CSS

Docente
Mario Perna
prof.perna.mario@darzo.net

A.S.
2025/2026
Materia
TEPSIT (Laboratorio)

Introduzione

L'acronimo **CSS** sta per **Cascading Style Sheets (fogli di stile a cascata)** e designa un linguaggio di stile per i documenti web. I CSS istruiscono un browser o un altro programma utente su come il documento debba essere presentato all'utente, per esempio definendone i font, i colori, le immagini di sfondo, il layout, il posizionamento delle colonne o di altri elementi sulla pagina, etc.

La storia dei CSS procede su binari paralleli rispetto a quelli di HTML, di cui vuole essere l'ideale **complemento**. Da sempre infatti, nelle intenzioni del W3C, HTML dovrebbe essere visto semplicemente come un linguaggio **strutturale**, alieno da qualunque scopo attinente la **presentazione** di un documento.

Strumenti necessari

Gli strumenti necessari per iniziare a sviluppare pagine web sono:

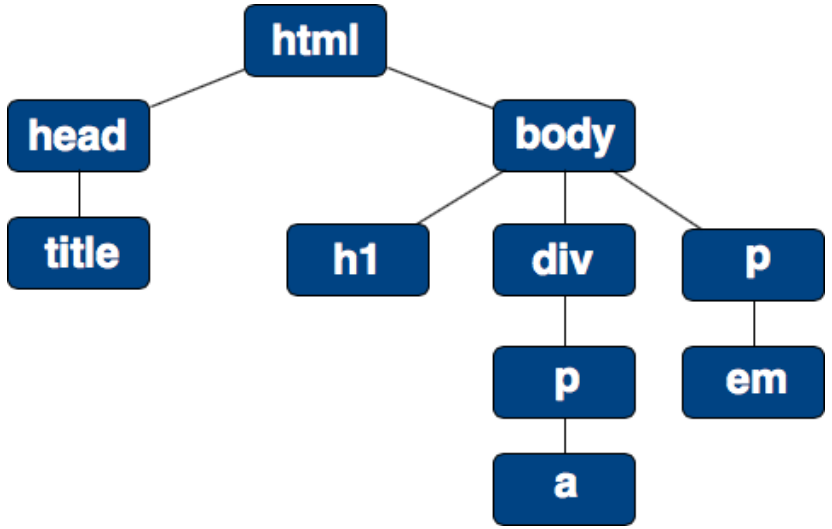
- Un browser per effettuare test, debug, valutare il risultato finale, etc. consiglio Google Chrome
- Un editor di testo, consiglio VS Code con le relative estensioni per il web development: https://code.visualstudio.com/download
- La bibbia quando si hanno dei dubbi: https://www.w3schools.com/ (tranne durante la verifica)

Struttura ad albero di un documento

Un concetto fondamentale da assimilare per una corretta applicazione dei CSS è quello della **struttura ad albero** di un documento. Il meccanismo fondamentale dei CSS è infatti **l'ereditarietà**. Esso fa sì che molte proprietà impostate per un elemento siano automaticamente ereditate dai suoi discendenti. Sapersi districare nella struttura ad albero significa padroneggiare bene questo meccanismo e sfruttare al meglio la potenza del linguaggio. Tutti i concetti che spiegheremo qui di seguito sono definiti nel cosiddetto **Document Object Model (DOM),** lo standard fissato dal W3C per la rappresentazione dei documenti strutturati.

Struttura ad albero di un documento

```
<html>
	<head>
		<title>Struttura del documento</title>
	</head>
	<body>
		<h1>Titolo</h1>
		<div>
			<p>Primo <a href="pagina.html">paragrafo</a>.</p>
		</div>
		<p>Secondo <em>paragrafo</em>.</p>
	</body>
</html>
```

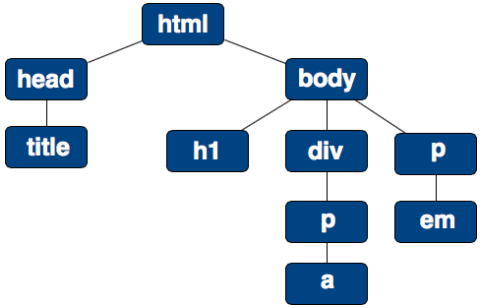


Struttura ad albero di un documento

Il documento è una perfetta forma di gerarchia ordinata in cui tutti gli elementi hanno tra di loro una relazione del tipo **genitore-figlio.** Ogni elemento è genitore e/o figlio di un altro.

Un elemento si dice **genitore** quando contiene altri elementi. Si dice **figlio** quando è racchiuso in un altro elemento. In base a queste semplici indicazioni possiamo analizzare il nostro documento.

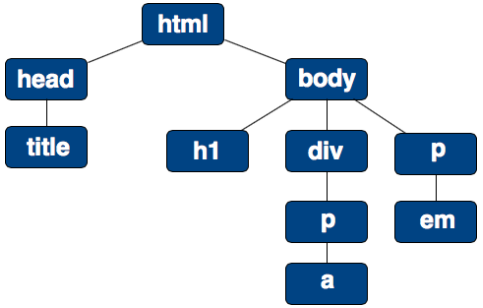
Ad esempio, <body> è figlio di <html>, ma è anche genitore di <h1>, e <p>. Quest'ultimo è a sua volta genitore di un elemento .



Struttura ad albero di un documento

Si potrebbe concludere che anche <body> sia in qualche modo genitore di . Non è esattamente così. Introduciamo ora un'altra distinzione, mutuata anch'essa dal linguaggio degli alberi genealogici, quella tra **antenato** e **discendente.**

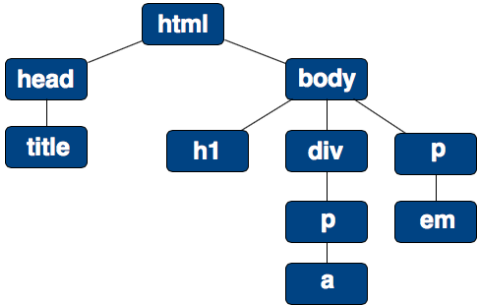
La relazione genitore-figlio è valida solo se tra un elemento e l'altro si scende di un livello. Esattamente come in un albero familiare si indica la relazione tra padre e figlio. Pertanto possiamo dire che <head> è figlio di <html>, che <a> è figlio di <p>, etc. Tra e <a>, invece, si scende di due livelli: diciamo allora che è un **antenato** di <a> e che questo è rispetto al primo un **discendente**.



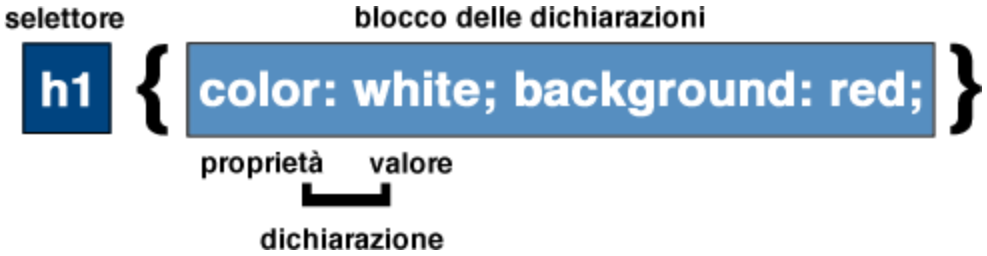
Struttura ad albero di un documento

L'albero del documento può essere letto non solo in senso verticale, ma anche orizzontale. In tal senso, gli elementi che sono posti sullo stesso livello, ovvero quelli che hanno lo stesso genitore, si dicono **fratelli**. Nel nostro esempio, h1, div e p sono fratelli rispetto all'elemento body.

Infine, c'è un solo elemento che racchiude tutti e non è racchiuso: <html>. Continuando con la metafora familiare potremmo dire che è il capostipite, ma in termini tecnici si dice che esso è l'elemento **radice**.



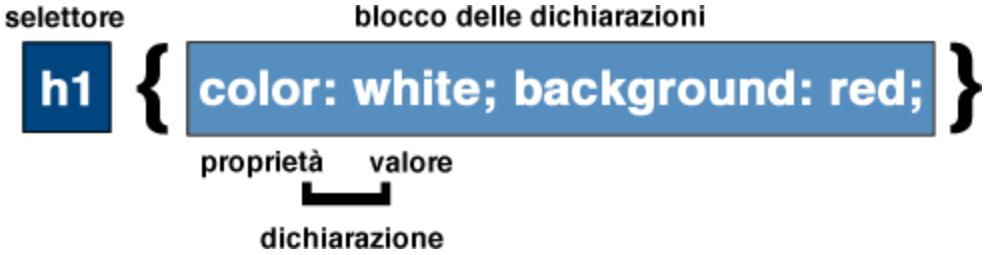
Regole CSS



L'illustrazione mostra la tipica struttura di una regola CSS. Essa è composta da due blocchi principali:

- **il selettore:** serve a definire la parte del documento cui verrà applicata la regola
- **il blocco delle dichiarazioni:** delimitato rispetto al selettore e alle altre regole da due parentesi graffe, la prima di apertura e la seconda di chiusura. Al suo interno possono trovare posto più dichiarazioni. Ognuna composta da un **nome di proprietà** e il relativo **valore**.

Regole CSS



La **proprietà** definisce un aspetto dell'elemento/selettore da modificare (margini, colore di sfondo, larghezza, etc.) secondo il valore espresso. Proprietà e valore devono essere separati dai due punti. Le dichiarazioni vanno invece separate con un punto e virgola. Non è obbligatorio, ma è buona norma mettere il punto e virgola anche dopo l'ultima dichiarazione del blocco.

Una limitazione fondamentale da rispettare è questa: per ogni dichiarazione non è possibile indicare più di una proprietà, mentre è spesso possibile specificare più valori. Questa regola è pertanto errata:

Regole CSS: esempi

```
/* Stili per il corpo della pagina*/
body {
	background: white;
	color: black;
}
/* Stili per i titoli h1 */
h1 {
	color: red;
	font: 36px Helvetica, Arial, sans-serif;
}
/* Colore del testo delle liste */
li {
	color: green;
}
```

I commenti in CSS

Le parti racchiuse tra i segni /* e */, rappresentano commenti.

```
/* Stili per i titoli h1 */
/* Colore del testo delle liste */
```

I commenti non sono interpretati dal browser. Sono utili nei CSS, come nei linguaggi di programmazione, per aggiungere annotazioni esplicative di vario tipo a beneficio di chi scrive e consulta il codice.

Inserire i fogli di stile CSS in un documento

Esistono tre modi per inserire le regole CSS in un documento HTML:

1. CSS **inline** (consigliato se si devono definire pochissime regole di stile)
2. CSS **interno** (consigliato se si devono definire un numero medio di regole di stile)
3. CSS **esterno** (consigliato se si devono definire un tante regole di stile)

Inserire i fogli di stile CSS: inline

Puoi aggiungere stili CSS direttamente all'interno di un elemento HTML utilizzando l'attributo style. Ad esempio:

```
<p style="color: blue; font-size: 16px;">
Questo è un paragrafo blu con un carattere di 16 pixel.
</p>
```

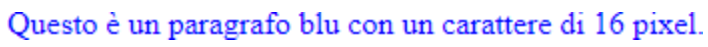
Questo è un paragrafo blu con un carattere di 16 pixel.

Tuttavia, l'utilizzo eccessivo di stili inline può rendere il codice HTML difficile da gestire e mischia il codice di **contenuto** della pagina con il codice di **presentazione (aspetto)** della pagina.

Inserire i fogli di stile CSS: interno

Per applicare stili a livello di pagina o per più elementi, è consigliabile utilizzare un elemento <style> all'interno della sezione <head> del documento HTML. Ad esempio:

```
<!DOCTYPE html>
<html>
<head>
    <style>
        p {
            color: blue;
            font-size: 16px;
        }
    </style>
</head>
<body>
    <p>Questo è un paragrafo blu con un carattere di 16 pixel.</p>
</body>
</html>
```



Inserire i fogli di stile CSS: esterno

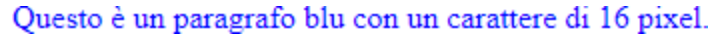
La pratica consigliata è separare il codice CSS in file esterni e quindi collegarli alle pagine HTML:

- Crea un file CSS separato con estensione .css, ad esempio "styles.css".
- All'interno del file CSS, definisci le regole di stile

```
/* styles.css */ p {
    color: blue; font-size:
    16px;
}
```

- Collega il file CSS alla pagina HTML utilizzando l'elemento <link> all'interno della sezione <head>

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <p>Questo è un paragrafo blu con un carattere di 16 pixel.</p>
</body>
</html>
```



Inserire i fogli di stile CSS: esterno

Attributo	Descrizione
`rel`	descrive il tipo di relazione tra il documento e il file collegato. È **obbligatorio**. Per i CSS due sono i valori possibili: `stylesheet` e `alternate stylesheet`.
`href`	serve a definire l'URL assoluto o relativo del foglio di stile. È **obbligatorio**
`type`	identifica il tipo di dati da collegare. Per i CSS il valore da usare è `text/css`. L'attributo non è più obbligatorio a partire dalla versione 5 del linguaggio HTML.

Esercizi 1

Esercizio 1
Inserisci CSS interno
Esercizio 2
Inserisci CSS esterno
Esercizio 3
Inserisci CSS inline
Esercizio 4
Dai uno sfondo ad un paragrafo dal tag style interno e cambia colore del testo con l'attributo style inline

Regole CSS: selettore universale

Il **selettore universale** serve a selezionare **tutti gli elementi** di un documento. Si esprime con il carattere * (asterisco).

```
* {color: red;}
```

La regola che abbiamo scritto assegna il colore rosso (red) a tutti gli elementi della pagina.

```
    <style>
        *{color: red}
    </style>
</head>
<body>
    <h1>Una mela danzante</h1>
    <p>Una mela rossa che balla</p>
</body>
```

Una mela danzante

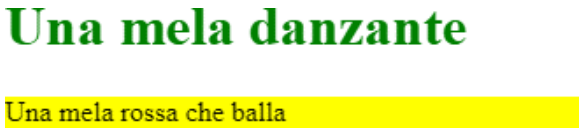
Una mela rossa che balla

Regole CSS: selettore di tipo o elemento

È costituito dal nome di uno **specifico elemento HTML**. Serve a selezionare **tutti gli elementi di quel tipo** presenti in un documento.

h1 {color: green;}
p {background-color: yellow;}

```
    <style>
        h1 {color: green;}
        p {background-color: yellow;}

    </style>
</head>
<body>
    <h1>Una mela danzante</h1>
    <p>Una mela rossa che balla</p>
</body>
```

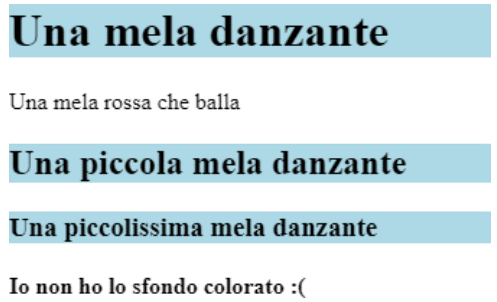


Regole CSS: raggruppare i selettori

È possibile nei CSS **raggruppare** diversi selettori al fine di semplificare il codice. I selettori raggruppati vanno separati da una **virgola**.

h1, h2, h3 {background: lightblue;}

```
    <style>
        h1, h2, h3 {background: lightblue;}

    </style>
</head>
<body>
    <h1>Una mela danzante</h1>
    <p>Una mela rossa che balla</p>
    <h2>Una piccola mela danzante</h2>
    <h3>Una piccolissima mela danzante</h3>
    <h4>Io non ho lo sfondo colorato :(</h4>
</body>
```



Esercizi 2
N.B. usare solo selettori di elemento

Esercizio 1
Crea una pagina HTML con diverse intestazioni e paragrafi Usa il selettore CSS per cambiare il colore del testo di tutte le intestazioni in blu e il colore del testo dei paragrafi in verde.

Esercizio 2
Crea una pagina HTML con un elenco non ordinato e un elenco ordinato, entrambi contenenti alcuni elementi di lista. Usa il selettore CSS per cambiare il colore del testo di tutti gli elementi di lista in rosso.

Esercizio 3
Crea una pagina HTML con un campo di input di tipo testo e un elemento di intestazione. Usa il selettore CSS per cambiare il colore del testo del campo di input in blu e il colore del testo dell'intestazione in verde.

Esercizio 4
Crea una pagina HTML con una tabella contenente alcune righe e colonne. Usa il selettore CSS per cambiare il colore del testo di tutte le celle della tabella in grigio.

Esercizio 5
Crea una pagina HTML con diversi paragrafi. Usa il selettore CSS per cambiare il colore del testo di tutti i paragrafi in blu.

Esercizio 6
Crea una pagina HTML con un link. Usa il selettore CSS per cambiare il colore del testo del link in arancione.

Regole CSS: selettore di classe e di id

I CSS non sarebbero uno strumento così potente senza questi tipi di **selettori**. *Id* e *classi* sono davvero una delle chiavi per sfruttare al meglio questo linguaggio. In HTML esistono due attributi globali applicabili a tutti gli elementi: sono ide class.

La differenza sta nel fatto che un **id** è assegnabile a **un solo elemento**, mentre una **classe** è assegnabile a **più di un elemento**.

