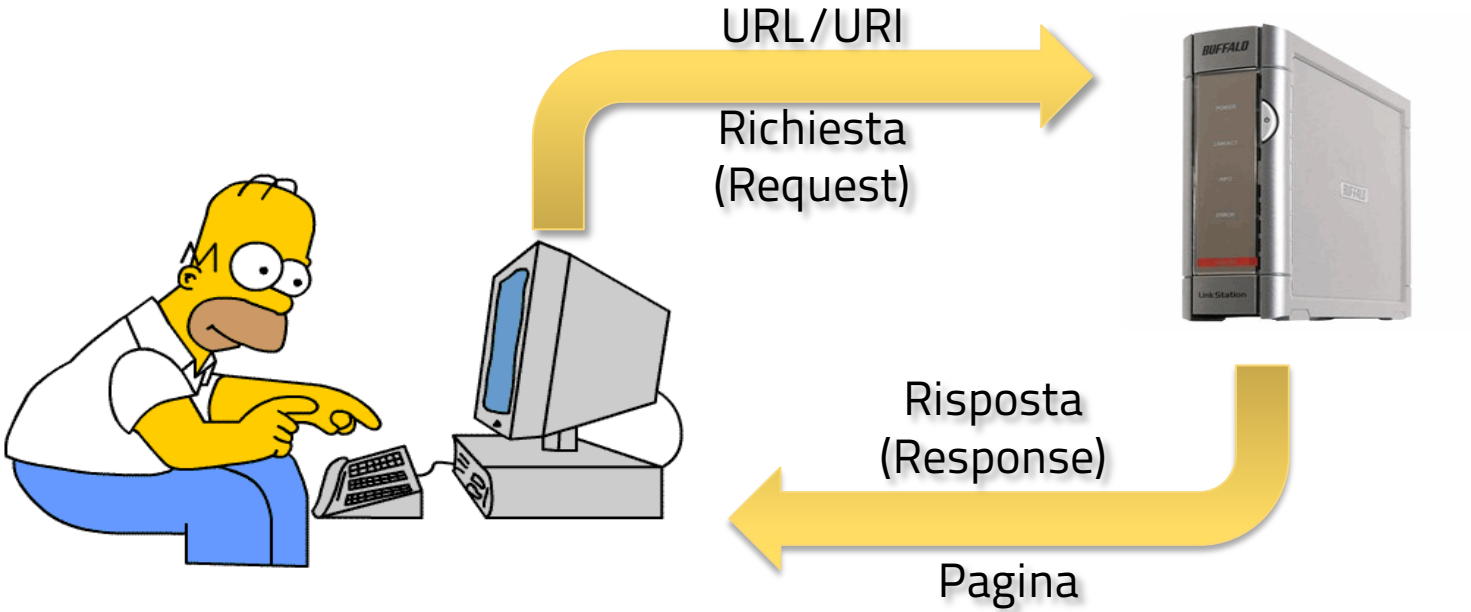
COME FUNZIONA INTERNET

ARCHITETTURA CLIENT SERVER



URL E URI

Uno **Uniform Resource Identifier** (URI, acronimo più generico rispetto ad "**URL**") è una stringa che identifica univocamente una risorsa generica che può essere un indirizzo Web, un documento, un'immagine, un file, un servizio, un indirizzo di posta elettronica, ecc. L'URL è un URI che indica una risorsa internet.

Un **Uniform Resource Locator** o **URL** è una sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa in Internet, come un documento o un'immagine.

