

Basi di HTML

Storia di HTML

- HTML nasce nel 1989 al Cern di Ginevra, insieme alla prima versione del World Wide Web.
- Nel 1994 nasce il W3C (World Wide Web Consortium) con lo scopo di sviluppare e standardizzare il linguaggio
- Nel 1995 viene pubblicata la versione 3.0
- Nel 1999 viene pubblicata la versione 4.0
- Nel 2014 viene pubblicata la versione 5.0
- La versione attuale è la 5.2 pubblicata il 14/12/2017
- E' circolata una bozza di versione 5.3 ma è stata definitivamente ritirata a gennaio 2021

Tag HTML

La struttura di un documento html è composta di elementi chiamati TAG, combinati in maniera opportuna per ottenere il layout e la struttura desiderata.

Tutti i tag condividono la stessa struttura, formata da un identificativo di apertura, un identificativo di chiusura e, opzionalmente, un contenuto e degli attributi (parametri aggiuntivi).

```
<tag [attributi]> Contenuto </tag>
```

oppure

```
<tag [attributi]/>
```

NB: HTML è un linguaggio case insensitive

Tag HTML

Il contenuto di un tag può essere una combinazione qualsiasi di due tipi:

- Un testo fisso
- Un altro tag

```
<tagA [attributi]>  
    <tagB>  
        HELLO WORLD  
    </tagB>  
</tagA>
```

Sebbene non obbligatorio indentare il contenuto rispetto al contenitore aumenta la leggibilità del codice.

Doctype

Un documento HTML deve iniziare con un tag speciale, DOCTYPE, per informare il browser che si tratta appunto di HTML, e deve essere salvato in un file con estensione .html (o .htm)

Per HTML 5 la sintassi è molto semplice:

```
<!DOCTYPE html>
```

In HTML 4 la sintassi è più complessa ma comunque fissa:

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

Nel proseguo del modulo utilizzeremo solamente la sintassi HTML 5.

Struttura di base di un documento HTML

Ogni pagina HTML ha una struttura fissa costituita, oltre che dal DOCTYPE anche dai tag HTML, HEAD e BODY:

```
<!DOCTYPE html>
```

```
<HTML>
```

```
    <HEAD>
```

```
    </HEAD>
```

```
    <BODY>
```

```
    </BODY>
```

```
</HTML>
```

- HTML è il tag contenitore che racchiude l'intera pagina.
- HEAD è il tag contenitore dentro al quale verranno inserite informazioni aggiuntive non direttamente visualizzate dal browser
- BODY è il tag contenitore dentro al quale verranno inseriti i tag che il browser visualizzerà sullo schermo

Il browser legge, interpreta e visualizza il codice HTML partendo dall'alto verso il basso.

Errori nella struttura di base di un documento HTML

Il browser HTML cerca di visualizzare anche pagine HTML contenenti errori (per esempio mancanza di tag di chiusura). La visualizzazione potrebbe pertanto essere differente da quanto previsto. In caso di errori "gravi", non risolvibili dal browser, non viene visualizzato nulla.

Il codice che segue sebbene errato viene correttamente visualizzato dal browser

```
<HTML>
```

```
    <BODY>
```

```
        HELLO WORLD!
```

