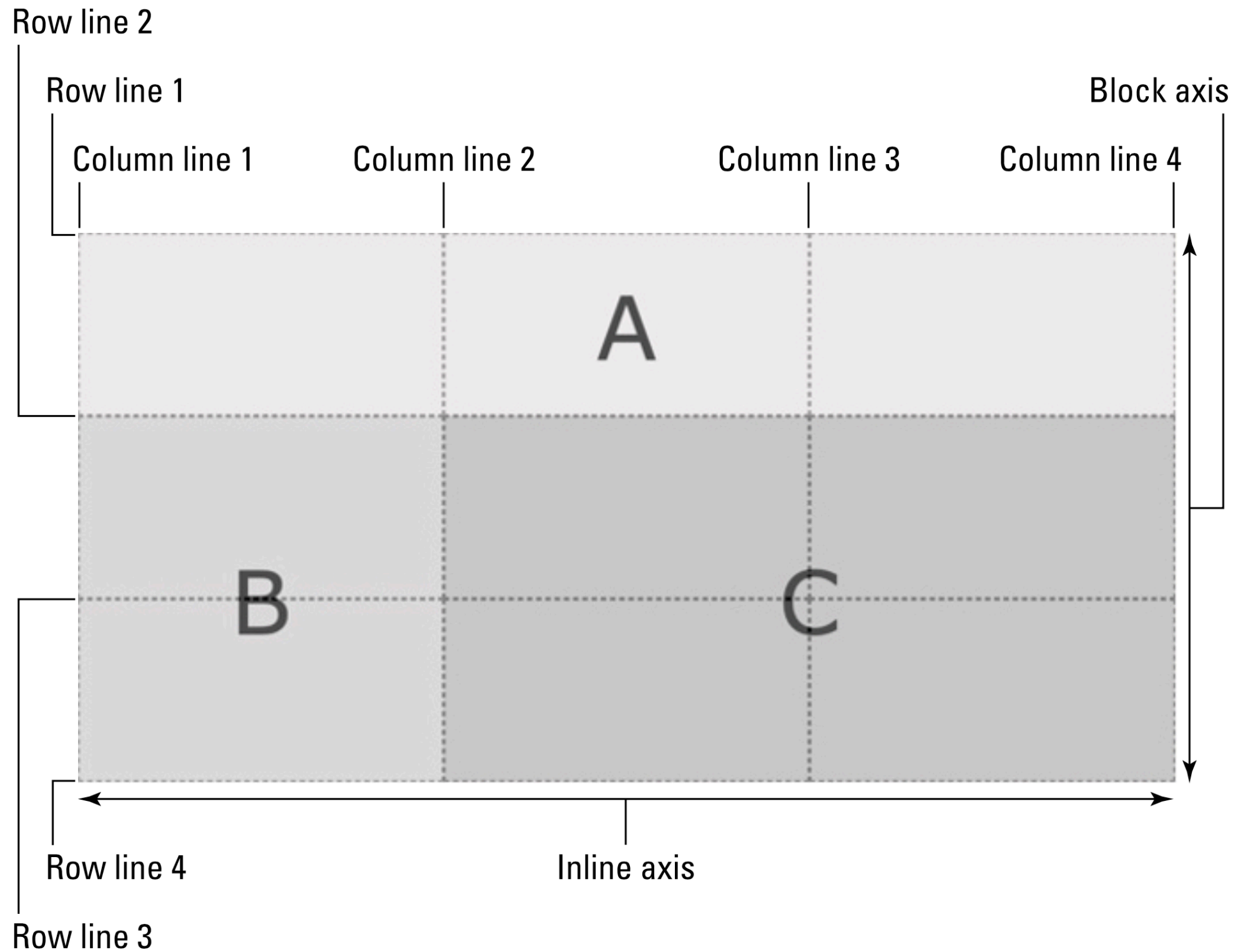
- Righe/colonne implicite quando non ci sono informazioni sulla struttura
- `grid-auto-columns`: dimensione per colonne implicite
- `grid-auto-rows`: dimensione per righe implicite

Grid

- Assegnazione e disposizione personalizzata di celle sulla griglia
- `grid-column-start`: Colonna di partenza della cella
- `grid-column-end`: Colonna di fine della cella
- `grid-row-start`: Riga di partenza della cella
- `grid-row-end`: Riga di fine della cella
- `grid-column`: Colonna di partenza / fine della cella
- `grid-row`: Riga di partenza / fine della cella