```
    <style>
        .black_and_white{ /* . specifica che è una classe */
            color: white;
            background-color: black;
        }

        #special_text{ /* # specifica che è un id */
            font-style: italic;
        }

    </style>
</head>
<body>
    <h1 class="black_and_white" id="special_text">Una mela danzante</h1>
    <p class="black_and_white">Una mela rossa che balla</p>
</body>
```

Regole CSS: selettore di classe e di id

E' possibile essere più specifici nei selettori di classe id aggiungendo anche lo specifico elemento HTML che vogliamo modificare.

```
    <style>
        p.black_and_white{ /* . specifica che è una classe */
            color: white;
            background-color: black;
        }

        h1#special_text{ /* # specifica che è un id */
            font-style: italic;
        }

        p#special_color{
            color: yellow;
        }

    </style>
</head>
<body>
   <h1 class="black_and_white" id="special_text">Una mela danzante</h1>
   <p class="black_and_white" id="special_color">Una mela rossa che balla</p>
</body>
```

Una mela danzante

Una mela rossa che balla

Regole CSS: selettore di classe e di id

Possiamo anche combinare id con classi

```
    <style>
        p.black_and_white{ /* . specifica che è una classe */
            color: white;
            background-color: black;
        }

        h1#special_text{ /* # specifica che è un id */
            font-style: italic;
        }

        p#special_color{
            color: yellow;
        }

        h1#special_text.big_text{
            font-size: 50px;
        }

    </style>
</head>
<body>
   <h1 class="black_and_white big_text" id="special_text">Una mela danzante</h1>
   <p class="black_and_white" id="special_color">Una mela rossa che balla</p>
</body>
```

Esercizi 3
N.B. usare solo selettori di classe

Esercizio 1
Crea una pagina HTML con un elemento di intestazione e un elemento di paragrafo. Aggiungi una classe "titolo" all'elemento di intestazione e una classe "testo" all'elemento di paragrafo. Usa i selettori di classi CSS per cambiare il colore del testo del titolo in blu e il colore del testo del paragrafo in verde.

Esercizio 2
Crea una pagina HTML con una lista non ordinata contenente tre elementi di lista. Aggiungi la classe "elenco" a ciascun elemento di lista. Usa il selettore di classe CSS per cambiare il colore del testo di tutti gli elementi di lista in rosso.

Esercizio 3
Crea una pagina HTML con un campo di input di tipo testo `input type="text"` e un elemento di intestazione. Aggiungi la classe "campo-input" al campo di input e la classe "titolo" all'elemento di intestazione. Usa i selettori di classe CSS per cambiare il colore del testo del campo di input in blu e il colore del testo dell'intestazione in verde.

Esercizio 4
Crea una pagina HTML con una tabella contenente alcune righe e colonne. Aggiungi la classe "cella" a tutte le celle della tabella. Usa il selettore di classe CSS per cambiare il colore del testo di tutte le celle della tabella in grigio.

Esercizio 5
Crea una pagina HTML con tre paragrafi. Aggiungi la classe "paragrafo" a ciascun paragrafo. Usa il selettore di classe CSS per cambiare il colore del testo di tutti i paragrafi in blu.

Esercizi 4
N.B. usare solo selettori di id

Esercizio 1
Crea una pagina HTML con un elemento di intestazione e un elemento di paragrafo. Aggiungi l'ID "titolo" all'elemento di intestazione e l'ID "testo" all'elemento di paragrafo. Usa i selettori di ID CSS per cambiare il colore del testo del titolo in blu e il colore del testo del paragrafo in verde.

Esercizio 2
Crea una pagina HTML con una lista non ordinata contenente tre elementi di lista `li`. Aggiungi l'ID "elenco" a tutto l'elemento `ul`. Usa il selettore di ID CSS per cambiare il colore del testo di tutti gli elementi di lista in rosso.

Esercizio 3
Crea una pagina HTML con un campo di input di tipo testo `input type="text"` e un elemento di intestazione. Aggiungi l'ID "campo-input" al campo di input e l'ID "titolo" all'elemento di intestazione. Usa i selettori di ID CSS per cambiare il colore del testo del campo di input in blu e il colore del testo dell'intestazione in verde.

Esercizio 4
Crea una pagina HTML con una tabella contenente alcune righe e colonne. Aggiungi l'ID "tabella" a tutto l'elemento tabella. Usa il selettore di ID CSS per cambiare il colore del testo di tutte le celle della tabella in grigio.

Esercizio 5
Crea una pagina HTML con tre paragrafi. Aggiungi l'ID "paragrafo" a ciascun paragrafo. Usa il selettore di ID CSS per cambiare il colore del testo di tutti i paragrafi in blu.

Regole CSS: selettore di relazione o combinatorio

Una categoria fondamentale di selettori CSS è rappresentata dai cosiddetti **combinatori** (detti anche selettori di **relazione**). Hanno la funzione di mettere in relazione elementi presenti all'interno dell'albero del documento. Sono quattro:

Selettore	Simbolo
Selettore di discendenti	
Selettore di figli	>
Selettore di fratelli adiacenti	+
Selettore generale di fratelli	~

Regole CSS: selettore di discendenti

Seleziona un elemento che è **discendente** di un altro elemento. Ricordiamo che un elemento è discendente di un altro **se è contenuto al suo interno**, **a qualsiasi livello**.

```
    <style>
        #contenitore p{
            font-weight: bold;
            color: orange;
        }
    </style>
</head>
<body>
    <div id="contenitore">
        <div id="sotto-contenitore">
            <p>
                Sono del testo semplice
            </p>
        </div>
    </div>
</body>
```

Sono del testo semplice

Regole CSS: selettore di figli

Il selettore di **figli (>)** consente di selezionare **un elemento che è figlio diretto dell'elemento padre.**

```
    <style>
        #sotto-contenitore > p{
            font-style: italic;
            color: salmon;
        }
    </style>
</head>
<body>
    <div id="contenitore">
        <div id="sotto-contenitore">
            <p>
                Sono del testo semplice
            </p>
        </div>
        <p>
            Sono altro testo semplice
        </p>
    </div>
</body>
```

Sono del testo semplice

Sono altro testo semplice

Regole CSS: selettore di fratelli adiacenti

Il selettore di **fratelli adiacenti (+)** serve a scorrere in orizzontale l'albero del DOM assegnando le regole CSS agli **elementi che si trovano allo stesso livello di un altro elemento**.

```
    <style>
        #sotto-contenitore + p{
            font-style: italic;
            color: salmon;
        }
    </style>
</head>
<body>
    <div id="contenitore">
        <div id="sotto-contenitore">
            <p>
                Sono del testo semplice
            </p>
        </div>
        <p>
            Sono altro testo semplice
        </p>
        <p>
            Sono altro testo semplice
        </p>
    </div>
</body>
```

Sono del testo semplice

Sono altro testo semplice

Sono altro testo semplice

Regole CSS: selettore generale di fratelli

L'ultimo **combinatore (~)** è una generalizzazione di quello visto in precedenza. Esso assegna uno stile a **tutti gli elementi che sono fratelli**.

```
    <style>
        #sotto-contenitore ~ p{
            font-style: italic;
            color: salmon;
        }
    </style>
</head>
<body>
    <div id="contenitore">
        <div id="sotto-contenitore">
            <p>
                Sono del testo semplice
            </p>
        </div>
        <p>
            Sono altro testo semplice
        </p>
        <p>
            Sono altro testo semplice
        </p>
    </div>
</body>
```

Sono del testo semplice

Sono altro testo semplice

Sono altro testo semplice

Esercizi 5

Esercizio 1
Crea una pagina HTML con una lista non ordinata contenente tre elementi di lista. Usa il combinatore "child" CSS per selezionare solo gli elementi che sono figli diretti della lista, quindi cambia il colore del testo di questi elementi in rosso.

Esercizio 2
Crea una pagina HTML con un elemento di intestazione seguito da un elemento di paragrafo. Usa il combinatore "discendente" CSS per selezionare solo il paragrafo che è discendente dell'intestazione, quindi cambia il colore del testo del paragrafo in blu.

Esercizio 3
Crea una pagina HTML con una lista non ordinata contenente tre elementi di lista. Usa il combinatore "fratello" CSS per selezionare solo gli elementi che sono fratelli diretti tra loro, quindi cambia il colore del testo di questi elementi in verde.

Esercizio 4
Crea una pagina HTML con una tabella contenente alcune righe e colonne. Usa il combinatore "discendente" CSS per selezionare solo le celle che sono discendenti delle righe `tr`, quindi cambia il colore del testo di queste celle in blu.

Esercizio 5
Crea una pagina HTML con un elemento di intestazione seguito da un elemento di paragrafo. Usa il combinatore "discendente" CSS per selezionare solo il paragrafo che è discendente diretto dell'intestazione, quindi cambia il colore del testo del paragrafo in blu.

Esercizio 7
Crea una pagina HTML con una lista non ordinata contenente tre elementi di lista. Usa il combinatore "fratello" CSS per selezionare solo il secondo e il terzo elemento `li` che sono fratelli diretti tra loro, quindi cambia il colore del testo di questi elementi in verde.

Esercizio 8
Crea una pagina HTML con una tabella contenente alcune righe e colonne. Usa il combinatore "discendente" CSS per selezionare solo le celle `td` che sono discendenti delle righe `tr`, quindi cambia il colore del testo di queste celle in blu.

Esercizio 9
Crea una pagina HTML con un campo di input di tipo testo e un bottone. Usa il combinatore "fratello" CSS per selezionare solo il bottone che segue immediatamente il campo di input, quindi cambia il colore del testo del bottone in verde.

Regole CSS: selettore di attributo

I selettori di **attributo** servono a selezionare gli elementi in base ai loro **attributi e/o al valore di tali attributi.**

La specifica CSS 2.1 prevede quattro tipi. Altri tre tipi sono stati invece definiti nei CSS3.

Nelle prossime slide vedremo i più diffusi:

- Selezione **in base alla presenza di un attributo**
- Selezione **con valore corrispondente**
- Selezione **in base a valori che contengono una stringa**
- Selezione **in base a valori che iniziano con una certa stringa**

Selezione in base alla presenza di un attributo

Con il primo tipo di selettore di attributo si selezionano **tutti gli elementi** che presentino nel codice HTML **un determinato attributo, a prescindere dal valore dell'attributo stesso**. La sintassi è:
elemento[attributo] {dichiarazioni;}

```
    <style>
        a[title] {
            color: green;
            text-decoration: underline;
        }
    </style>
</head>
<body>
    <p>Ecco un <a href="#" title="link">link</a>.</p>
    <p>Ancora un <a href="#">link</a> ma senza l'attributo title.</p>
    <p>Un altro <a href="#" title="">link</a>, con title vuoto.</p>
    <p>Ultimo <a href="#" titles="link">link</a>.</p>
</body>
```

Ecco un link.

Ancora un link ma senza l'attributo title.

Un altro link, con title vuoto.

Ultimo link.

Selezione con valore corrispondente

Questo tipo di selettore individua **tutti gli elementi** che abbiano **come valore dell'attributo specificato la stringa di testo impostata nella regola CSS**. La sintassi è:
elemento[attributo="valore"] {dichiarazioni;}

```
    <style>
    a[title="lorem ipsum"] {
        color: green;
        text-decoration: underline;
    }
    </style>
</head>
<body>
    <p>Ecco un <a href="#" title="lorem ipsum">link</a>.</p>
    <p>Ancora un <a href="#" title="link">link</a>.</p>
    <p>Un altro <a href="#" title="loremipsum">link</a>.</p>
    <p>Ultimo <a href="#" title="LOREM IPSUM">link</a>.</p>
</body>
```

Ecco un link.

Ancora un link.

Un altro link.

Ultimo link.

Selezione in base a valori che contengono una stringa

Questo tipo seleziona **tutti gli elementi** con un **attributo che contenga una lista di parole separate da spazi, una delle quali corrisponde esattamente al valore definito nella regola CSS**. La sintassi è:
elemento[attributo~="valore"] {dichiarazioni;}

```
    <style>
    a[title~="lorem"] {
        color: green;
        text-decoration: underline;
    }
    </style>
</head>
<body>
    <p>Ecco un <a href="#" title="lorem ipsum sit">link</a>.</p>
    <p>Ancora un <a href="#" title="link lorem">link</a>.</p>
    <p>Un altro <a href="#" title="ipsum sit">link</a>.</p>
    <p>Ultimo <a href="#" title="LOREM link">link</a>.</p>
</body>
```

Ecco un link.

Ancora un link.

Un altro link.

Ultimo link.

Selezione in base a valori che iniziano con una certa stringa

Si tratta di un selettore poco utilizzato e di scarsa utilità. Individua **tutti gli elementi in cui uno specifico attributo contiene una lista di parole separate da trattini, una delle quali corrisponde al valore definito nella regola CSS**. La sintassi è: elemento[attributo|="valore"] {dichiarazioni;}

```
    a[title|="lorem"] {
        color: green;
        text-decoration: underline;
    }
    </style>
</head>
<body>
    <p>Ecco un <a href="#" title="lorem-ipsum-dolor">link</a>.</p>
    <p>Ancora un <a href="#" title="ipsum-lorem-dolor">link</a>.</p>
</body>
```

Ecco un link.

Ancora un link.

Esercizi 6

Esercizio 1
Crea una pagina HTML con un campo di input di tipo testo e un bottone. Utilizza il selettore di attributo CSS per selezionare solo il bottone che ha l'attributo "disabled" e cambia il colore del testo del bottone in rosso.

Esercizio 2
Crea una pagina HTML con una lista non ordinata contenente tre elementi di lista. Utilizza il selettore di attributo CSS per selezionare solo gli elementi che hanno l'attributo "class" con il valore "importante" e cambia il colore del testo di questi elementi in blu.

Esercizio 3
Crea una pagina HTML con un campo di input di tipo email e un campo di input di tipo password. Utilizza il selettore di attributo CSS per selezionare solo il campo di input email, quindi cambia il colore del testo di questo campo in verde.

Esercizio 4
Crea una pagina HTML con un campo di input di tipo testo e un bottone. Utilizza il selettore di attributo CSS per selezionare solo il bottone che ha l'attributo "type" con il valore "submit", quindi cambia il colore del testo del bottone in verde.

Esercizio 5
Crea una pagina HTML con una tabella contenente alcune righe e colonne. Utilizza il selettore di attributo CSS per selezionare solo le celle che hanno l'attributo "data-value" e il valore "important", quindi cambia il colore del testo di queste celle in blu.

Le pseudo-classi

Il concetto di **pseudo-classe** è di assoluto rilievo nel contesto dei CSS. Una pseudo-classe non definisce la presentazione di un elemento ma di un **particolare stato di quest'ultimo**. In buona sostanza, grazie alle pseudo-classi possiamo impostare uno stile per un elemento al verificarsi di certe condizioni.

Per dichiarare una pseudo-classe si usa questa sintassi:

```
selettore:pseudo-classe {dichiarazioni;}
```

Le pseudo-classi per i link

Le prime due **pseudo-classi** che prendiamo in considerazione sono associate ai **link**, ovvero all'elemento a. Servono a impostare l'aspetto dei collegamenti ipertestuali presenti in una pagina:

- **:link**

La pseudo-classe :link definisce l'aspetto di partenza dei link, ovvero quello che ci troviamo davanti quando non sono stati visitati.

- **:visited**

La pseudo-classe :visited serve a modificare l'aspetto dei link quando siano stati visitati.

