





# HTML

Scrivere pagine web





# Cos'è l'HTML

L'HTML è un *linguaggio di markup* che definisce la struttura dei tuoi contenuti.

È costituito da una serie di elementi che si utilizzano per racchiudere o avvolgere parti diverse del contenuto per farlo apparire in un certo modo o agire in un certo modo.

Ad esempio, la seguente riga di contenuto:

Chi si sveglia presto arriva lontano.

Rappresenta un paragrafo di testo: in una pagina html, viene avvolto dal tag di apertura e dal tag di chiusura

Chi si sveglia presto arriva lontano.



- HTML è il linguaggio di markup standard per la creazione di pagine Web;
- HTML sta per Hyper Text Markup Language;
- HTML descrive la struttura e il contenuto delle pagine Web usando il markup;
- I tag HTML etichettano parti di contenuto come "heading", "paragraph", "table"...;
- I browser non visualizzano i tag HTML, ma li usano per presentare il contenuto della pagina nel modo appropriato;
- Gli elementi HTML sono gli elementi costitutivi delle pagine HTML.



## Elemento HTML



- Il tag di chiusura è scritto come il tag di apertura, nome dell'elemento tra parentesi angolari,
   ma con una forward slash "/" inserita prima del nome dell' elemento;
- Il *contenuto*: questo è il contenuto dell'elemento, che in questo caso è solo testo.
- L'*elemento*: il tag di apertura, il tag di chiusura e il contenuto insieme costituiscono l'elemento HTML.



#### Elementi HTML (nidificazione)

Gli elementi possono essere inseriti all'interno di altri elementi. Questo si chiama nidificazione<sup>1</sup>.

```
Chi si sveglia molto presto arriva <strong>molto</
strong> lontano.
```

Usando il tag <strong> abbiamo aggiunto un elemento per rinforzare il concetto del testo.

Il modo corretto di nidificare l'elemento <strong> è di racchiuderlo all'interno dell'elemento

Esempio sbagliato di eseguire l'annidamento:

```
Chi si sveglia molto presto arriva <strong>molto
lontano.</strong>
```

<sup>&</sup>lt;sup>1</sup> I tag devono essere nidificati in modo tale che gli elementi siano tutti completamente uno dentro l'altro, senza sovrapporsi.



#### Elementi **Block** vs elementi **inline**

Ci sono due importanti categorie di elementi in HTML:

- elementi a livello di blocco (block);
- elementi in linea (inline).

Queste separazione determina le regole di nidificazione per una corretta formattazione dei documenti HTML¹

<sup>1</sup> HTML5 ha ridefinito le categorie di elementi: vedi <u>Categorie di contenuto degli elementi</u>. Queste nuove definizioni sono più accurate ma sono molto più complicate da comprendere rispetto a block e inline (<u>fonte</u>).



#### Elementi Block

Un elemento a livello di blocco **inizia sempre su una nuova riga** e **occupa l'intera larghezza disponibile** *indipendentemente dalla lunghezza del contenuto*(si estende a sinistra e a destra fino ad *occupare tutto lo spazio disponibile*).

Qualsiasi contenuto che segue un elemento a livello di blocco viene visualizzato su **una nuova riga**.

Un elemento a livello di blocco *può essere nidificato all'interno di un altro elemento* a livello di blocco. NON può essere nidificato all'interno di un elemento in linea.



#### **Elementi Inline**

Gli elementi del tipo inline si dispongono uno di fianco all'altro e occupano solo la la larghezza necessaria per mostrare il contenuto.

Un elemento in linea non farà apparire una nuova riga nel documento.

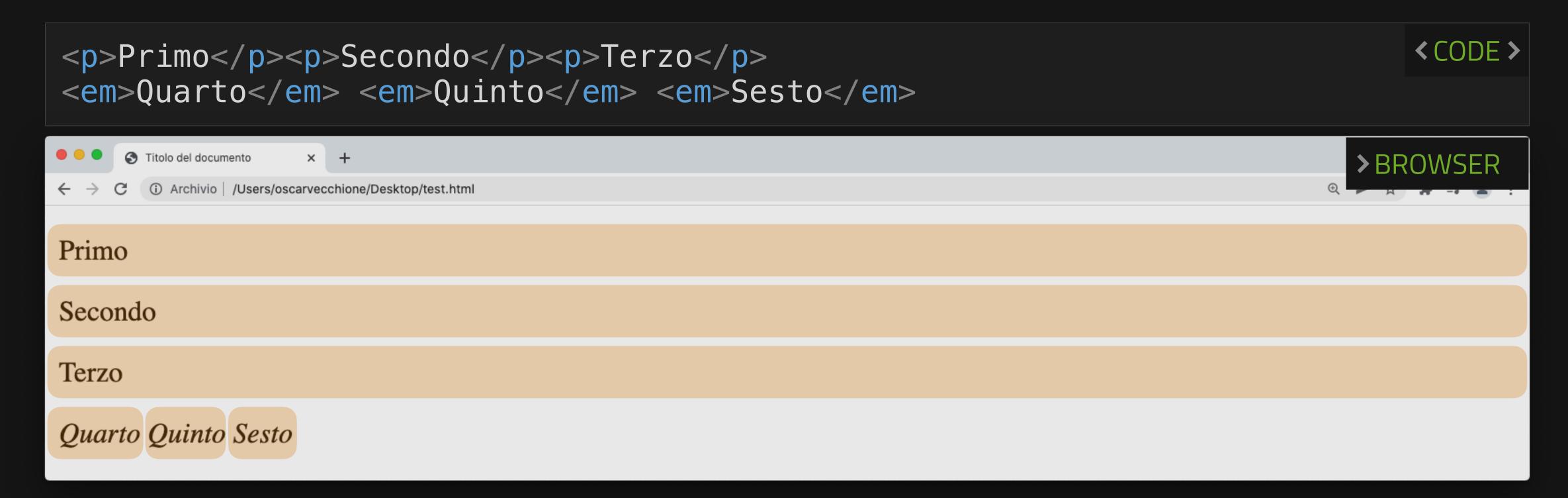
Gli elementi in linea *sono contenuti all'interno di elementi a livello di blocco* e "avvolgono" solo piccole parti di contenuto del documento.

Un elemento **inline** può essere nidificato *all'interno di un altro elemento in linea*.

Ad esempio l'elemento <a> crea un collegamento ipertestuale (link); <em> o <strong> enfatizzano il contenuto che avvolgono.



#### Consideriamo l'esempio di seguito:



elemento di blocco: occupa l'intera riga indipendentemente dalla lunghezza del contenuto, iniziano in una nuova riga;

<em> elemento inline: occupa lo spazio necessario al contenuto, i contenuti si affiancano.



#### Elementi vuoti

Gli elementi HTML senza contenuto sono chiamati elementi vuoti.



<br/>
<br/>
è un elemento vuoto senza un tag di chiusura, definisce un'interruzione di riga nel testo;



<hr> disegna una riga, per interrompere le sezioni tematiche;



per inserire un'immagine...;

per inserire un campo di input di testo all'interno di un modulo.



Tutti gli elementi HTML definiscono il tipo di contenuto che dovrà essere rappresentato dal browser.

```
<h1> descrive un titolo di primo livello,
 per scrivere un paragrafo di testo,
```

<img> per visualizzare un'immagine...

#### Elementi HTML generici

Ci sono due elementi HTML che *non* sono legati ad alcun tipo di contento:

```
<div> [elemento di tipo block]
```

<span> [elemento di tipo inline]

L'elemento <div> viene utilizzato per creare strutture HTML complesse e per disegnare il layout delle pagine: lo possiamo definire un *contenitore generico*.

L'elemento <span> viene utilizzato per inserire contenuto in linea: testo, immagini, glifi...



# Attributo HTML

Gli elementi possono anche avere attributi.

```
Attributo
class="massime">Chi si sveglia presto arriva lontano.
```

Gli attributi contengono informazioni aggiuntive sull'elemento che non verranno visualizzate nel contenuto. L' attributo *class* assegna un nome identificativo utilizzato per associare le regole di stile da applicare.

#### L'attributo deve avere:

- Uno spazio tra esso e il nome dell'elemento. (Per più di un attributo, tutti gli attributi devono essere separati da uno spazio .)
- Il nome dell'attributo, seguito da un segno di uguale.
- Un valore di attributo, racchiuso tra virgolette di apertura e chiusura.



#### Gli attributi possono avere

- un valore globale (possono essere usati per tutti gli elementi)
- un valore locale (solo per specifici elementi)
- Gli attributi sono sempre specificati all'interno del tag di inizio
- Gli attributi di solito vengono rappresentati da coppie nome / valore, tipo: id = "myparagraph"
- In alcuni casi l'attributo è fondamentale per la visualizzazione del contenuto, per esempio per stabilire dove si trova la risorsa immagine da visualizzare

```
<img src="assets/img/myphoto.jpg">
```



#### **Aggiunta di attributi**: elemento <a>

L'elemento <a> sta per ancora (ancor) . Un'ancora può trasformare il testo che racchiude in un *collegamento ipertestuale* (link).

L'elemento <a> può assumere una serie di attributi, i più usati sono i seguenti:

**href**: il valore di questo attributo specifica l'indirizzo web per il collegamento. L'attributo *href* sta per: **H**ypertext **REF**erence.

**title**: specifica informazioni aggiuntive sul collegamento, ad esempio una descrizione della pagina a cui è collegato. Questa appare come un suggerimento quando un cursore passa sopra l'elemento<sup>1</sup>.

**target**: l' attributo target specifica il contesto di navigazione utilizzato per visualizzare il collegamento. Ad esempio, target="\_blank" visualizzerà il collegamento in una nuova scheda.

```
<a href="https://www.mozilla.org/it/" title="vai al sito Mozilla" target="_blank">Sito Mozilla</a>
```

<sup>&</sup>lt;sup>1</sup> Approfondimento sull'utilità dell'attributo title



#### Attributi booleani

Gli *attributi booleani* possono avere un solo valore, che generalmente è lo stesso del nome dell'attributo.

L' attributo disabled disabilita l'input del modulo in modo che l'utente non possa inserire voci.

```
<input type="text" disabled="disabled">
```

In HTML5 è accettabile scrivere anche così:

```
<input type="text" disabled>
```



### Uso delle virgolette per i valori dell'attributo

Lo standard HTML5 non richiede virgolette attorno ai valori degli attributi.