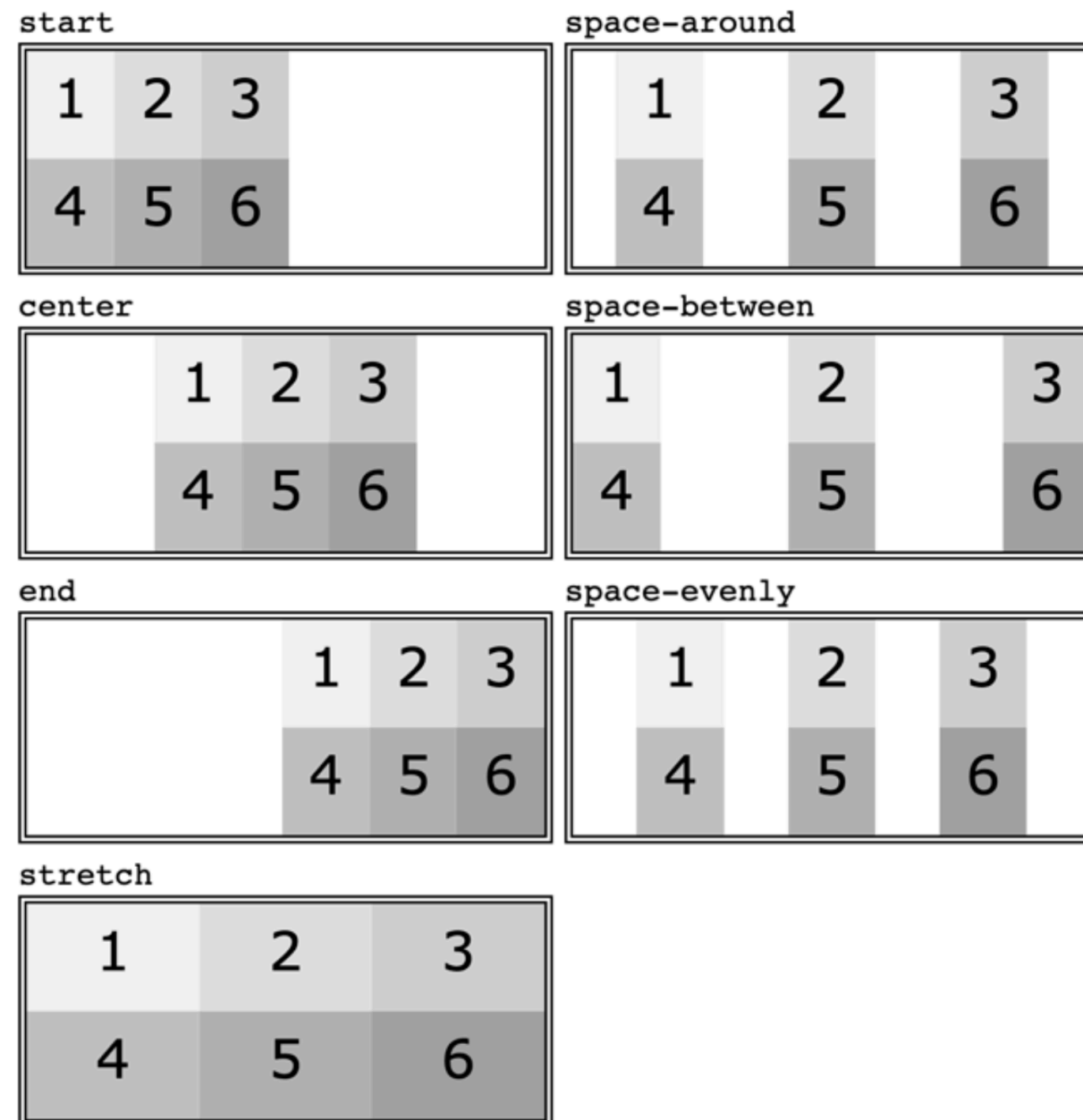
<https://codepen.io/danielrampanelli/pen/zYgLoKe>