```
a:link {color: blue; text-decoration: none; font-weight: bold;}
a:visited {color: green;}
```

Le pseudo-classi dinamiche

Il secondo gruppo di **pseudo-classi** definite nella specifica è quello delle cosiddette pseudo-classi **dinamiche**. Anch'esse servono a impostare e modificare la presentazione di un particolare stato di un elemento, ma ciò avviene **in seguito ad un'azione dell'utente**.

Pur trovando un campo di applicazione molto comune a livello dei link, possono essere applicate anche ad altri tipi di elementi, **non solo dunque all'elemento a:**

- **`:hover`**

Questa pseudo-classe viene applicata quando si passa con il cursore del mouse (o altro dispositivo di puntamento) su un elemento senza attivarlo, ovvero senza cliccarci sopra

- **`:active`**

Questa pseudo-classe serve a impostare la presentazione di un elemento quando e mentre esso viene attivato dall'utente.

- **`:focus`**

Questa pseudo-classe viene attivata quando un elemento riceve il focus.

Le pseudo-classi dinamiche

```
a:link {color: blue; text-decoration: none; font-weight: bold;}
a:visited {color: green;} a:hover {color: orange;}
a:active {color: red;} input:focus {background:
yellow;}
```

Essendo dinamiche, per essere apprezzate devono essere provate in diretta ;)

Le pseudo-classi

Infine esistono altre psudo-classi come:

- :first-child il primo elemento figlio di un altro elemento.

```
    <style>

        p:first-child{
            color: red;
            font-size: 25px;
        }

        p:nth-child(3){
            color: green;
            font-weight: bold;
        }
    </style>
</head>
<body>
    <div>
        <p>Paragrafo 1</p>
        <p>Paragrafo 2</p>
        <p>Paragrafo 3</p>
        <p>Paragrafo 4</p>
    </div>
</body>
```

Paragrafo 1

Paragrafo 2

Paragrafo 3

Paragrafo 4

Le pseudo-classi

Infine esistono altre psudo-classi come:

- :first-letter la prima lettera di qualunque elemento contenente del testo

```
    <style>

        p:first-letter{
            color: red;
            font-size: 25px;
        }

    </style>
</head>
<body>
    <div>
        <p>Paragrafo 1</p>
        <p>Paragrafo 2</p>
        <p>Paragrafo 3</p>
        <p>Paragrafo 4</p>
    </div>
</body>
```

Paragrafo 1

Paragrafo 2

Paragrafo 3

Paragrafo 4

Le pseudo-classi

Infine esistono altre psudo-classi come:

- :first-line la prima riga di un elemento contenente del testo

```
    <style>

        p:first-line{
            color: red;
            font-size: 25px;
        }

    </style>
</head>
<body>
    <div>
        <p>Paragrafo 1: tante cose da dire e così poco spazio per raccontarle tutte,
        accidenti se soltanto avessi più spazio...</p>
        <p>Paragrafo 2: tante cose da dire e così poco spazio per raccontarle tutte,
            accidenti se soltanto avessi più spazio...</p>
        <p>Paragrafo 3: tante cose da dire e così poco spazio per raccontarle tutte,
            accidenti se soltanto avessi più spazio...</p>
        <p>Paragrafo 4: tante cose da dire e così poco spazio per raccontarle tutte,
            accidenti se soltanto avessi più spazio...</p>
    </div>
</body>
```

Paragrafo 1: tante cose da dire e così poco spazio per raccontarle tutte, accidenti se soltanto avessi più spazio...

Paragrafo 2: tante cose da dire e così poco spazio per raccontarle tutte, accidenti se soltanto avessi più spazio...

Paragrafo 3: tante cose da dire e così poco spazio per raccontarle tutte, accidenti se soltanto avessi più spazio...

Paragrafo 4: tante cose da dire e così poco spazio per raccontarle tutte, accidenti se soltanto avessi più spazio...

Gli pseudo-elementi

Esistono delle varianti delle pseudo-classi chiamati **pseudo-elementi** che, in aggiunta, permettono di inserire del contenuto agli elementi HTML già esistenti.

A differenza delle pseudo-classi, la sintassi utilizza i simboli :: e non :

```
selettore::pseudo-elemento {content: "..."; dichiarazioni;}
```

Gli pseudo-elementi

Infine esistono pseudo-elementi molto diffusi come:

- ::before per inserire del contenuto prima del contenuto dell'elemento desiderato con il relativo CSS
- ::after per inserire del contenuto dopo il contenuto dell'elemento desiderato con il relativo CSS

```
    <style>

        p::before{
            content: "--> ";
            background-color: yellow;


        }

        p::after{
            content: " <--";
            background-color: cyan;
        }

    </style>
</head>
<body>
    <div>
        <p>Paragrafo 1</p>
        <p>Paragrafo 2</p>
        <p>Paragrafo 3</p>
        <p>Paragrafo 4</p>
    </div>
</body>
```

--> Paragrafo 1 <--

--> Paragrafo 2 <--

--> Paragrafo 3 <--

--> Paragrafo 4 <--

Esercizi 7

Esercizio 1
Crea una pagina HTML con una lista non ordinata contenente diversi elementi. Applica una pseudo-classe CSS in modo che gli elementi diventino rossi quando il cursore del mouse vi passa sopra.
Esercizio 2
Crea una pagina HTML con un link e applica una pseudo-classe CSS in modo che il colore del link diventi verde quando viene visitato.
Esercizio 3
Crea una pagina HTML con un pulsante. Applica una pseudo-classe CSS in modo che il colore del pulsante diventi blu quando il cursore del mouse vi passa sopra e rosso quando il pulsante è cliccato.
Esercizio 4
Crea una pagina HTML con una tabella contenente diverse righe. Applica una pseudo-classe CSS in modo che le righe pari della tabella abbiano un colore di sfondo grigio.
Esercizio 5
Crea una pagina HTML con una lista non ordinata contenente diversi elementi. Applica una pseudo-classe CSS in modo che gli elementi diventino grigi quando sono selezionati.
Esercizio 6
Crea una pagina HTML con una lista ordinata contenente diverse righe. Applica una pseudo-classe CSS in modo che il testo delle righe pari diventi grassetto.
Esercizio 7
Crea una pagina HTML con una lista non ordinata contenente diversi elementi. Utilizza un pseudo-elemento CSS per aggiungere delle parentesi graffe intorno a ogni elemento della lista.

Ereditarietà degli stili

Meccanismo con cui le impostazioni di stile applicate ad un elemento vengono ereditate anche dai suoi discendenti. Almeno fino a quando, per un elemento discendente, non si imposta esplicitamente un valore diverso per quella proprietà

```
    <style>

        body{
            color: green;
            text-decoration: underline;
        }

        p { /* sostituisce lo stile del body */
            color: red;
        }
    </style>
</head>
<body>
    <div>
        <p>Paragrafo</p>
    </div>
</body>
```

Paragrafo

Priorità degli stili

Meccanismo con cui le impostazioni di stile applicate ad un elemento hanno un ordine di priorità e in caso di conflitti prevale lo stile con priorità maggiore. Le variabili che contribuiscono al calcolo di questa priorità sono riassunti mediante la tabella sottostante:

Selector	Specificity Value	Calculation
p	1	1
p.test	11	1 + 10
p#demo	101	1 + 100
<p style="color: pink;">	1000	1000
#demo	100	100
.test	10	10
p.test1.test2	21	1 + 10 + 10
#navbar p#demo	201	100 + 1 + 100
*	0	0 (the universal selector is ignored)

Il box model

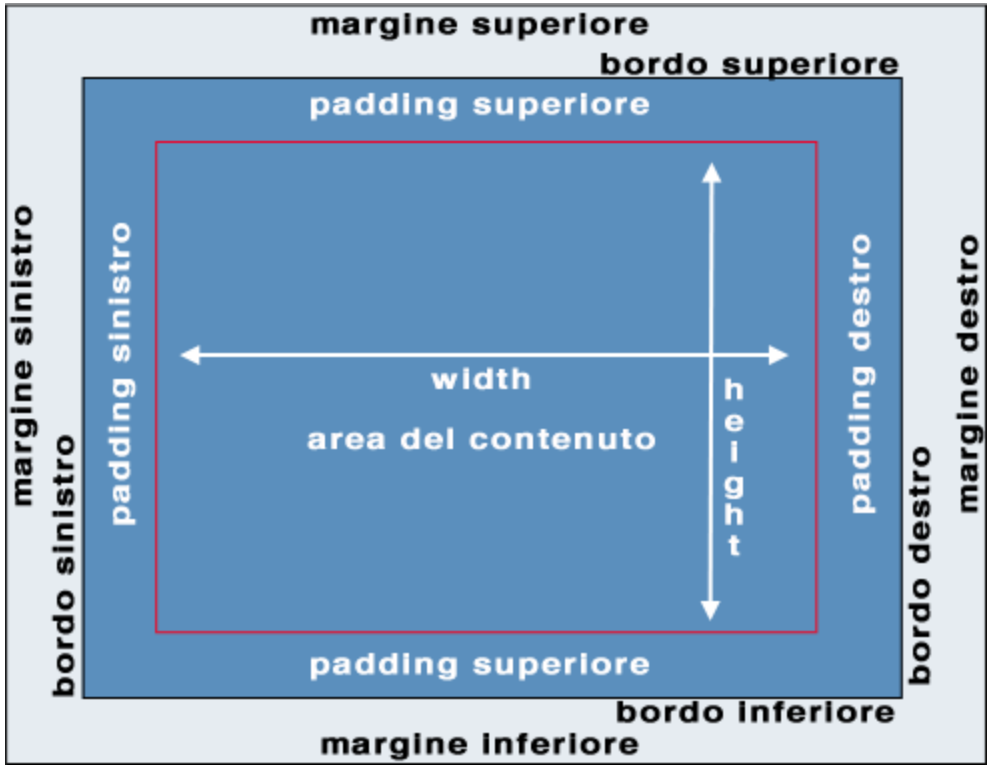
Dopo aver preso in esame i selettori, è giunto il momento di iniziare l'analisi delle proprietà CSS. Prime però è necessario soffermarsi su un fondamentale argomento propedeutico: il **box model**.

Si tratta del meccanismo che governa la presentazione dei vari elementi di una pagina. Abbiamo visto che una pagina HTML non è altro che un insieme di box rettangolari, che si tratti di elementi **blocco** o di elementi **inline**.

Elementi CSS block-level	Elementi CSS inline
• Un elemento block-level può contenere altri elementi block-level e anche elementi inline, mentre un elemento inline può contenere solo altri elementi inline • Ad un elemento block-level si possono attribuire delle dimensioni • Un elemento block level di dimensioni non specificate occupa tra margini, bordi, padding e contenuto, tutta la larghezza messa a disposizione del suo box contenitore. In verticale occuperà l'altezza necessaria al suo contenuto	• Ad un elemento inline, a meno che questo non venga dichiarato float, posizionato o modificandone la sua natura con la proprietà display, non si possono attribuire delle dimensioni • Elementi inline adiacenti vengono disposti orizzontalmente, mentre elementi blocco vengono disposti verticalmente • Un elemento inline occuperà sia in orizzontale che in verticale l'altezza necessaria al suo contenuto

Il box model

Ogni box comprende un certo numero di componenti di base, ciascuno modificabile con proprietà dei CSS. La figura qui sotto mostra visivamente tali componenti:



Partendo dall'interno abbiamo:

- **L'area del contenuto:** zona in cui trova spazio il contenuto vero e proprio: testo, immagini, video, etc. Le dimensioni orizzontali dell'area possono essere modificate con la proprietà width. Quelle verticali con height.
- **Il padding:** spazio vuoto che può essere creato tra l'area del **contenuto** e il **bordo** dell'elemento. Come nella figura, se si imposta un colore di sfondo per un elemento, esso si estende dall'area del contenuto alla zona di padding.
- **Il bordo**: linea di dimensione, stile e colore variabile che circonda la zona del padding e l'area del contenuto.
- **Il margine**: spazio di dimensioni variabili che separa un dato elemento da quelli adiacenti.

Il box model (contenuto)

E' possibile modificare le **dimensioni** del **contenuto** con gli attributi width(larghezza) e height (altezza).
Alcuni dei possibili formati accettati sono dei valori espressi in pxo in percentuale.

```
<head>
    <style>
        div{
            width: 300px;
            height: 200px;
            border: 1px solid black;
        }
    </style>
</head>
<body>

    <div>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Aenean pretium ullamcorper sagittis. Nunc laoreet enim nisi,
        ut congue nibh luctus quis. Nunc hendrerit odio et elit bibendum ornare.
        Donec tincidunt faucibus sapien non pharetra. Vivamus laoreet lacus at nisi finibus,
        id vehicula libero elementum. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Fusce ut ligula leo. Mauris interdum fermentum rutrum.
    </div>
</body>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean pretium ullamcorper sagittis. Nunc laoreet enim nisi, ut congue nibh luctus quis. Nunc hendrerit odio et elit bibendum ornare. Donec tincidunt faucibus sapien non pharetra. Vivamus laoreet lacus at nisi finibus, id vehicula libero elementum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce ut ligula leo. Mauris interdum fermentum rutrum.

Il box model (contenuto)

E' anche possibile modificare le **dimensioni massime** e **minime** del **contenuto** con gli attributi [max/min]-width (larghezza) e [max/min]- height (altezza).
Alcuni dei possibili formati accettati sono dei valori espressi in pxo in percentuale.

```
<head>
    <style>
        div{
            min-width: 300px;
            max-width: 500px;
            min-height: 200px;
            max-height: 300px;
            border: 1px solid black;
        }
    </style>
</head>
<body>

    <div>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Aenean pretium ullamcorper sagittis. Nunc laoreet enim nisi,
        ut congue nibh luctus quis. Nunc hendrerit odio et elit bibendum ornare.
        Donec tincidunt faucibus sapien non pharetra. Vivamus laoreet lacus at nisi finibus,
        id vehicula libero elementum. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Fusce ut ligula leo. Mauris interdum fermentum rutrum.
    </div>
</body>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean pretium ullamcorper sagittis. Nunc laoreet enim nisi, ut congue nibh luctus quis. Nunc hendrerit odio et elit bibendum ornare. Donec tincidunt faucibus sapien non pharetra. Vivamus laoreet lacus at nisi finibus, id vehicula libero elementum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce ut ligula leo. Mauris interdum fermentum rutrum.

Il box model (contenuto) – overflow

Quando il contenuto eccede dalle sue dimensioni esplicite, possono essere applicate diverse politiche mediante il valore dell'attributo overflow

```
<head>
    <style>
        div{
            width: 300px;
            height: 100px;
            border: 1px solid black;
        }
    </style>
</head>
<body>

    <div>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Aenean pretium ullamcorper sagittis. Nunc laoreet enim nisi,
        ut congue nibh luctus quis. Nunc hendrerit odio et elit bibendum ornare.
        Donec tincidunt faucibus sapien non pharetra. Vivamus laoreet lacus at nisi finibus,
        id vehicula libero elementum. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Fusce ut ligula leo. Mauris interdum fermentum rutrum.
    </div>
</body>
```

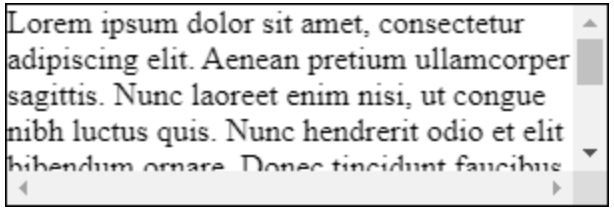
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean pretium ullamcorper sagittis. Nunc laoreet enim nisi, ut congue nibh luctus quis. Nunc hendrerit odio et elit bibendum ornare. Donec tincidunt faucibus sapien non pharetra. Vivamus laoreet lacus at nisi finibus, id vehicula libero elementum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce ut ligula leo. Mauris interdum fermentum rutrum.

Il box model (contenuto)

I valori possono essere espressi con le parole chiave:

- visible:valore iniziale, il contenuto eccedente rimane visibile
- hidden: il contenuto eccedente non viene mostrato
- scroll:il browser crea barre di scorrimento che consentono di fruire del contenuto eccedente
- auto:il browser tratta il contenuto eccedente secondo le sue impostazioni predefinite; di norma dovrebbe mostrare una barra di scorrimento laterale.