L' attributo href può essere scritto senza virgolette:

```
<a href=https://www.mioblog.it>
<a href="https://www.mioblog.it">
```

Ma a volte è necessario usare le virgolette. L'esempio seguente non mostrerà correttamente l'attributo titolo, perché contiene degli spazi:



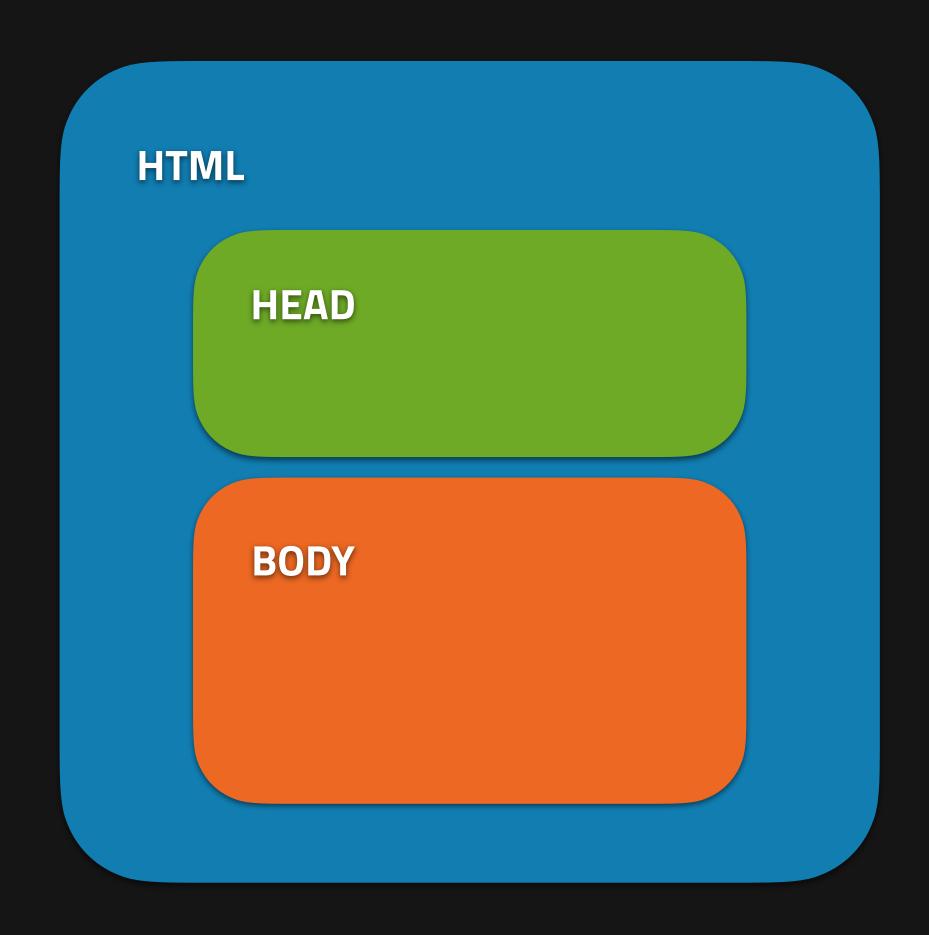
## Virgolette singole o doppie?

Le doppie virgolette attorno ai valori degli attributi sono le più comuni in HTML, ma è possibile utilizzare anche le virgolette singole.

In alcune situazioni, quando il valore dell'attributo contiene le virgolette, è necessario utilizzare le virgolette singole:



# Struttura di una pagina HTML



Un documento HTML è composto dall'elemento HTML (considerato l'*elemento radice*) che contiene due elementi:

**HEAD**: **intestazione della pagina** -> contiene le meta informazioni sul documento, le dipendenze CSS per gli stili, Javascript per i comportamenti... non visibili nel browser

BODY: corpo della pagina -> contenuti del documento, testo, immagini, audio, video...

visibili nel browser



## Esempio di pagina HTML

```
<!DOCTYPE html>
<html lang="it">
    <head>
        <meta charset="utf-8">
        <title>Titolo della pagina</title>
    </head>
    <body>
        <h1>Titolo del paragrafo</h1>
       Testo del paragrafo.
    </body>
</html>
```



#### <!DOCTYPE>

La dichiarazione *<!DOCTYPE>* indica la versione HTML utilizzata nel documento e aiuta i browser a interpretare, e quindi visualizzare correttamente le pagine Web.

Deve apparire solo una volta, nella parte superiore della pagina (prima di qualsiasi tag HTML).

Dichiarazione per i documenti HTML5:

<!DOCTYPE html>



# Versioni HTML

### Un po' di storia

Versione	Anno
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01 (introduzione fogli di stile: CSS)	1999
XHTML 1.0	2000
HTML5	2014



#### <! DOCTYPE>

alcuni Esempi

#### HTML5

<!DOCTYPE html>

#### HTML 4.01 Strict

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

#### HTML 4.01 Transitional

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

#### HTML 4.01 Frameset

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">



## SCHEMA DI LETTURA: es. HTML 4.01 Transitional

Elemento	Descrizione
HTML	il tipo di linguaggio utilizzato è l'HTML
PUBLIC	il documento è pubblico
W3C	il documento fa riferimento alle specifiche rilasciate dal W3C
	(è il segno "meno") le specifiche non sono registrate all'ISO (organizzazione di standardizzazione internazionale). Se lo fossero state, ci sarebbe stato un "+"
DTD HTML 4.01 Transitional	il documento fa riferimento a una DTD ("Document Type Definition" cioè "Definizione del tipo di documento"); la versione di HTML supportata è la 4.01 "transitional"
EN	la lingua con cui è scritta la DTD è l'inglese

La dichiarazione <!DOCTYPE html> dell'html5 non fa riferimento ad alcun DTD, document type definition.



#### <html></html>

Questo elemento avvolge tutto il contenuto della pagina.

È anche conosciuto come l'elemento radice: *root element*.

#### <head></head>

Questo elemento è un contenitore per i metadati. I metadati HTML sono dati relativi al documento HTML. I metadati non vengono visualizzati.

i metadati generalmente definiscono: *il titolo del documento, il set di caratteri, gli stili, gli script* e altre meta informazioni.

#### <body></body>

Questo contiene tutto il contenuto visualizzato sulla pagina, inclusi testo, immagini, video, tracce audio...



#### Spazi bianchi nel contenuto

Indipendentemente dalla quantità di spazi bianchi utilizzati all'interno del contenuto dell'elemento HTML (che può includere uno o più caratteri spazio, ma anche interruzioni di riga), il parser HTML riduce ogni sequenza di spazi bianchi a un singolo spazio durante il rendering del codice.

Usare spaziature nel codice serve solo a formattare il tuo codice per renderlo più leggibile.

```
>
   Questo paragrafo
    contiene molte linee
    nel codice sorgente,
    ma il browser
     le ignora
 >
           paragrafo
 Questo
    contiene molti
                         spazi
             codice
      nel
                      sorgente,
                     browser
       ma
    li
            ignora
```





## HTML entities

#### **Caratteri riservati**

Le **HTML entities** (entità carattere) *vengono spesso utilizzate per visualizzare caratteri riservati in HTML* (che altrimenti verrebbero interpretati come *codice HTML*: ", ', &, <, >) e caratteri invisibili (come spazi unificatori).

Possono anche essere utilizzate al posto di altri caratteri difficili da digitare con una tastiera standard (simboli matematici, tecnici e di valuta e altri comprese le emoji<sup>1</sup>...).

L' **HTML entity** è una porzione di testo ("*stringa*"), che inizia con una e commerciale ( & ) e termina con un punto e virgola ( ;).

< nel browser verrà renderizzato come: <

Per visualizzare questi caratteri come testo bisogna sostituirli con le corrispondenti HTML entities<sup>2</sup>.

- 1. Simboli ed Emoji: <a href="https://www.w3schools.com/charsets/ref\_emoji.asp">https://www.w3schools.com/charsets/ref\_emoji.asp</a>
- 2. Elenco completo: <a href="https://html.spec.whatwg.org/multipage/named-characters.html#named-character-references">https://html.spec.whatwg.org/multipage/named-characters.html#named-character-references</a>





# HTML entities

Result	Descrizione	Entity Name	Entity Number
	Spazio		
<	Minore di	<	<
<b>&gt;</b>	Maggiore di	>	>
&	E commerciale	&	&
=	Virgolette	"	"
-	Virgoletta singola (apostrofo)	'	'
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	&сору;	©
®	Marchio registrato	®	®
	Emoji, faccina con i cuoricini		😍



#### **Commenti HTML**

I tag di commento vengono utilizzati *per inserire commenti nel codice sorgente* HTML.

Puoi aggiungere commenti al tuo sorgente HTML usando la seguente sintassi:

```
<!-- Qui puoi inserire i tuoi commenti -->
```

Il punto esclamativo! va inserito solo nel tag di apertura.

I commenti *non vengono visualizzati dal browser* quindi possono aiutare a documentare il codice HTML.

Con i commenti puoi inserire notifiche e promemoria nel tuo codice:

```
Questo è un paragrafo
<!-- Ricordarsi di aggiungere un pulsante "more info" -->
```

Puoi usare i commenti per il debug, puoi commentare le righe di codice HTML, una alla volta, per cercare errori:

```
<!-- <img src="torino.jpg" alt="Foto panoramica di Torino"> -->
```



## Strumenti

- Le pagine Web possono essere create e modificate utilizzando un semplice editor di testo come Notepad
   (PC) o TextEdit (Mac).
- Tuttavia, per la scrittura di codice HTML si raccomanda l'uso di editor professionali come:
  - Visual Studio Code, Sublime Text, Atom, Brackets...
     (Gli editor professionali aiutano nella scrittura e lettura del codice e offrono funzioni di ricerca e sostituzione molto potenti, scorciatoie per scrivere velocemente codice<sup>1</sup>)...
- I browser:
  - Chrome, Firefox, IE, Edge, Opera, Safari...
     I browser sono software molto potenti, tra le diverse caratteristiche useremo le funzioni di sviluppo
- 1) scorciatoie per scrivener velocemente html: <a href="https://code.visualstudio.com/docs/editor/emmet">https://code.visualstudio.com/docs/editor/emmet</a>



#### Strumenti web:

- https://validator.w3.org/ (validatore del codice HTML)
- https://caniuse.com/ (compatibilità browser)

#### o documentazione:

- W3C II World Wide Web Consortium (W3C) è una comunità internazionale in cui i membri lavorano insieme per sviluppare standard Web.
- https://html.spec.whatwg.org/ (Standard HTML)
- MDN comunità di sviluppatori Mozilla
- W3school comunità di sviluppatori



# ELEMENTI HTML



## Heading (intestazioni)

I titoli sono definiti con i tag da <h1> </h1> a <h6> </h6>

```
<h1> </h1> definisce l'intestazione più importante.
```

<h6> </h6> definisce l'intestazione meno importante.

```
<h1>Titolo 1</h1>
<h2>Titolo 2</h2>
<h3>Titolo 3</h3>
<h4>Titolo 4</h4>
<h5>Titolo 5</h5>
<h6>Titolo 6</h6>
```



## Le intestazioni sono importanti

- I motori di ricerca utilizzano i titoli per indicizzare la struttura e il contenuto delle tue pagine web.
- Gli utenti scorrono velocemente le pagine in base ai titoli.
- È importante utilizzare i titoli per mostrare la struttura del documento.
- le intestazioni <h1> dovrebbero essere usate per le intestazioni
   principali, seguite dal <h2>, poi le meno importanti <h3> ... e così via.

