

# Web Design

prof.ssa Cristina Iurissevich  
cristinaiurissevich@abacatania.it

## Indice lezione:

1. HTML
  - a. Head
  - b. Immagini
  - c. Tabelle
2. CSS
  - a. Testo
  - b. Immagini e media
  - c. Unità di misura assoluta
  - d. Unità di misura relativa
  - e. I colori
3. Esercizio in aula
4. Esercizi a casa

# HTML

# Lista della spesa

## La mia lista della spesa

Ricordati di comprare la verdura al mercato

- zucchine
- pomodori
- pasta
- riso

## Spesa per strudel di sabato

1. cannella
2. pasta sfoglia
3. mele
4. uova
5. noci

# Head

---

L'intestazione (**<head>**) di un documento HTML è la parte che non viene visualizzata nel browser web quando viene caricata la pagina. Contiene informazioni come il <title> della pagina, i collegamenti ai fogli di stile CSS, collegamenti a favicon personalizzate e altri tipi di metadati (come ad esempio l'autore ed eventuali parole chiave importanti che descrivere il documento).

I browser Web utilizzano le informazioni contenute nell'intestazione per visualizzare correttamente il documento HTML.

# <head>

```
<!doctype html>
<html lang="en-US">
  <head>
    <meta charset="utf-8" />
    <title>La mia pagina</title>
  </head>
  <body>
    <p>Pagina web</p>
  </body>
</html>
```

L'intestazione HTML è il contenuto dell'elemento <head>. A differenza del contenuto dell'elemento <body> (che viene visualizzato sulla pagina quando viene caricato in un browser), il contenuto dell'head non viene visualizzato sulla pagina. All'interno di <head> sono contenuti i metadati del documento.

Nell'esempio all'interno dell'head troviamo il titolo della pagina (title) che sarà visibile nella finestra del browser e la codifica dei caratteri (charset="utf-8")

# <link>

```
<link rel="stylesheet" href="foglio-stile.css" />  
<link rel="icon" href="favicon.ico" type="image/x-icon" />
```

I link ci danno la possibilità di creare dei collegamenti tra il foglio html e altri collegamenti esterni (come ad esempio i fogli di stile, la favicon, le font,...)

Nel primo esempio troviamo link stylesheet con il collegamento al foglio di stile.

Nel secondo link icon con il collegamento alla favicon.

# viewport

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

La viewport è l'area visibile dall'utente di una pagina web che varia a seconda del dispositivo.

Questa meta fornisce al browser istruzioni su come controllare le dimensioni e il ridimensionamento della pagina.

La parte *width=device-width* imposta la larghezza della pagina in modo che segua la larghezza dello schermo del dispositivo.

La parte *initial-scale=1.0* imposta il livello di zoom iniziale quando la pagina viene caricata per la prima volta dal browser.



# Immagini

---

Per inserire una semplice immagine in una pagina web, utilizziamo l'elemento **<img>**. Questo è un elemento vuoto (ovvero, non può avere alcun contenuto all'interno e non ha un tag di chiusura) e richiede due attributi per essere utile: **src** e **alt**.

L'attributo **src** (source) contiene un URL che punta all'immagine che desideri incorporare nella pagina. L'attributo src può essere un URL relativo o assoluto. Senza un attributo src, un elemento img non ha immagini da caricare.

L'attributo **alt** (alternative text) è una descrizione testuale dell'immagine, che verrà utilizzata in situazioni in cui l'immagine non può essere vista/visualizzata o richiede molto tempo per essere visualizzata a causa di una connessione Internet lenta.

# Attributi <img>

```

```

Gli attributi larghezza (**width**) e altezza (**height**) vengono utilizzati per specificare la larghezza e l'altezza dell'immagine. Sono dati come numeri interi senza unità e rappresentano la larghezza e l'altezza dell'immagine in pixel.

L'attributo titolo (**title**) alle immagini creerà un tooltip al passaggio del mouse. Questo attributo presenta una serie di problemi di accessibilità.

# Tabelle

— — —

```
<table>
  <tr>
    <th>Prima cella</th>
    <th>Seconda cella</th>
    <th>Terza cella</th>
    <th>Quarta cella</th>
  </tr>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
    <td>4</td>
  </tr>
</table>
```

Una tabella è un insieme strutturato di dati composto da righe e colonne (dati tabellari). Una tabella consente di cercare in modo semplice e veloce valori che indicano una sorta di connessione tra diversi tipi di dati.

