

HTML

Prospettiva

- Scrivere codice well formed, ovvero ordinato
- Seguire la semantica degli elementi
- Separare la presentazione dal contenuto
- Scrivere codice accessibile
- Scrivere codice portabile

– Elementi tutti in minuscolo:



• Sì: `<html>`



• No: `<HTML>`

– Elementi con contenuto aperti e chiusi, ed elementi vuoti (senza contenuto) scritti correttamente:



• Sì: `<html>` con `</html>` e `
`



• No: `<html>` senza `</html>` e `
`

**Well
Formed**

Separare la presentazione dal contenuto

- **Separiamo bene:**

- il contenuto (che è specificato dal markup HTML5)
- dagli aspetti presentazionali, come la resa grafica (specificato dal foglio di stile, con CSS3).



- **Esempio:**



– Sì: Uso un elemento `` per dare caratteristiche più evidenti ad una porzione di testo che non è titolo.



– No: uso un elemento `<h4>` per dare caratteristiche più evidenti a una porzione di testo che non è titolo.

- HTML5 ha nuovi elementi:
 - che consentono di **strutturare** il contenuto attribuendo alle parti di documento una **semantica**. La pagina deve quindi essere divisa in parti a secondo del ruolo che queste hanno e questi ruoli (header, footer, ecc.) devono essere specificate.
 - che definiscono bene elementi di controllo, menù e strumenti di navigazione
- Noi li useremo più possibile:
 -  – Sì: Uso un elemento `<article>` per indicare il contenuto specifico di un post di un blog.
 -  – No: il contenuto specifico di un post di un blog lo inserisco in un paragrafo `<p>` generico, eventualmente distinto graficamente via CSS.

Rispettare la semantica degli elementi

Accessibilità: è la capacità dei sistemi informatici di erogare servizi e di fornire informazioni fruibili, senza discriminazioni, anche da parte di coloro che a causa di disabilità necessitano di tecnologie assistive o configurazioni particolari.

Elementi HTML5

<!DOCTYPE>: è la prima riga di codice di un documento html, serve per indicare al browser il tipo di documento. Per specificare l'utilizzo di HTML5 basta aggiungere html dopo !DOCTYPE, come nell'esempio accanto.

<html> </html>: tag che delimitano l'inizio e la fine di un documento html. Al loro interno sono contenuti tutti gli elementi della pagina. E' **buona norma** aggiungere l'attributo **lang** al tag html per indicare il linguaggio utilizzato nella pagina.

<head> </head>: intestazione del documento. Riporta informazioni sulla pagina o sulle relazioni con altri documenti.

Il tag `<title></title>` racchiude al suo interno il titolo della pagina, verrà mostrato nel tab della finestra.

Il tag `<meta charset="UTF-8"/>` specifica il charset in uso.

<body> </body>: corpo del documento che racchiude il contenuto della pagina.

Struttura documento

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Nome</title>
```

```
<meta
```

```
  charset="UTF-8"/>
```

```
</head>
```

```
<body>
```

```
  Content of the  
  document ...
```

```
</body>
```

```
</html>
```

Metadati: consentono di descrivere il documento specificandone caratteristiche, comportamento, presentazione, relazioni con altri documenti.

Bisogna includerli nell'**head del documento**.

<title>: rappresenta il nome o il titolo del documento.

<base>: deve essere univoco all'interno del documento, se ci sono più elementi <base> viene preso in considerazione solo il primo. Lo scopo principale è di indicare il **path base** del documento che servirà per risolvere gli URL relativi sia in termini di **href** che di **target**.

Esempio:

```
<head>
<base href="http://www.w3schools.com/images/"
target="_blank"/>
</head>
<body>
```

<link>: utilizzato per creare **relazioni** con altri documenti o risorse. Serve principalmente per creare la relazione con il documento **CSS**.

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

<style>: permette di includere stili all'interno del documento.

Il rendering del documento sarà il risultato dei **<link>**, **<style>** e degli eventuali (**meglio di no**) stili inline utilizzati.