**Nota**: utilizzare le intestazioni HTML solo per i titoli. Non usare le intestazioni per rendere il testo di dimensioni maggiori o in grassetto.



Questo elemento definisce un paragrafo.

I paragrafi sono generalmente rappresentati come blocchi di testo, ma i paragrafi HTML possono essere qualsiasi raggruppamento strutturale di contenuto correlato, come immagini o campi di input all'interno di un form.

I paragrafi sono elementi a livello di blocco e, in particolare, si chiuderanno automaticamente se un altro elemento a livello di blocco viene analizzato prima del tag di chiusura

```
primo paragrafo di testo, elemento di blocco
secondo paragrafo di testo, elemento di blocco
<h2>Sottotitolo</h2>
```





L'elemento <br >br > definisce un'interruzione di riga.

Il tag <br/>br> è un tag vuoto, il che significa che non ha tag di fine.

Utilizzare <br/>
se si desidera un'interruzione di riga (una nuova riga) senza iniziare un nuovo paragrafo:

Questo testo<br>è un paragrafo<br>con interruzioni di riga.

Non utilizzare l'elemento <br/> per creare spazi tra gli elementi HTML, usalo per andare a capo all'interno di un testo.





Il tag <hr> rappresenta un'interruzione tematica tra gli elementi a livello di paragrafo: ad esempio, un cambio di scena in una storia o uno spostamento di argomento all'interno di una sezione e viene spesso visualizzata come una riga orizzontale.

§1: La prima regola del Fight Club è: non parlare mai del Fight Club.

<hr>

§2: La seconda regola del Fight Club è: porta sempre i cupcakes.





L'elemento definisce il testo pre-formattato.

Il testo all'interno di un elemento viene
visualizzato in un font a larghezza fissa (monospace), in
genere *Courier*, e conserva spazi e interruzioni di riga:

Non si possono inserire paragrafi all'interno del tag

'Doloroso quello che porti e chi ti adorna, o persona civile, toglilo se vuoi respirare'

```
R A
      DOU
      LOU
     REUSE
     QUE TU
     PORTES
   ORNE O CI
    VILISÉ
   OTE-
         TU VEUX
    LA
          BIEN
   SI
           RESPI
           RER
```

**Titolo**: La Cravatte, tratto da i 'Calligrammes' **Autore**: Guillaume Apollinaire



#### Semantica del testo

HTML definisce elementi speciali per la definizione del testo con un significato speciale.

Gli elementi di formattazione sono stati progettati per visualizzare tipi di testo speciali:

```
<b>Testo in grassetto</b>
<strong>Testo importante</strong>
<i>>Testo in corsivo</i>
<i>>Testo in corsivo</i>
<mark>Testo enfatizzato</mark>

<br/>
<b
```

Questi elementi sono di tipo inline.



#### Citazioni



L'elemento <q> HTML definisce una breve citazione all'interno di un paragrafo. Elemento di tipo: inline.

I browser di solito inseriscono virgolette attorno all'elemento <q> .

L'azienda XYZ opera al 100% di energie rinnovabili: <q cite="http://www.xyz.com/green">Così proteggiamo il mondo dai cambiamenti climatici.

#### <blockquote> </blockquote>

L'elemento <blockquote> definisce una citazione estesa, elemento di tipo: block.

I browser di solito applicano un rientro per gli elementi <blockquote>.

```
Ecco una citazione dal sito web del WWF:
<blockquote cite="http://www.worldwildlife.org/who/index.html">
  Da 50 anni il WWF protegge il futuro della natura. Il WWF, la principale organizzazione
  mondiale per la conservazione, opera in 100 paesi ed è supportato da 1,2 milioni di membri negli
  Stati Uniti e da quasi 5 milioni a livello globale.
</blockquote>
```

L'uso dell'attributo cite consente di specificare la fonte della citazione.



## IMMAGINI





L'elemento <img> incorpora un'immagine¹ nel documento.

```
<img src="immagine.jpeg" width="720" height="375" alt="Descrizione
dell'immagine">
```

- L'attributo src è obbligatorio e contiene il percorso dell'immagine da incorporare.
- L'attributo alt è consigliato e contiene un testo alternativo descrittivo dell'immagine rappresentata.
- Gli attributi width e height sono consigliati per impostare la dimensione intrinseca dell'immagine, consentendole di occupare spazio *prima del caricamento*, per mitigare gli spostamenti del layout del contenuto.

I valori relativi alle dimensioni sono da intendersi in pixel: width="720" significa larghezza 720 pixel.

<sup>&</sup>lt;sup>1</sup> Formati di immagini supportati: https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Image\_types



#### Percorsi di file HTML

Un percorso file descrive la posizione di un file nella struttura di cartelle di un sito web. I percorsi dei file vengono utilizzati durante il collegamento a file esterni come:

- pagine web
- immagini
- Fogli di stile
- JavaScript

Path	Descrizione	
<img src="immagine.jpg"/>	l'immagine si trova nella stessa directory della pagina corrente	
<img src="./img/immagine.jpg"/>	l'immagine si trova nella directory delle img (directory figlia rispetto alla directory corrente)	
<img src="./assets/img/immagine.jpg"/>	l'immagine si trova nella directory delle img collocata a sua volta dentro la directory assets	
<img src="/ immagine.jpg"/>	l'immagine si trova nella directory padre rispetto alla directory corrente	

con il simbolo [.] vi muovete tra le directory ([.] per la directory corrente e [..] per la directory padre).



#### Percorsi file assoluti

Un percorso file assoluto è l'URL completo di un file:

```
<img src="https://cdn.unito.it/unito-loghi/logo_sigillo.svg"alt="immagine che
rappresenta il sigillo dell'Università degli Studi di Torino">
```

#### Percorsi file relativi

Un percorso relativo punta a un file relativo alla pagina corrente.

In questo esempio, il path punta a un file nella directory delle immagini situata nella directory corrente:

```
<img src="./assets/img/homer-felice.gif" alt="Homer Simpson, personaggio dei cartoni,
che saltella felice" width="150" height="150">
```



## LINK





In HTML, i link (collegamenti) sono definiti con il tag <a> :

```
<a href="https://www.miosito.com/blog/">Visita il mio blog</a>
```

L'attributo href specifica l'indirizzo di destinazione (https://www.miosito.com/blog/) del collegamento.

Il testo *contenuto* del collegamento è la parte visibile (Visita il mio blog).

Quando il mouse passa sopra un link, il puntatore si trasforma in una piccola mano



Nota: senza la barra in avanti (forward slash) alla fine delle sottocartelle(.../blog/) i server web aggiungeranno automaticamente la barra all'indirizzo, generando una nuova richiesta.



L'attributo href rappresenta l'URL a cui punta il collegamento iper-testuale.

I collegamenti non sono limitati agli URL basati sul protocollo HTTP: possono utilizzare qualsiasi schema URL supportato dai browser:

- Un URL assoluto: punta a un altro sito web (come href = "http://www.example.com/form.html");
- Un URL relativo: punta a un file all'interno di un sito web (come href = "form.html");
- Collegamento a un elemento con un ID specificato all'interno della pagina (come href = "# section2");
- Altri protocolli (come https: //, mailto :, tel :, sms: ecc ..)

```
<a href="mailto:oscar.vecchione@icloud.com">scrivimi via email</a>
```

<a href="tel:3331112233">chiamami</a>



L'attributo target specifica dove aprire il documento collegato.

Può avere i seguenti valori maggiormente usati:

\_blank: apre il documento collegato in una nuova finestra o scheda

\_self: apre il documento collegato nella stessa finestra / scheda (questo è il valore predefinito)

Questo esempio aprirà il documento collegato in una nuova finestra / scheda del browser:

```
<a href="https://www.miosito.com/blog/"target="_blank">Visita il mio blog</a>
```

L'attributo globale title descrive il link inserito nell'href, più ingenerale descrive la funzione del collegamento.

```
<a href="https://www.miosito.it/blog/" title="Questo è il collegamento al mio
blog">Visita il mio blog</a>
```



L'attributo download specifica che la risorsa collegata verrà scaricata quando l'utente fa clic sul collegamento ipertestuale;

Il valore dell'attributo sarà il nome del file scaricato;

l'attributo href deve essere impostato.

Se il valore dell'attributo viene omesso, viene utilizzato il nome file originale.

```
<a href="assets/document/lez-I.pdf">Visualizza le slide relative alla prima lezione</a>
<br/>
<a href="assets/document/lez-I.pdf" download>Scarica le slide relative alla prima lezione</a><br/>
<a href="assets/document/lez-I.pdf" download="Prima lezione">Scarica le slide relative alla prima lezione con il nome del file specificato nell'attributo</a>
```

Non funziona in locale. Dovete usare un web server, come il LiveServer di Visual Studio Code.

Nota: download funziona solo per URL della stessa origine (https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin\_policy)



#### Link HTML - Segnalibri (anchor)

I segnalibri HTML vengono utilizzati per consentire di passare a sezioni specifiche di una pagina Web, utili se la pagina web è molto lunga.

Quando si fa clic sul collegamento, la pagina scorrerà fino alla posizione del segnalibro.

Per creare un segnalibro si assegna l'attributo id al primo elemento della sezione a cui vuoi rimandare il lettore:

```
<h2 id="capitolo3">Capitolo 3</h2>Lorem ipsum sit dolor...
```

Aggiungi un collegamento al segnalibro ("Vai al capitolo 3"), nella stessa pagina, in genere in un indice rappresentato in una colonna laterale della pagina web:

```
<a href="#capitolo3">Vai al capitolo 3</a>
```

Oppure aggiungi un link al segnalibro "Vai al capitolo 3" che si trova in un'altra pagina:

```
<a href="capitoli3_4.html#capitolo3">Vai al capitolo 3 in un'altra pagina</a>
```



## Immagine come collegamento

Per utilizzare un'immagine come collegamento (link ad altra pagina), inserisci il tag <img> all'interno del tag <a> :

```
<a href="https://it.wikiquote.org/wiki/Proverbi_latini"
title="vai alla pagina dedicata ai proverbi latini">
        <img src="assets/img/boh.gif" alt="omino pensieroso"
width="60" height="60">
    </a>
```



# LISTE



### 

Il tag ul> rappresenta una lista non ordinata (unordered list).