Prima cella	Seconda cella	Terza cella	Quarta cella
1	2	3	4

# Tag tabelle

Il contenuto di ogni tabella è racchiuso da questi due tag: **<table></table>**.

Il contenitore più piccolo all'interno di una tabella è una cella di tabella, creata da un elemento **<td>** (dati della tabella). Le celle non sono posizionate una sotto l'altra, ma sono automaticamente allineate tra loro sulla stessa riga. Ogni elemento **<td>** crea una singola cella e insieme costituiscono la prima riga. Ogni cella che aggiungiamo allunga la riga.

Per fermare la crescita di questa riga e iniziare a posizionare le celle successive su una seconda riga, dobbiamo utilizzare l'elemento **<tr>** (riga della tabella).

L'elemento **<th>** (intestazione tabella) funziona esattamente allo stesso modo di **<td>**, tranne per il fatto che denota un'intestazione e non una cella normale.

# Racconto

## Ospiti e letti

La casa in cui vivo con la mia famiglia è *piccola* ma **accogliente**, che sia piccola è un dato di fatto, che sia accogliente invece lo hanno decretato in massa i miei parenti.

Questo fine settimana, ad esempio, arrivano alcuni cugini di non so quale grado a farci visita.

L'arrivo di questi ospiti segna inevitabilmente l'inizio di spossanti giochi di potere all'interno della famiglia, doppi e tripli giochi sono del tutto naturali.

L'obiettivo ultimo è quello di accaparrarsi un letto decente per la notte.

A causa delle mie pessime doti diplomatiche e di una cronica incapacità di negoziare alcunchè spesso sono proprio io ad aggiudicarmi il peggiore letto della casa...

CSS

# Testo

---

Il contenuto testuale si comporta effettivamente come una serie di elementi in linea, essendo disposti su righe adiacenti l'una all'altra e non creando interruzioni di riga finché non viene raggiunta la fine della riga o a meno che non si forzi manualmente un'interruzione di riga utilizzando l'elemento `<br>`.

Le proprietà CSS utilizzate per definire lo stile del testo generalmente rientrano in due categorie:

**Stili di carattere:** proprietà che influiscono sul carattere di un testo, ad esempio quale font viene applicato, la sua dimensione e se è in grassetto, corsivo, ecc.

**Stili di layout del testo:** proprietà che influiscono sulla spaziatura e su altre caratteristiche di layout del testo, consentendo la modifica dello spazio tra linee e lettere e del modo in cui il testo è allineato all'interno del box contenuto.

# color

La proprietà **color** imposta il colore del contenuto in primo piano degli elementi selezionati, che solitamente è il testo, ma può anche includere una sottolineatura o una sovrallinea posizionata sul testo utilizzando la proprietà text-decoration.

La proprietà color accetta qualsiasi valore colore.

```
p {  
  color: green;  
}
```



# font-family

Per impostare un carattere diverso per il testo si utilizza la proprietà **font-family** che consente di specificare un carattere (o un elenco di caratteri) che il browser applicherà agli elementi selezionati. Il browser applicherà un carattere solo se è disponibile, in caso contrario utilizzerà semplicemente un carattere predefinito del browser.

Solo un certo numero di caratteri sono generalmente disponibili su tutti i sistemi e possono quindi essere utilizzati senza troppe preoccupazioni (**web safe fonts**): Arial, Courier New, Georgia, Times New Roman, Trebuchet MS e Verdana.

```
p {  
  font-family: Arial;  
}
```

# font-family

I CSS definiscono cinque nomi generici per i caratteri: serif, sans-serif, monospace, cursive e fantasy. Questi sono molto generici e l'esatto carattere utilizzato da questi nomi generici può variare a seconda del browser e del sistema operativo su cui vengono visualizzati.

Poiché non è possibile garantire la disponibilità dei caratteri (anche un carattere web potrebbe non funzionare per qualche motivo), puoi fornire più scelte di caratteri in modo che il browser possa scegliere. Ciò implica un valore font-family costituito da più nomi di font separati da virgole. In tal caso, il browser si avvia all'inizio dell'elenco e controlla se il carattere è disponibile sulla macchina. Se lo è, applica quel carattere agli elementi selezionati. In caso contrario, passa al carattere successivo e così via.

```
p {  
  font-family: "Trebuchet MS", Verdana, sans-serif;  
}
```

# font-size

Le unità più comuni per ridimensionare il testo sono:

**px** (pixel): il numero di pixel di altezza che vuoi che sia il testo.

**em**: 1 em è uguale alla dimensione del carattere impostata sull'elemento genitore dell'elemento corrente a cui stiamo applicando lo stile.