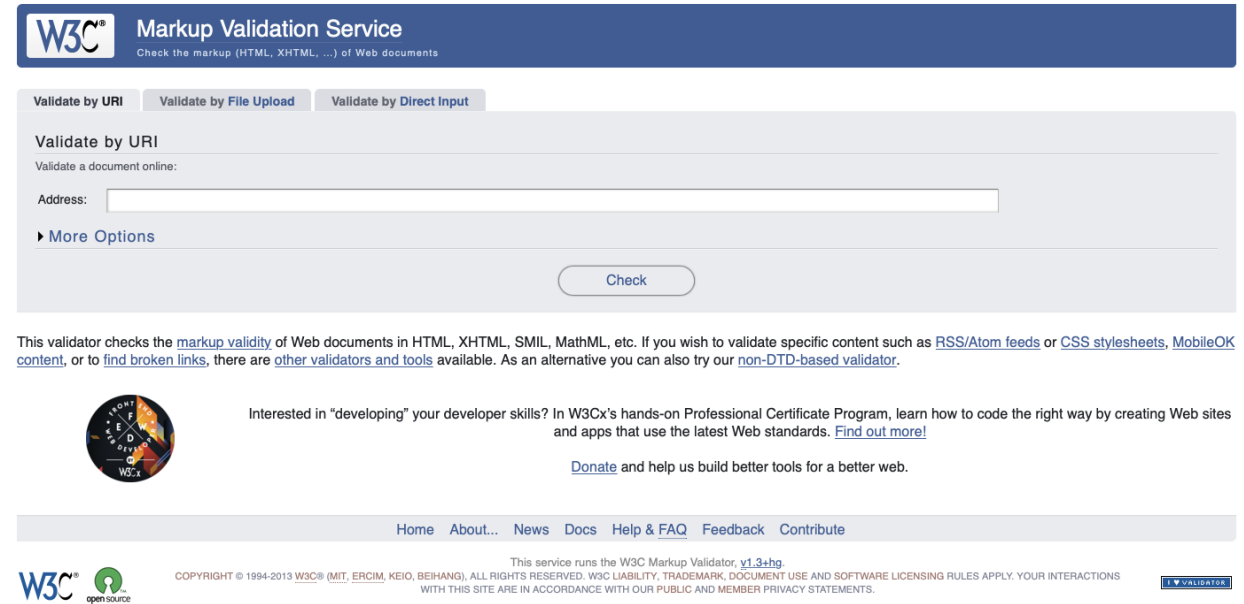
```
    </BODY>
```

ATTENZIONE: L'aiuto del browser non è un motivo per scrivere codice HTML non corretto.

Validazione di un documento HTML

All'url validator.w3.org è possibile validare un documento HTML, in tre modi:

1. Inserendo l'url della pagina html
2. Caricando il file .html
3. Digitando direttamente il codice HTML



The screenshot shows the W3C Markup Validation Service interface. At the top, there's a blue header with the W3C logo and the text "Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below this, there are three tabs: "Validate by URI", "Validate by File Upload", and "Validate by Direct Input". The "Validate by URI" tab is selected. Under this tab, there's a section titled "Validate by URI" with the instruction "Validate a document online:". Below this, there's a text input field labeled "Address:" and a "Check" button. There's also a link for "More Options". Below the input field, there's a paragraph of text explaining the validator's purpose and providing links to other resources. At the bottom, there's a footer with the W3C logo, a copyright notice, and a link to the validator's source code.

W3C[®] Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI Validate by File Upload Validate by Direct Input

Validate by URI


Validate a document online:

Address:

► More Options


Check

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).

 Interested in "developing" your developer skills? In W3C's hands-on Professional Certificate Program, learn how to code the right way by creating Web sites and apps that use the latest Web standards. [Find out more!](#)

[Donate](#) and help us build better tools for a better web.

[Home](#) [About...](#) [News](#) [Docs](#) [Help & FAQ](#) [Feedback](#) [Contribute](#)

W3C[®]  COPYRIGHT © 1994-2013 W3C[®] (MIT, ERCIM, KEIO, BEIHANG), ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.

This service runs the W3C Markup Validator, [v1.3+hg](#).

[W3C Validator](#)

Il validator evidenzia errori strutturali (mancanza di tag) e errori di configurazione (attributi mancanti). Segnala inoltre warning, ovvero situazioni che non costituiscono errore ma che è comunque opportuno sistemare

Formattazione del codice

HTML non impone un formato di scrittura dei tag.

I tag possono anche essere scritti tutti su una riga, a patto di rispettare l'ordine di apertura e di chiusura.

Per esempio, il codice che segue è valido:

```
<HTML><BODY>HELLO WORLD!</BODY></HTML>
```

Tuttavia, per una miglior lettura e manutenzione del codice si consiglia di utilizzare l'indentazione:

Il contenuto di ogni tag va scritto a capo, indentandolo verso destra di una tabulazione (tasto tab)

Tag di intestazione (HEAD)

All'interno del tag HEAD vengono inserite informazioni aggiuntive, non direttamente visualizzate sullo schermo ma comunque necessarie per la corretta gestione della pagina.

E' in questa sezione che si specificano, per esempio, i riferimenti al css e al javascript da utilizzare.