```
<style>
  h1 {color:red;}
  p {color:blue;}
</style>
```

<meta>: vengono usati per aggiungere altri **metadati** al documento. Il tipo di metadati è specificato dall'attributo **name**.

Sono spesso usati dai motori di ricerca.

```
<head>
  <meta charset="UTF-8"/>
  <meta name="description"
        content="Free Web tutorials"/>
  <meta name="keywords"
        content="HTML,CSS,XML,JavaScript"/>
  <meta name="author" content="Hege Refsnes"/>
</head>
```

Sectioning: gli elementi di questa categoria hanno **funzione strutturale**, ovvero dividono la pagina in parti con ruoli diversi a seconda dell'elemento utilizzato.

- `<article>`
- `<aside>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<nav>`
- `<section>`



<section>: definisce una **sezione** del documento. Una home page potrebbe essere suddivisa in 3 sezioni:

- una per l'introduzione
- una per il contenuto
- una per le informazioni sui contatti

<article>: definisce informazioni indipendenti e auto-contenute. Un **articolo** dovrebbe essere un elemento con un suo senso proprio da leggere in modo indipendente dal resto della pagina.

Esempio: post su un blog, forum o un social network.

<header>: definisce l'**intestazione** del documento o di una sua sezione. Può essere usato come contenitore di un contenuto di tipo introduttivo. Possono essere presenti più header nella stessa pagina.

<footer>: definisce il **footer** di un documento o di una sua sezione. Il footer contiene informazioni sull'autore del documento, le informazioni sul copyright, link ai termini d'uso, informazioni sui contatti etc... Possono essere presenti più footer in una pagina

<nav>: definisce un insieme di **link di navigazione**(menù o toolbar).

```
<!DOCTYPE html>
<html>
...
  <body>
    <nav>
      <a href="/html/">HTML</a> |
      <a href="/css/">CSS</a> |
      <a href="/js/">JavaScript</a> |
      <a href="/jquery/">jQuery</a>
    </nav>
  </body>
</html>
```

<aside>: definisce un contenuto **a latere** rispetto quelli principali. Può essere utilizzato per contenere i contenuti di una barra laterale, oppure per contenuti collaterali che non si posizionano necessariamente a lato.

Heading: introducono i **titoli** delle diverse sezioni del documento.

- gli heading vanno da **<h1>** a **<h6>** a seconda della loro rilevanza
- **h1** è quello con maggiore rank e deve rappresentare il titolo principale della sezione
- i titoli di rank inferiore devono intestare sottosezioni

Phrasing: rientra il contenuto che rappresenta il **testo** del documento.

<p>: inserisce un paragrafo **testuale**. Va utilizzato quando non esiste un elemento più specifico o semanticamente più idoneo per descrivere quel testo.

```
<section>
  <p> Primo paragrafo di testo.</p>
  <p> Secondo paragrafo di testo.</p>
  <p> Terzo paragrafo di testo.</p>
</section>
```

**
: rappresenta un **interruzione di linea.

- è un elemento vuoto, va scritto con /
- deve essere utilizzato solo per interruzioni che sono effettivamente parte del contenuto, quindi **non per ottenere effetti grafici**

```
<p>Nel mezzo del cammin di nostra vita<br/>
mi ritrovai per una selva oscura<br/>
ché la diritta via era smarrita.</p>
```

<div>: non ha un significato proprio, ha lo scopo di **rappresentare gli elementi in esso annidati e specificare per loro gli attributi** -----> *class, lang e title*

****: simile al div ma opera a livello di testo (elemento inline).

<main>: raggruppa gli elementi di struttura (come **<section>** o **<article>**) che rappresentano il contenuto principale del documento. Ogni documento deve avere solo un **<main>**.

Ruolo del testo: elementi che attribuiscono **ruoli** al testo.

- **<i>:** **testo in voce alternativa**, termini tecnici, frasi idiomatiche, pensieri, testo in altra lingua.
- **:** **stress emphasis**, un testo o una frase che si pronuncia in modo differente dal resto.
- **:** **strong importance**, testo importante.
- **:** **offset text**, testo più visibile.
- **<small>:** **note a margine**.

<sub> e **<sup>**: corrispondono ad **apice** e **pedice**.