```
<head>
    <style>
        div{
            width: 300px;
            height: 100px;
            border: 1px solid black;
            overflow: scroll;
        }
    </style>
</head>
<body>

    <div>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Aenean pretium ullamcorper sagittis. Nunc laoreet enim nisi,
        ut congue nibh luctus quis. Nunc hendrerit odio et elit bibendum ornare.
        Donec tincidunt faucibus sapien non pharetra. Vivamus laoreet lacus at nisi finibus,
        id vehicula libero elementum. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Fusce ut ligula leo. Mauris interdum fermentum rutrum.
    </div>
</body>
```



Il box model (padding)

Esiste uno spazio tra il contenuto dell'elemento e il suo bordo. Tale spazio viene chiamato **padding**.

E' possibile specificare la dimensione del padding mediante la seguente dichiarazione (dove valore può essere espresso in px o in percentuale):

```
selettore {padding: valore;}
```

Oppure specificare sono il padding di uno dei quattro lati dell'elemento mediante:

```
selettore {
        padding-top: <valore>; padding-bottom: <valore>; padding-left: <valore>;
        padding-right: <valore>;
}
```

Il box model (padding)

```
7       <style>
8           div{
9               background-color: lightblue;
10              padding: 20px;
11          }
12      </style>
13  </head>
14  <body>
15      <div>Sono un paragrafo!</div>
16  </body>
```


```
7       <style>
8           div{
9               background-color: lightblue;
10              padding-top: 20px;
11          }
12      </style>
13  </head>
14  <body>
15      <div>Sono un paragrafo!</div>
16  </body>
```

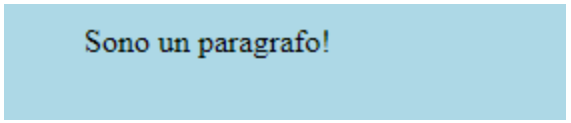
Il box model (padding)

Inoltre è possibile specificare i diversi padding nel seguente modo:

selettore {padding: <valore-1> <valore-2> <valore-3> <valore-4>;}

- se si usano tre valori, il primo si riferisce al margine superiore, il secondo a quelli sinistro e destro, il terzo a quello inferiore
- se si usano due valori, il primo si riferisce ai lati superiore e inferiore, il secondo al sinistro e al destro
- se si usa un solo valore, un uguale distanza sarà applicata ai quattro lati.

```
7       <style>
8           div{
9               background-color: lightblue;
10              padding: 10px 20px 30px 40px;
11          }
12      </style>
13  </head>
14  <body>
15      <div>Sono un paragrafo!</div>
16  </body>
```



Il box model (border)

La definizione dei bordi con i CSS può risultare a prima vista un po' intricata per il numero di proprietà coinvolte: se infatti consentono all'autore la massima flessibilità, rendono a volte complicata la gestione del codice.

In linea di massima possiamo suddividere le proprietà relative ai bordi in due categorie: **proprietà singole** e **proprietà a sintassi abbreviata**. Le prime definiscono singoli aspetti di ciascuno dei quattro bordi. Le seconde consentono di riunire in una sola regola le diverse impostazioni.

Esempio di proprietà singole: border-top-color, border-top-style, border-top- width, border-bottom-color, ...

Esempio di proprietà a sintassi abbreviata: border, border-bottom, border-top, ...

Il box model (border)

Sintassi per modificare le singole proprietà di un bordo (colore, stile e spessore):

```
selettore {
border-<lato>-color: <valore>; border-<lato>-
style: <valore>; border-<lato>-width:
<valore>;
}
```

O in modo abbreviato:

```
selettore {
    border-<lato>: <valore width> <valore style> <valore color>;
}
```

Il box model (border)

```
7       <style>
8           div{
9               background-color: lightblue;
10              border-top-style: solid;
11              border-top-width: 5px;
12              border-top-color: blue;
13          }
14      </style>
15  </head>
16  <body>
17      <div>Sono un paragrafo!</div>
18  </body>
```

Sono un paragrafo!

```
7       <style>
8           div{
9               background-color: lightblue;
10              border-bottom: 3px dotted red;
11          }
12      </style>
13  </head>
14  <body>
15      <div>Sono un paragrafo!</div>
16  </body>
```

Sono un paragrafo!

Il box model (border)

E’ anche possibile definire le proprietà per tutti e quattro i bordi:

selettore {
    border: <valore width> <valore style> <valore color>;
}

```
7       <style>
8           div{
9               background-color: lightblue;
10              border: 5px dashed purple;
11          }
12      </style>
13  </head>
14  <body>
15      <div>Sono un paragrafo!</div>
16  </body>
```

Il box model (border)

E’ anche possibile definire le proprietà per tutti e quattro i bordi:

selettore {
    border: <valore width> <valore style> <valore color>;
}

```
7       <style>
8           div{
9               background-color: lightblue;
10              border: 5px dashed purple;
11          }
12      </style>
13  </head>
14  <body>
15      <div>Sono un paragrafo!</div>
16  </body>
```

Il box model (border)

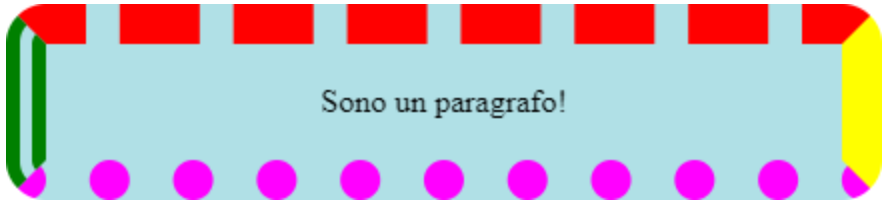
Gli stili supportati per i bordi sono i seguenti:

Stile bordo	Descrizione
`none`	l'elemento non presenta alcun bordo e lo spessore equivale a 0
`hidden`	equivalente a `none`
`dotted`	bordo a puntini
`dashed`	bordo a lineette
`solid`	bordo solido e continuo
`double`	bordo solido, continuo e doppio
`groove`	tipo di bordo in rilievo
`ridge`	altro tipo di bordo in rilievo
`inset`	effetto 'incastonato'
`outset`	effetto 'sbalzato'

Il box model (border)

Combinando assieme tutte queste caratteristiche posso ottenere anche effetti molto «esotici»

```
7       <style>
8           #super_bordo{
9               text-align: center;
10              border-top: 20px dashed red;
11              border-bottom: 20px dotted magenta;
12              border-left: 20px double green;
13              border-right: 20px solid yellow;
14              border-radius: 20px;
15              margin: 50px;
16              padding: 20px;
17              background-color: powderblue;
18          }
19      </style>
20  </head>
21  <body>
22      <div id="super_bordo">Sono un paragrafo!</div>
23  </body>
```



Il box model (margini)

Nei CSS, le proprietà legate ai margini consentono di impostare con precisione la **distanza tra gli elementi**. Possiamo infatti definire il margine come la **distanza tra il bordo di un elemento e gli elementi adiacenti**.

Sono cinque le proprietà con cui è possibile definire un margine. Quattro di esse sono **singole** e impostano la distanza per ciascun lato del box. L'ultima, margin, è una **proprietà a sintassi abbreviata** utile per definire con una sola dichiarazione tutte le impostazioni per i quattro lati.

I possibili valori possono essere:

- espressi in px
- espressi in percentuale
- auto(cerca di centrare orizzontalmente l'elemento se ha una larghezza)

Il box model (margini)

Sintassi per modificare un singolo margine:

```
selettore {
margin-top: <valore>; margin-
bottom: <valore>; margin-left:
<valore>; margin-right: <valore>;
}
```

O in modo abbreviato per tutti e quattro i margini:

```
selettore {margin: <valore>;}
```

Il box model (margini)

```
7       <style>
8           div{
9               width: 100px;
10              height: 100px;
11              background-color: lightblue;
12              border: 1px solid black;
13              margin: 100px;
14          }
15      </style>
16  </head>
17  <body>
18      <div>Testo1</div>
19      <div>Testo2</div>
20      <div>Testo3</div>
```

Il box model

Possiamo definire la dimensione di un elemento come:

- Larghezza = width + padding + border
- Altezza = height + padding + border

Esiste una proprietà che aiuta a calcolare le dimensioni effettive di un elemento senza aggiungere il contributo di padding e border al contenuto.

Atteverso la proprietà box-sizing: border-box di default impostata come box-sizing: content-box

Esercizi 8

Esercizio 1
Crea una pagina HTML con un paragrafo all'interno. Usa il CSS per impostare un margine esterno di 20px e un padding interno di 10px al paragrafo.
Esercizio 2
Crea una pagina HTML con un link all'interno. Usa il CSS per impostare un margine interno di 5px e un padding esterno di 15px al link.
Esercizio 3
Crea una pagina HTML con un div all'interno. Usa il CSS per impostare un margine esterno di 30px e un padding interno di 10px al div.
Esercizio 4
Crea una pagina HTML con una lista non ordinata (ul) contenente alcuni elementi di lista (li). Usa il CSS per impostare un margine interno di 5px e un padding esterno di 10px agli elementi di lista.
Esercizio 5
Crea una pagina HTML con un titolo all'interno. Usa il CSS per impostare un margine esterno di 40px e un padding interno di 10px al titolo.
Esercizio 6
Crea una pagina HTML con un link all'interno. Usa il CSS per impostare un margine interno di 5px e un padding esterno di 15px al link, con il testo in grassetto.

Esercizio 7
Crea una pagina HTML con un paragrafo all'interno. Usa il CSS per impostare un bordo blu di 2px solo nella parte superiore del paragrafo.
Esercizio 8
Crea una pagina HTML con un link all'interno. Usa il CSS per impostare un bordo rosso di 1px con raggio di 5px solo intorno al link.
Esercizio 9
Crea una pagina HTML con un div all'interno. Usa il CSS per impostare un bordo nero di 3px con raggio di 10px solo nella parte inferiore del div.
Esercizio 10
Crea una pagina HTML con una lista non ordinata (ul) contenente alcuni elementi di lista (li). Usa il CSS per impostare un bordo verde di 2px solo a destra degli elementi di lista.
Esercizio 11
Crea una pagina HTML con un titolo all'interno. Usa il CSS per impostare un bordo tratteggiato nero di 1px solo nella parte inferiore del titolo.

Gestione del colore

I colori possono essere espressi in vari modi nel contesto di una regola CSS:

- Parola chiave

black | navy | blue | maroon | purple | green | red | teal | fuchsia | olive | gray | lime | aqua | silver | yellow | white …

- Notazione esadecimale: #RRGGBB

color: #CC0000

- Notazione RGB

color: #C00

- Notazione decimale con RGB

color: rgb(0, 0, 0)

Gestione dello sfondo

Ecco la lista delle proprietà per lo sfondo:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

Ciascuna di essa definisce un solo, particolare aspetto relativo allo sfondo di un elemento. La proprietà background, invece, è una proprietà a sintassi abbreviata con cui possiamo definire sinteticamente e con una sola dichiarazione tutti i valori per lo sfondo. La analizzeremo alla fine. Prima è necessario chiarire significato e sintassi delle proprietà singole.

Gestione dello sfondo: background-color

Definisce il colore di sfondo di un elemento. Questa proprietà non è ereditata.

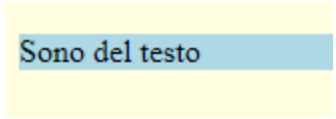
Sintassi generale:
selettore {background-color: valore;}

I valori possone essere un qualunque colore o la parola chiave `transparent`.

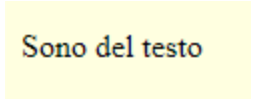
Usando `transparent`come valore un elemento avrà come colore quello dell'elemento parente.

Gestione dello sfondo: background-color

```
7       <style>
8           p{
9               background-color: lightblue;
10          }
11
12          body{
13              background-color: lightyellow;
14          }
15      </style>
16  </head>
17  <body>
18      <p>Sono del testo</p>
19  </body>
```




```
7       <style>
8           p{
9               background-color: transparent;
10          }
11
12          body{
13              background-color: lightyellow;
14          }
15      </style>
16  </head>
17  <body>
18      <p>Sono del testo</p>
19  </body>
```



Gestione dello sfondo: background-image

Definisce l'URL di un'immagine da usare come sfondo di un elemento. Questa proprietà non è ereditata.

Sintassi generale:
selettore {background-image: url(valore);}

I valori possono essere un URL assoluto/relativo o la parola chiave none. Valore di default.

Usare noneequivale a non impostare la proprietà: nessuna immagine verrà applicata come sfondo.

Un consiglio. Una buona norma e il buon senso vogliono che quando si definisce un'immagine come sfondo si usi sempre anche un colore di sfondo e che questo colore consenta una lettura agevole del testo.

Gestione dello sfondo: background-image

```
7       <style>
8           p{
9               background-image: url(bumblebee.png);
10              color: white;
11              width: 500px;
12              height: 500px;
13              border: 1px solid black;
14          }
15      </style>
16  </head>
17  <body>
18      <p>Sono del testo</p>
19  </body>
```

Gestione dello sfondo: background-repeat

Con questa proprietà iniziano le novità. Essa consente di definire la direzione in cui l'immagine di sfondo viene ripetuta. Proprietà non ereditata.

Sintassi generale:

```
selettore {background-repeat: valore;}
```

I valori possono essere:

- repeatL'immagine viene ripetuta in orizzontale e verticale. È il comportamento standard
- repeat-xL'immagine viene ripetuta solo in orizzontale
- repeat-yL'immagine viene ripetuta solo in verticale
- no-repeatL'immagine non viene ripetuta

Gestione dello sfondo: background-repeat

```
7       <style>
8           p{
9               background-image: url(bumblebee.png);
10              color: white;
11              width: 500px;
12              height: 500px;
13              border: 1px solid black;
14              background-repeat: no-repeat;
15          }
16      </style>
17  </head>
18  <body>
19      <p>Sono del testo</p>
20  </body>
```

Gestione dello sfondo: background-attachment

Con questa proprietà si imposta il comportamento dell'immagine di sfondo rispetto all'elemento cui è applicata e all'intera finestra del browser. Si decide, in pratica, se essa deve scorrere insieme al contenuto o se deve invece rimanere fissa. Proprietà non ereditata.

Sintassi generale:

```
selettore {background-attachment: valore;}
```

I valori possono essere:

- scroll L'immagine scorre con il resto del documento quando si fa lo scrolling della pagina
- fixed L'immagine rimane fissa mentre il documento scorre

Da vedere in azione per apprezzare le differenze.

Gestione dello sfondo: background-position

Proprietà un po' complessa. Definisce il punto in cui verrà piazzata un'immagine di sfondo non ripetuta o da dove inizierà la ripetizione di una ripetuta. Si applica solo agli elementi blocco o rimpiazzati. Attenzione alla compatibilità e alla resa, non omogeena, tra i vari browser. Proprietà non ereditata.

Sintassi generale:

```
selettore {background-position: valore_oriz valore_vert;}
```

I valori specificano le coordinate di un punto sull'asse verticale e su quello orizzontale e possono essere espressi con diverse unità di misura e modalità:

- con valori in percentuale
- con valori espressi con unità di misura
- con le parole chiave top, left, bottom, right

Gestione dello sfondo: background-position

```
7       <style>
8           p{
9               background-image: url(bumblebee.png);
10              width: 500px;
11              height: 500px;
12              border: 1px solid black;
13              background-repeat: no-repeat;
14              background-position: center bottom;
15          }
16      </style>
17  </head>
18  <body>
19      <p></p>
20  </body>
```

```
7       <style>
8           p{
9               background-image: url(bumblebee.png);
10              width: 500px;
11              height: 500px;
12              border: 1px solid black;
13              background-repeat: no-repeat;
14              background-position: 10px 50px;
15          }
16      </style>
17  </head>
18  <body>
19      <p></p>
20  </body>
```

Gestione dello sfondo: background

Esaminiamo finalmente la proprietà background. Con essa, ricordiamolo, possiamo definire in un colpo solo tutti gli aspetti dello sfondo. Per essere valida, la dichiarazione non deve contenere necessariamente riferimenti a tutte le proprietà viste finora, ma deve contenere almeno la definizione del colore di sfondo.

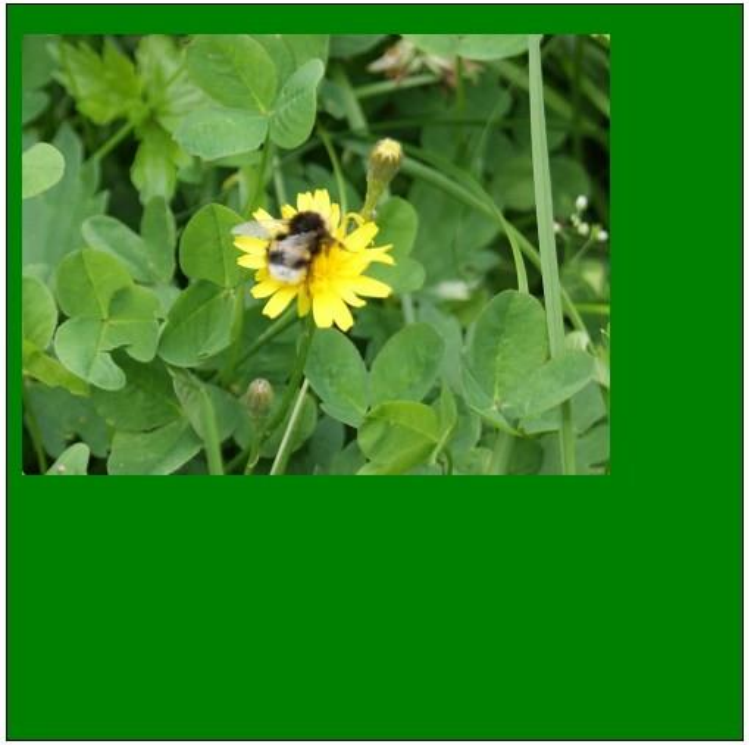
Sintassi generale:

```
selettore {background: background-color
                                background-image background-
                                repeat background-
                                attachment background-
                                position;
                }
```

I valori non vanno separati da virgole. L'ordine non è importante, ma quello dato è il più logico e andrebbe rispetatto.

Gestione dello sfondo: background

```
7       <style>
8           p{
9               background: url(bumblebee.png) green no-repeat scroll 10% 10%;
10              width: 500px;
11              height: 500px;
12              border: 1px solid black;
13          }
14      </style>
15  </head>
16  <body>
17      <p></p>
18  </body>
```



Gestione dello sfondo: background-size

La proprietà background-sizeserve a impostare le dimensioni delle immagini usate come sfondo su un elemento della pagina.

Sintassi generale:

selettore {background-size: valore;}

La proprietà può assumere cinque tipi di valori:

- La parola chiave auto, l'immagine mantiene le sue dimensioni originali. Comportamento di default.
- una dimensione espressa in percentuale
- una dimensione espressa con un'unità di misura
- la parola chiave cover, l'immagine viene ridimensionata per **coprire** interamente l'elemento a cui viene applicata
- la parola chiave contain, l'immagine viene ridimensionata per **adattarsi** all'area dell'elemento cui viene applicata

Gestione dello sfondo: background-size

```
7       <style>
8           p{
9               background: url(bumblebee.png);
10              width: 500px;
11              height: 500px;
12              border: 1px solid black;
13              background-size: cover;
14          }
15      </style>
16  </head>
17  <body>
18      <p></p>
19  </body>
```

Esercizi 9

Esercizio 1
Crea una pagina HTML con un paragrafo all'interno. Usa il CSS per impostare un'immagine di sfondo nel paragrafo.
Esercizio 2
Crea una pagina HTML con un link all'interno. Usa il CSS per impostare un'immagine di sfondo al link.
Esercizio 3
Crea una pagina HTML con un div all'interno. Usa il CSS per impostare un'immagine di sfondo nel div e ripetila verticalmente.
Esercizio 4
Crea una pagina HTML con una lista non ordinata (ul) contenente alcuni elementi di lista (li). Usa il CSS per impostare un'immagine di sfondo a tutti gli elementi di lista.
Esercizio 5
Crea una pagina HTML con un link all'interno. Usa il CSS per impostare un'immagine di sfondo al link e ripetila solo orizzontalmente.
Esercizio 6
Crea una pagina HTML con un titolo all'interno. Usa il CSS per impostare un'immagine di sfondo al titolo e ripetila solo orizzontalmente

Gestione del tipo di presentazione: display

Questa proprietà serve per controllare il modo in cui un elemento viene rappresentato dal browser.

Abbiamo imparato che ogni elemento HTML ha un suo comportamento predefinito rispetto alla presentazione: si parla di elementi blocco, elementi in linea, elementi di lista e elementi di tabella.

Ebbene, con la proprietà display possiamo modificare, quando e se necessario, questo comportamento di default. La proprietà display è ereditata.

Sintassi generale:

```
selettore { display: valore; }
```

Il valore può essere rappresentato unicamente da una parola chiave. Nella pratica comune, i valori in assoluto più utili e usati sono mostrati nella slide successiva

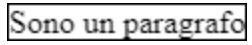
Gestione del tipo di presentazione: display

Valore	Descrizione
`block`	l'elemento viene reso come un elemento **blocco**
`inline`	l'elemento a cui viene applicata assume le caratteristiche degli elementi **inline**
`inline-block`	l'elemento può assumere, come gli elementi blocco, dimensioni esplicite (larghezza e altezza), margini e padding, ma come tutti gli elementi inline, si disporrà orizzontalmente e non verticalmente, potendo essere circondato dal testo ed essendo sensibile all'allineamento verticale
`none`	l'elemento non viene mostrato; o meglio: è come se non fosse nemmeno presente nel documento, in quanto non genera alcun box; l'uso del valore `none` è uno dei mezzi con cui, nei CSS, si può nascondere un elemento

Gestione del tipo di presentazione: display

```
7       <style>
8           p{
9               border: 1px solid black;
10          }
11      </style>
12  </head>
13  <body>
14      <p>Sono un paragrafo</p>
15  </body>
```


```
8           p{
9               border: 1px solid black;
10              display: inline;
11          }
12      </style>
13  </head>
14  <body>
15      <p>Sono un paragrafo</p>
16  </body>
```



Esercizi 10

Esercizio 1
Crea una pagina HTML con uno span all'interno. Usa il CSS per impostare lo span in modo che occupi l'intera larghezza del contenitore.
Esercizio 2
Crea una pagina HTML con un'immagine all'interno. Usa il CSS per impostare il display dell'immagine come "inline" in modo che il testo scorra attorno ad essa.
Esercizio 3
Crea una pagina HTML con una lista non ordinata (ul) contenente alcuni elementi di lista (li). Usa il CSS per impostare il display degli elementi di lista come "inline-block" in modo che siano visualizzati su una singola riga e possano avere una larghezza specificata.
Esercizio 4
Crea una pagina HTML con un paragrafo all'interno. Usa il CSS per impostare il display del paragrafo come "none" in modo che non sia visibile sulla pagina.
Esercizio 5
Crea una pagina HTML con due link all'interno. Usa il CSS per impostare il display dei link come "block" in modo che siano visualizzati su righe separate, ognuno con larghezza specificata.
Esercizio 6
Crea una pagina HTML con una lista non ordinata (ul) contenente alcuni elementi di lista (li). Usa il CSS per impostare il display della lista come "block" in modo che ogni elemento sia visualizzato su una riga separata.

Gestione del tipo di presentazione: float e clear

Con questa proprietà è possibile rimuovere un elemento dal normale flusso del documento e spostarlo su uno dei lati (destro o sinistro) del suo elemento contenitore. Il contenuto che circonda l'elemento scorrerà intorno ad esso sul lato opposto rispetto a quello indicato come valore di float. La proprietà non è ereditata.

Sintassi generale:

```
selettore {float: valore;}
```

I possibili valori sono:

- left: l'elemento viene spostato sul lato sinistro del box contenitore, il contenuto scorre a destra
- right: l'elemento viene spostato sul lato destro, il contenuto scorre a sinistra
- none: valore iniziale e di default in mancanza di una dichiarazione esplicita; l'elemento mantiene la sua posizione normale

Gestione del tipo di presentazione: float e clear

Gli elementi con float, di qualsiasi natura essi siano, possono avere margini, bordi e padding come tutti gli altri elementi blocco. E soprattutto, possono avere dimensioni esplicite. Questo perché un elemento dichiarato float viene reso automaticamente con la proprietà display:block.

Un elemento float, pur essendo un elemento blocco a tutti gli effetti, se non dichiarato attraverso dimensioni esplicite, assumerà la dimensione orizzontale massima per il suo contenuto, e quella verticale necessaria.

Una compagna quasi inseparabile della proprietà floatè la proprietà clearche, se applicata ad un elemento successivo ad uno reso float, impedisce che questo subisca il float.

Gestione del tipo di presentazione: float e clear

La proprietà clear serve a impedire che al fianco di un elemento compaiano altri elementi con il float. Si applica solo agli elementi blocco e non è ereditata.
L'origine di tale proprietà è questa: visto che il float sposta un elemento dal flusso normale del documento, è possibile che esso venga a trovarsi in posizioni non desiderate, magari al fianco di altri elementi che vogliamo invece tenere separati. clear risolve questo problema.

Sintassi generale:

```
selettore {clear: valore;}
```

I possibili valori sono:

- none: gli elementi con float possono stare a destra e sinistra dell'elemento
- left: si impedisce il posizionamento a sinistra
- right: si impedisce il posizionamento a destra
- both: si impedisce il posizionamento su entrambi i lati (il più diffuso)

Gestione del tipo di presentazione: float e clear

```
7       <style>
8           p{
9               border: 1px solid black;
10              float: right;
11          }
12      </style>
13  </head>
14  <body>
15      <p>Sono un paragrafo</p>
16
17      <div>Sono una div</div>
18  </body>
```

Sono una div

Sono un paragrafo

```
7       <style>
8           p{
9               border: 1px solid black;
10              float: right;
11          }
12
13          div{
14              clear: both;
15          }
16      </style>
17  </head>
18  <body>
19      <p>Sono un paragrafo</p>
20
21      <div>Sono una div</div>
22  </body>
```

Sono un paragrafo

Sono una div

Esercizi 11

Esercizio 1
Crea una pagina HTML con due div all'interno. Usa il CSS per impostare il float di entrambi i div come "left" in modo che si posizionino uno a fianco all'altro.
Esercizio 2
Crea una pagina HTML con tre div all'interno. Usa il CSS per impostare il float dei primi due div come "left" e "right" rispettivamente, in modo che si posizionino uno a fianco all'altro, mentre il terzo div non viene influenzato dal float.
Esercizio 3
Crea una pagina HTML con due div all'interno e un paragrafo successivo. Usa il CSS per impostare il float dei div come "left" e "right" rispettivamente, in modo che il paragrafo si posizioni sotto di essi.
Esercizio 4
Crea una pagina HTML con tre div all'interno. Usa il CSS per impostare il float dei primi due div come "left" in modo che si posizionino uno a fianco all'altro, mentre il terzo div viene posizionato sotto di essi con la proprietà "clear".
Esercizio 5
Crea una pagina HTML con due div all'interno e un paragrafo successivo. Usa il CSS per impostare il float dei div come "left" e "right" rispettivamente, in modo che il paragrafo si posizioni tra i due div.
Esercizio 6
Crea una pagina HTML con quattro div all'interno. Usa il CSS per impostare il float dei primi tre div come "left" in modo che si posizionino uno a fianco all'altro, mentre il quarto div viene posizionato sotto di essi con la proprietà "clear".

Posizione degli elementi: position

positionè la proprietà fondamentale per la gestione della posizione degli elementi: determina la modalità di presentazione di un elemento sulla pagina. Si applica a tutti gli elementi e non è ereditata.

Sintassi generale:

selettore {position: valore;}

I valori con cui è possibile definire la modalità di posizionamento sono quattro:

- static
- relative
- absolute
- fixed

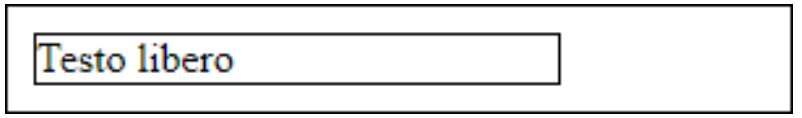
Ciascuno di questi valori merita un spiegazione approfondita da cui emergeranno i concetti di base che governano le regole sul posizionamento.

Posizione degli elementi: position static

- **position: static**

È il valore di default, quello predefinito per tutti gli elementi non posizionati secondo un altro metodo. Rappresenta la posizione normale che ciascuno di essi occupa nel flusso del documento.

```
7       <style>
8
9           #d1{
10              width: 300px;
11              border: 1px solid black;
12              position: static;
13          }
14
15          #d2{
16              width: 200px;
17              margin: 10px;
18              border: 1px solid black;
19              position: static;
20          }
21      </style>
22  </head>
23  <body>
24
25      <div id="d1">
26          <div id="d2">Testo libero</div>
27      </div>
28
29  </body>
```

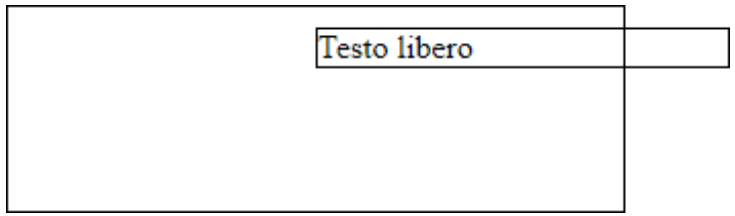


Posizione degli elementi: position relative

- **position: relative**

L'elemento viene posizionato relativamente al suo box contenitore. In questo caso il box contenitore è rappresentato dal posto che l'elemento avrebbe occupato nel normale flusso del documento. La posizione viene impostata con le proprietà top, left, bottomo right. Nel posizionamento relativo esse non indicano un punto preciso, ma l'ammontare dello spostamento in senso orizzontale e verticale rispetto al box contenitore.