Vedremo nel dettaglio i tag utilizzati in HEAD nelle prossime lezioni:

- Metadati: `<META>`
- Css: `<STYLE>...</STYLE>`
- Script: `<SCRIPT>...</SCRIPT>`
- Titolo della pagina: `<TITLE>...</TITLE>`

Tag di intestazione (HEAD) - Metadati

I metadati permettono di esprimere informazioni sulla pagina web. Alcuni esempi:

- `<meta charset="UTF-8">` → Sistema di codifica dei caratteri
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">` → Dimensione della parte visibile e dello zoom
- `<meta keywords="HTML, CSS, Js">` → Parole chiave della pagina, ai fini dei motori di ricerca (oggi poco rilevanti)
- `<meta description="Qua il testo descrittivo della pagina">` → Descrizione estesa della pagina ai fini dei motori di ricerca

Tag del corpo (BODY)

All'interno del tag BODY vengono inseriti i tag necessari per specificare la struttura della pagina, ovvero quella che il browser presenterà sullo schermo.

I tag utilizzabili nel BODY possono essere suddivisi in due categorie:

- Tag block level: Vengono visualizzati sullo schermo andando a capo, prima e dopo il tag
- Tag inline: Vengono visualizzati senza andare a capo, all'interno del tag contenitore, nella posizione in cui sono stati inseriti

```
<tagBlockLevel>                                </tagBlockLevel>
```

```
<tagBlockLevel>...
```

```
...
```

```
... <tagInline>                                </tagInline>
```

```
...
```

```
...
```

```
</tagBlockLevel>
```

Tag per la gestione del testo

Tag Block level:

- Intestazioni: da <H1>...</H1> a <H6>...</H6>
- Paragrafi: <p>...</p>

Tag Inline

- Grassetto: ... o ...
- Codice: <code>...</code>
- Citazione: <cite>...</cite>
- Interruzione di linea:
 o
 (da usarsi solo dentro un tag <p>)
- Testo in corsivo: <i>...</i>
- Testo più piccolo: <small>...</small> o testo più grande: <big>...</big>
- Testo con enfasi: ...

Headings H1 ... H6

Sono disponibili 6 livelli di titoli, di dimensioni decrescenti da H1 (il più grande) a H6 (il più piccolo).

I titoli devono essere utilizzato in modo strutturato, in base alla loro importanza e non in base alle loro dimensioni. E' un aspetto fondamentale in quanto i motori di ricerca fanno affidamento su questo fatto.

Rispettare una serie di semplici regole:

- Massimo un titolo H1 in ogni pagina
- Non saltare dei livelli di titolo. Per esempio mettendo solo H1 e H3 saltando H2
- I titoli devono essere sempre posizionati dopo i titoli di dimensione maggiore. Non mettere prima H2 e poi H1

Testo e caratteri speciali

Non è possibile utilizzare nel testo caratteri che sono interpretati dal browser come facenti parti del linguaggio. Per esempio i caratteri `<` e `>`.

In alcune situazioni il browser potrebbe anche avere problemi nel rappresentare caratteri "speciali", anche se non facenti parti del linguaggio html. Per esempio le lettere accentate à è é...

HTML prevede una codifica dedicata per specificare i caratteri speciali. E' una codifica che inizia con `&`, prosegue con un codice identificativo del carattere e termina con `;`

Per esempio:

- `>` e `<` per `>` e `<`
- `à` e `è` per `à` e `è`
- ` ` per spazi che non interrompono le parole

L'elenco completo su <https://dev.w3.org/html5/html-author/charref>

Tag per creare elenchi (block level)

Elenchi numerati

```
<ol>  
  <li> Contenuto </li>  
  <li> Contenuto </li>  
</ol>
```

1. **Contenuto**
2. **Contenuto**

Elenchi puntati

```
<ul>  
  <li> Contenuto </li>  
  <li> Contenuto </li>  
</ul>
```

- **Contenuto**
- **Contenuto**

Tag per creare elenchi (block level)

Elenchi numerati e puntati possono anche essere innestati l'uno dentro l'altro

```
<ul>
  <li> Contenuto
    <ul>
      <li> Sotto Contenuto </li>
    </ul>
  </li>
</ul>
```

- **Contenuto**
 - **Sotto Contenuto**

Tag per creare tabelle (block level)

Le tabelle in HTML sono create con il tag `<table>` e sono composte da intestazione, corpo e piede, specificati rispettivamente con i tag `<thead>`, `<tbody>`, `<tfoot>`. Il corpo è obbligatorio mentre gli altri sono opzionali.

```
<table>  
    <thead> </thead>  
    <tbody> </tbody>  
    <tfoot> </tfoot>  
</table>
```

All'interno di ogni sezione vengono definite una o più righe, tramite tag `<tr>` e uno o più campi per ogni riga, tramite il tag `<td>` (`<th>` per l'intestazione)