```
<p>This text contains  
  <sub>subscript</sub> text.</p>  
<p>This text contains  
  <sup>superscript</sup> text.</p>
```

This text contains subscript text.

This text contains superscript text.

Citazioni: ci sono vari elementi per inserire citazioni.

- **<blockquote>:** per parti di contenuto che vengono **citare** da una sorgente esterna
- **<q>:** simile a **blockquote** ma agisce su un breve testo. Si può usare l'attributo *cite*
- **<cite>:** per citare i riferimenti ad un **lavoro creativo**.

<ins> e ****: servono a marcare le **modifiche al documento**. In particolare **<ins>** si usa quando si vuole aggiungere del contenuto, mentre **** si usa quando si vuole cancellare del contenuto dal documento.

E' consigliato l'utilizzo degli attributi **cite** e **datetime**.

Embedded: ha lo scopo di importare risorse o contenuto dentro il documento.

Alcuni elementi embedded prevedono un contenuto **fallback** che viene usato quando la risorsa non può essere utilizzata.

<figcaption>: definisce la **didascalia** di una immagine.

<figure>: raggruppa l'immagine e la sua didascalia.

```
<figure>  
    
  <figcaption>Fig.1 - Pulpit Rock,  
    Norvegia.</figcaption>  
</figure>
```



****: definisce le **immagini inline**. Gli attributi obbligatori sono:

- **src**: specifica l'URL del **file contenente l'immagine**
- **alt**: testo alternativo, viene visualizzato in caso il browser non riesca a mostrare l'immagine e in caso di immagini disabilitate

<video>: definisce un modo standard per **includere un video** in una pagina web. Attributi:

- **controls**: aggiunge il controllo per il video: play, pausa, volume, ecc
- **autoplay**: per far partire il video automaticamente
- **width**: definisce la larghezza del video
- il testo contenuto in **<video></video>** viene mostrato nel caso in cui il browser non supporti l'elemento

<audio>: definisce un modo standard per **includere un audio** in una pagina web. Attributi:

- **controls**: aggiunge il controllo per il video: play, pausa, volume, ecc
- il testo contenuto in **<audio></audio>** viene mostrato nel caso in cui il browser non supporti l'elemento

<object>: definisce un oggetto incluso in un documento HTML. Supportato da tutti i browser, usato per includere audio, video, animazioni e plug-in nelle pagine web.

<embed>: è un elemento vuoto, definisce un oggetto incluso in un documento HTML come **<object>**.

<canvas>: consente di disegnare sulla pagina, fornisce le API necessarie per la generazione e il rendering dinamico di grafico, diagrammi, immagini e animazioni.

Definisce un'area rettangolare in cui disegnare direttamente immagini bidimensionali e **modificarle in relazione a eventi, tramite funzioni Javascript**.

I **commenti** in html si inseriscono tra **<!-- e -->** .

Link

<a>: consente di inserire àncore nel documento, ovvero **punti di partenza di un link**.

- la destinazione si specifica con un URI tramite l'utilizzo dell'attributo **href**
- il punto di arrivo viene specificato utilizzando l'attributo **id** su qualsiasi elemento


```

<nav>
  <ul>
    <li> <a href="/">Home</a> </li>
    <li> <a href="/news">News</a> </li>
    <li> <a href="http://www.google.it">Google</a> </li>
    <li> <a href="#articolo">Articolo</a> </li>
  </ul>
</nav>