**rem**: funzionano proprio come em, tranne per il fatto che 1 rem è uguale alla dimensione del carattere impostata sull'elemento radice del documento (ovvero <html>) e non sull'elemento genitore. L'elemento radice dell'intero documento — <html> — ha una dimensione standard del carattere impostata su 16px su tutti i browser. Alcuni elementi potrebbero avere dimensioni predefinite diverse (ad esempio l'elemento h1 ha una dimensione impostata di default su 2em, quindi avrà una dimensione finale di 32px)

[https://codepen.io/cristina\\_iurissevich/pen/XWQKXzb](https://codepen.io/cristina_iurissevich/pen/XWQKXzb)

# Font style, font weight, text transform e text decoration

- **font-style:** usato per lo stile normale o corsivo. Valori: normal, italic, oblique
- **font-weight:** peso del font. Valori:
  - normal, bold: peso normale o grassetto.
  - lighter, bolder: più leggero o pesante rispetto all'elemento genitore.
  - 100 – 900: valori di grassetto numerico che forniscono un controllo più preciso.
- **text-transform:** permette di trasformare il testo. Valori:
  - none: evita qualsiasi trasformazione.
  - uppercase: trasforma tutto in maiuscolo.
  - lowercase: trasforma tutto in minuscolo.
  - capitalize: trasforma tutte le parole in modo che la prima lettera sia maiuscola.
- **text-decoration:** imposta/annulla le decorazioni del testo. Valori:
  - none: elimina le decorazioni presenti.
  - underline: sottolineatura del testo.
  - overline: sovralineatura del testo.
  - line-through: testo barrato.

# Text align

La proprietà **text-align** viene utilizzata per controllare il modo in cui il testo viene allineato all'interno della casella che lo contiene. Valori disponibili:

- **left**: giustifica il testo a sinistra.
- **right**: giustifica il testo a destra.
- **center**: centra il testo.
- **giustifica**: allarga il testo, variando gli spazi tra le parole in modo che tutte le righe di testo abbiano la stessa larghezza.

# Line height

La proprietà **line-height** imposta l'altezza di ogni riga di testo.

Questa proprietà può utilizzare la maggior parte delle unità di lunghezza e dimensione, ma può anche assumere un valore senza unità, che funge da moltiplicatore ed è generalmente considerata l'opzione migliore.

Con un valore senza unità, la dimensione del carattere viene moltiplicata e risulta nell'altezza della riga. Il corpo del testo generalmente ha un aspetto migliore ed è più facile da leggere quando le righe sono distanziate. L'altezza della linea consigliata è di circa 1,5 - 2 (interlinea doppia). Per impostare le nostre righe di testo su 1,6 volte l'altezza del carattere, utilizzeremo:

```
p {  
  line-height: 1.6;  
}
```

# Immagini e media

---

Tutto nel CSS genera un box. Se si posiziona un'immagine all'interno di un riquadro più piccolo o più grande delle dimensioni del file immagine, apparirà più piccolo del riquadro o traboccherà rispetto al riquadro, di conseguenza bisogna gestire quello che succede con l'**overflow** (cioè l'avanzo).

Una tecnica comune consiste nel rendere la larghezza massima, **max-width**, di un'immagine pari al 100%. Ciò consentirà all'immagine di diventare più piccola della scatola ma non più grande. Questa tecnica funzionerà anche con altri elementi sostituiti come <video> o <iframe>.

Se invece vogliamo ridimensionare un'immagine in modo che copra completamente una scatola possiamo utilizzare la proprietà **object-fit**.

# Object-fit

```
object-fit: fill;
```

**Fill:** L'immagine viene ridimensionata per riempire il box. Se le proporzioni dell'oggetto non corrispondono al box, l'oggetto verrà stretchato per adattarsi.

**Contain:** L'immagine viene ridimensionata per mantenere le proporzioni mentre si adatta all'interno del box. Se le proporzioni non corrispondono il box avrà degli spazi.

**Cover:** L'immagine viene ridimensionata per mantenere le proporzioni mentre riempie l'intero box. Se le proporzioni non corrispondono l'oggetto verrà ritagliato.

