

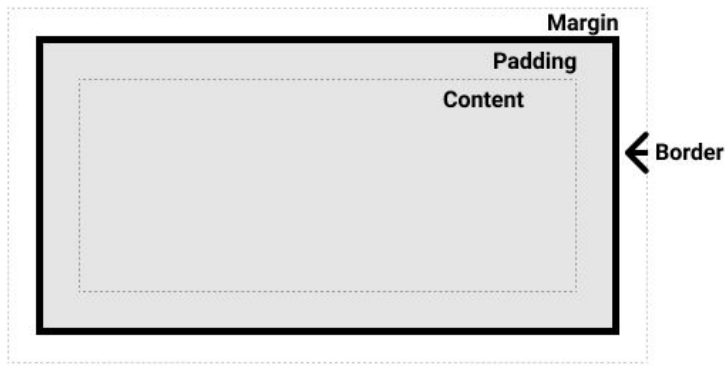
# Web Design

prof.ssa Cristina Iurissevich  
cristina.iurissevich@abacatania.it

# CSS Layout

# Box Model

— — —



## Content

L'area in cui vengono visualizzati i contenuti; può essere dimensionato utilizzando le proprietà `width` e `height`

## Padding

Il padding si trova attorno al contenuto come spazio vuoto.

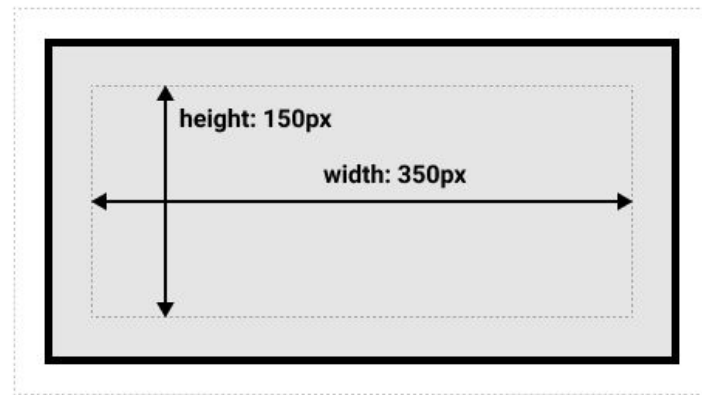
## Border

Il bordo che avvolge il contenuto e l'eventuale spaziatura interna

## Margin

Il margine è il livello più esterno che avvolge contenuto, riempimento e bordo e li distanzia da altri elementi

```
.box {  
  width: 350px;  
  height: 150px;  
  margin: 10px;  
  padding: 25px;  
  border: 5px solid black;  
}
```



Lo spazio effettivo occupato dal box sarà largo 410px ( $350 + 25 + 25 + 5 + 5$ ) e alto 210px ( $150 + 25 + 25 + 5 + 5$ ).

# Background

La proprietà **background-color** definisce il colore di sfondo su qualsiasi elemento nei CSS. La proprietà accetta qualsiasi <colore> valido. Un colore di sfondo si estende sotto il contenuto e il riquadro di riempimento dell'elemento.

```
.box {  
    background-color:red;  
}
```

# Background-image / Background-repeat

La proprietà **background-image** consente la visualizzazione di un'immagine sullo sfondo di un elemento.

```
.box {  
  background-image: url(star.png);  
  background-repeat: no-repeat;  
}
```

La proprietà **background-repeat** viene utilizzata per controllare il comportamento dell'affiancamento delle immagini. I valori disponibili sono:

- **no-repeat**: impedisce la ripetizione completa dello sfondo.
- **repeat-x**: ripeti orizzontalmente.
- **repeat-y**: ripeti verticalmente.
- **repeat**: l'impostazione predefinita; ripetere in entrambe le direzioni.

# Background-size

La proprietà **background-size** può assumere valori `<length>` o `<percentage>` che servono a ridimensionare l'immagine per adattarla allo sfondo.