```
    <style>

        #d1{
            width: 300px;
            height: 100px;
            border: 1px solid black;
            position: static;
        }

        #d2{
            width: 200px;
            border: 1px solid black;
            position: relative;
            left: 150px;
            top: 10px;
        }
    </style>
</head>
<body>

    <div id="d1">
        <div id="d2">Testo libero</div>
    </div>

</body>
```

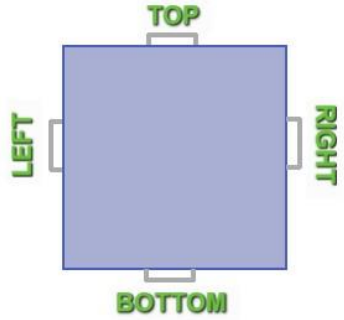


Posizione degli elementi: position relative

La metafora più semplice per comprendere il funzionamento è immaginare che sui quattro lati di un elemento posizionato relativamente ci siano quattro maniglie che si possono tirare o spingere: *un valore positivo equivale a spingere, mentre un valore negativo equivale a tirare l'elemento*.

Per esempio, se viene specificato `left: 30px` significa che l'elemento viene spinto da sinistra di 30 pixel. Dichiarando invece `top: -20px` è come se tirassimo l'elemento dal suo lato superiore, con il conseguente effetto di traslarlo verso l'alto di 20 pixel.

Con il posizionamento relativo andrebbero specificati al massimo uno o due valori tra `top`, `left`, `bottom` e `right`.



N.B. i possibili valori accettati sono in px (o comunque un unità di misura), percentuale ed auto (default)

Posizione degli elementi: position absolute

- **position: absolute**

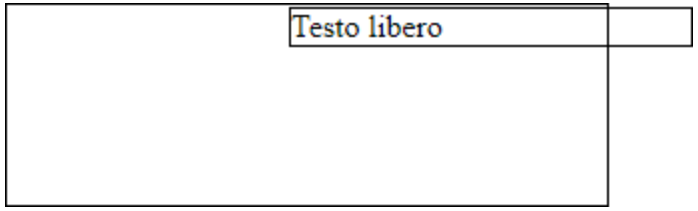
L'elemento, o meglio, il box dell'elemento, viene rimosso dal flusso del documento ed è posizionato in base ai valori forniti con le proprietà top, left, bottom o right.

Il posizionamento absoluteavviene sempre rispetto al box contenitore dell'elemento. Questo è rappresentato dal primo elemento antenato che abbia un posizionamento diverso da static.

Se tale elemento non esiste il posizionamento assoluto avviene in base all'elemento radice html, che in condizioni standard coincide con l'area del browser che contiene il documento e che ha inizio dall'angolo superiore sinistro di tale area. Un elemento posizionato in modo assoluto scorre insieme al resto del documento.

Posizione degli elementi: position absolute

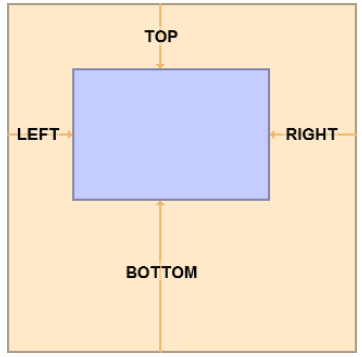
```
7     <style>
8
9         #d1{
10            width: 300px;
11            height: 100px;
12            border: 1px solid black;
13            position: static;
14        }
15
16        #d2{
17            width: 200px;
18            border: 1px solid black;
19            position: absolute;
20            left: 150px;
21            top: 10px;
22        }
23    </style>
24 </head>
25 <body>
26
27    <div id="d1">
28        <div id="d2">Testo libero</div>
29    </div>
30
31 </body>
```



Un elemento posizionato assolutamente è di default reso block-level, indipendentemente dalla sua natura iniziale. È come se, insieme a position: absolute dichiarassimo ogni volta implicitamente display: block.

Posizione degli elementi: position absolute

Per quanto riguarda le proprietà top, left, bottom o right, si potrebbe pensare che questi valori siano da intendersi come delle coordinate. In realtà non è proprio così, ed è più facile pensare i valori che queste proprietà possono assumere come vere distanze. Un'immagine aiuterà a capire il concetto:



Posizione degli elementi: position fixed

- **position: fixed**

Usando questo valore, il box dell'elemento viene, come per absolute, sottratto al normale flusso del documento. La differenza sta nel fatto che per fixed il box contenitore è sempre la cosiddetta **viewport**.

Con questo termine si intende la finestra principale del browser, ovvero l'area del contenuto. Altra differenza fondamentale: un box posizionato con fixed non scorre con il resto del documento. Rimane, appunto, fisso al suo posto.

In questo caso bisogna usare un esempio concreto per apprezzare la differenza ;)

N.B. esiste anche una variante di fixed, denominata sticky che ha un comportamento leggermente diverso

Esercizi 12

Esercizio 1
Crea una pagina HTML con un div all'interno. Usa il CSS per impostare la posizione del div come comportamento predefinito.
Esercizio 2
Crea una pagina HTML con un div all'interno. Usa il CSS per impostare la posizione del div come "relative" e spostalo di 50px verso il basso e 20px verso destra rispetto alla sua posizione originale.
Esercizio 3
Crea una pagina HTML con un div all'interno. Usa il CSS per impostare la posizione del div come "absolute" e posizionalo rispetto al bordo superiore destro della finestra del browser con una distanza di 30px.
Esercizio 4
Crea una pagina HTML con un div all'interno. Usa il CSS per impostare la posizione del div come "fixed" e posizionalo rispetto al bordo inferiore destro della finestra del browser con una distanza di 20px.
Esercizio 5
Crea una pagina HTML con un div all'interno. Usa il CSS per impostare la posizione del div come "sticky" e posizionalo rispetto al bordo superiore della finestra del browser in modo che segua lo scorrimento verticale.
Esercizio 6
Crea una pagina HTML con un div all'interno. Usa il CSS per impostare la posizione del div come "absolute" e posizionalo rispetto al suo contenitore (un altro div) con una distanza di 20px dal bordo superiore e 30px dal bordo sinistro.

La proprietà visibility

Questa proprietà determina se un elemento debba essere visibile o nascosto. Si applica a tutti gli elementi e non è ereditata.

Una nota fondamentale riguarda la differenza tra l'uso di visiblitye della dichiarazione display: none per nascondere un elemento. Usando visiblity, l'elemento non viene rimosso dal flusso del documento. Significa che, pur essendo invisibile, il box che genera occupa comunque lo spazio dettato dalle sue dimensioni. Potremmo dire, semplificando, c'è ma non si vede. Con display: none, invece, come abbiamo visto, l'elemento viene rimosso dal flusso del documento e non occupa alcuno spazio sulla pagina.

Sintassi generale:

```
selettore {visibility: valore;}
```

La proprietà visibility

I valori possibili sono:

- visible: valore iniziale e di default, l'elemento è visibile
- hidden: l'elemento è nascosto, ma mantiene il suo posto nel layout dove apparirà come una zona vuota
- collapse: usato solo per elementi di tabella (righe, colonne, celle)

```
        #d1{
            width: 300px;
            height: 100px;
            border: 1px solid black;
            position: static;

        }

        #d2{
            width: 200px;
            border: 1px solid black;
            visibility: hidden;
        }
    </style>
</head>
<body>

    <div id="d1">
        <div id="d2">Testo libero</div>
    </div>

</body>
```

La proprietà z-index

Delle tre proprietà analizzate in questa lezione è certamente quella più utile e usata. Con essa si imposta **l'ordine di posizionamento** dei vari elementi sulla base di una scala di livelli.

I posizionamenti (assoluti, relativi o fissi che siano) permettono di sistemare o traslare elementi lungo **due dimensioni (verticale e orizzontale).** C'è in realtà nei CSS una sorta di profondità, o **terza dimensione**: lo z-index, appunto. Stabilisce la disposizione degli elementi posizionati lungo l'asse perpendicolare allo schermo.

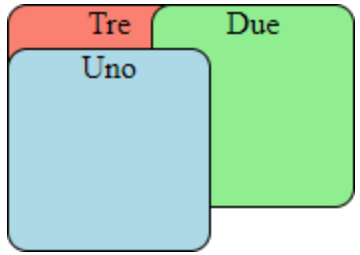
Sintassi generale:

```
selettore {z-index: valore;}
```

Lo z-index assume valori interi senza unità di misura, che possono essere positivi o negativi. Elementi con z-index maggiore vengono disposti sopra (o ancor meglio, davanti) ad elementi con z-index minore.

La proprietà z-index

```
    <style>

        .box{
            height: 100px;
            width: 100px;
            margin: 10px;
            border-radius: 10px;
            text-align: center;
            border: 1px solid black;
        }

        .due{
            position: absolute;
            background-color: lightgreen;
            left: 80px;
            /* l'elemento con z-index maggiore viene mostrato sopra */
            z-index: 5;
        }

        .tre{
            background-color: salmon;
            position: absolute;
            z-index: 1;
        }

        .uno{
            background-color: lightblue;
            top: 30px;
            position: absolute;
            z-index: 6;
        }
    </style>
</head>
<body>

    <div class="box due">Due</div>
    <div class="box tre">Tre</div>
    <div class="box uno">Uno</div>

</body>
```



Gestione del testo: proprietà di base

La gestione del testo e della tipografia è un aspetto essenziale dei CSS. Le proprietà che definiscono il modo in cui il testo appare sullo schermo sono tante e abbiamo deciso di suddividere l'argomento in due lezioni. Iniziamo quindi dalle proprietà di base. Sono quelle che definiscono i seguenti aspetti:

- **il font da usare**
- **la sua dimensione**
- **la sua consistenza**
- **l'interlinea tra le righe**
- **l'allineamento del testo**
- **la sua decorazione (sottolineature, etc.)**

Gestione del testo: proprietà font-family

La proprietà font-familyserve a impostare il tipo di carattere tipografico per una qualunque porzione di testo. Si applica a tutti gli elementi ed è ereditata.

Il CSS permette di impostare non un unico font, ma un elenco di caratteri alternativi. Il meccanismo funziona così:

```
p {font-family: Arial, Verdana, sans-serif;}
```

Quando la pagina viene caricata, il browser tenterà di usare il primo font della lista. Se questo non è disponibile sul dispositivo dell'utente userà il secondo. In mancanza anche di questo, verrà utilizzato il font principale della famiglia sans-serif presente sul sistema. La spiegazione di tutto ciò è semplice: ovviare al problema dei diversi font presenti sulle varie piattaforme.

Gestione del testo: proprietà font-family

Dunque: quando si imposta la proprietà font-familysi possono usare tutti i font che desideriamo, valutando la loro presenza sui vari sistemi e non dimenticando mai di inserire alla fine l'indicazione di una famiglia generica. Esse sono cinque (tra parentesi riportiamo i caratteri predefiniti su ciascuna sui sistemi Windows):

- serif (Times New Roman)
- sans-serif (Arial)
- cursive (Comic Sans)
- fantasy (Allegro BT)
- monospace (Courier)

Gestione del testo: proprietà font-family

I nomi dei font della lista vanno separati dalla virgola. I caratteri con nomi composti da più parole vanno inseriti tra virgolette.

Sintassi:

selettore {font-family: <font 1>, <font2>, ...,<famiglia generica>;}

Esempio:

div {font-family: Georgia, "Times New Roman", serif;}

Gestione del testo: proprietà font-size

Insieme a font-familyè la proprietà considerata essenziale nella definizione dell'aspetto del testo, di cui definisce le dimensioni. È applicabile a tutti gli elementi ed ereditata.

Sintassi:

```
selettore {font-size: valore;}
```

I valori delle dimensioni del testo possono essere espressi in vari modi. I diversi sistemi si possono distinguere a seconda che definiscano le dimensioni **in senso assoluto o relativo**.

- **Dimensione assoluta** significa che essa non dipende da nessun altro elemento ed è quella definita dall'unità di misura usata.
- **Dimensione relativa** significa che essa viene calcolata in base alla dimensione del testo dell'elemento parente.

Gestione del testo: proprietà font-size

Sono valori assoluti le sette parole chiave xx-small, x-small, small, medium, large,
x. large, xx-large.

Sono valori relativi:
- le parole chiave smallere larger
- quelli espressi in em (relativo all'elemento padre, ad esempio 2em significa grande il doppio dell'elemento padre)
- quelli espressi in percentuale

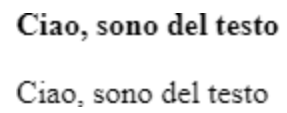
Nelle pratiche più comuni, la scelta del dimensionamento dei font viene fatta tra pixel, em e percentuale.

p {font-size: 12px;} div {font-size:
50%;} h2 {font-size: 1.2em;}

Gestione del testo: proprietà font-weight

Serve a definire la consistenza o "peso" visivo del testo. Si applica a tutti gli elementi ed è ereditata.

Sintassi: selettore {font-weight: valore;}



Il "peso" visivo di un carattere può essere espresso con una scala numerica o con parole chiave:

- **valori numerici**: 100 - 200 - 300 - 400 - 500 - 600 - 700 - 800 - 900 ordinati in senso crescente (dal più leggero al più pesante)
- normal: valore di default, è l'aspetto normale del font ed equivale al valore 400
- bold: il carattere acquista l'aspetto che definiamo in genere 'grassetto'; equivale a 700
- bolder: misura relativa; serve a specificare che una determinata porzione di testo dovrà apparire più pesante a livello visuale rispetto al testo dell'elemento parente
- lighter: misura relativa; il testo sarà più leggero di quello dell'elemento parente

Gestione del testo: proprietà font-style

Imposta le caratteristiche del testo in base ad uno di questi tre valori:

- normal: il testo mantiene il suo aspetto normale
- italic: formatta il testo in corsivo
- oblique: praticamente simile a italic

La proprietà si applica a tutti gli elementi ed è ereditata.

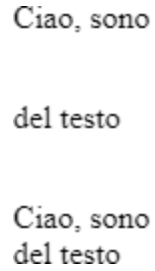
Sintassi: selettore {font-style: valore;}

Esempio: p {font-style: italic;}

Ciao, sono del testo

Gestione del testo: proprietà line-height

Grazie a line-heightè possibile usare nelle nostre pagine un sistema di interlinea paragonabile a quello dei word-processor. La proprietà, in realtà, serve a definire l'altezza di una riga di testo all'interno di un elemento blocco. Ma l'effetto ottenuto è appunto quello di impostare uno spazio tra le righe. La proprietà si applica a tutti gli elementi ed è ereditata.



Sintassi: selettore {line-height: valore;}

I valori che è possibile utilizzare sono:

- normal: il browser separerà le righe con uno spazio ritenuto "ragionevole"
- **un valore numerico**: usando valori numerici tipo 1.2, 1.3, 1.5 si ottiene questo risultato: l'altezza della riga sarà uguale alla dimensione del font moltiplicata per questo valore
- **un valore numerico con unità di misura**: l'altezza della riga sarà uguale alla dimensione specificata
- **percentuale**: l'altezza della riga viene calcolata come una percentuale della dimensione del font.

Gestione del testo: proprietà font

La proprietà fontè una proprietà a sintassi abbreviata che serve a impostare con una sola dichiarazione tutte le principali caratteristiche del testo. Le proprietà definibili in forma abbreviata con font sono:

- font-family
- font-size
- line-height
- font-weight
- font-style

Ciao, sono del testo

Sintassi: selettore {font: valori;}

Esempio: p {font: bold 12px/1.5 Georgia, "Times New Roman", serif;}
Dove font-size/line-height

Gestione del testo: proprietà text-align

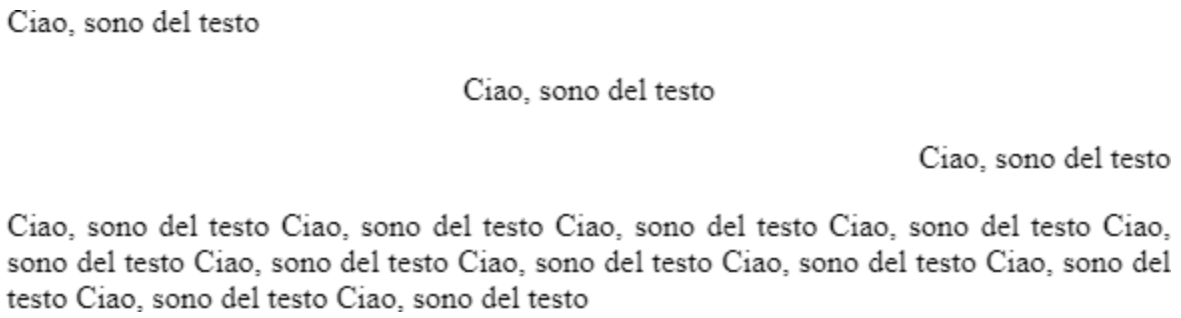
La proprietà serve a impostare l'allineamento del testo. È ereditata e si applica a tutti gli elementi.

Sintassi: selettore {text-align: valore;}

I valori possono essere rappresentati da una delle seguenti parole chiave:

- left: allinea il testo a sinistra
- right: allinea il testo a destra
- center: centra il testo
- justify: giustifica il testo

p {text-align: center;} div {text-align: justify;}

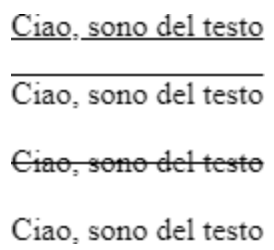


Gestione del testo: proprietà text-decoration

Imposta particolari decorazioni e stili per il testo. Ereditabile e applicabile a tutti gli elementi.

Sintassi: selettore {text-decoration: valore o valori;}

I valori che è possibile usare sono:



- none: il testo non avrà alcuna decorazione particolare
- underline: il testo sarà sottolineato
- overline: il testo avrà una linea superiore
- line-through: il testo sarà attraversato da una linea orizzontale al centro