STRUTTURA DELL'URI

- La tipica struttura di un **URI** è:

  protocollo://indirizzo_risorsa

- In un URL **indirizzo_risorsa** può contenere informazioni aggiuntive:

  nomehost [:porta][/percorso][?querystring]

ESEMPIO DI URL

ARCHITETTURA CLIENT-SERVER

- Server
  - Programma *in ascolto* su una porta (punto di accesso)
  - Quando arriva una richiesta da un client, il server analizza questa richiesta (eventualmente con l'aiuto di altri programmi), elabora una risposta (anche in questo caso, eventualmente con l'aiuto di altri programmi) e la invia al client.
  - Un server, generalmente, può servire più client contemporaneamente
- Client
  - Un client è un programma che si connette ad un server, fa una richiesta ed aspetta una risposta

WEB SERVER

- Un **Web Server (server che fornisce** servizi sul Web) è sostanzialmente un **HTTP Server (server che comunica** mediante il protocollo HTTP) e **gestisce 2 flussi di informazioni:**
  - **le richieste in arrivo dai client** (HTTP request)
  - **le risposte del server, inviate** ai client (HTTP response)

BROWSER

- Un Web browser è un **HTTP client, cioè un** programma, dotato di interfaccia grafica, che:
  - **interagisce con un HTTP server , richiedendone** i servizi (per es. pagine Web)
  - riceve i dati dal server e li ricompone
  - **visualizza le pagine Web (ipertesti),** mostrandone il contenuto e interpretando correttamente i linguaggi che vengono utilizzati per descriverne i contenuti

LE APPLICAZIONI WEB

- Il passaggio di semplici documenti HTML tra il server e il client non permette lo sviluppo di applicazioni web complesse che coinvolgano una fase di elaborazione oltre che di passaggio di dati.
- Per questo motivo sono state sviluppate tecnologie che permettano una maggiore interazione dell'utente con il server web e una capacità di elaborazione sia del server che del client web.

PAGINE STATICHE E DINAMICHE

- Quando ci connettiamo ad una risorsa in rete, identificata da un URL:
  - Nel caso più semplice l'indirizzo di una pagina (generalmente scritta in HTML) il cui contenuto è fisso (STATICA);
  - In altri casi, l'URL può contenere l'indirizzo di una pagina "dinamica" (per esempio scritta in ASP, PHP, o JSP) il cui contenuto viene generato (selezionato, composto) al momento della richiesta;

CONCLUDENDO

1. Le informazioni che ci arrivano da **Internet** ci arrivano in formato **pagina web**

CONCLUDENDO

2. Ci arrivano codificate:
   - Una specifica codifica per i caratteri
   - Specifiche Codifiche per le varie risorse multimediali

CONCLUDENDO

3. Come nella comunicazione tradizionale le informazioni sono organizzate semanticamente in **indici**, **titoli**, **sottotitoli**, **paragrafi**, **figure**, ecc., in modo che siano intellegibili.

CONCLUDENDO

4. Questa formattazione semantica viene fatta utilizzando **HTML**, un linguaggio che contiene istruzioni che servono, appunto, ad organizzare i contenuti di una pagina web.

CONCLUDENDO

5. La pagina web per essere fruita deve essere presentata in maniera piacevole e leggibile su vari tipi di schermo e di dispositivo. A tal fine è stato costruite un altro linguaggio che ha il compito di definire le regole di visualizzazione della varie parti di una pagina: **CSS**

CONCLUDENDO

6. Infine le pagine web sono interattive:
   - Interattività ipertestuale immediata (HTML)
   - Interattività lato server (Invio richieste e ricevo risposte)
   - Interattività lato client: utilizzo di programmi più o meno complessi scritti in **javascript** per rendere più semplice e produttiva l'attività di navigazione dell'utente.

HTML

HTML

- **HTML** è l'acronimo di **H**yper**T**ext **M**arkup **L**anguage ("Linguaggio a marcatori per gli Ipertesti").
- Non è un linguaggio di programmazione non ha, cioè, meccanismi che consentono di prendere delle decisioni ("in questa situazione fai questo, in quest'altra fai quest'altro"), e non è in grado di compiere delle iterazioni ("ripeti questa cosa, finché non succede questo"), né ha altri costrutti propri della programmazione.
- Si tratta invece di un linguaggio di contrassegno (o 'di marcatura'), che permette di articolare gli elementi di una pagina in blocchi le cui caratteristiche vengono definite attraverso degli appositi marcatori, detti "**tag**".