Ogni elemento della lista (*list item*) inizia con il tag e si chiude con

Gli elementi dell'elenco saranno contrassegnati con punti elenco (piccoli cerchi neri):

```
Pane
Latte
Frutta
Carne
```

Nota: gli elementi dell'elenco possono contenere un nuovo elenco e altri elementi HTML, come immagini e collegamenti, ecc.



Le liste HTML possono essere abbinate ad elenchi di link per creare un menu di navigazione.

```
<l
 <
  <a href="#home">Home</a>
 <|i>>
  <a href="#news">Notizie</a>
 <
  <a href="#contact">Contatti</a>
 <|i>>
  <a href="#about">Chi siamo</a>
```



Il tag rappresenta una lista ordinata di elementi (ordered list).

Ogni elemento della lista (*list item*) inizia con il tag e si chiude con

Gli elementi dell'elenco saranno in genere visualizzati come un elenco numerato:

```
    Mescolare farina, lievito, zucchero e sale. 
    In un'altra ciotola, mescola uova, latte e olio. 
    Mescola le due miscele insieme. 
    Riempire la teglia per muffin per 3/4. 
    Cuoci in forno per 20 minuti.
```



Lista ordinata 
 /ol> , l'attributo type

L'attributo type del tag , definisce il tipo del marcatore dell'elemento della lista:

tipo	descrizione
type = "1"	Le voci dell'elenco saranno numerate con i numeri (impostazione predefinita)
type = "A"	Le voci dell'elenco saranno numerate con lettere maiuscole
type = "a"	Le voci dell'elenco saranno numerate con lettere minuscole
type = "I"	Le voci dell'elenco saranno numerate con numeri romani maiuscoli
type = "i"	Le voci dell'elenco saranno numerate con numeri romani minuscoli

```
     Mescolare farina, lievito, zucchero e sale. 
     In un'altra ciotola, mescola uova, latte e
olio. 
     Mescola le due miscele insieme. 
     Riempire la teglia per muffin per 3/4. 
     Cuoci in forno per 20 minuti.
```

lettere maiuscole numeri romani



Lista ordinata , l'attributo start

Per impostazione predefinita, un elenco ordinato inizierà il conteggio da 1.

Se si desidera iniziare il conteggio da un numero specificato, è possibile utilizzare l'attributo start



Lista ordinata , l'attributo reversed

Questo attributo booleano specifica che gli elementi dell'elenco sono in ordine inverso. Gli elementi saranno numerati dal più alto al più basso.

```
<h2>Countdown</h2>

  Tre
  Due
  Uno
```



### Elenchi di descrizione HTML

HTML supporta anche gli elenchi di descrizioni.

Un elenco di descrizioni è una lista di termini, con una descrizione di ciascun termine, per esempio per scrivere un glossario.

- <dl> </dl> definisce l'elenco di descrizione;
- <dt></dt> contiene il termine (nome);
- <dd></dd> contiene la descrizione del termine;

Link: <u>criptide</u>



# TABELLE



#### 

L' elemento HTML rappresenta i dati tabulari, ovvero le informazioni presentate in una tabella bidimensionale composta da righe e colonne di celle contenenti dati.

È strutturata in sezioni e può avere:

- <caption> elemento opzionale (titolo della tabella);
- zero o più elementi <colgroup> , elemento opzionale (raggruppa semanticamente le colonne);
- un elemento <thead> opzionale (intestazione della tabella);
- elemento opzionale (corpo della tabella);
- uno o più elementi 

   (righe della tabella), contenenti i , celle dei dati o , celle delle intestazioni;
- un elemento <tfoot> opzionale, footer della tabella.

Per impostazione predefinita, le intestazioni delle tabelle sono in *grassetto* e *centrato*.

Di default la tabella viene renderizzata *senza bordi* e con le *celle separate*.

Il layout è impostato su *auto*: le dimensioni delle celle *dipendono dai contenuti*.



#### Tabella semplice

```
VersioneAnno
HTML1991
HTML 2.01995
HTML 3.21997
HTML 4.01 (introduzione fogli di stile: CSS)1999
XHTML 1.02000
HTML52014
```



#### Tabella con elementi opzionali utili per lo styling

```
<caption>L'HTML nel tempo</caption>
<thead>
 VersioneAnno
 </thead>
HTML1991
 HTML 2.01995
 HTML 3.21997
 HTML 4.01 (introduzione fogli di stile: CSS)1999
 XHTML 1.02000
 HTML52014
```





L' elemento definisce una cella come intestazione di un gruppo di celle della tabella.

L'attributo **scope** specifica se una cella di intestazione è un'intestazione per una colonna, riga o gruppo di colonne o righe.

#### Può avere i seguenti valori:

- row: L'intestazione si riferisce a tutte le celle della riga a cui appartiene.
- col : L'intestazione si riferisce a tutte le celle della colonna a cui appartiene.
- rowgroup: L'intestazione appartiene a un gruppo di righe e si riferisce a tutte le sue celle.
- colgroup: L'intestazione appartiene a un gruppo di colonne e si riferisce a tutte le sue celle.
- auto



Tabella con intestazioni di riga e di colonna:

```
<caption>Grandi campioni di calcio</caption>
 Player
  Presenze
  Goal
 CR-7
  1023
  748
 Messi
  914
  723
 Maradona
  588
  312
```



### <colgroup> e <col>

L' elemento <colgroup> definisce un gruppo di colonne all'interno di una tabella.

L' elemento **col** definisce una colonna all'interno di una tabella e viene utilizzato per definire la semantica comune su tutte le celle comuni.

Si trova generalmente all'interno di un elemento <colgroup> .

Il tag <colgroup> deve essere figlio di un elemento , dopo ogni elemento <caption> e prima di qualsiasi elemento <thead> , , <tfoot> e

span (attributo)

Questo attributo contiene un numero intero positivo che indica il numero di colonne consecutive su cui si estende l' elemento.

Se non presente, il suo valore predefinito è 1.

Vedere esempio nella slide successiva.



```
<caption>Supereroi e loro aiutanti
 <colgroup>
 <col>
 <col span="2" class="batman">
 <col span="2" class="flash">
</colgroup>
 <
 Batman
 Robin
 The Flash
 Kid Flash
Skill
 Intelligenza
 Agilità
 Super velocità
 Super velocità
```

fonte: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/col



#### **Tabella HTML:** celle che si estendono su più colonne e/o righe

#### colspan

Per fare in modo che una cella si estenda su più di una colonna si usa l'attributo colspan:

Questo attributo contiene un valore intero non negativo che indica per quante colonne si estende la cella.

Il suo valore predefinito è 1.

#### rowspan

Per fare in modo che una cella si estenda su più di una riga si usa l'attributo rowspan :

Questo attributo contiene un valore intero non negativo che indica per quante righe si estende la cella.

Il suo valore predefinito è 1.



#### Tabella HTML: celle che si estendono su più colonne e/o righe

```
<caption>Articoli venduti nel 2016
Vestiti
  Accessori
 Pantaloni
  Gonne
  Bracciali
  Anelli
 Torino
  Roma
  56
  22
  72
  23
 Vittorio
  46
  18
  61
  15
 Milano
  Duomo
  89
  34
  85
  38
 Navigli
  80
  12
  36
  19
```

Articoli venduti nel 2016									
		Vestiti		Accessori					
		Pant	aloni Gonne	Bracciali	Anelli				
Torino	Roma	56	22	72	23				
	Roma Vittorio	46	18	61	15				
Milano	Duomo		34	85	38				
	Navigli	80	12	36	19				



# FORM



### <form>

L'elemento <form> definisce un modulo che viene utilizzato per raccogliere l'input dell'utente:

```
<form>
<!-- elementi del form --->
</form>
```

Un form (modulo) HTML contiene elementi di immissione dati.

Gli elementi del modulo rappresentano diversi tipi di elementi di **input**, come *campi di testo, caselle* di controllo, elenchi selezionabili, aree di testo, pulsanti di opzione, pulsanti di invio e altro.

L'elemento <form> non è visibile, sono visibili alcuni suoi elementi.

Non si possono annidare più form.



#### Elementi del form

<input> , <select> e <textarea> .

L' elemento HTML <input> viene utilizzato per creare controlli interattivi per moduli basati sul Web al fine di accettare dati dall'utente; è disponibile un'ampia varietà di tipi di dati di input e widget di controllo, a seconda del dispositivo e dello user-agent.

L' input può essere visualizzato in diversi modi, a seconda dell'attributo type .

L' elemento <input> è uno dei più potenti e complessi in tutto l'HTML a causa dell'enorme numero di combinazioni di *tipi di input* e *attributi*.



### <Input>

```
type="text"
<input type="text"> definisce un campo a una riga per l'immissione di testo :
   <label for="user">User:</label><br>
   <input type="text" name="user" id="user"><br>
type="password"
<input type="password"> definisce un campo password :
   <label for="password">Password:</label><br>
```

I caratteri in un campo password vengono mostrati come asterischi o pallini a seconda del browser.

<input type="password" name="password" id="password"><br>



```
type="submit"
```

<input type="submit"> definisce un pulsante per l'invio dei dati del modulo a un gestore del form.

Il **form-handler** (gestore del form) è in genere un file sul server con uno script per l'elaborazione dei dati di input.

Il form-handler è specificato nell'attributo action del form:

```
<form action="/action.php">

<label for="firstname">Nome:</label><br>
<input type="text" name="firstname" id="firstname"><br>
<label for="lastname">Cognome:</label><br>
<input type="text" name="lastname" id="lastname"><br>
<input type="submit" value="Invia"></form>
```

Se ometti l'attributo del valore del pulsante di invio, il pulsante verrà mostrato con il testo predefinito: "invia".



```
<button type="submit">Invia</button>
```

<button type="submit"> definisce un pulsante per l'invio dei dati del modulo al posto di input.

```
<form action="/action.php">
 <label for="firstname">Nome:</label><br>
 <input type="text" name="firstname" id="firstname"><br>
 <label for="lastname">Cognome:</label><br>
 <input type="text" name="lastname" id="lastname"><br>
 <button type="submit">Invia</button>
</form>
```

L'elemento button viene preferito per via di un maggior controllo sugli stili.



Organizzazione del form: <label>



Il tag <label> rappresenta una didascalia per un elemento in un'interfaccia utente.

L'elemento <label> non viene visualizzato come qualcosa di speciale per l'utente.

Fornisce un miglioramento dell'usabilità se arricchito dall'attributo **for** : quando l'utente fa clic sul testo all'interno dell'elemento **<label>** , attiva il controllo.