```
p {text-decoration: none;}
a {text-decoration: underline;}
```

Gestione del testo: proprietà font-variant

Consente di trasformare il testo in maiuscoletto (lettere in maiuscolo rese con dimensioni uguali ai caratteri minuscoli). Proprietà ereditata.

Sintassi: selettore {font-variant: valore;}
I valori possibili sono solo due:

- normal: il testo ha il suo aspetto normale; valore iniziale e di default
- small-caps: trasforma il testo in maiuscoletto

CIAO, SONO DEL TESTO

h2 {font-variant: small-caps;}

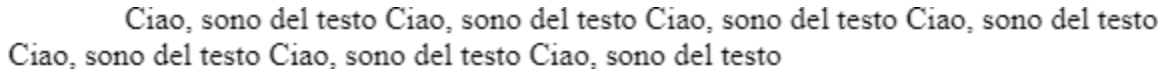
Gestione del testo: proprietà text-indent

Definisce l'indentazione della prima riga in ogni elemento contenente del testo. Proprietà ereditata.

Sintassi: selettore {text-indent: valore;}

Si può esprimere il valore con:

- **un valore numerico con unità di misura**
- **un valore in percentuale**



Come al solito, il valore con unità di misura è assoluto, quello in percentuale è relativo. In questo caso il valore è relativo alla larghezza dell'area del contenuto. In pratica, se per un paragrafo largo 200px imposto un'indentazione uguale al 10%, essa sarà uguale a 20px.

```
p {text-indent: 10px;} div {text-
indent: 10%;}
```

Gestione del testo: proprietà text-transform

Questa proprietà serve a cambiare gli attributi del testo relativamente a tre aspetti: maiuscolo, minuscolo, prima lettera maiuscola. È una proprietà ereditata.

Sintassi: selettore {text-transform: valore;}

Il valore può essere:

- none: valore di default; nessuna trasformazione viene applicata
- capitalize: la prima lettera di ogni parola viene trasformata in maiuscolo
- uppercase: tutto il testo diventa maiuscolo
- lowercase: tutto il testo è minuscolo

```
p {text-transform: capitalize;} h1 {text-
transform: uppercase;}
```

Ciao, sono del testo

Ciao, Sono Del Testo

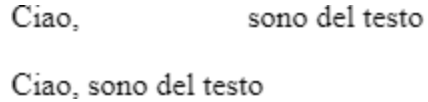
ciao, sono del testo

CIAO, SONO DEL TESTO

Gestione del testo: proprietà white-space

Questa proprietà serve a gestire il trattamento degli spazi bianchi presenti in un elemento. E' ereditata. È una cosa nota a chi ha un minimo di dimestichezza con il codice HTML. Se nel codice di una pagina si lasciano spazi bianchi tra le parole, essi saranno automaticamente convertiti in un solo spazio quando la pagina viene visualizzata nel browser. A meno di non racchiudere il testo con tag come <pre> o <code>. In questo caso, spazi bianchi e fine riga saranno rispettati e manterranno l'aspetto che essi hanno nel codice. Uguale meccanismo è implementabile nei CSS con white-space.

Sintassi: selettore {white-space: valore;}



I valori possibili sono:

- none: valore di default; gli spazi bianchi sono ridotti a uno;
- pre: gli spazi bianchi e l'inizio di nuove righe sono mantenuti così come sono nel codice HTML;
- nowrap: gli spazi bianchi sono ridotti a uno e l'andata a capo è disabilitata.

Gestione del testo: proprietà letter-spacing

Aumenta lo spazio tra le lettere di una parola. Proprietà ereditata.

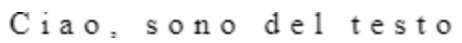
Sintassi: selettore {letter-spacing: valore;}

Per i valori si può scegliere tra:

- normal: valore di default; le lettere mantengono il loro spazio normale
- **un valore numerico con unità di misura**: le lettere saranno spaziate secondo la distanza impostata

È possibile anche impostare valori negativi. Ciò farà sì che le lettere appaiano sempre più compresse.

p {letter-spacing: 5px;}



Gestione del testo: proprietà word-spacing

Proprietà complementare a letter-spacing. Serve ad aumentare lo spazio tra le parole comprese in un elemento. Proprietà ereditata.

Sintassi: selettore {word-spacing: valore;}

Per i valori possiamo usare:

- normal: valore di default; le parole mantengono il loro spazio normale
- **un valore numerico con unità di misura**: le parole saranno spaziate secondo la distanza impostata

p {word-spacing: 1.2em;}

Ciao, sono del testo

Esercizi 13

Esercizio 1
Crea una pagina HTML con un paragrafo all'interno. Usa il CSS per impostare il colore del testo a blu.
Esercizio 2
Crea una pagina HTML con un link all'interno. Usa il CSS per impostare il colore del testo a rosso e il font a Arial.
Esercizio 3
Crea una pagina HTML con un elemento di intestazione (h1) all'interno. Usa il CSS per impostare la dimensione del testo a 24px.
Esercizio 4
Crea una pagina HTML con un paragrafo all'interno. Usa il CSS per impostare il grassetto sul testo del paragrafo.
Esercizio 5
Crea una pagina HTML con un link all'interno. Usa il CSS per impostare lo stile del testo a corsivo.
Esercizio 6
Crea una pagina HTML con un elemento di intestazione (h2) all'interno. Usa il CSS per impostare il colore di sfondo del testo a giallo.
Esercizio 7
Crea una pagina HTML con un paragrafo all'interno. Usa il CSS per impostare la dimensione del testo a 18px e il colore di sfondo del testo a verde.
Esercizio 8
Crea una pagina HTML con un link all'interno. Usa il CSS per impostare il testo in maiuscolo.

Esercizio 9

Crea una pagina HTML con un elemento di intestazione (h3) all'interno. Usa il CSS per impostare il testo sottolineato.

Esercizio 10

Crea una pagina HTML con un paragrafo all'interno. Usa il CSS per impostare lo spazio tra le lettere (letter-spacing) a 2px.

Gestione delle liste: proprietà di base

Grazie ai CSS possiamo definire in vari modi l'aspetto delle liste HTML. In realtà, tutte le proprietà che andremo ad esaminare si riferiscono non tanto ai tag usati per inserire una lista ordinata (<ol>) o non ordinata (<ul>), ma ai singoli elementi che le compongono, ovvero gli elementi <li>.

In effetti è solo questo tipo di elemento che trova una rappresentazione effettiva sulla pagina, mentre ole ulsono semplici dichiarazioni del tipo di lista usato.

Gestione delle liste: proprietà list-style-image

Definisce l'URL di un'immagine da usare come marcatore di un elemento <li>. Proprietà ereditata.

Sintassi: selettore {list-style-image: url(<url_immagine>);}
Nella definizione della sintassi per questa e per le altre proprietà che vedremo, possiamo impostare la regola a partire dall'elemento/selettore li:

li {list-style-image: url(immagine.png);}

Ma anche usando come selettore l'elemento che definisce la lista, dal momento che la proprietà è ereditata:

ul {list-style-image: url(immagine.png);} ol {list-style-image:
url(immagine.png);}

Gestione delle liste: proprietà list-style-image

Più correttamente e per un fatto di rigore strutturale (lo stile si applica ai list-item), è preferibile, usando ulo ol, affidarsi ad un selettore del discendente:

```
ul li {list-style-image: url(immagine.png);}
```

La soluzione è ottimale se si prevede di usare sempre uno stesso stile per tutte le liste. Se invece si pensa di usare stili diversi, ci si affida alle classi o agli id, che applichiamo di volta in volta secondo le necessità. La sintassi consigliata è questa:

```
ul.nome-classe li {list-style-image: url(immagine.png);}
```

I valori possono essere rappresentati da:

- **un URL assoluto o relativo** che punti ad un'immagine
- none: non viene usata nessuna immagine e il marcatore è quello di default

Gestione delle liste: proprietà list-style-position

Imposta la posizione del marcatore rispetto al testo dell'elemento <li>.Proprietà ereditata. Si applica agli elementi <li>.

Sintassi: selettore {list-style-position: valore;}

Il valore può corrispondere ad una di queste due parole chiave:

- outside: valore di default; è il comportamento standard, il marcatore è collocato all'esterno del testo;
- inside: il marcatore diventa parte integrante del testo e ne rappresenta in un certo senso il primo carattere; se il testo va a capo il marcatore apparirà all'interno del box.

```
li {list-style-position: inside;}
#lista li {list-style-position: outside;}
```

Gestione delle liste: esempio

```
10      #lista_img > li{
11              list-style-image: url(star.svg);
12              list-style-position: inside;
13
14          }
15
16      </style>
17  </head>
18  <body>
19      <ul id="lista_img">
20          <li>Cioccolato</li>
21          <li>Vaniglia</li>
22          <li>Fragola</li>
23          <li>Menta</li>
24          <li>Biscotto</li>
25      </ul>
26  </body>
```

Gestione delle liste: proprietà list-style-type

Definisce l'aspetto del marcatore. Proprietà ereditata. Si applica agli elementi <li>.

Sintassi: selettore {list-style-type: valore;}

La scelta dei valori possibili è lunghissima. Definiamo nei dettagli solo i più importanti. Tali valori sono stati inseriti per venire incontro alle esigenze di numerazione di lingue come l'ebraico o il giapponese, che usano sistemi diversi dal nostro:

• none	nessun marcatore
• disc	un cerchietto pieno colorato; il colore può essere modificato per tutti i tipi con la proprietà color
• circle	un cerchietto vuoto
• square	un quadratino
• decimal	sistema di conteggio decimale 1, 2, 3, ...
• decimal-leading-zero	sistema di conteggio decimale ma con la prima cifra preceduta da 0 (01, 02, 03, ...)
• lower-roman	cifre romane in minuscolo (i, ii, iii, iv, ...)
• upper-roman	cifre romane in maiuscolo (I, II, III, IV, ...)
• lower-alpha	lettere ASCII minuscole (a, b, c, ...)
• upper-alpha	lettere ASCII maiuscole (A, B, C, ...)
• lower-latin	simile a lower-alpha
• upper-latin	simile a upper-alpha
• lower-greek	lettere minuscole dell'alfabeto greco antico

Gestione delle liste: proprietà list-style-type

1. Cioccolato
2. Vaniglia
3. Fragola
4. Menta
5. Biscotto

a. Cioccolato
b. Vaniglia
c. Fragola
d. Menta
e. Biscotto

I. Cioccolato
II. Vaniglia
III. Fragola
IV. Menta
V. Biscotto

- Cioccolato
- Vaniglia
- Fragola
- Menta
- Biscotto

- Cioccolato
- Vaniglia
- Fragola
- Menta
- Biscotto

- Cioccolato
- Vaniglia
- Fragola
- Menta
- Biscotto

Gestione delle liste: esempio

```
    <style>

    #lista_img > li{
            background-image: url(images/star.svg);
            list-style-position: outside;
            background-size: 1rem 1rem;
            background-repeat: no-repeat;
            background-position: 0 0;
            list-style-type: none;
            padding-left: 5%;
            padding-bottom: 5%;

        }

    </style>
</head>
<body>
    <ul id="lista_img">
        <li>Cioccolato</li>
        <li>Vaniglia</li>
        <li>Fragola</li>
        <li>Menta</li>
        <li>Biscotto</li>
    </ul>
```

Gestione delle liste: proprietà list-style

Proprietà a sintassi abbreviata con cui si definiscono tutti gli aspetti e gli stili di un elemento <li>.Proprietà ereditata.
Sintassi:

```
selettore {
        list-style: valore-tipo valore-posizione valore-immagine;
}
```

I valori possibili sono quelli visti in precedenza per le proprietà singole. A rigor di logica solo due delle tre proprietà singole dovrebebro trovare posto in questa dichiarazione abbreviata: per definire l'aspetto del marcatore, infatti, o decido di usare un'immagine o propendo per un marcatore a carattere. Se si inseriscono indicazioni per tipo e immagine, prevale l'immagine e il tipo è ignorato.

```
ul li {list-style: square inside;}
li {list-style: outside url(imamgine.png);}
```

Esercizi 14

Esercizio 1
Crea una pagina HTML con una lista non ordinata (ul) contenente alcuni elementi di lista (li). Usa il CSS per rimuovere i punti elenco e cambiare il colore del testo degli elementi di lista in blu.
Esercizio 2
Crea una pagina HTML con una lista ordinata (ol) contenente alcuni elementi di lista (li). Usa il CSS per cambiare il tipo di numerazione degli elementi di lista in caratteri romani minuscoli.
Esercizio 3
Crea una pagina HTML con una lista non ordinata (ul) contenente alcuni elementi di lista (li). Usa il CSS per cambiare il tipo di simboli degli elementi di lista in quadrati.
Esercizio 4
Crea una pagina HTML con una lista ordinata (ol) contenente alcuni elementi di lista (li). Usa il CSS per impostare l'immagine di sfondo personalizzata per gli elementi di lista numerati.
Esercizio 5
Crea una pagina HTML con una lista non ordinata (ul) contenente alcuni elementi di lista (li). Usa il CSS per impostare un'immagine di sfondo personalizzata per gli elementi di lista puntati.
Esercizio 6
Crea una pagina HTML con una lista ordinata (ol) contenente alcuni elementi di lista (li). Usa il CSS per cambiare il colore del testo degli elementi di lista numerati in rosso e aggiungere uno sfondo grigio chiaro.

Gestione delle tabelle: proprietà table-layout

Questa proprietà imposta il metodo di layout di una tabella. Non è ereditata. Si applica solo alle tabelle.

Sintassi: selettore {table-layout: valore;}

A	B	C
D	E	F

I valori della proprietà possono essere i seguenti:

- auto: il layout della tabella viene definito automaticamente dal browser
- fixed: le regole di presentazione sono quelle impostate dall'autore nel CSS

Nel caso del valore auto, tutto è affidato al meccanismo di rendering del browser. Usando invece fixedpossiamo innanzitutto definire la larghezza della tabella tramite la proprietà width.
Volendo creare una tabella di 400px, quindi, scriveremo questa regola:
table {table-layout: fixed; width: 400px;}

Gestione delle tabelle: proprietà border-collapse

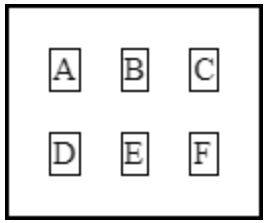
Attraverso questa proprietà possiamo stabilire in che modo trattare i bordi e gli spazi tra le celle di una tabella. Si applica solo alle tabelle ed è ereditata.

Sintassi: selettore {border-collpse: valore;}

Si può usare uno tra questi due valori:

- collapse: se viene impostato un bordo, le celle della tabella lo condividono
- separate: se viene impostato un bordo, ogni cella ha il suo, separato dalle altre; lo spazio tra le celle e tra i bordi si imposta con la proprietà border-spacing