A COSA SERVE

- Comunica al browser di quali risorse ha bisogno per comporre la pagina web
- Definisce e articola il contenuto della pagina:
  - Struttura semantica della pagina
  - Elementi di formattazione
  - Componenti multimediali
  - Componenti funzionali

COME LO FA: I TAG

- I **tag** vanno inseriti tra parentesi uncinate: **<TAG>**
- La chiusura del tag viene indicata con una barra: **</TAG>**
- Il contenuto che il tag modifica va inserito tra l'apertura e la chiusura del tag medesimo:

  Questa ****parola**** è in grassetto.
- che nel rendering verrà reso:

  Questa **parola** è in grassetto.
- Alcuni tag non hanno (o possono non avere) contenuto (**empty tag**) .
  Ad esempio l'interruzione di linea la indico così:

  **
**

GLI ELEMENTI

- Ogni tag definisce un **elemento** (element) del documento HTML
- Esistono vari tipi di elementi:
  - Elementi visibili che definiscono componenti semantiche del testo (titoli, intestazioni, piè di pagina, paragrafi, ecc.)
  - Elementi visibili che definiscono componenti che consentono l'interazione con l'utente (link, campi che consentono l'immissione di testo, bottoni, ecc.)
  - Elementi visibili che servono a formattare parti di testo (grassetto, corsivo, ecc.)
  - Elementi visibili e non visibili che consentono l'inserimento di risorse multimediali (immagini, audio, video, ecc.)
  - Elementi non visibili che servono a caricare risorse funzionali necessarie alla pagina web (script, fogli di stile)
  - Elementi non visibili che racchiudono codice eseguibile o istruzioni di stile.

GLI ATTRIBUTI

- Le caratteristiche di un tag vengono determinate dagli attributi del tag. Ogni tag ha per i suoi attributi dei valori predefiniti che io posso modificare

```
<TAG att                                                    </TAG>
```

- Alcuni attr ecc.), altri sono speci

```
<IMG wi                                                    " />
```

- Una caratt ossono essere ann altri.
- È quindi opportuno usare l'indentazione. Grazie ad essa il codice HTML risulta più leggibile.

```
<TAG1 attributo = "valore">
        contenuto 1
        <TAG2>
                contenuto 2
        </TAG2>
</TAG1>
```

ENTITY

- Per rappresentare i caratteri non coficabili con lo standard ASCII si è introdotta una codifica particola detta **entity**.
- Un entity è così composta:

&nnnn;

- Quando il parser HTML incontra una parola che inizia con **&** legge i successivi caratteri fino ad in contrare "**;**" e tenta di interpretare il tutto come un carattere secondo la tabella di codici definita dal W3C.

COMMENTI

- Un strategia importante, per rendere il nostro codice più leggibile è quella di inserire dei "commenti" nei punti più significativi:
- Un commento è un'indicazione significativa per il webmaster, ma invisibile al browser. Inserendo i commenti in punti specifici del documento ci permette di mantenere l'orientamento anche in file molto complessi e lunghi.
- La sintassi è la seguente:

```
<!-- questo è un commento -->
```

IL W3C

- L'organizzazione che si occupa di standardizzare la sintassi del linguaggio HTML (il W3C: World Wide Web Consortium).
- Ha rilasciato diverse versioni di questo linguaggio (HTML 2.0, HTML 3.2, HTML 4.0….);
- Allo stato attuale abbiamo a che fare con 3 versioni:
  - HTML 4.01 (24/12/1999)
  - XHTML 1.0 (01/08/2002)
  - HTML 5 (Bozza di lavoro: 19/10/2010)

HTML 5

STRUTTURA DELLA PAGINA

```
<!DOCTYPE html>
<html lang="it">
<head>
  <script src="mioscripr.js"></script>
  <link rel="stylesheet" href="styles.css">
  <meta charset="utf-8">
  <meta name="google" value="notranslate">
</head>
    <body>
        <h1>Benvenuto!</h1>
        <p>Questo &egrave; il mondo di XHMTL!</p>
    </body>
</html>
```