<article id="articolo">
  <p> testo dell'articolo .... </p>
</article>

```

partenza

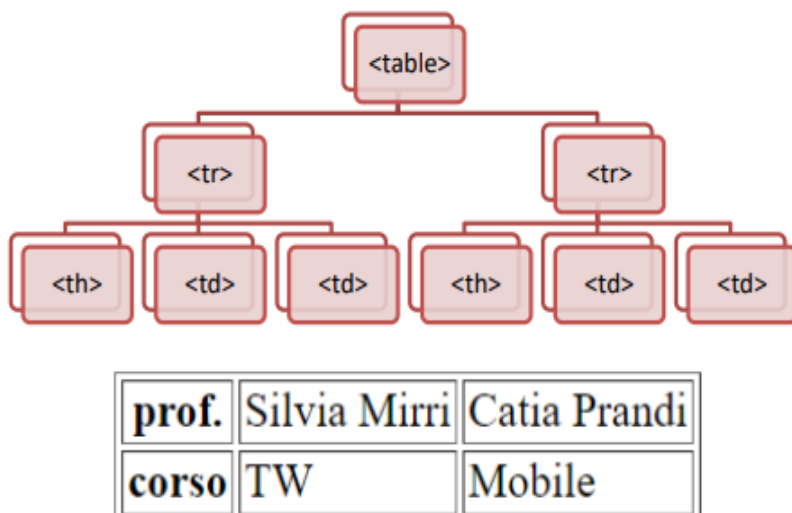
arrivo

Flow

<table>: elemento che serve per realizzare le **tabelle** in HTML5.

- sono organizzate per righe, utilizzando l'elemento **<tr>**, table row
- ogni riga è poi divisa in celle
 - celle normali, in questo caso si utilizza l'elemento **<td>**, table data
 - celle di intestazione, sono realizzate con l'elemento **<th>**, table header

Struttura delle tabelle



```

<table>
  <tr>
    <th>prof.</th>
    <td>Silvia Mirri</td>
    <td>Catia Prandi</td>
  </tr>
  <tr>
    <th>corso</th>
    <td>TW</td>
    <td>Mobile</td>
  </tr>
</table>

```


<caption>: con questo elemento è possibile dare un titolo alla tabella

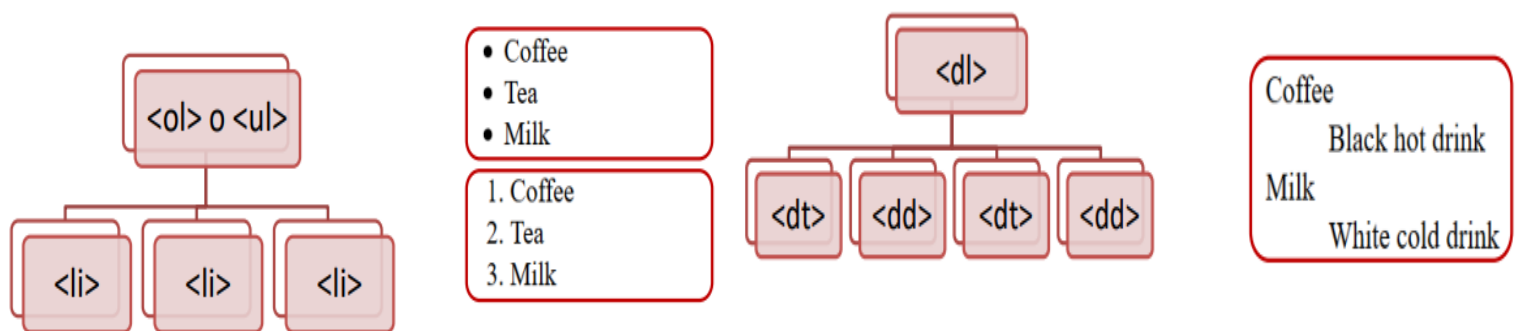
Una cella di tipo **<td>** o **<th>** può occupare più righe o più colonne utilizzando rispettivamente l'attributo **colspan** e **rowspan**.

Liste in HTML5:

- Liste non ordinate, **** (unordered list)
- Liste ordinate, **** (ordered list)
- Liste di definizioni, **<dl></dl>** (definition list)

Nelle liste ordinate e non ordinate ogni item è definito da ****.

Struttura delle liste



- Nelle liste **ordinate** possono essere specificati: **start** -> il valore iniziale della numerazione, **type** -> il tipo di numerazione utilizzata, **reversed** -> la numerazione è inversa
- Le liste definite da **<dl></dl>** sono liste **associative**, pensate per correlare un **concetto** con uno o più altri **termini** che sono specificati da:
 - **<dt>** termini descrittivi
 - **<dd>** uno o più definizioni,

Le liste possono essere annidate l'una nell'altra.

Interactive: categoria associata ai form, meccanismo di input (e in parte di output) di valori di elaborare.

<form>: parte della pagina web che contiene controlli di input, come per esempio campi di testo, bottoni e checkbox.

- può essere processata lato client (via **javascript**)
- può essere processata lato server
- entrambe le cose

Gli **attributi** di form sono:

- **action** -> specifica l'URL dell'applicazione server-side che riceverà i dati
- **method** -> specifica il metodo **HTTP** che deve essere usato per i dati
 - **GET**
 - **POST**
- **name** -> specifica il nome del controllo, viene definito mediante l'attributo

- **required** -> è un attributo booleano dei controlli che indica che è obbligatorio inserire un valore per quel campo
- **autofocus** -> mette il focus su questo controllo
- **autocomplete** -> è un attributo booleano di input che consente di attivare l'autocompletamento

<label>: ogni controllo deve avere una **<label>** che descriva il controllo all'utente:

– La presenza di **<label>** è fondamentale per l'accessibilità, se non si vuole che sia visibile, va resa invisibile ma senza rimuoverla.

– La label può essere associata al controllo:

- Annidando il controllo nella **<label>**.