L'attributo for del tag <label> deve riferirsi all'attributo id dell'elemento correlato.

```
<label for="fisrstname">Nome:</label><br>
<input type="text" name="firstname" id="fisrstname"><br>
<label for="lastname">Cognome:</label><br>
<input type="text" name="lastname" id="lastname">
```

Un'etichetta può essere associata anche posizionando l'elemento <input> all'interno dell'elemento

```
<label><input type="radio" name="gender"> M </label>
```



Organizzazione del form: <fieldset>

Il tag <fieldset> viene utilizzato per raggruppare diversi controlli ed etichette all'interno di un modulo web.

# <legend>

definisce una didascalia per il <fieldset> genitore.

```
<fieldset>
  <legend>Anagrafica:</legend>
  <label for="firstname">Nome:</label><br>
  <input type="text" name="firstname" id="firstname"><br>
  <label for="lastname">Cognome:</label><br>
  <input type="text" name="lastname" id="lastname">
  </fieldset>
```

Disegna un bordo attorno agli elementi correlati.



```
type="radio"
<input type="radio"> definisce una opzione di scelta.
```

I pulsanti di opzione consentono all'utente di selezionare solo una tra un numero limitato di scelte:

```
<label for="male"><input type="radio" name="gender" id="male" value="M"> M </label><br><label for="female"><input type="radio" name="gender" id="female" value="F"> F </label>
```

```
type="checkbox"
```

```
<input type="checkbox"> definisce una casella di controllo .
```

Le caselle di controllo consentono di selezionare più opzioni da un numero limitato di scelte.

```
<fieldset>
    <legend>Veicoli posseduti:</legend>
    <label for="cycle"><input type="checkbox" id="cycle" name="vehicle" value="cycle"> Bicicletta</label><br>
    <label for="car"><input type="checkbox" id="car" name="vehicle" value="car"> Automobile</label></fieldset>
</fieldset></fieldset>
```

Per gli input checkbox e radio il value è obbligatorio per passare il valore al form-handler.



```
type="file"
```

<input type="file"> consente all'utente di scegliere uno o più file dalla memoria del
dispositivo e di inviarlo al server.

L'attributo accept definisce uno o più identificatori di tipo di file univoci che identificano i tipi di file da consentire;

```
<label for="avatar">Scegli l'avatar:</label>
<input type="file"id="avatar" name="avatar" accept="image/png, image/jpeg">
```

```
type="button"
```

<input type="button"> definisce un pulsante:

```
<input type="button" onclick="alert('Ciao!');" value="Clicca qui!">
```



```
type="reset"
```

<input type="reset"> definisce un pulsante di ripristino che riporterà tutti i valori del modulo ai valori predefiniti.

```
<form>
    <label for="firstname">Nome:</label><br>
    <input type="text" name="firstname" id="firstname"><br>
    <label for="firstname">Cognome:</label><br>
    <input type="text" name="lastname" id="lastname"><br>
        <input type="reset" value="Cancella">
        <input type="submit" value="Invia">
        </form>
```

Se si modificano i valori di input e quindi si fa clic sul pulsante "Ripristina", i campi dati vengono svuotati.

"Ripristina" è il valore di default del pulsante <input type="reset">



## type="hidden"

Gli input nascosti possono essere utilizzati ovunque si desideri includere dati che l'utente non può vedere o modificare insieme al modulo quando viene inviato al server.

Uno degli usi più comuni degli input nascosti è tenere traccia di quali record del database devono essere aggiornati quando viene inviato un modulo di modifica.



```
5
```

```
Nuovi <input type="...">
```

```
type="date"
```

<input type="date"> stabilisce che il dato che verrà immesso è di tipo data.

A seconda del supporto del browser viene visualizzato un selettore di date.

```
<label for="bday">Scegli una data:</label>
<input type="date" name="bday" id="bday">
```

```
type="url"
<input type="url">
```

viene utilizzato per i campi di input che dovrebbero contenere un indirizzo URL.

A seconda del supporto del browser, il campo dell'URL può essere automaticamente convalidato quando inviato.

Alcuni smartphone lo riconoscono e aggiungono ".com" alla tastiera per far corrispondere l'input dell'URL.

```
<label for="user_home">Inserisci la tua home page:</label>
<input type="url" name="url" id="user_home">
```



```
type="email"
```

<input type="email"> stabilisce che il dato che verrà immesso è di tipo email.

A seconda del supporto del browser, l'indirizzo e-mail può essere automaticamente convalidato quando inviato.

Alcuni smartphone riconoscono il tipo di email e aggiungono ".com" alla tastiera per abbinare l'input e-mail.

```
<label for="email">Email:</label>
<input type="email" name="email" id="email">
```

type="number"

<input type="number"> stabilisce che il dato che verrà immesso è di tipo numero.

È possibile impostare restrizioni sui numeri accettati. Nell'esempio seguente viene visualizzato un campo di input numerico, in cui è possibile immettere un valore compreso tra 18 e 30:

```
<label for="quantity">Scegli una quantità:</label>
<input type="number" name="quantity" min="18" max="30" id="quantity">
```



```
type="range"
<input type="range">
```

Definisce un controllo (come un cursore) per immettere un numero il cui valore esatto non è importante.

Il range di default va da 0 a 100. Tuttavia, è possibile impostare restrizioni sui numeri con il min , max :

```
<label for="score">Scegli un punteggio:</label>
<input type="range" name="score" min="0" max="50" id="score">
```

```
type="search"
<input type="search">
```

Sono campi di testo progettati per consentire all'utente di inserire query di ricerca (un campo di ricerca si comporta come un campo di testo normale, ma possono essere resi differentemente dai diversi browser).

```
<label for="search">Cerca:</label>
<input type="search" name="ricerca" id="search">
```



```
type="time"
```

```
<input type="time">
```

permette all'utente di selezionare un tempo (ore e minuti).

A seconda del supporto del browser, è possibile visualizzare un time picker nel campo di input.

```
<label for="user_time">Scegli un ora:</label>
<input type="time" name="user_time" id="user_time">
```

```
type="color"
```

<input type="color"> viene utilizzato per i campi di input in cui l'utente deve scegliere un colore.

Un selettore di colori viene visualizzato per il campo di input.

```
<label for="mycolor">Scegli un colore:</label>
<input type="color" name="mycolor" id="mycolor">
```





L'elemento <select> rappresenta un controllo che fornisce un menu di opzioni, disegnato con un elenco a discesa:

Gli elementi <option> definiscono l'opzione che può essere selezionata.

Per impostazione predefinita, il primo elemento nell'elenco a discesa è selezionato.

Per definire un'opzione preselezionata, aggiungere l'attributo selected

```
<option value="fiat" selected>Fiat
```





<optgroup> viene utilizzato per raggruppare le opzioni correlate in un elenco a discesa.

```
<label for="cars">Scegli la marca di automobili preferita</label><br>
<select name="cars" id="cars">
 <option value="">--Scegli tra le opzioni--</option>
 <optgroup label="Auto italiane">
   <option value="fiat">Fiat
   <option value="alfa-romeo">Alfa Romeo</option>
   <option value="ferrari">Ferrari
   <option value="maserati">Maserati
 </optgroup>
 <optgroup label="Auto tedesche">
   <option value="mercedes">Mercedes
   <option value="audi">Audi</option>
   <option value="opel">Opel</option>
  </optgroup>
</select>
```

Se hai una lunga lista di opzioni, i gruppi di opzioni correlate sono più facili da gestire per un utente.



#### Selezioni multiple:

Utilizzare l'attributo multiple per consentire all'utente di selezionare più di un valore:

#### Valori visibili:

Utilizza l'attributo size per specificare il numero di valori visibili quando viene mostrata la select:



## <textarea>

L'elemento <textarea> rappresenta un controllo di modifica di testo normale su più righe, utile quando si desidera consentire agli utenti di immettere una quantità considerevole di testo in formato libero, ad esempio un commento su un modulo di recensione o feedback.

```
<label for="message">Lascia un commento:</label><br></textarea name="message" id="message" rows="10" cols="30"></textarea>
```

L'attributo rows specifica il numero visibile di linee in un'area di testo.

L'attributo cols specifica la larghezza visibile di un'area di testo.

Non supporta l'attributo value.



#### Attributi per l'invio del modulo

#### action

Il form-handler, gestore del form (file che recupera il dato inviato), è specificato nell'attributo action del form.

#### method

L'attributo method definisce il metodo HTTP con cui inviare il modulo.

Gli unici metodi/valori consentiti sono (senza distinzione tra maiuscole e minuscole):

- GET (predefinito): dati del modulo aggiunti all'URL action con un separatore ? Utilizza questo metodo quando il modulo non ha effetti collaterali.
- POST: dati del modulo inviati come corpo della richiesta

```
<form action="filehandler.php" method="post">
```

#### enctype

L'attributo enctype: è essenziale perché consente al browser di inviare dati binari, come i file, tramite il protocollo HTTP o HTTPS. utilizzalo se il modulo contiene elementi <input> con type="file"

```
<form action="filehandler.php" enctype="multipart/form-data" >
```



### Attributi importanti degli elementi del form

#### name

L'attributo name specifica il nome di un elemento del form, è fondamentale per l'invio dei dati del modulo.

```
<input type="text" name="firstname">
```

#### value

#### L'attributo value:

specifica *il valore di default per un campo di input*: Il campo "Nome" si presenterà già pre-compilato con il testo "Paolo"

```
<input type="text" name="firstname" value="Paolo">
```

specifica *il testo custom del pulsante submit* 

```
<input type="submit" value="Registrati">
```

Se non specifichi value, il pulsante avrà un'etichetta predefinita, scelta dall'*user agent* : "Invia" o "Invia richiesta".

specifica il valore da inviare al form per gli input radio e checkbox e per i valori option delle select

```
<input type="radio" name="gender" id="male" value="M"> M
```



Altri attributi degli elementi del form

## readonly

L'attributo readonly specifica che il campo di input è di sola lettura (non può essere modificato):

```
<input type="text" name="firstname" value="Paolo" readonly>
```

## disabled

L'attributo disabled specifica che il campo di input è disabilitato.

Un campo di input disabilitato è inutilizzabile e il suo valore non verrà inviato quando si invia il modulo:

```
<input type="text" name="firstname" value="Paolo" disabled>
```



#### size

L'attributo size specifica la dimensione (in caratteri) per il campo di input:

```
<input type="text" name="firstname" value="John" size="40">
```

#### cols, rows

Gli attributi cols e rows specificano rispettivamente la lunghezza e l'altezza massima consentita per il campo textarea.

```
<textarea name="message" cols="100" rows="5"></textarea>
```

#### maxlength

L'attributo maxlength specifica la quantità massima consentita di caratteri per il campo di input:

```
<input type="text" name="firstname" maxlength="10">
```

Con maxlength il campo di input non accetterà più del numero consentito di caratteri.

maxlength non fornisce alcun feedback all'utente. Se si desidera informare l'utente, è necessario scrivere del codice javascript.