Possiamo anche utilizzare i valori:

- **cover**: il browser renderà l'immagine abbastanza grande da coprire completamente l'area del riquadro pur mantenendo le proporzioni (parte dell'immagine potrebbe essere tagliata).
- **contain**: il browser renderà l'immagine della dimensione giusta per adattarsi alla casella (potrebbero esserci degli spazi vuoti su entrambi i lati o nella parte superiore e inferiore dell'immagine).

```
.box {  
  background-image: url(balloons.jpg);  
  background-repeat: no-repeat;  
  background-size: 100px 10em;  
}
```

# Background-position

La proprietà **background-position** permette di scegliere la posizione in cui appare l'immagine di sfondo. Questo utilizza un sistema di coordinate in cui l'angolo in alto a sinistra della casella è (0,0) e la casella è posizionata lungo gli assi orizzontale (x) e verticale (y).

Possiamo utilizzare valori come **top** e **right**, lunghezze e percentuali oppure un mix delle due.

```
.box {  
  background-image: url(star.png);  
  background-repeat: no-repeat;  
  background-position: 20px 10%;  
}
```

```
.box {  
  background-image: url(star.png);  
  background-repeat: no-repeat;  
  background-position: top center;  
}
```



# Border

In genere quando aggiungiamo bordi a un elemento con CSS utilizziamo una proprietà abbreviata che imposta la larghezza, lo stile e il colore del bordo in una riga di CSS.

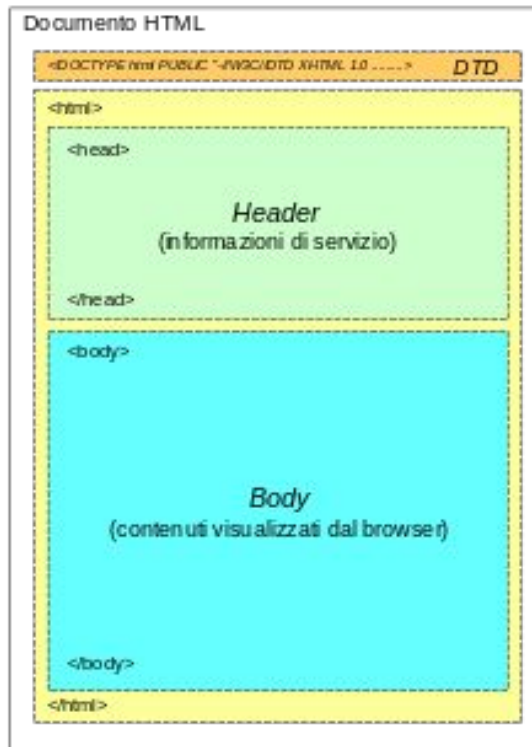
```
.box {  
  border: 1px solid black;  
}
```



```
.box {  
  border-width: 1px;  
  border-style: solid;  
  border-color: black;  
}
```

# Precisazioni

# Anatomia



`<!DOCTYPE html>` dichiara che questo è un foglio html.

`<html>` rappresenta l'inizio (e la fine del nostro codice).

```
1  <!DOCTYPE html>
2  <html lang="it">
3  >   <head> ...
5      </head>
6  >   <body> ...
15      </body>
16  </html>
```

In **<head>** abbiamo tutte le informazioni per il browser.

In **<body>** tutto ciò che verrà visualizzato all'interno della finestra del browser.

# Elementi <head>

```
<head>
  <!-- Codifica caratteri -->
  <meta charset="utf-8" />
  <!-- Il titolo visualizzato nella "linguetta" del browser -->
  <title>La mia pagina</title>
  <!-- Collegamento al foglio di stile -->
  <link rel="stylesheet" type="text/css" href="foglio-stile.css" />
  <!-- Aggiunta favicon -->
  <link rel="icon" href="favicon.ico" type="image/x-icon" />
  <!-- Adattabilità alla viewport -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
</head>
```

Ovviamente ce ne sono anche molti altri.

# Percorsi

---

Un collegamento deve portare ad una sola destinazione perciò nel nostro link (attributo href o src) dobbiamo indicare la posizione esatta del collegamento.

Si utilizza un **percorso assoluto** se il collegamento si trova sullo stesso sito web già pubblicato online o su un sito web esterno.

Si utilizza un **percorso relativo** se si trova sullo stesso sito web e vogliamo essere liberi di poter modificare l'indirizzo web.