Tag per creare tabelle (block level)

```
<table>

  <thead>  </thead>

  <tbody>

    <tr>

      <td>Contenuto cella</td>

      <td>Contenuto cella</td>

    </tr>

    <tr>

      <td>Contenuto cella</td>

      <td>Contenuto cella</td>

    </tr>

  </tbody>

  <tfoot>  </tfoot>

</table>
```

Tag per creare tabelle (block level)

`<table>`

`<thead>`

`<tr>`

`<th>...</th>`

`<th>...</th>`

`<th>...</th>`

`</tr>`

`</thead>`

`<tbody>`

`<tr>`

`<td>...</td>`

`<td>...</td>`

`<td>...</td>`

`</tr>`

`<tr>`

`<td>...</td>`

`<td>...</td>`

`<td>...</td>`

`</tr>`

`<tr>`

`<td>...</td>`

`<td>...</td>`

`<td>...</td>`

`</tr>`

`</tbody>`

`<tfoot>`

`<tr>`

`<td>...</td>`

`<td>...</td>`

`<td>...</td>`

`</tr>`

`</tfoot>`

`</table>`

Tag per creare tabelle (block level)

L'attributo `colspan` permette di far estendere una cella per più colonne. Per esempio `colspan="2"` renderà la cella larga due colonne.

`<table>`

`<thead>`

`<tr>`

`<th colspan="2">...</th>`

`<th>...</th>`

`</tr>`

`</thead>`

`<tbody>`

`<tr>`

`<td>...</td>`

`<td>...</td>`

`<td>...</td>`

`</tr>`

`<tr>`

`<td>...</td>`

`<td>...</td>`

`<td>...</td>`

`</tr>`

`<tr>`

`<td>...</td>`

`<td>...</td>`

`<td>...</td>`

`</tr>`

`</tbody>`

`<tfoot>`

`<tr>`

`<td>...</td>`

`<td>...</td>`

`<td>...</td>`

`</tr>`

`</tfoot>`

`</table>`

Tag per creare link (inline)

La possibilità di navigare verso altre pagine avviene tramite l'inserimento di un "link", ovvero di una porzione di testo (o di altri elementi, lo vedremo in seguito) cliccabile.

Il link è realizzato con il tag inline `<a>` e richiede di specificare, come attributo, l'indirizzo della pagina verso la quale navigare

```
<a href="percorso pagina">Contenuto</a>
```

Il contenuto del tag rappresenta ciò che sarà effettivamente cliccabile.

Tag per creare link (inline)

Il percorso associato ad un link è prevalentemente specificato nella forma di url, che nella sua forma più completa è espresso come:

<https://www.example.com/search?keywords=sql>

Dove:

- https = protocollo
- www.example.com = dominio (risolto in indirizzo ip + porta)
- search = percorso della risorsa all'interno del dominio
- keywords=sql = parametri aggiuntivi

Tag per creare link (inline)

E' possibile specificare percorsi relativi, che fanno riferimento alla posizione in cui si trova la pagina in cui c'è il tag img:

```
./nomePagina.html
```

```
../nomeCartella/nomePagina.html
```

Oppure percorsi assoluti:

```
/nomeCartella/nomeSottocartella/nomePagina.html
```

```
C:\nomeCartella\nomeSottocartella\nomePagina.html
```


Tag per creare link (inline)

Il link può anche puntare a contenuti differenti rispetto ad una pagina HTML, per esempio immagini o pdf:

```
<a href="../../../images/selfie.png">Foto Selfie</a>
```

E' anche possibile fare in modo che il click scarichi l'allegato, utilizzando l'attributo download

```
<a href="../../../images/selfie.png" download>Foto Selfie</a>
```

Download è un attributo booleano, il cui valore è true o false. In questo caso non è necessario scrivere download="true" ma è sufficiente la presenza dell'attributo. La sua assenza viene invece interpretata come download="false"

Tag per creare link (inline)

Il tag `<a>` prevede anche un ulteriore attributo, `target`, utilizzato per specificare la modalità di apertura del link:

```
<a href="percorso pagina" target="">Contenuto</a>
```

Il parametro `target` può valere:

- `_blank` per aprire il link in una nuova tab del browser
- `_self`, `_parent`, `_top` per aprire il link nei frame presenti nel documento

In caso di assenza dell'attributo, o qualora fosse vuoto, l'apertura del link avviene nella tab attuale.

Tag per inserire immagini (inline)

L'inserimento di una immagine avviene tramite il tag `` e richiede di specificare, obbligatoriamente, l'indirizzo dell'immagine da visualizzare. Opzionalmente può essere specificata anche l'altezza e la larghezza in pixel.

```

```

Il percorso dell'immagine è specificato con le stesse modalità viste per il percorso della pagine html.