**Nota**: I controlli sui valori di input (validazione del dato) vengono effettuati lato server dal linguaggio di scripting dell'applicativo.



autofocus 😈

L'attributo **autofocus** specifica quale campo di input ottiene automaticamente lo stato attivo al caricamento della pagina, il "focus".

Al caricamento della pagina il browser *attiverà* il campo del **form**> che ha l'attributo **autofocus** evidenziandolo con un bordo (*outline*) azzurro:

```
<label for="fname">Nome:</label><br><input type="text" name="fname" id="fname" autofocus>
```

min e max 🥫

Gli attributi min e max specificano i valori minimo e massimo per un elemento di <input> .

Gli attributi min e max funzionano con i seguenti tipi di input: *number, range, date, datetime-local / time, month* e *week*.

```
<label for="bday_pre">Inserisci una data antecedente al 1980-01-01:</label><br>
<input type="date" name="bday" id="bday_pre" max="1979-12-31"><br>
<label for="quantity">Quantità (da 1 a 5):</label><br>
<input type="number" name="quantity" id="quantity" min="1" max="5"><br>
```



# placeholder 🥃

L'attributo placeholder specifica un suggerimento che descrive *il valore atteso* di un campo di input (un valore di esempio o una breve descrizione del formato).

Il suggerimento viene visualizzato nel campo di *input* prima che l'utente inserisca un valore.

```
<input type="email" name="email" id="email" placeholder="esempio:
example@email.com">
```

L'attributo placeholder funziona con i seguenti tipi di **input**\*:

- text
- search
- url
- tel

- email
- password
- textarea\*



## **NOTA:** placheolder e <textarea>

Tra i tag di apertura <textarea> e di chiusura </textarea> dell'elemento non devono esserci spazi, più a capo o tabulazioni, altrimenti ll placeholder non verrà visualizzato poiché l'area di immissione contiene già del contenuto (uno spazio o una tabulazione rappresentano, tecnicamente, contenuto valido).



Si

```
<textarea placeholder="inserisci il testo"></textarea><br><textarea placeholder="inserisci il testo"></textarea><br></textarea><br></
```



 $N_0$ 

```
<textarea placeholder="inserisci il testo">
</textarea><br>
```

<sup>\*</sup> https://www.w3schools.com/tags/att\_input\_placeholder.asp





L'attributo required specifica che un campo di input **deve** essere compilato prima di inviare il modulo.

```
<input type="text" name="username" required>
```

L'attributo required funziona con i seguenti tipi di input, oltre a «select» e «textarea» :

- text
- search
- url
- · tel
- email
- password

- date pickers
- number
- checkbox
- radio
- file





Un <iframe> viene utilizzato per visualizzare una pagina Web all'interno della pagina Web.

```
<iframesrc="https://it.wikipedia.org/wiki/HTML"width="800"height="600"></iframe>
```

L'attributo src specifica l'URL (indirizzo web) della pagina frame inline.

Utilizzare gli attributi height e width per specificare la dimensione\* dell'iframe.

```
<iframesrc="page.html"width="800"height="500"></iframe>
```

Usare title per etichettare il contenuto dell'iframe:

```
<iframe title="pagina di Wikipedia sull'HTML" src="https://it.wikipedia.org/
wiki/HTML" width="800" height="500"></iframe>
```

<sup>\*</sup> Le dimensioni si possono controllare successivamente con l'uso dei CSS.



Per impostazione predefinita, un iframe ha un bordo.

Per rimuovere il bordo, aggiungi l'attributo style e usa la proprietà CSS **border**:

```
<iframe title="pagina di Wikipedia sull'HTML"
src="https://it.wikipedia.org/wiki/HTML" style="border:0"
width="800" height="500"></iframe>
```

Di solito gli iframe si usano per incorporare nel proprio sito servizi forniti da terzi: *mappe di google, stream di Facebook, twitter, instagram, servizio meteo, info-grafiche,* ecc...



### Nuovi elementi semantici in HTML5

Nel realizzare siti Web molti sviluppatori hanno usato codice tipo:

```
<div id="nav">...</div>
<div id="header">...</div>
<div id="footer">...</div>
```

per indicare rispettivamente la navigazione, l'intestazione e il piè di pagina.

Ciò ha reso impossibile per i motori di ricerca identificare il contenuto corretto delle pagine web.

HTML5 offre nuovi elementi semantici per definire parti diverse di una pagina Web:

```
<article> <aside> <details> <figcaption> <figure> <footer> <header> <mark> <nav> <section> <summary> <time>
```



# HTML5 offre nuovi elementi per una migliore struttura del documento:

tag	Descrizione del tag
<article></article>	Definisce un articolo in un documento
<aside></aside>	Definisce il contenuto a parte il contenuto della pagina
<details>*</details>	Definisce i dettagli aggiuntivi che l'utente può visualizzare o nascondere
<figcaption>**</figcaption>	Definisce una didascalia per un elemento figure (in <figure>)</figure>
<figure>**</figure>	Definisce contenuto autonomo
<footer></footer>	Definisce un piè di pagina per un documento o una sezione
<header></header>	Definisce un'intestazione per un documento o una sezione
<main></main>	Definisce il contenuto principale di un documento
<mark></mark>	Definisce il testo evidenziato
<nav></nav>	Definisce un contenitore per i collegamenti di navigazione
<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	Rappresenta lo stato di avanzamento di un'attività
<section></section>	Definisce una sezione in un documento
<summary>*</summary>	Definisce un'intestazione visibile per un elemento di dettagli (in <details>)</details>





L'elemento <header> specifica un'intestazione per un documento o una sezione.

L' <header> dovrebbe essere usato come contenitore per i contenuti introduttivi.

Puoi avere diversi <header> in un documento.

L'esempio seguente definisce un'intestazione per un articolo:





L' elemento <nav> rappresenta una sezione di una pagina il cui scopo è fornire collegamenti di navigazione, all'interno del documento corrente o ad altri documenti.

Esempi comuni di sezioni di navigazione sono i menu, i sommari e gli indici..

Si noti che NON tutti i collegamenti di un documento devono essere all'interno di un elemento <nav> . È inteso solo per la navigazione.

Possono esserci più elementi nav nel documento HTML.





L'elemento <section> rappresenta una sezione autonoma - che non ha un elemento semantico più specifico per rappresentarlo - contenuta in un documento HTML.

In genere le sezioni hanno un'intestazione.





L'elemento <article> specifica il contenuto indipendente e autonomo.

Un articolo dovrebbe avere senso da solo, e dovrebbe essere possibile leggerlo indipendentemente dal resto del sito web.

<article> può essere utilizzato per:

- Post sul forum
- Post sul blog
- Articolo di giornale
- ...



## Annidare <article> in <section> o viceversa?

<article> specifica il contenuto indipendente e autonomo.

<section> definisce la sezione in un documento.

Sulla base di queste definizioni però non consegue una regola di annidamento degli elementi.

Non esiste differenza, nello standard HTML5, tra <article> , <section> e <div> .

Infatti possiamo scrivere pagine HTML con <section> contenente <article> ed <article> contenenti <section> .

Possiamo inserire tutto in un <div> o inserire <div> nelle <section> o negli <article> .

Per esempio in un giornale di tecnologia un autore può scrivere un articolo, una recensione all'interno della quale è presente una sezione tecnica.





L' elemento <aside> rappresenta una parte di un documento il cui contenuto è solo indirettamente correlato al contenuto principale del documento.

Gli <aside> sono spesso presentati come barre laterali o riquadri di richiamo.





L'elemento <footer> specifica un piè di pagina per un documento o una sezione.

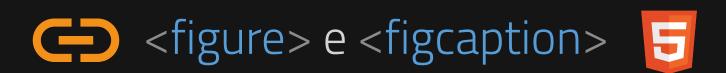
Un elemento <footer> dovrebbe contenere informazioni sul suo elemento contenitore.

Un footer in genere contiene l'autore del documento, informazioni sul copyright, collegamenti a termini di utilizzo, informazioni di contatto, ecc.

Potresti avere diversi <footer> in un documento.

```
<footer>
    Creato da: Mario Bianchi
    Contatto: <a
href="mailto:mario.bianchi@gmail.com">mario.bianchi@gmail.com</a>.
</footer>
```





Di solito a <figure> è un'immagine, un'illustrazione, un diagramma, uno snippet di codice, ecc., a cui si fa riferimento nel flusso principale di un documento, ma che può essere spostato in un'altra parte del documento o in un'appendice senza influire sul flusso principale.

<figcaption> Aggiunge una didascalia.

In HTML5, un'immagine e una didascalia possono essere raggruppate in un elemento <figure> :

```
<figure>
  <img src="assets/img/spartito.png" alt="spartito musicale..." width="300" height="111">
  <figcaption>In figura lo spartito della IX Sinfonia di LW.Beethowen</figcaption>
  </figure>
```

L'elemento <img> definisce l'immagine,

l'elemento <figcaption> definisce la didascalia: è facoltativo.





L'elemento HTML <picture> contiene zero o più elementi <source> e uno elemento <img> per offrire versioni alternative di un'immagine per diversi scenari di display/dispositivo.

Il browser prenderà in considerazione ogni elemento <source> figlio e sceglierà l'immagine migliore corrispondenza tra di questi. Se non vengono trovate corrispondenze o il browser non supporta l'elemento <picture>, viene selezionato l'URL dell'attributo src dell'elemento <img>.

```
<picture>
    <source media="(min-width: 768px)" srcset="assets/img/wave-background768.jpg">
        <source media="(min-width: 480px)" srcset="assets/img/wave-background480.jpg">
        <source srcset="assets/img/wave-background-small.jpg">
        <img src="assets/img/wave-background-small.jpg" alt="immagine di sfondo">
        </picture>
```

## Casi d'uso comuni per <picture>:

- Direzione artistica. Ritaglio o modifica di immagini per diverse condizioni media (ad esempio, caricamento di una versione più semplice di un'immagine che contiene troppi dettagli, su display più piccoli).
- Offrire formati di immagine alternativi, per i casi in cui determinati formati non sono supportati.
   Ad esempio, formati più recenti come AVIF o WEBP presentano molti vantaggi, ma potrebbero non essere supportati dal browser.
- Risparmiare larghezza di banda e accelerare i tempi di caricamento della pagina caricando l'immagine più appropriata per la visualizzazione dello spettatore.