```
table{
      border: 2px solid black; border-
      collapse: separate; border-spacing:
      20px;
}
```



Gestione delle tabelle: proprietà empty-cells

Gestisce il trattamento delle celle di tabella senza contenuto. Agisce solo su quelle che non presentino al loro interno alcun tipo di markup. Proprietà ereditata.

Sintassi: `selettore {empty-cells: valore;}`

Anche in questo caso due i valori possibili:

- show: mostra i bordi della cella
- hide: i bordi non vengono mostrati e apparirà solo uno spazio vuoto

A		C
D		F

Gestione delle tabelle: proprietà caption-side

Le buone norme dell'accessibilità vogliono che una tabella sia sempre preceduta da una sorta di titolo/riassunto. In HTML questa funzione è demandata al tag <caption>.
La proprietà caption-sidedefinisce il lato su cui vogliamo far comparire tale titolo. È ereditata.

Sintassi: selettore {caption-side: valore;}

I valori che possono essere impostati fanno riferimenti ai quattro lati della tabella:

- top: caption sul lato superiore
- bottom: caption sul lato inferiore

```
table {caption-side : bottom;}
```

A	B	C
D	E	F

Lettere

Esercizi 15

Esercizio 1
Crea una tabella HTML semplice con alcune righe e colonne. Usa il CSS per cambiare il colore di sfondo delle celle della tabella in grigio chiaro e modifica la larghezza di ogni cella a 40px.

Esercizio 2
Crea una tabella HTML con alcune righe e colonne. Usa il CSS per aggiungere bordi a tutte le celle della tabella.

Esercizio 3
Crea una tabella HTML con alcune righe e colonne. Usa il CSS per centrare il testo nelle celle della tabella e aggiungi uno spazio tra le celle di 20px.

Esercizio 4
Crea una tabella HTML con alcune righe e colonne. Usa il CSS per cambiare il colore del testo nelle celle della tabella in bianco e il colore di sfondo in nero.

Esercizio 5
Crea una tabella HTML con alcune righe e colonne. Usa il CSS per cambiare il colore del testo nelle celle della tabella quando il mouse ci passa sopra.

Esercizio 6
Crea una tabella HTML con alcune righe e colonne. Usa il CSS per aggiungere uno sfondo grigio alle righe pari della tabella.

Esercizio 7
Con una tabella 1x3 realizza la bandiera italiana e francese. Con una tabella 3x1 quella tedesca.

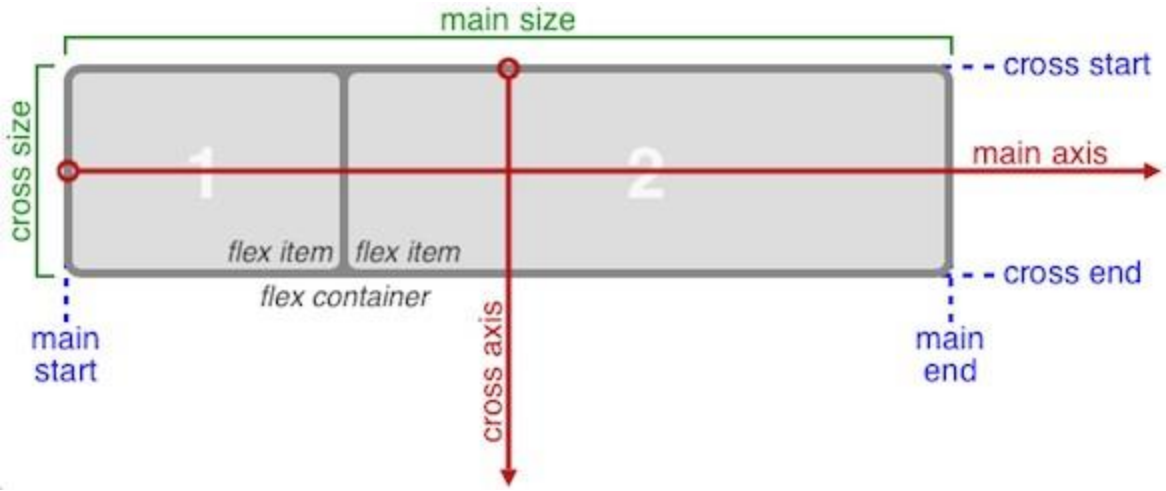
SEZIONE BONUS

Gestione del layout

L'impaginazione è uno degli aspetti fondamentali di un sito internet. Dalla diffusione dei fogli di stile, e in seguito alle raccomandazioni del W3C, la tendenza di creare layout senza l'uso di tabelle si è andata man mano consolidando, fino a diventare una realtà in continua espansione.

Il dibattito tra i sostenitori di tabelle e quelli dei fogli di stile per il layout è ancora molto acceso. Ci sono molti argomenti validi da entrambe le parti, ma attualmente si può affermare quasi con certezza che l'ago della bilancia pende verso i fogli di stile.

Layout flexbox



Il sistema è fondato su due elementi costitutivi:

- **un elemento contenitore (flex container):** è quello che contiene i box flessibili e viene creato usando il valore flex per la proprietà display
- **box flessibili (flex items):** sono gli elementi figli dell'elemento contenitore che assumono come loro caratteristica fondamentale quella della 'flessibilità'; su questi box non hanno effetto le proprietà float, clear e vertical-align

Di fatto, il modello di layout del flexbox prevede in prima istanza la dichiarazione di un elemento contenitore e delle sue proprietà di base, quindi l'impostazione del comportamento dei singoli box al suo interno.

Le tredici proprietà che compongono il modulo possono pertanto essere suddivise in due gruppi: quelle che si applicano al contenitore e quelle che si applicano ai box flessibili.

Layout flexbox

Proprietà del contenitore:

Proprietà	Valori	Descrizione
display	flex	è il valore della proprietà display con cui si imposta un contenitore flessibile
flex-direction	row | row-reverse | column | column- reverse	specifica la direzione dell'asse principale su cui si dispongono i box flessibili nel contenitore
flex-wrap	nowrap | wrap | wrap-reverse	specifica se i box all'interno del contenitore si dispongono su una riga o su più righe in base allo spazio disponibile
flex-flow		è una proprietà a sintassi abbreviata con cui esprimere insieme i valori per flex-directione flex-wrap
justify-content	flex-start | flex-end | center | space- between | space-around	stabilisce la modalità di allineamento dei box flessibili sull'asse principale del contenitore
align-items	stretch | flex-start | flex-end | center | baseline	gestisce l'allineamento dei box flessibili lungo l'asse perpendicolare all'asse principale
align-content	stretch | flex-start | flex-end | center | space-between | space-around	gestisce l'allineamento di una riga di box flessibili lungo l'asse perpendicolare; ha effetto solo su contenitori multi-riga

Layout flexbox

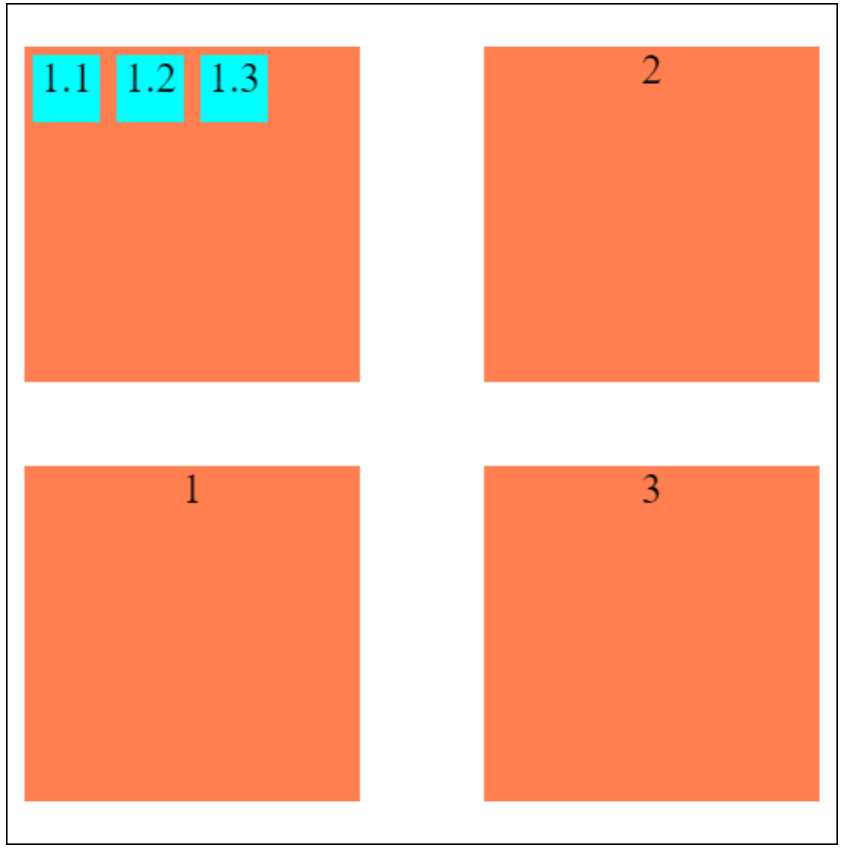
Proprietà dei box flessibili:

Proprietà	Valori	Descrizione
order	unnumero	specifica l'ordine in cui viene mostrato un box rispetto agli altri definiti nel contenitore
align-self	auto | flex-start | flex-end | center | baseline | stretch	consente di specificare per un singolo box l'allineamento sull'asse perpendicolare, superando caso per caso le impostazioni definite con la proprietà align-items
flex-grow	unnumero	consente di impostare il fattore di ingrandimento di un box rispetto agli altri presenti nel contenitore quando si distribuisce lo spazio disponbile
flex-shrink	unnumero	è la proprietà opposta rispetto a flex-grow, dal momento che agisce sul fattore di restringimento relativo tra i box
flex-basis	unnumero	consente di specificare la dimensione principale iniziale (in valori assoluti o in percentuale) di un box
flex		proprietà a sintassi abbreviata per dichiarare in un'unica regola i valori per flex-grow, flex-shrink e flex-basis.

Layout flexbox: esempio

```css
<style>

    section{
        display: flex;
        flex-flow: row wrap;
        height: 500px;
        border: 1px solid black;
        justify-content: space-between; /* orizzontalmente */
        align-items: center; /* verticalmente */
    }

    div{
        height: 200px;
        width: 200px;
        background-color: coral;
        margin: 10px;
        font-size: 25px;
        text-align: center;
    }

    .box4{
        order: 1;
        display: flex;
        flex-direction: row;
    }

    .box2{
        order: 2;
    }

    .box1{
        order: 3;
    }

    .box3{
        order: 4;
    }

    .box_nested{
        width: 40px;
        height: 40px;
        margin: 5px;
        background-color: cyan;
    }

</style>
```

```html
<body>

    <section>
        <div class="box1">1</div>
        <div class="box2">2</div>
        <div class="box3">3</div>
        <div class="box4">
            <div class="box_nested">1.1</div>
            <div class="box_nested">1.2</div>
            <div class="box_nested">1.3</div>
        </div>
    </section>


</body>
```



Layout gridlayout

Il gridlayout di CSS è un layout model che consente di tenere sotto controllo la dimensione e il posizionamento di elementi box così come del loro contenuto.

Il gridlayout rappresenta una soluzione alternativa rispetto a flexbox e diversi layout possono essere definiti privilegiando uno di questi due approcci ottenendo il medesimo risultato; dal punto di vista dei progetti basati sui layout grid-based entrambi i due moduli presentano dei costrutti simili, ma gridlayout offre interessanti opportunità per esplicitare la collocazione degli elementi in una griglia e consente di apportare modifiche sostanziali alla struttura di un layout senza richiedere cambiamenti rilevati a livello di markup.

Sostanzialmente gridlayout prevede la realizzazione di una griglia formata da un contenitore principale da definire tramite display: grid, e da dei componenti figli che lo sviluppatore può organizzare il posizionamento e l'ordine di questi ultimi indipendentemente dalla loro collocazione nel markup di pagina.

Layout gridlayout

Proprietà del contenitore:

Proprietà	Valori	Descrizione
display	grid	è il valore della proprietà display con cui si imposta un contenitore a matrice
grid-template-columns	valore-1, …, valore-n in px o in percentuale	specifica il numero e la larghezza delle colonne
grid-template-rows	valore-1, …, valore-n in px o in percentuale	specifica il numero e la larghezza delle righe
grid-auto-columns	valore in px o in percentuale	specifica una larghezza delle colonne di default
grid-auto-rows	valore in px o in percentuale	specifica una larghezza delle righe di default
row-gap	valore in px o in percentuale	specifica la distanza tra le righe
column-gap	valore in px o in percentuale	specifica la distanza tra le colonne
gap		specifica la distanza tra le righe e le colonne, raggruppa row-gap e column-gap

Layout gridlayout

Proprietà dei box flessibili:

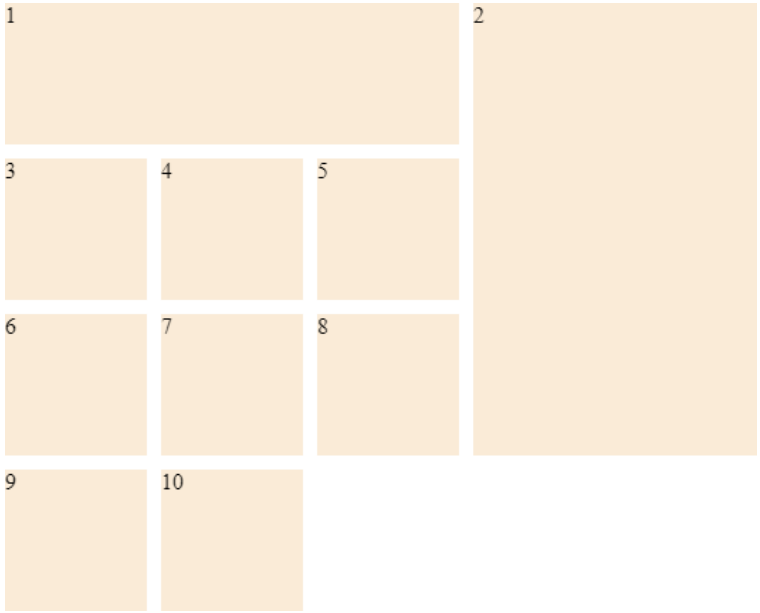
Proprietà	Valori	Descrizione
grid-row	Uno o due numeri	Specifica la riga del box (cella) o l'inizio e la fine del box
grid-column	Uno o due numeri	Specifica la colonna del box (cella) o l'inizio e la fine del box

N.B. se non specificata riga e/o colonna viene presa la prima disponibile

Layout gridlayout: esempio

```css
.grid_container{
    display: grid;
    grid-template-columns: 100px 100px 100px 200px;
    grid-auto-rows: 100px;
    gap: 10px;
}

.grid_container > div{
    background-color: antiquewhite;
}

.box1{
    grid-column: 1 / 4;
    grid-row: 1;
}

.box2{
    grid-row: 1 / 4;
    grid-column: 4;
}
```

```html
<div class="grid_container">
    <div class="box1">1</div>
    <div class="box2">2</div>
    <div class="box3">3</div>
    <div class="box4">4</div>
    <div class="box5">5</div>
    <div class="box6">6</div>
    <div class="box7">7</div>
    <div class="box8">8</div>
    <div class="box9">9</div>
    <div class="box10">10</div>
</div>
```



Per i più curiosi…

Esistono un’infinità di layout online. Un altro molto diffuso è il columnlayout:
https://www.w3schools.com/css/css3_multiple_columns.asp

Sitografia

https://www.html.it/guide/guida-css-di-base/
https://www.w3schools.com/css/