Grid



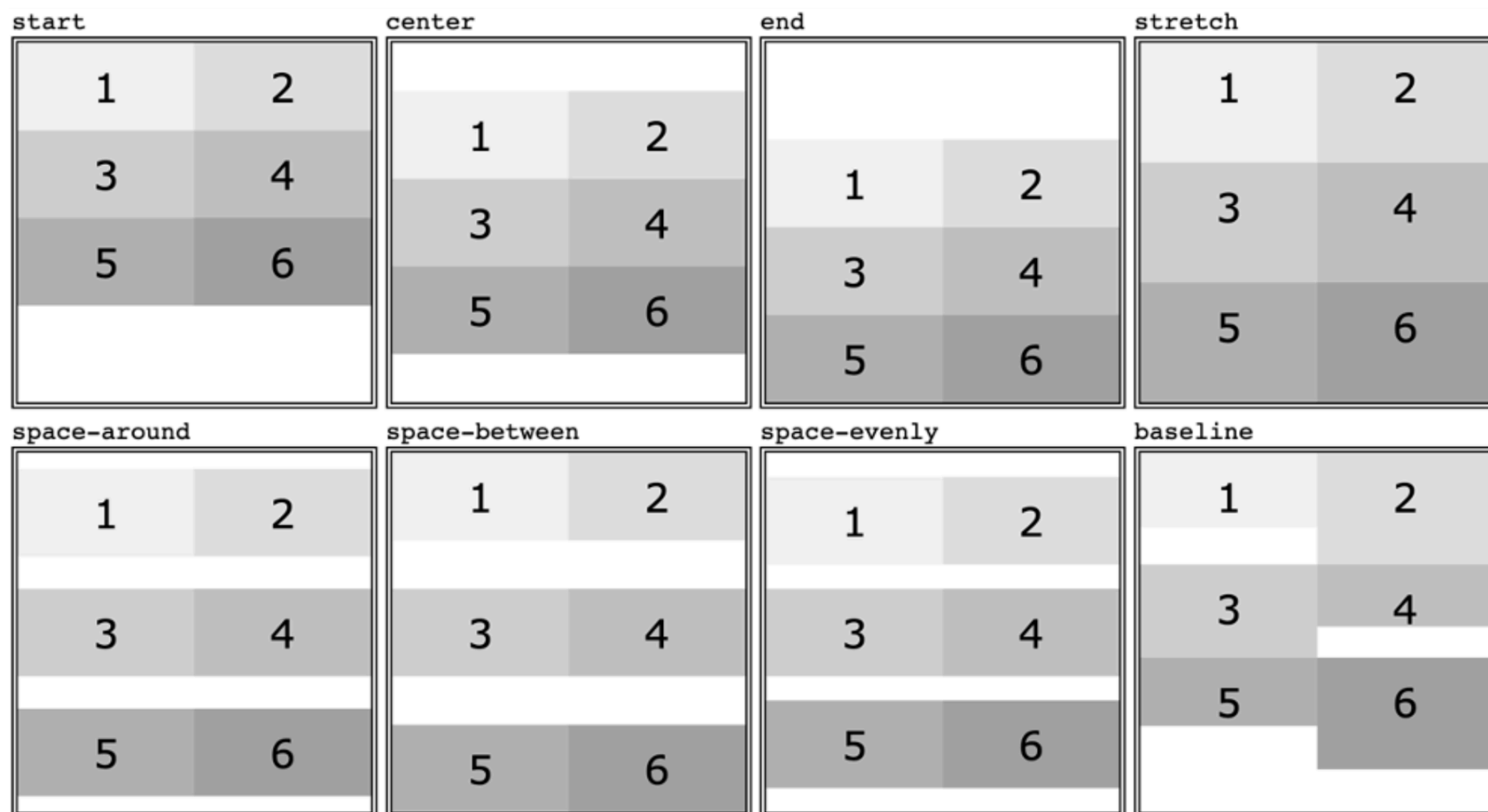
Grid: justify-content

Allineamento sull'asse "inline" delle colonne della griglia.



Grid: align-content

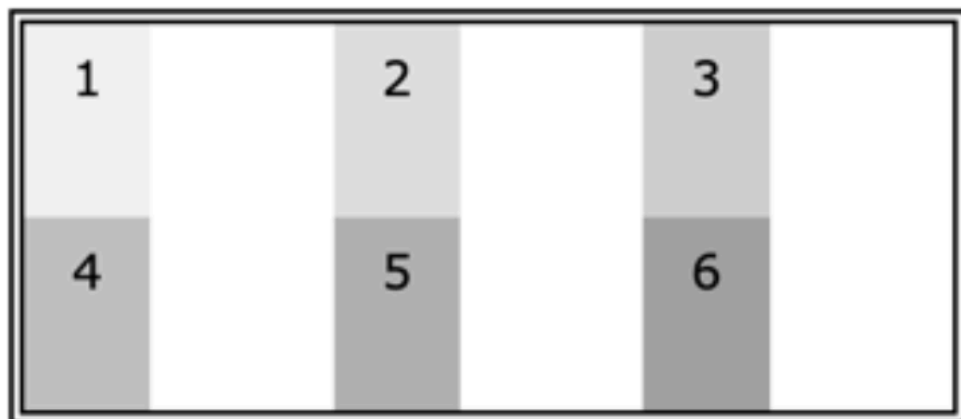
Allineamento sull'asse "block" delle righe della griglia.