## <figure>, <figcaption> e <picture>

È possibile combinare gli elementi <figure> e <picture> all'interno di un documento HTML.

Questa combinazione è utile quando si desidera incorporare immagini responsive all'interno di una struttura che include anche una didascalia per l'immagine.

```
<figure>
    <picture>
    <source media="(min-width: 768px)" srcset="assets/img/wave-background768.jpg">
        <source media="(min-width: 480px)" srcset="assets/img/wave-background480.jpg">
        <source srcset="assets/img/wave-background-small.jpg">
        <img src="assets/img/wave-background-small.jpg" alt="immagine di sfondo">
        </picture>
        <figcaption>Immagine di sfondo</figcaption>
        </figure>
```

In questo esempio, l'elemento <figure> racchiude l'elemento <picture> che contiene diverse fonti di immagini utilizzando l'elemento <source> per fornire alternative in base alle dimensioni dello schermo o del viewport.

L'immagine è specificata nell'elemento <img> all'interno di <picture>.

Al di fuori di <picture> c'è l'elemento <figcaption> che fornisce la didascalia associata all'immagine.



# Elementi multimediali

	Tag	Description
	<video></video>	Definisce un contenuto video
	<audio></audio>	Definisce un contenuto audio
	<source/>	Definisce più risorse multimediali per elementi multimediali ( <video> e <audio>)</audio></video>
Ð	<track/>	Definisce le tracce di testo per gli elementi multimediali ( <video> e <audio>)</audio></video>
	<embed/> *	Definisce un contenitore per un'applicazione esterna (non HTML, esempio pdf) I browser Web hanno supportato l'elemento <embed/> per un lungo periodo. Tuttavia, non ha fatto parte delle specifiche HTML prima di HTML5.

<sup>\* &</sup>lt;object> (html 4)



## Media type

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

## Supporto del browser

Browser	MP4	WebM	Ogg
Internet Explorer	SI	NO	NO
Chrome	SI	SI	SI
Firefox	SI	SI	SI
Safari	SI	NO	NO
Opera	SI (da versione 25)	SI	SI





L'elemento <video> specifica un modo standard per incorporare un video in una pagina Web.

L'attributo controls aggiunge i controlli video, come riproduzione, pausa e volume.

L'elemento < source > consente di specificare i file video alternativi che il browser può scegliere. Il browser utilizzerà il primo formato riconosciuto.

Il testo nel tag <video> verrà visualizzato solo nei browser che non supportano l'elemento <video> .



## <video>

L'attributo autoplay consente di avviare automaticamente il contenuto:

```
<video width="600" height="400" controls autoplay>
```

L'attributo **poster** specifica un'immagine da mostrare durante il download del video, o finché l'utente non preme il pulsante di riproduzione. Se poster non è specificato, verrà utilizzato il primo fotogramma del video.

```
<video controls poster="./assets/img/poster.png">
```

L'attributo muted è un attributo booleano.

```
<video controls muted>
```

Quando presente, specifica che l'uscita audio del video deve essere disattivata.

**ATTENZIONE!!!** L'attributo **autoplay** non funziona su dispositivi mobili come iPad e iPhone.

L' **autoplay** può infastidire il visitatore e causare più danni che benefici.

Dall'aprile 2018 non funziona su Chrome perché Google ha modificato le policy dell'autoplay.



#### video di YouTube\*

- Carica il video su YouTube;
- Prendi nota dell'id del video;
- Definisci un elemento <iframe> nella tua pagina web;
- inserisci l'URL del video nell'attributo src;
- Usa gli attributi width e height per specificare la dimensione del player;
- Aggiungi altri parametri all'URL (&autoplay=valore, &loop=valore,...)

```
<iframe width="420" height="315" src="https://www.youtube.com/embed/tgbNymZ7vqY"></iframe>
```

## autoplay, loop e controls

<iframe width="420" height="315" src="https://www.youtube.com/embed/tgbNymZ7vqY?&controls=0"></iframe>

**Valore= 0**: nasconde i controlli.

**Valore= 1**: mostra i controlli (default).



## Media type

File Format	Media Type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

## Supporto del browser

Browser	MP3	WAV	OGG
Internet Explorer	SI	NO	NO
Chrome	SI	SI	SI
Firefox	SI	SI	SI
Safari	SI	SI	NO
Opera	SI	SI	SI





L'elemento <audio> specifica un modo standard per incorporare l'audio in una pagina Web.

Riprodurre un file audio in HTML: tag <audio> :

```
<audio controls>
    <source src="assets/media/canzone.ogg" type="audio/ogg">
        <source src="assets/media/canzone.mp3" type="audio/mpeg">
        Il tuo browser non supporta l'elemento audio.
        </audio>
```

L'attributo controls aggiunge controlli audio, come riproduzione, pausa e volume.

L'elemento **source** consente di specificare i file audio alternativi che il browser può scegliere. Il browser utilizzerà il primo formato riconosciuto.

Per avviare automaticamente il file audio si può usare l'attributo autoplay \*

Il testo nel tag <audio> verrà visualizzato solo nei browser che non supportano l'elemento <audio> .

<sup>\*</sup> ATTENZIONE!!! L'attributo autoplay non funziona su dispositivi mobili come iPad e iPhone. L'autoplay può infastidire il visitatore e causare più danni che benefici. Dall'aprile 2018 non funziona su Chrome perché Google ha modificato le policy dell'autoplay

**E** 



## **Grafica HTML5**

Tipo	Esempio
<svg></svg>	Scalable Vector Graphics - Disegna grafica vettoriale scalabile
<canvas></canvas>	Disegna la grafica, al volo, tramite Javascript

Esempio che utilizza il tag «svg» per creare un cerchio, nel secondo esempio l' «svg» diventa un link.



## Differenze tra SVG e Canvas

<svg> è un linguaggio per descrivere la grafica 2D in XML.

<canvas> disegna grafica 2D, al volo (con un JavaScript).

SVG è basato su XML, il che significa che ogni elemento è disponibile all'interno del DOM SVG.

Con <svg>, ogni forma disegnata viene ricordata come un oggetto. Se vengono modificati gli attributi di un oggetto SVG, il browser può automaticamente eseguire il rendering della forma.

<canvas> viene disegnato pixel per pixel.

Con <canvas>, una volta terminato il disegno questo viene dimenticato dal browser. Se la posizione deve essere cambiata, è necessario ridisegnare da capo.



#### Esempio uso canvas

```
<script>
function creaF(){
   var disegno=document.getElementById('disegno');
   var context=disegno.getContext('2d');
   context.arc(100,100,50,0,Math.PI,true);
   context.lineWidth=5;
   context.strokeStyle="rgba(0,0,255,1)";
   context.stroke();
   context.fillStyle="rgba(255,255,0,1)";
   context.fill();
</script>
<button onclick="creaf()" type="button">Crea arco</button>
<canvas id="disegno" width="400" height="100">
Spiacente non supporto i canvas di HTML5 </canvas>
```

Il disegno viene creato o al click del pulsante (esempio) o al caricamento della pagina, ma comunque in seguito ad un evento che scatena il codice javascript.



## <head></head>

L'elemento <head> è un contenitore per i metadati. I metadati HTML sono dati relativi al documento HTML. I metadati non vengono visualizzati.

L'elemento <head> è posizionato tra il tag <html> e il tag <body> :

```
<!DOCTYPE html>
<html>
<head>
<title>La mia pagina web</title>
<meta charset="UTF-8">

...
</head>
<body>
```

i metadati generalmente definiscono: il titolo del documento, il set di caratteri, gli stili, i collegamenti, gli script e altre meta informazioni.



## Approfondimento sull'elemento <head>

L'elemento <head> , come abbiamo già visto, è un contenitore per i metadati, posizionato tra il tag <html> e il tag <body>.

I metadati sono dati sul documento HTML, non vengono quindi visualizzati.

I metadati generalmente definiscono il titolo del documento, il set di caratteri, gli stili, i collegamenti, gli script e altre meta informazioni.

I seguenti tag descrivono i metadati: <title> , <style> , <meta> , <link> , <script> .

### <title>

Definisce il titolo del documento ed è richiesto in tutti i documenti HTML / XHTML.

#### L'elemento <title> :

- definisce un titolo nella scheda del browser
- fornisce un titolo per la pagina quando viene aggiunto ai preferiti
- visualizza un titolo per la pagina nei risultati del motore di ricerca



## lang

La lingua viene dichiarata con l'attributo lang aggiunto all'interno del tag <html> .

Dichiarare una lingua è importante per le applicazioni di accessibilità (screen reader) e per i motori di ricerca:

Le prime due lettere specificano la lingua (en). Se c'è un dialetto, si usano altre due lettere (US) \*.

Una pagina in lingua italiana avrà la dichiarazione seguente:

```
<html lang="it">
```

<sup>\* &</sup>lt;a href="https://www.w3schools.com/tags/ref\_language\_codes.asp">https://www.w3schools.com/tags/ref\_language\_codes.asp</a>





<meta> viene utilizzato per specificare diversi tipi di dati: set di caratteri, descrizione della pagina, parole chiave, autore...

I metadati sono utilizzati dai browser (come visualizzare il contenuto), dai motori di ricerca (parole chiave) e altri servizi web.

Definisci il set di caratteri usato:

```
<metacharset="UTF-8"> <!-- nelle prime righe del doc -->
```

Definisci una descrizione della tua pagina web:

```
<meta name="description" content="il mio blog">
```

Definisci le parole chiave per i motori di ricerca:

```
<meta name="keywords" content="HTML, CSS, XML, JavaScript">
```

Definisci l'autore di una pagina:

```
<meta name="author" content="0scar Vecchione">
```



## Impostazione di Viewport

HTML5 ha introdotto un metodo per consentire ai web designer di assumere il controllo sulla vista, attraverso il tag <meta> .

Il viewport è l'area visibile dell'utente di una pagina Web. Varia a seconda del dispositivo e sarà più piccolo su un telefono cellulare che sullo schermo di un computer.

Bisogna includere il seguente elemento <meta> di visualizzazione in tutte le tue pagine web:

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

La proprietà "viewport" dell'attributo name fornisce al browser le istruzioni su come controllare le dimensioni e il ridimensionamento della pagina.

**width**=*device*-*width* imposta la larghezza della pagina sulla base della larghezza dello schermo del dispositivo (che varierà a seconda del dispositivo).