L'altezza e la larghezza sono specificate in pixel. Omettendone una delle due l'immagine verrà "scalata" calcolando la dimensione mancante in modo proporzionale.

Omettendole entrambe l'immagine assumerà le sue dimensioni originali.

Tag per inserire immagini (inline)

Il tag `` prevede anche l'attributo `alt` tramite il quale specificare un testo associato all'immagine.

E' un attributo facoltativo ma caldamente raccomandato, in quanto:

- Rappresenta il testo che verrà visualizzato in caso di mancato caricamento dell'immagine
- Rappresenta il testo che viene letto dagli screen reader utilizzati dai non vedenti.

```

```

Tag per inserire video (inline)

L'inserimento di una video avviene tramite il tag `<video>`, utilizzando una modalità analoga a quella vista per le immagini. Richiede di specificare, obbligatoriamente, l'indirizzo del video da visualizzare. Opzionalmente può essere specificata anche l'altezza e la larghezza in pixel.

```
<video src="percorso video" height="y" width="x"/>
```

Il percorso dell'immagine è specificato con le stesse modalità viste per il percorso della pagine html e per le immagini.

L'altezza e la larghezza sono specificate in pixel. Omettendone una delle due il video verrà "scalata" calcolando la dimensione mancante in modo proporzionale.

Omettendole entrambe l'immagine assumerà le sue dimensioni originali.

Tag per inserire video (inline)

Il tag <video> contempla degli attributi booleani:

- controls: Per visualizzare i controlli di riproduzione
- autoplay: Per avviare automaticamente il video al caricamento della pagina
- loop: Per eseguire il video in modo continuo

NB: Non tutti i browser supportano completamente il tag video. La prassi più comune è quella di inserire un testo nel corpo del tag. Testo che verrà visualizzato quando il tag non è supportato.

```
<video src="percorso video" height="y" width="x">Il tuo browser non  
supporta i video </video>
```

Tag per linee di interruzioni (block)

Il tag `<hr>` permette di inserire una linea orizzontale utile per dividere i contenuti.

La linea è larga tanto quanto il contenitore (se non specificato diversamente) e il tag `hr` non prevede la necessità di un tag di chiusura.

`<hr>`

Tag contenitori

HTML prevede due specifici tag "contenitori" il cui unico compito è quello di raggruppare altri tag.

Non hanno una propria grafica e sono principalmente utilizzati per una la distribuzione degli spazi e per una migliore applicazione degli stili css.

- `...` tag inline per raggruppare elementi inline
- `<div>...</div>` tag block level per raggruppare elementi di qualsiasi tipo

Tag Semantici HTML5

Tag Semantici

Una delle più grandi novità di HTML è la semantica, ovvero la possibilità di rendere il codice HTML più comprensibile e in grado di descrivere meglio i contenuti. Nelle versioni precedenti i tag semantici erano molto pochi (form, table per esempio) e una grossa parte di formattazione era fatta con div, indipendentemente dal contenuto.

HTML5 introduce nuovi tag contenitore in grado di meglio specificare il contenuto che ospitano

<header>	Definisce l'intestazione di una pagina
<footer>	Definisce il piede di una pagina
<main>	Definisce il contenuto principale di una pagina (massimo 1 per pagina)
<section>	Definisce una sezione all'interno di una pagina
<article>	Definisce un blocco di codice auto contenuto, spesso con header e footer
<nav>	Definisce una sezione con link di navigazione (menù)
<aside>	Definisce una sezione rappresentata come sidebar laterale
<figure>	Definisce una sezione che contiene una immagine
<figcaption>	Definisce una sezione che contiene la didascalia di una figure

Tag Semantici

La maggior parte dei tag semantici non ha una propria grafica e si comporta esattamente come un div.

I vantaggi di utilizzare i tag semantici sono di altra natura:

- Possibilità di migliorare la comprensione della pagina da parte dei motori di ricerca
- Possibilità di migliorare la comprensione della pagina da parte degli screen reader
- Possibilità di utilizzare attributi specifici del singolo tag semantico
- Possibilità di assegnare stili css sui tag semantici con semplici selettori a livello di tag

Tag Semantici

Esistono anche alcuni tag semantici di tipo inline:

<code><mark></code>	Definisce una part di testo evidenziata
<code><time></code>	Definisce una data con orario