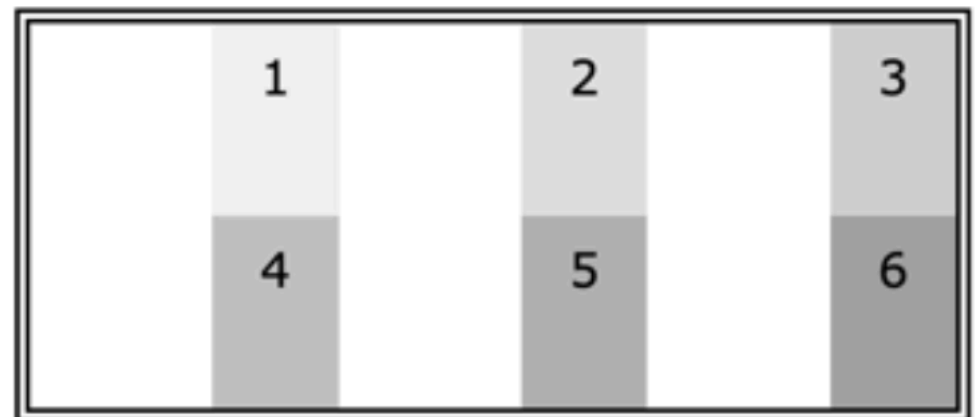
Grid: justify-items

Allineamento sull'asse "inline" dei singoli elementi nelle loro rispettive celle.