```
<form>
  <p><label>Customer name:
  <input...../></label></p>
</form>
```

- Mettendo un **id** nel controllo e relazionandolo alla **<label>** mediante l'attributo **for**.

```
<form>
  <p><label for="CN">Customer name: </label>
  <input .... id="CN" /></p>
</form>
```



<fieldset>: raggruppa più controlli che hanno semantica comune:

```
<form>
  <fieldset> <legend>dati personali:</legend>
    <label>Nome:
    <input type="text" name="nome"/></label><br/>
    <label>Email:
    <input type="email" name="mail"/> </label ><br/>
    <label>Data di nascita:
    <input type="date" name="date"/></label >
  </fieldset><br/>
  <input type="submit"
    value="Invia"/>
</form>
```

<input>: consente di inserire nella pagina molti tipi diversi di controllo, il **tipo** è specificato mediante l'attributo **type** che può assumere diversi valori:

- **reset**: genera un bottone per ripristinare i valori di default della form
- **submit**: genera un bottone per sottomettere la form
- **button**: genera un bottone a cui ci può essere associato uno script Javascript con l'attributo **onclick**
- **value**: in tutti e 3 è possibile utilizzare questo attributo per modificare il testo sul bottone

```
</form>
  <input type="button" value="Bottone" onclick="..." />
  <input type="reset" value="Reimposta" />
  <input type="submit" value="Invia" />
</form>
```

A screenshot of three buttons arranged horizontally. The first button is labeled 'Bottone', the second 'Reimposta', and the third 'Invia'. They are all light gray with black text. A red rectangular box highlights the entire group of buttons.

- **image**: ha la stessa funzione di un bottone
 - L'immagine è usata per la resa grafica.
 - Va inserito l'elemento alt per rendere accessibile il bottone.
 - Invia le coordinate del click (x e y) al server.

- **Esempio:**

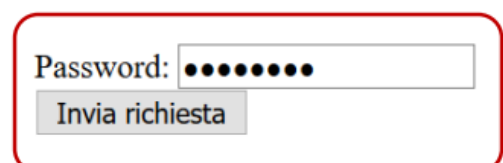
```
<form action="demo_form.asp">
  <label>nome: <input type="text" name="fname" />
</label><br/>
  <input type="image" src="img_submit.gif"
    alt="Submit" width="48" height="48" />
</form>
```

A screenshot of a form. It has a label 'nome:' followed by a text input field. Below the input field is a green circular button with a white right-pointing arrow. A red rectangular box highlights the entire form area.

- **hidden**: sono inseriti nella form per creare un'associazione variabile valore invisibile all'utente. Il controllo **<input>** con **type="hidden"** non viene visualizzato.
- **password**: sono inseriti nella form per creare un'associazione variabile valore invisibile all'utente.
- Quando l'utente riempie il controllo, la visualizzazione della password viene oscurata.

- **Esempio:**