**None:** L'immagine non viene ridimensionata.

**Scale-down:** L'immagine viene ridimensionata come NONE o CONTAIN, a seconda di quale risulterebbe più piccola.

[https://codepen.io/cristina\\_iurissevich/pen/QWPEyJR](https://codepen.io/cristina_iurissevich/pen/QWPEyJR)



# Valori: unità di misura assoluta

— — —

<b>cm</b>	Centimeters	$1\text{cm} = 37.8\text{px} = 25.2/64\text{in}$
<b>mm</b>	Millimeters	$1\text{mm} = 1/10\text{th of } 1\text{cm}$
<b>Q</b>	Quarter-millimeters	$1\text{Q} = 1/40\text{th of } 1\text{cm}$
<b>in</b>	Inches	$1\text{in} = 2.54\text{cm} = 96\text{px}$
<b>pc</b>	Picas	$1\text{pc} = 1/6\text{th of } 1\text{in}$
<b>pt</b>	Points	$1\text{pt} = 1/72\text{nd of } 1\text{in}$
<b>px</b>	Pixels	$1\text{px} = 1/96\text{th of } 1\text{in}$

# Valori: unità di misura relativa

— — —

em	Relativo alla dimensione dell'elemento direttamente superiore (2em significa 2 volte la dimensione del font corrente)
rem	Relativo alla dimensione del carattere dell'elemento radice (elemento base), che di base corrisponde a 16px per molti browser.
vw	Dimensione relativa alla larghezza della viewport, si basa sulla percentuale (es. 50vw = 50% della larghezza della viewport)
vh	Dimensione relativa all'altezza della viewport, si basa sulla percentuale (es. 25vw = 25% della larghezza della viewport)
%	Percentuale relativo rispetto ad un altro valore, in genere quello espresso per l'elemento genitore.

# I colori

---

## **Proprietà:**

Color

Background-color

Border-color

....

## **Valori:**

nome colore

RGB o RGBa

Esadecimali

HSL

HWB

# Nomi colori

Il CSS supporta 145 nomi di colori.

<https://www.w3.org/wiki/CSS/Properties/color/keywords>.

```
h1 { color: red;}
```

Sono valori che non distinguono tra maiuscole e minuscole e che rappresentano un colore specifico, ad esempio rosso, blu, nero o verde mare. Sebbene i nomi descrivano più o meno i rispettivi colori, sono essenzialmente artificiali, senza una logica rigorosa dietro i nomi usati.

# Valori RGB o RGBA

```
h1 { color: rgb(255, 0, 0);}  
  
h1 { color: rgba(255, 0, 0, .8);}
```

Il valore **rgb()** esprime un colore in base ai suoi componenti rosso, verde e blu che sono rappresentati in ordine dai tre dati numerici valutati da 0 a 255.

Il componente alfa facoltativo, **rgba()**, rappresenta l'opacità del colore (nell'esempio sopra riportato abbiamo un valore di opacità dell'80%).

Esempio di calcolatore RGB/RGBA: <https://www.hexcolortool.com/>

# Valori esadecimali (HEX)

```
h1 { color: #ff0000;}
```

Il sistema di colori può essere scritto come numero **esadecimale**, da #000000 (nero puro) a #FFFFFF (bianco puro). I valori RGB sono espressi da “00” a “FF”.

Ad esempio, #FF0000 significa che il rosso è “255”, il verde è “0”, il blu è “0”. Il sistema di colori Hex supporta anche il colore a 3 cifre, ad esempio il colore #e1a equivale a #ee11aa e il valore che abbiamo inserito nell’esempio potrebbe essere scritto anche #F00.

La sintassi non fa distinzione tra maiuscole e minuscole: #00ff00 è uguale a #00FF00.

Esempio di calcolatore RGB/RGBA, HEX: <https://www.hexcolortool.com/>

# Prima pagina web:

Creiamo una pagina WEB (HTML e CSS) con le seguenti caratteristiche:

- Colore di sfondo alla pagina (body) diverso dal bianco
- Scegliamo un logo png con trasparenze e salviamolo in una cartella “img”
- Inseriamo il logo appena scaricato nella pagina e impostiamo l'altezza a 35px
- Sotto il logo occorre inserire un titolo <h1> di 32px
- Inseriamo un testo fittizio (lorem ipsum) di almeno 10 righe: il testo deve essere giustificato, avere un'interlinea di 1.5 e un font-size minore di 16px.

# Esercizio a casa

1. Su <https://www.freecodecamp.org/> continuiamo la certificazione “Responsive Web Design” facendo i moduli:
  - a. **“Learn CSS Colors by Building a set of Colored Markers”**
  - b. **“Learn Typography by Building a Nutrition Label”**