



# Introduzione a HTML

*Programmazione*

**HTML (HyperText Markup Language) è il linguaggio di base usato per creare la struttura e il contenuto di una pagina web.**

**Funziona attraverso tag che definiscono elementi come paragrafi, titoli, immagini e collegamenti.**

**Ogni pagina HTML parte con una struttura gerarchica che comprende il tag <html>, al cui interno troviamo <head> (informazioni e risorse della pagina) e <body> (contenuto visibile all'utente).**



- **Linguaggio di markup**

**HTML (HyperText Markup Language) non è un linguaggio di programmazione, ma un linguaggio di markup:**

**serve a descrivere la struttura e il significato dei contenuti di una pagina web (titoli, paragrafi, immagini, link), non a eseguire logica o calcoli.**

- **Basato su tag**

**HTML utilizza tag racchiusi tra < > per definire gli elementi della pagina.**

**Ogni tag rappresenta un contenuto o una funzione semantica (es. <p>, <h1>, <a>), spesso con tag di apertura e chiusura.**



- **Struttura gerarchica (DOM)**

**Gli elementi HTML sono organizzati in una struttura ad albero, detta DOM (Document Object Model).**

**Questa gerarchia permette al browser e a JavaScript di navigare, modificare e manipolare la pagina.**

- **Semantica**

**HTML moderno (HTML5) introduce tag semantici come <header>, <nav>, <article>, <section>, <footer>.**

**Questi migliorano:**

- **accessibilità**
- **SEO**
- **leggibilità del codice**

**perché descrivono il significato del contenuto, non solo l'aspetto.**



- **Indipendente dalla presentazione**  
**HTML si occupa solo del contenuto e della struttura, non dello stile grafico.**  
**La presentazione è demandata ai CSS, mentre il comportamento dinamico è gestito da JavaScript (separazione dei ruoli).**
- **Supporto agli ipertesti**  
**HTML nasce per il concetto di ipertesto:**  
**grazie al tag <a> permette di collegare documenti, risorse e sezioni, creando la rete di pagine del Web.**



- **Standardizzato**

**HTML è uno standard definito dal W3C e dal WHATWG,  
garantendo:**

- **compatibilità tra browser**
- **evoluzione controllata**
- **retro-compatibilità con versioni precedenti**

- **Interpretato dal browser**

**Il codice HTML viene interpretato direttamente dal browser,  
senza compilazione.**

**Il browser legge il file, costruisce il DOM e renderizza la pagina.**



- **<body> – Contenuto della pagina**

**Il tag <body> contiene tutto ciò che viene visualizzato nel browser: testi, immagini, link, liste, tavole, ecc.**

**È il “contenitore principale” del contenuto visibile e rappresenta il punto di partenza per CSS e JavaScript quando si lavora sull’interfaccia utente.**

1. **<body>**
2. **<h1>Il mio sito</h1>**
3. **<p>Benvenuto nella mia pagina web.</p>**
4. **</body>**



- **<h1> (e <h1>–<h6>)** – Titoli

I tag **<h1>** fino a **<h6>** rappresentano i titoli gerarchici della pagina.

**<h1>** è il titolo principale (di solito uno solo per pagina), mentre gli altri indicano sottosezioni.

Sono fondamentali per accessibilità, SEO e struttura logica del contenuto

1. **<h1>Corso HTML</h1>**
2. **<h2>Introduzione</h2>**
3. **<h3>Cos'è HTML</h3>**



- **< p >** – Paragrafo di testo

**Il tag `<p>` serve a rappresentare un blocco di testo.**

**Ogni paragrafo è separato logicamente dagli altri e il browser applica automaticamente spaziatura verticale.**

**È l'elemento più usato per contenuti testuali descrittivi.**

1. **< p >**
2. **HTML è il linguaggio del Web per strutturare i contenuti.**
3. **</ p >**



- **<a> – Collegamento (link)**

**Il tag <a> permette di creare collegamenti ipertestuali verso altre pagine, sezioni della stessa pagina o risorse esterne.**

**È uno degli elementi chiave che rende il Web una rete interconnessa di documenti.**

1. **<a href="https://www.wikipedia.org">**
2. **Visita Wikipedia**
3. **</a>**



- **<img> – Immagini**

**Il tag <img> serve a visualizzare immagini all'interno della pagina web.**

**È un tag auto-chiudente (non ha chiusura) e utilizza attributi per definire il percorso dell'immagine e il testo alternativo, fondamentale per accessibilità e SEO.**

1. ****

2.



- **<div> – Contenitore generico**

**Il tag <div> è un contenitore generico di tipo blocco usato per raggruppare elementi.**

**Non ha significato semantico, ma è estremamente usato per layout, styling CSS e logica JavaScript.**

1. **<div>**
2. **<h2>Sezione notizie</h2>**
3. **<p>Ultimi aggiornamenti dal sito.</p>**
4. **</div>**
- 5.



