



# **Struttura dei tag HTML**

## *Sintassi del linguaggio*

Sviluppo di siti web – UD02

*prof. Giovanni Borga*

# I «mattoncini» della pagina HTML: i tag

Come abbiamo detto in precedenza, all'interno di ogni pagina è presente una **serie di marcatori**.

**I marcatori di cui si compone una pagina HTML prendono il nome di TAG**

Ai tag viene affidata la formattazione e disposizione dei contenuti.

Hanno differenti nomi a seconda della loro funzione.

*<TAG attributi>contenuto</TAG>*

## **Sintassi di base:**

I TAG vanno inseriti con le seguenti regole sintattiche:

- L'apertura del tag è una stringa tra **parentesi uncinate** (<TAG>)
- La chiusura del tag viene indicata con una "/" (è il simbolo comunemente detto "**slash**". </TAG>).
- Il primo elemento dopo l'apertura della prima parentesi è detto «**nome**» del tag e può essere seguito da «attributi»
- Nel tag di chiusura non può esserci nulla tranne lo slash e il nome del tag
- Ciò che sta tra il tag di apertura e il tag di chiusura è detto «**contenuto**» del tag.

## Esempi di tag e struttura degli attributi

Ecco un esempio, con una sintassi che serve a disporre un testo giustificato a destra:

```
<p align="right">testo</p>
```

Dall'esempio è evidente che la **struttura di un attributo** è: attributo="valore" ; la gamma di attributi è definito e dipende dallo specifico tag. I valori degli attributi vanno indicati tra doppi apici.

In definitiva la **struttura complessiva di un tag** è la seguente:

```
<TAG attributo_1="valore1" attributo_2="valore2">contenuto</TAG>
```

Alcuni particolari tag non hanno contenuto - perché ad esempio indicano solo la posizione di alcuni elementi (ad es. il tag delle immagini), conseguentemente questi tag non hanno la chiusura. Vengono detti appunto **tag vuoti**.

La loro forma in origine era dunque: <TAG attributi>, tuttavia con l'adozione delle specifiche XHTML si rende necessario lo slash prima della seconda parentesi; ecco un esempio di tag per inserimento di un'immagine:

```
<IMG widht="20" height="20" src="logo.gif" alt="logo aziendale" />
```

## Disposizione dei tag

Una pagina HTML può essere vista come un «flusso» sequenziale di tag dove però occorre rispettare la gerarchia di apertura e chiusura dei tag senza intersecarli.

Questa sequenza è corretta:

```
<TAG1>contenuto 1<TAG2>contenuto 2</TAG2></TAG1>
```

Quest'altra no:

```
<TAG1>contenuto 1<TAG2></TAG1>contenuto 2</TAG2>
```

Nella sequenza corretta è altrettanto corretto affermare che il TAG2 è un contenuto del TAG1. Tecnicamente il TAG2 è annidato nel TAG1. Possiamo scrivere il codice anche andando a capo per renderlo più leggibile:

```
<TAG1>  
contenuto 1  
<TAG2>contenuto 2</TAG2>  
</TAG1>
```

Si consideri infine la proprietà di base per cui **i contenuti nel flusso HTML tutte le interruzioni di riga e tutti gli spazi vengono renderizzati come un unico spazio.**

Per inserire più spazi o più righe esistono apposite codifiche. Vedremo più avanti che alcuni tag costituiscono interruzione del flusso e quindi mandano il contenuto a nuova riga mentre altri no.

## Annidamento e indentazione

La logica di tipo strettamente gerarchico dell'HTML fa sì che sia quantomeno conveniente «indentare» le righe di codice in modo da rendere più leggibile il listato e limitare l'inserimento di errori come ad esempio l'omissione del tag di chiusura o l'errato annidamento.

Ad esempio il seguente frammento di codice:

```
<TAG1 attributi>contenuto 1<TAG2>contenuto 2</TAG2></TAG1>
```

E' più leggibile se scritto così:

```
<TAG1 attributi>  
contenuto 1  
  <TAG2>  
    contenuto 2  
  </TAG2>  
</TAG1>
```

Ma ancor meglio così:

```
<TAG1 attributi>  
  contenuto 1  
    <TAG2>  
      contenuto 2  
    </TAG2>  
  </TAG1>
```

In questo molti editor aiutano lo sviluppatore con l'indentazione automatica e la colorazione del codice. Per indentare si utilizza generalmente il carattere di tabulazione (TAB).

# Commenti

Un strategia adottata universalmente da tutti gli sviluppatori per **rendere più comprensibile un listato** (esiste una sintassi per i commenti per ogni linguaggio), è quella di inserire dei «commenti» nei punti significativi.

Si tratta di indicazioni significative per lo sviluppatore, ma sostanzialmente «invisibili» al browser; i commenti quindi non hanno alcuna influenza nel rendering della pagina.

La sintassi dei commenti HTML è la seguente:

```
<!-- questo è un commento HTML -->
```

Il commento può anche comprendere più righe di codice:

```
<!--  
questo è un commento  
che comprende più righe di testo  
<b>testo in grassetto</b>  
-->
```

*Si noti che tutti i tag compresi nei commenti non vengono renderizzati.*

# Commenti

```
<html>
```

```
<!-- qui inizia la pagina -->
```

```
<head>
```

```
<!-- qui l'intestazione -->
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
<title>
```

```
<!-- qui il titolo -->
```

```
HTML.it
```

```
</title>
```

```
</head>
```

```
<body>
```

```
<!-- qui il contenuto visibile della pagina-->
```

```
Lorem ipsum factum ...
```

```
</body>
```

```
</html>
```

## Maiuscolo / minuscolo

L'**HTML** è «**case insensitive**», cioè indipendente dal maiuscolo e minuscolo. E' quindi del tutto indifferente scrivere tag e attributi in maiuscolo o in minuscolo.

**<P ALIGN="RIGHT">**

e

**<p align="right">**

vengono renderizzati allo stesso modo dal browser.

*NB: naturalmente i contenuti sono invece «case sensitive» in quanto sono ciò che viene visualizzato materialmente nella pagina.*

Fino a qualche tempo fa, per aumentare la leggibilità del codice, era consuetudine scrivere in maiuscolo il nome del tag (es: <P>) e in minuscolo gli attributi (es: align="right"). Quindi: <P align="right">. Tuttavia oggi, per analogia con l'XHTML (che è figlio dell'XML e dell'HTML ed è «case sensitive») **è consigliabile scrivere tutto in minuscolo**, fermo restando che se per errore dovessimo scrivere dei tag in maiuscolo ciò non costituisce un errore.