# Percorsi assoluti

In un percorso assoluto si comunica al browser di andare a prendere la destinazione partendo direttamente dal dominio e aggiungendo la posizione del file che si vuole raggiungere.

Esempio:

<https://www.abacatania.it/elenco-professori/>

```
<p><a href="https://www.abacatania.it/elenco-professori/">Elenco professori</a></p>
```

# Percorsi relativi

Può capitare di fare riferimento a documenti situati all'interno del sito e non avere ancora un indirizzo web, oppure di dover spostare un sito da un'indirizzo a un altro e perciò non voler reimpostare da capo tutti i link. In questi casi è necessario utilizzare i percorsi relativi.

I percorsi relativi fanno riferimento alla posizione degli altri file rispetto al documento in cui ci si trova in quel momento.

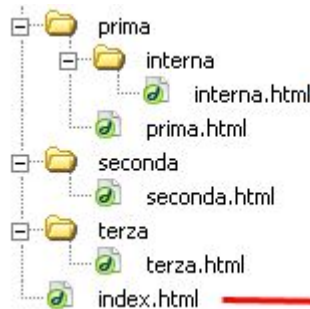
Per linkare due pagine che si trovano all'interno della stessa directory è sufficiente scrivere:

```
<a href="pagina2.html">Pagina 2</a>
```

# /sottodirectory

Per far riferimento a un file contenuto in una cartella di livello inferiore alla posizione corrente (cioé al file a cui vogliamo collegarlo), bisogna nominare la cartella (o le cartelle divise tra loro dallo /) seguita dal carattere / , e poi il nome del file.

Secondo la formula: **cartella/nomeFile.estensione**

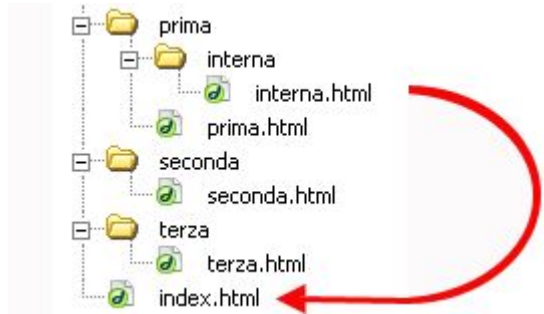


```
<a href="prima/interna/interna.html">Pagina interna</a>
```



## /directory di livello superiore

Se dalla pagina interna vogliamo far riferimento a una pagina (index.html) che si trova più in alto dobbiamo utilizzare la dicitura `../`



```
<a href="../../index.html">Pagina interna</a>
```

# Indentatura HTML

```
1  <!DOCTYPE html>
2  <html lang="it">
3      <head>
4          <title>Ciao Mondo!</title>
5      </head>
6      <body>
7          <header>
8              <h1>Ciao Mondo!</h1>
9          </header>
10         <main>
11             <section>
12                 <p>Questa è la nostra prima pagina HTML!</p>
13             </section>
14         </main>
15     </body>
16 </html>
```

# Indentatura CSS

— — —

```
1  body {
2    |   font-family: 'Roboto', sans-serif;
3    |   text-align: justify;
4    |
5    | }
6  section {
7    |   margin-left: 40px;
8    | }
9  .parent {
10   |   display: flex;
11   |   flex-direction: row;
12   |   flex-wrap: wrap;
13   |   align-items: center;
14   | }
15  .parent figure {
16   |   margin: 6px 6px 20px 6px;
17   | }
18  .parent img {
19   |   width: 430px;
20   |   height: 457px;
21   |   object-fit: cover;
22   | }
```

# Commenti

— — —

```
1  <!DOCTYPE html>
2  <html lang="it">
3  |   <!-- Questo è un commento, non influisce sul documento ma ci può aiutare -->
4  >   <head> ...
6  |   </head>
7  >   <body> ...
16 |   </body>
17 </html>
```

```
1  /*Questo è un commento, non influisce sul documento ma ci può aiutare*/
2  body {
3  |   font-family: 'Roboto', sans-serif;
4  |   text-align: justify;
5  |
6  | }
```

I commenti ci servono per segnare determinate note all'interno del nostro file senza andare a intaccare il codice.

In html il commento si scrive: `<!-- -->`

Nel css invece: `/* */`