PROLOGO

```
<!DOCTYPE html >
```

- Con HTML 5 il prologo si riduce alla semplice dichiarazione del tipo di documento come **HTML**

ELEMENTO RADICE (ROOT)

```
<html lang="it" >
</html>
```

- <HTML> è obbligatorio
- L'elemento <html> può assumere questi attributi:
  - **dir** Determina la direzione del testo
  - **lang** Specifica il linguaggio di base dell'elemento quando è interpretato come HTML

HEAD

```
<head>
  <script src="mioscripr.js"></script>
  <link rel="stylesheet" href="styles.css">
  <meta charset="utf-8">
  <meta name="google" value="notranslate">
</head>
```

- La sezione <head> contiene informazioni che non vengono direttamente visualizzate nella pagina:
  - <link> Contiene informazioni su documenti esterni collegati. Usato soprattutto per i CSS.
  - <meta> Specifica informazioni di vario tipo sul documento.
  - <noscript> Usato per visualizzazioni alternative nei browser che non supportano gli script.
  - <object> Racchiude un oggetto.
  - <script> Contiene script di programmazione o carica uno script di programmazione esterno .
  - <style> Definisce le regole di formattazione per il documento corrente
  - <title> Specifica il titolo del documento che compare nella barra del titolo del browser

BODY

```
<body>
    <h1>Benvenuto!</h1>
    <p>Questo &egrave; il mondo di XHMTL!</p>
</body>
```

- Il corpo del documento è la sezione in cui si sviluppa il contenuto. È racchiusa, come in HTML, tra i tag <body>...</body>.
- Gli elementi che possono comparire all'interno del corpo sono in genere suddivisi in due categorie:
  - **elementi blocco** ed gli elementi blocco sono quelli che definiscono la struttura del documento. Possono contenere altri elementi blocco, elementi inline o testo. Quando sono inseriti danno origine ad una nuova riga nel flusso del documento.
  - **elementi inline**: quando sono inseriti non danno origine a una nuova riga e possono contenere solo dati (essenzialmente testo) o altri elementi inline.

ELEMENTI SEMANTICI

ELEMENTI SEMANTICI

- Gli elementi semantici sono quelli che ci aiutano a dare una struttura logica ad un documento
- Normalmente sono elementi di tipo blocco
- Nella maggior parte dei casi i browser non prevendono alcuna formattazione di default e vengono resi sullo schermo come testo normale

TITOLI (HEADINGS)

- Gli heading (titoli) sono elementi blocco contrassegna dai tag h1-h6 e possono essere utilizzati per strutturare gerarchicamente un documento.
- Per default i browser rendono i titoli con corpi crescenti da h6 (più piccolo del testo normale) a h1.

DIV

- L'elemento è un generico elemento blocco che racchiude una porzione di documento
- Non è prevista alcuna formattazione di default

```
<div>
      Testo o qualsiasi altro elemento
</div>
```

P

- L'elemento <p> è un elemento blocco che racchiude una paragrafo
- Normalmente i browser rendono l'elemento p assegnandogli un margine inferiore

```
<p>
      Testo o qualsiasi altro elemento
</p>
```

UL

- L'elemento <ul> rappresenta una lista non ordinata.
- Gli elementi della lista sono rappresentati da elementi
- Per default una lista <ul> viene resa come lista puntata

```
<ul>
  <li><a href="#">link</a></li>
  <li><a href="#">link</a></li>
</ul>
```

OL

- L'elemento <ol> rappresenta una lista ordinata.
- Gli elementi della lista sono rappresentati da elementi
- Per default una lista <ol> viene resa come lista numerata

```
<ul>
  <li><a href="#">link</a></li>
  <li><a href="#">link</a></li>
</ul>
```

HEADER

- L'elemento <header> ha come scopo quello di racchiudere una porzione di documento che avrà ruolo di testata nella pagina o in una sua parte.
- Non è prevista lacuna formattazione di default

```
<header>
      <!-- il logo ed eventuale intestazione al sito -->
</header>
```