- **<span> – Contenitore inline**

**Il tag <span> è un contenitore inline, usato per evidenziare o modificare porzioni di testo senza interrompere il flusso del contenuto.**

**È spesso utilizzato insieme a CSS o JavaScript.**

1. **<p>**
2. **Questo testo è <span>importante</span>.**
3. **</p>**
- 4.



- **<ul> / <li>** – Liste non ordinate

**Il tag <ul> crea una lista non ordinata, mentre <li> rappresenta ciascun elemento della lista.**

**È molto usato per menu di navigazione, elenchi puntati e strutture ripetitive.**

1. **<ul>**
2. **<li>HTML</li>**
3. **<li>CSS</li>**
4. **<li>JavaScript</li>**
5. **</ul>**
- 6.



```
1.<!DOCTYPE html>
2.<html lang="it">
3.<head>
4. <meta charset="UTF-8">
5. <title>Pagina di esempio HTML</title>
6.</head>
7.
8.<body>
9.
10.<h1>Benvenuto nel mio sito</h1>
11.
12.<p>
13. Questa è una pagina di esempio scritta in
14. <span>HTML</span>, il linguaggio base del Web.
15.</p>
16.
17.<div>
18.<h2>Argomenti trattati</h2>
19.
20.<ul>
21. <li>Struttura HTML</li>
22. <li>Tag principali</li>
23. <li>Collegamenti</li>
24.</ul>
25.</div>
26.
27.<div>
28.<h2>Logo del sito</h2>
29.
30.</div>
31.
32.<p>
33. Per approfondire visita
34. <a href="https://developer.mozilla.org">
35. la documentazione ufficiale
36.</a>.
37.</p>
38.
39.</body>
40.</html>
```



## Spiegazione dell'esempio

**Questa pagina rappresenta la struttura tipica di un documento HTML reale.**

- **<!DOCTYPE html> indica al browser che il documento segue lo standard HTML5.**
- **<html> è il contenitore principale di tutta la pagina.**
- **<head> contiene metadati non visibili, come il titolo della pagina e la codifica dei caratteri.**
- **<body> racchiude tutto il contenuto visibile della pagina.**

**All'interno del <body>:**

- **<h1> definisce il titolo principale della pagina.**
- **<p> contiene il testo descrittivo.**
- **<span> evidenzia una porzione di testo inline, senza creare un nuovo blocco.**
- **<div> raggruppa sezioni logiche della pagina.**
- **<h2> introduce sottosezioni.**
- **<ul> e <li> creano una lista non ordinata, tipica di menu o elenchi.**
- **<img> mostra un'immagine, con alt per accessibilità.**
- **<a> crea un collegamento ipertestuale a una risorsa esterna.**

## Consigli pratici per principianti in HTML

- Parti sempre dalla struttura base

Ogni pagina HTML dovrebbe avere doctype, html, head e body.  
Scrivere subito una struttura corretta aiuta il browser a interpretare la pagina senza comportamenti imprevisti.

- Usa i tag per il loro significato, non per l'aspetto

Non scegliere un `<h1>` perché “è grande” o un `<p>` perché “va a capo”.

Usa i tag in base al ruolo semantico del contenuto: lo stile si gestisce con CSS.

- Un solo `<h1>` per pagina

È una buona pratica usare un solo titolo principale e poi strutturare il resto con `<h2>`, `<h3>`, ecc.

Questo migliora accessibilità e SEO.



## Consigli pratici per principianti in HTML

- Indenta il codice e mantienilo leggibile

Un HTML ben indentato è più facile da leggere, capire e mantenere.

La leggibilità è una competenza fondamentale, non un dettaglio estetico.

- Usa sempre alt nelle immagini

L'attributo alt:

- aiuta chi usa screen reader
- appare se l'immagine non viene caricata
- migliora l'indicizzazione

È obbligatorio dal punto di vista dell'accessibilità.



## Consigli pratici per principianti in HTML

- **Evita <div> ovunque quando esistono alternative**  
**<div> è utile, ma non sempre la scelta migliore.**  
**Quando possibile, usa tag più specifici (<p>, <ul>, <section>, ecc.) per rendere il codice più chiaro.**
- **Salva spesso e testa nel browser**  
**HTML è immediato: apri il file nel browser dopo ogni modifica.**  
**Imparare a osservare cosa cambia è il modo migliore per capire come funziona.**



## Consigli pratici per principianti in HTML

- Impara a leggere l'HTML degli altri  
**Usa Ispeziona elemento nel browser per:  
capire come sono fatte le pagine reali  
imparare nuove strutture  
vedere come HTML, CSS e JS lavorano insieme**

- HTML, CSS e JavaScript hanno ruoli diversi  
**Ricorda sempre:**

- HTML → struttura
- CSS → stile
- JavaScript → comportamento

**Non mescolare i ruoli: è la base di un buon codice web.**