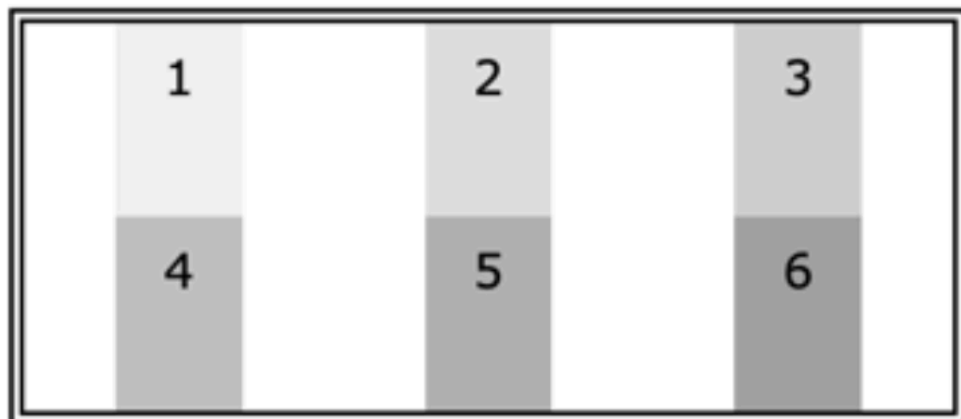
start



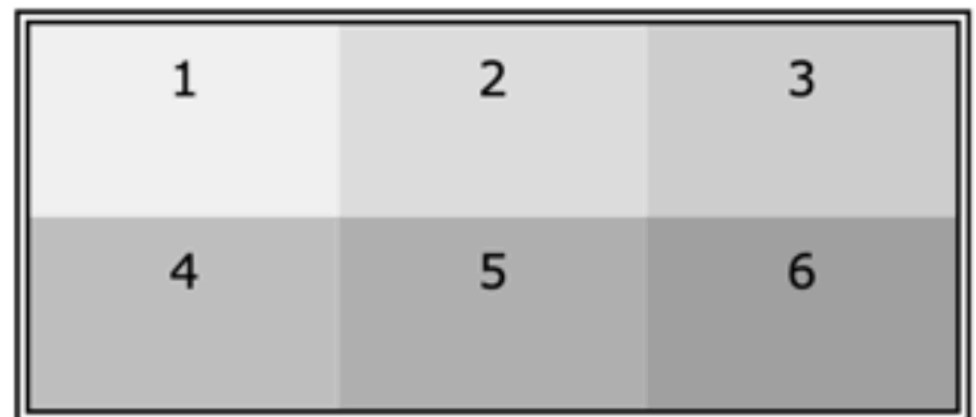
end



center

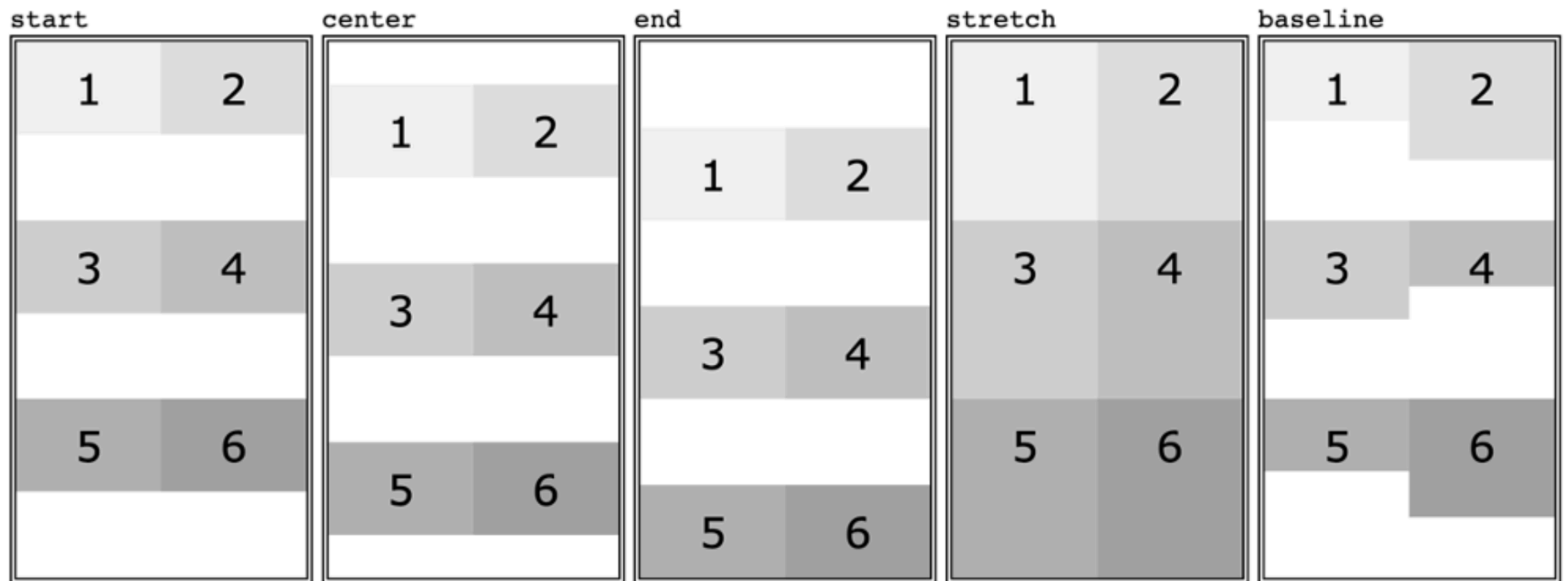


stretch



Grid: align-items

Allineamento sull'asse "block" dei singoli elementi nelle loro rispettive celle.



Grid: Proprietà CSS

display	grid
grid-template-columns	200px repeat(2, 1fr)
grid-template-rows	100px 200px 100px
gap	10px 20px
row-gap	20px
column-gap	10px
grid-template-areas	"A B C" "A D E"
grid-area	A

Grid: Proprietà CSS

grid-auto-rows	200px
grid-auto-columns	100px
grid-column-start grid-column-end	1
grid-column	1 / 2
grid-row-start grid-row-end	2
grid-row	2 / 4
justify-content	start center end stretch space-around space-between space-evenly
align-content	start center end stretch space-around space-between space-evenly baseline
justify-items	start center end stretch
align-items	start center end stretch baseline

Nesting

- Sviluppo di CSS in maniera più concisa e modulare
- Maggiore facilità nel leggere e mantenere il codice

```
article {  
  padding: 20px;  
}  
  
article .title {  
  color: coral;  
}  
  
article .title.bold {  
  font-weight: bold;  
}
```

```
article {  
  padding: 20px;  
  
  .title {  
    color: coral;  
  
    &.bold {  
      font-weight: bold;  
    }  
  }  
}
```

Esercizio

- Implementazione di 3 sezioni (o tutte) di uno dei mockup a disposizione
- Utilizzo di Visual Studio Code
- Applicazione delle strutture corrette, i font, colori e immagini fornite
- Obiettivo: replicare la grafica nel modo più dettagliato possibile