ARTICLE

- L'elemento <article> racchiude una porzione di codice semanticamente indipendente dal resto della pagina

```
<article>
  <header>
     <h1>Titolo del sito</h1>
     <h2>Sottotitolo</h2>
  </header>
</article>
```

TIME

- L'elemento <time> rappresenta una data di pubblicazione

```
<time datetime="2010-04-21" pubdate>21 Aprile 2009</time>
```

NAV

- L'elemento <nav> specifica una porzione di documento destinato alla navigazione per esempio la lista di un menu.

```
<nav>
  <ul>
    <li><a href="#">link</a></li>
    <li><a href="#">link</a></li>
  </ul>
</nav>
```

ASIDE

- L'elemento <aside> viene utilizzato per definire una porzione di codice correlata al contenuto ma separata come per esempio una sidebar.

FOOTER

- L'elemento <footer> definisce una porzione di documento che rappresenta il piede della pagina o di una parte specifica del documento.
- Non è prevista lacuna formattazione di default

```
<footer>&copy;2012 Mia Azienda Tutti i diritti riservati</footer>
```

FIGURE

- Il tag <figure> specifica un contenuto indipendente dal testo, come illustrazioni, diagrammi, foto, esempi di codice, ecc
- Mentre il contenuto dell'elemento <figure> è in relazione al flusso principale, la sua posizione è indipendente dal flusso principale, e se rimosso non dovrebbe influenzare il flusso del documento.

```
<figure>
  <img src="img_pulpit.jpg" alt="The Pulpit Rock"
width="304" height="228">
</figure>
```

ELEMENTI INLINE

SPAN

- L'elemento è una generico elemento *inline* che definisce una porzione di testo.
- Non è prevista lacuna formattazione di default.

```
<p>Questa <span style="font-wieight:bold">parola</span> è
                    in grassetto.</p>
```

FORMATTAZIONE

- Esistono una serie di elementi *inline* che definiscono la formattazione o la funzione di una porzione di testo.
  - Testo corsivo
  - Testo importante
  - <dfn> Definizione
  - <code> Codice
  - <samp> Esempio di output
  - <kbd> Imput da tastiera
  - <var> Variabile
- Non è prevista lacuna formattazione di default.

INTERATTIVITÀ

A

- Il tag <a> definisce un collegamento ipertestuale, che viene utilizzato per collegare una risorsa internet.
- L'attributo più importante dell'elemento <a> è l'attributo **href**, che indica la destinazione del collegamento.
- Per impostazione predefinita, i collegamenti verranno visualizzati come segue in tutti i browser:
  - Un collegamento non visitato è sottolineato e blu
  - Un link visitati è sottolineato e viola
  - Un collegamento attivo è sottolineato e rosso

```
<a href="http://www.accademiadiurbino.it">Visita
    l'accademia di Belle Arti di Urbino!</a>
```

FORM

- Il tag <form> viene utilizzato per creare un modulo HTML per l'input dell'utente.
- L'elemento <form> può contenere uno o più dei seguenti elementi del modulo:
  - <input>
  - <textarea>
  - <button>
  - <select>
  - <option>
  - <optgroup>
  - <fieldset>
  - <label>

```
<form action="demo_form.asp" method="get">
  Nome: <input type="text" name="nome"><br>
  Cognome: <input type="text" name="cognome"><br>
  <input type="submit" value="Invia">
</form>
```

INPUT

- Il tag <input> specifica un campo di input in cui l'utente può inserire i dati.
- Gli elementi <input> vengono utilizzati all'interno di un elemento <form>
- La funzione di imput cambia a secondo del valore definito dall'attributo type:
  - button, checkbox, color, date, datetime, datetime-local, email, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week

```
<form action="demo_form.asp" method="get">
  Nome: <input type="text" name="nome"><br>
  Cognome: <input type="text" name="cognome"><br>
  <input type="submit" value="Invia">
</form>
```