```
<form action="...">
  <label>Password:
    <input type="password" name="pwd"
      maxlength="8" /></label>
  <br/><br/>
  <input type="submit" value="Invia richiesta" />
</form>
```

A screenshot of a form. It has a label 'Password:' followed by a password input field filled with dots. Below the input field is a button labeled 'Invia richiesta'. A red rectangular box highlights the entire form area.

- **radio**: realizzano un radio button. Le diverse opzioni possibili sono definite inserendo più **radio** con lo stesso **name**.

```
<form action="demo_form.asp">
  <label>
    <input type="radio" name="gender" value="male"/>
    Male</label> <br/>
  <label>
    <input type="radio" name="gender" value="female"/>
    Female</label> <br/>
  <label>
    <input type="radio" name="gender" value="other"/>
    Other</label> <br/>

  <input type="submit" value="Submit"/>
</form>
```

- **checkbox**: realizzano una checkbox. Le diverse opzioni possibili sono definite inserendo più **checkbox** con lo stesso **name**.

```
<form action="demo_form.asp">
  <label><input type="checkbox" name="vehicle1"
    value="Bike"/>I have a bike</label><br/>
  <label><input type="checkbox" name="vehicle2"
    value="Car"/> I have a car</label><br/>
  <label><input type="checkbox" name="vehicle3"
    value="Boat"/> I have a boat</label><br/>
  <input type="submit" value="Submit"/>
</form>
```

- **text**: sono utilizzati come controlli per input di tipo testo (generico).

```
<form action="demo_form.asp"/>
  <label>First name:
    <input type="text" name="fname"/>
  </label><br/>
  <label>Last name:
    <input type="text" name="lname"/>
  </label><br/>
  <input type="submit" value="Invia richiesta"/>
</form>
```

<textarea>: definisce un controllo di input per testi multilinea. Gli attributi **cols** e **rows** specificano la dimensione della textarea.

<select> e <option>: <select> è usato per realizzare un menù a tendina. Le diverse opzioni sono introdotte attraverso l'elemento <option>.

- Esempio:

```
<form>
  <label> scegli un insegnamento<br/>
  <select name="insegnamento">
    <option value="SO">Sistemi Operativi</option>
    <option value="TW">Tecnologie Web</option>
    <option value="SM">Sistemi Multimediali</option>
  </select></label>
</form>
```

- Attributi per **<select>** sono:
 - **multiple**, booleano per effettuare selezioni multiple
 - **size**, definisce le opzioni da mostrare all'utente
- Attributi per **<option>** sono:
 - **selected**, booleano è true se quella **<option>** è il valore di default, altrimenti la prima option è il default. Se si vuole un default nullo, si deve aggiungere una **<option>** vuota.
 - **value**, indica il valore quando diverso dal testo della opzione, per esempio un codificato o abbreviato.

<optgroup>: è usato per raggruppare le **option** in una **select**.

<datalist>: l'elemento **<option>** assieme all'elemento **<datalist>** può essere usato anche per predisporre dei suggerimenti per i valori di un campo testuale.

<output>: rappresenta il risultato di un calcolo o azione dell'utente.