

# 1.0 HTML

Sta per Hyper-Text-MarkUp-Language. Un testo non deve essere come un libro, l'idea di un Hyper-Testo è quella di non avere sequenzialità, ma poter saltare da una parte all'altra in modo fluido. E' un formato aperto, ognuno può aprirlo e leggerlo, definito da uno standard IEEE.

## 1.1 Struttura a Marcatori

Di base abbiamo una struttura ASCII con dei marcatori che definiscono delle sezioni che avranno delle funzioni o particolarità in base al marcatore utilizzato. I marcatori possono avere una struttura annidata.

- **Elemento:** una pagina è costituita da tanti elementi, ogni elemento è un blocco composto da marcatore di apertura e relativo marcatore di chiusura.

E.g: Tutto il documento deve stare all'interno dei marcatori HTML, a sua volta tutto il campo informativo deve stare all'interno di HEAD, e il titolo all'interno dell'elemento TITLE.

Vediamo alcuni elementi:

- **Paragrafo (p):** un documento è principalmente composto da paragrafi, dove al loro interno inseriamo un contenuto, che può essere testuale.
- **Capo (br):** Sta per "break", elemento vuoto che non ha contenuto, manda semplicemente a capo. Esempio più semplice di elemento vuoto.
- **Caratteri scomodi (uso delle Entity):** Non potendo direttamente utilizzare i caratteri che costituiscono sintassi HTML, esistono diverse entity (codici) che corrispondono ad un determinato carattere e che quindi ne permettono l'uso nel documento renderizzato. queste stesse entity possono essere scritte anche come `&#CODICE_ASCII_CARATTERE`. tra questi caratteri sono presenti lettere maiuscole accentate, apici (><), ecc.
- **Ancora (a):** Elemento HTML che significa "Ancora", permette di inserire collegamenti ipertestuali utilizzando href (hyper-reference). si possono utilizzare anche delle ancore (tag "#")
- **Form:** Elemento che permette di contenere altri pulsanti o label, utilizzato per inviare i dati al server. fa uso dei campi di `<input type=".." name=".." value=".."/>`, dove vengono specificati tipo, nome e valore del campo di input. il campo `type` può assumere valore `text`, `number`, `email`, `password`, `Submit` (pulsante di invio), `reset`, `checkbox`, `radio` (opzioni con pallini). Possiamo introdurre un elemento di tipo **option**, che consente di specificare liste di selezione, contiene tanti elementi `option` quante sono le opzioni che vogliamo aggiungere

( `<option value=".."> text </option>` ); Oppure elementi **textarea** che permettono la definizione di aree di testo libero su più righe

`<textarea name=".." rows=".." cols=".."> Testo iniziale </textarea>` (da usare con POST).

## 1.2 Percorsi Relativi

I percorsi relativi semplificano i riferimenti alle risorse locali:

- `../` sale di un livello
- `A/` scende nella cartella A
- `./` si riferisce alla cartella corrente

## 1.3 Elementi Strutturali

### Elemento `<div>`

Crea una divisione o sezione della pagina senza effetti particolari. Può contenere qualsiasi elemento HTML.

### Elemento `<nav>`

Definisce la barra di navigazione, dedicata ai menu del sito con link ad altre sezioni. Utile per applicare uno stile specifico.

### Elemento `<footer>`

Definisce il piè di pagina, contenente informazioni generali del sito o del proprietario. rimane sempre nella parte bassa della pagina.

## Attributi Importanti

### Attributo `id`

Identifica univocamente un elemento, utile per JavaScript e per l'elemento `<label>`, possiamo dare un id al campo label che lo (in questo caso) associa al campo input.

```
<label for="cg">Cognome</label>
<input type="text" name="cognome" id="cg"/>
```

## Attributi style e class

- `style` specifica al volo lo stile diretto usando CSS.
- `class` specifica la classe di stile da applicare, le classi si trovano nel file CSS.

## CSS - Cascading Style-Sheets

I fogli di stile permettono uno stile uniforme modificabile centralmente. "Cascading" indica che lo stile si propaga dagli elementi esterni a quelli interni (tutto ciò che è contenuto in un elemento eredita il suo stile).

## Colori

```
color: red; /* Testo */
background-color: black; /* Sfondo */
```

I colori possono essere specificati come:

- Nome in inglese: `red` , `black`
- RGB: `rgb(255,0,0)`
- Esadecimale: `#FF0000`

## Font

```
font-size: 15px; /* Dimensione */
font-family: Arial; /* Tipo */
```

## Implementazione CSS

### Elemento <style>

Inserito nel <head> HTML, contiene le specifiche degli stili (si comporta come un foglio CSS esterno, ma è integrato nell'HTML).

```
<head>
  <style type="text/css">
    body {background-color: white;}
    p {color: red;}
  </style>
</head>
```

## Classi di Stile

Proprietà non associate a elementi specifici, definite con punto e applicate con l'attributo `class`.

```
.bordato { border: 1px solid black; }
```

```
<div class="bordato">Contenuto</div>
```

## Selettori CSS

Il selettore può essere:

- Nome elemento: `p { color: blue; }`
- Classe: `.bordato { border: 1px solid black; }`
- ID: `#myname { background-color: yellow; }`

## Pseudo-classi CSS

Hanno formato `selettore:pseudo-classe{...}`.

Indicano stati degli elementi:

- `:hover` - mouse sopra l'elemento
- `:active` - elemento attivato
- `:focus` - elemento con focus
- `:link` e `:visited` - stati dei link

In questo modo si può specificare lo stile quando un elemento si trova in un determinato stato.

## Fogli di Stile Esterni

I file `.css` possono essere esterni, quindi usati da più pagine HTML.

```
<link rel="stylesheet" type="text/css" href="stili.css"/>
```

rel viene utilizzato per indicare all'HTML il motivo della relazione.

## Aggiungere Nuovi Font

```
@font-face {  
    font-family: Oswald;  
    src: url(./fonts/Oswald/Oswald.ttf);  
}
```

## Layout

### Dimensioni

```
width: 100px;    /* Pixel */  
width: 20%;      /* Percentuale */  
max-width: 500px; /* Massima */
```

### Padding e Margine

Margine interno tra contenuto e bordo esterno dell'oggetto (top, right, left, bottom).

```
/* Versione compatta di top, right, bottom, left */  
padding: 10px 20px 15px 25px;  
margin: 10px 20px 15px 25px;
```

### Posizionamento

- static - flusso normale (default)
- relative - spostato rispetto alla posizione normale
- absolute - posizionato rispetto al contenitore
- fixed - posizionato rispetto alla finestra

Per specificare lo spostamento:

```
left: 10px; /* Orizzontale */  
top: 10px; /* Verticale */
```