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

Questa istruzione da sola non adatta la pagina web al dispositivo, consente solo che la pagina non venga ridimensionata dal browser. Occorrerà intervenire con i css per completare l'adattamento in modo che i contenuti siano fruibili anche da un cellulare.



```
<!DOCTYPE html>
<html lang="it">
<head>
    <title>Il blog di Oscar Vecchione</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="author" content="0scar Vecchione">
    <meta name="description" content="il mio blog sulle tecnologie front-end">
    <meta name="keywords" content="HTML,CSS,XML,JavaScript">
</head>
<body>
    <h1>Titolo del testo</h1>
    Contenuto...
</body>
</html>
```



## <style></style>

L'elemento <style> viene utilizzato per definire le informazioni di stile per la singola pagina HTML:

#### k>

L'elemento <link> è usato per collegare fogli di stile esterni:

```
k rel="stylesheet" href="style.css" type="text/css">
```

L'attributo rel specifica la relazione tra il documento corrente e il documento collegato.

o per collegare la "favicon", icona utilizzata dai browser nei segnalibri, nei titoli delle schede...

```
k rel="shortcut icon" href="assets/img/favicon.ico">
```



## <script>

L'elemento <script> è usato per racchiudere codice javascript per la pagina.

Questo semplice javascript al click di un button inserisce la scritta "Ciao! Benvenuto Oscar!" in un elemento html identificato da id="msg":

```
<script>
  function saluta(){
    var user;
    user = "Oscar";
    document.getElementById("msg").innerHTML = "Ciao! Benvenuto " + user;
  }
  </script>

    id="msg">
    <button type="button" onclick="saluta()">Premi qui!</button>
```

o per richiamare sorgenti javascript esterne, nell'esempio JQuery versione 3.7.1:

```
<script src="http://code.jquery.com/jquery-3.7.1.js"></script>
```



### **Colori HTML**

I colori HTML vengono specificati utilizzando **nomi di colori** predefiniti o valori **RGB**, **HEX**, **HSL**, **RGBA**, **HSLA**.

In HTML, un colore può essere specificato usando un *nome di colore*:

```
aliceblue,
beige,
cadetblue,
darkblue,
firebrick,
ghostwhite,
hotpink,
indigo,
```

```
p {color: indigo}
```

HTML supporta **140 nomi di colori** standard.

Elenco completo: <a href="https://www.w3schools.com/colors/colors\_names.asp">https://www.w3schools.com/colors/colors\_names.asp</a>



### **Valore RGB**

In HTML, un colore può essere specificato come valore RGB, utilizzando questa formula:

```
rgb(0,0,0)
```

Ogni parametro (canale rosso, verde e blu) definisce l'intensità (luminosità) del colore tra 0 e 255.

Ad esempio, rgb(255,0,0) visualizza il rosso, poiché il canale del rosso è impostato sul valore più alto (255) e gli altri sono impostati su 0.

Per visualizzare il colore nero, tutti i *canali dei colori* devono essere impostati su 0 :

```
p {color: rgb(0,0,0)}
```

Per visualizzare il colore bianco, tutti i *canali dei colori* devono essere impostati su 255:

```
p {color: rgb(255,255,255)}
```

Le sfumature di grigio sono spesso definite utilizzando valori uguali per tutti e 3 i canali:

```
p {color: rgb(120,120,120)}
```



### Valore esadecimale (HEX)

In HTML, un colore può essere specificato utilizzando un valore **esadecimale**:

## #rrggbb

Dove rr (rosso), gg (verde) e bb (blu) rappresentano valori esadecimali compresi tra 00 e ff (0123456789abcdef).

#ff0000 visualizza il rosso, poiché il rosso è impostato sul valore più alto (ff) e gli altri sono impostati sul valore più basso (00).

#0000ff visualizza il **blu**, poiché il blu è impostato sul valore più alto (ff) e gli altri sono impostati sul valore più basso (00)

Le sfumature di grigio sono spesso definite utilizzando valori uguali per tutte e 3 le fonti di luce:

```
p {color:#000000} /* nero */
p {color:#3c3c3c} /* grigio scuro */
p {color:#787878} /* grigio intermedio */
p {color:#b4b4b4} /* grigio chiaro */
p {color:#f0f0f0} /* grigio chiarissimo */
p {color:#ffffff} /* bianco */
```

https://developer.mozilla.org/en-US/docs/Web/CSS/color\_value



#### **Valore HSL**

In HTML, un colore può essere specificato usando hue (tonalità), saturation (saturazione) e lightness (luminosità) (HSL):

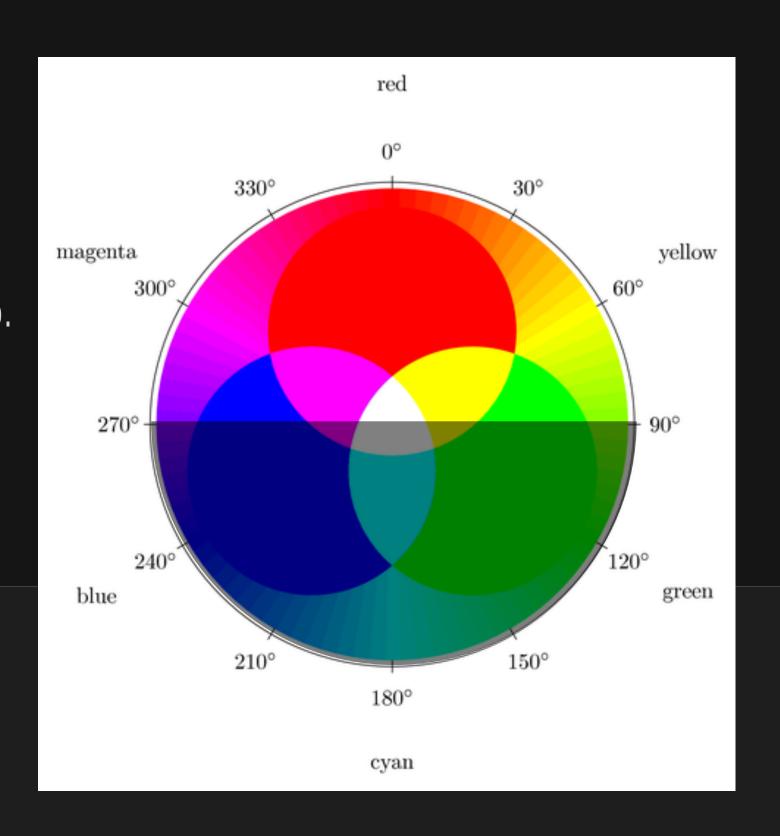
hsl(hue, saturation(%), lightness%)

La tonalità è rappresentata in gradi sulla *ruota dei colori* da 0 a 360. 0 è rosso, 120 è verde e 240 è blu.

La saturazione è un valore percentuale, lo 0% indica una sfumatura di grigio (assenza di colore) e il 100% indica il colore pieno.

Anche la luminosità è una percentuale, lo 0% è nero (assenza di luminosità), il 50% non è né chiaro né scuro, il 100% è bianco (massima luminosità)

```
p {color:hsl(0, 100%, 50%)} /* rosso */
p {color:hsl(240, 100%, 50%)} /* blu */
p {color:hsl(147, 50%, 47%)} /* verde */
p {color:hsl(39, 100%, 50%)} /* arancione */
```





## Saturazione

La saturazione può essere descritta come l'*intensità di un colore* (**purezza**).

Il 100% è di colore puro, senza sfumature di grigio

Il 50% è grigio al 50%, ma è ancora possibile vedere il colore.

0% è completamente grigio, non puoi più vedere il colore.

```
p {color:hsl(0, 100%, 50%)} /* rosso */
p {color:hsl(0, 20%, 50%)} /* rosso-grigio */
p {color:hsl(0, 0%, 50%)} /* grigio */
```



### Luminosità

La luminosità di un colore può essere descritta come *quanta luce vuoi dare al colore*, dove **0%** significa assenza di luce (**nero**), **50%** significa 50% luce (né buio né luce -**grigio**) **100%** significa luminosità massima (**bianco**).

Le *sfumature di grigio* vengono spesso definite impostando la *tonalità* e la *saturazione* su 0 e regolando la *luminosità* da 0% a 100% per ottenere tonalità più scure / chiare:

#### nero:

```
p {color:hsl(0, 100%, 0%)}
p {color:hsl(0, 0%, 0%)}
```

#### bianco:

```
p {color:hsl(0, 100%, 100%)}
p {color:hsl(0, 0%, 100%)}
```

## livelli di grigio:

```
p {color:hsl(0, 0%, 24%)}
p {color:hsl(0, 0%, 47%)}
p {color:hsl(0, 0%, 71%)}
p {color:hsl(0, 0%, 94%)}
```



#### **Valore RGBA**

I valori di colore RGBA sono un'estensione dei valori di colore RGB con un canale alfa, che specifica l'opacità di un colore.

Un valore di colore RGBA è specificato con:

```
rgba(rr,gg,bb,alfa)
```

Il parametro alfa è un numero compreso tra 0.0 (trasparente) e 1.0 (opaco):

colore trasparente (il colore è assente):

```
p {color:rgba(255, 99, 71, 0)}
```

il colore ha una trasparenza, il tono del colore si vede (rosso-arancione):

```
p {color:rgba(255, 99, 71, 0.6)}
```

il colore è pieno, opaco:

```
p {color:rgba(255, 99, 71, 1)}
```



#### Valore HSLA

I valori di colore HSLA sono un'estensione dei valori di colore HSL con un canale alfa, che specifica l'opacità di un colore.

Un valore di colore HSLA è specificato con:

hsla(hue, saturation(%), lightness%, alfa)

Il parametro alfa è un numero compreso tra 0.0 (trasparente) e 1.0 (opaco):

colore trasparente (il colore è assente):

```
p {color:hsla(9, 100%, 64%, 0)}
```

il colore ha una trasparenza, il tono del colore si vede (rosso-arancione):

```
p {color:hsla(9, 100%, 64%, 0.6)}
```

il colore è pieno, opaco:

```
p {color:hsla(9, 100%, 64%, 1)}
```



#### Valore esadecimale (HEX) con canale alfa

Anche con i valori di colore esadecimale si può aggiungere un canale alfa, che specifica l'opacità del colore.

Un valore di colore HEX con canale alfa è specificato con l'aggiunta al fondo di un ulteriore valore esadecimale (verificare compatibilità dei browser):

```
#00000066 /* nero con opacità al 40% */
```

Il parametro alfa è un valore compreso tra 00 (trasparente) e ff (opaco):

Rosso con opacità al 40%:

```
p {color:#ff000066}
```

Verde con opacità al 40%:

```
p {color:#00ff0066}
```

Blu con opacità al 40%:

```
p {color:#0000ff66}
```

Valore esadecimale e percentuale di opacità: <a href="https://gist.github.com/lopspower/03fb1cc0ac9f32ef38f4">https://gist.github.com/lopspower/03fb1cc0ac9f32ef38f4</a>