TEXTAREA

- Il tag <textarea> definisce un controllo di input di testo multilinea.
- Un'area di testo può contenere un numero illimitato di caratteri.
- Per default il testo viene reso in un font a larghezza fissa (di solito Courier).

```
<textarea rows="4" cols="50">
        Inserisci un testo.
</textarea>
```

BUTTON

- Il tag <button> definisce un pulsante cliccabile.
- Contrariamente che per l'elemento input all'interno di un elemento <button> Posso inserire qualsiasi tipo di contenuto.
- Browser diversi utilizzano diversi tipi di default per l'elemento <button>.

```
<button type="button">Cliccami!</button>
```

SELECT

- L'elemento <select> viene utilizzato per creare un elenco a discesa.
- I tag <option> all'interno dell'elemento <select> definiscono le opzioni disponibili nella lista.

```
<select>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</select>
```

ELEMENTI MULTIMEDIALI

IMG

- Il tag <img> definisce un'immagine in una pagina HTML.
- Il tag <img> ha due attributi obbligatori: **src** e **alt**.
- Le immagini non sono tecnicamente inserite in una pagina HTML, ma collegate. Il tag <img> crea lo spazio di per l'immagine di riferimento

```
<img src="smiley.gif" alt="Smiley face" height="42" width="42">
```

VIDEO

- Il tag <video> consente di inserire un video in una pagina WEB, ad esempio un clip filmato o un video in streaming.
- Attualmente sono tre i formati video supportati dall'elemento <video>: MP4 (Internet Explorer, Chrome, Safari, Firefox su Windows), WebM (Chrome, Firefox, Opera) e Ogg. (Chrome, Firefox, Opera).

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Il tuo browser non supporta il tag video.
</video>
```

AUDIO

- Il tag <audio> consente di inserire una risorsa audio in una pagina WEB, ad esempio una clip audioo un audio in streaming.
- Attualmente sono tre i formati audio supportati dall'elemento <audio>: MP3 (Internet Explorer, Chrome, Safari, Firefox su Windows), Wav(Chrome, Firefox, Opera. Safari) e Ogg. (Chrome, Firefox, Opera).

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Il tuo browser non supporta il tag audio.
</audio>
```

CANVAS

- Il tag HTML5 <canvas> viene usato per disegnare la grafica, al volo, tramite scripting (di solito JavaScript).
- L'elemento <canvas> non ha capacità di disegno proprie (è solo un contenitore per la grafica). È necessario utilizzare uno script per disegnare effettivamente la grafica.

ELEMENTI BLOCCO

- Il primo elemento della gerarchia dovrebbe essere , che definisce in pratica una sezione del documento. Al suo interno trovano posto gli altri elementi. Bisogna evitare annidamenti errati, che i browser fanno passare senza problemi, ma che il validatore segnala impietosamente in quanto violano le regole delle DTD.
- Esempio:

~~ Qui inserisco il mio testo ~~

```
<div><p>Qui inserisco il mio testo</p></div>
```

ELEMENTI BLOCCO

Elemento	Descrizione	DTD
<address>	Definisce un blocco di testo destinato a indirizzi, firme, indicazioni sull'autore. Non può contenere altri elementi blocco.	STF
<blockquote>	Usato per riportare citazioni da altri documenti. Il testo inserito viene indentato. Può contenere tutti gli elementi blocco.	STF
	Centra il testo che racchiude. Sconsigliato in HTML 4.0.	TF
<dir>	Crea una lista di tipo directory. Sconsigliato in HTML 4.0	TF
	Definisce un blocco di contenuto generico o una sezione del documento. Può contenere tutti gli elmenti blocco.	STF
<dl>	Crea una lista di definizione. Può contenere solo gli elementi <dt> e<dd>.	STF
<fieldset>	Usato per raggruppare campi di un form.	STF
<form>	Definisce un form. Può contenere i classici elementi dei form ma anche elementi blocco.	STF
<h1>..<h6>	Definiscono titoli e sottotitoli. Non possono contenere altri elementi blocco.	STF
<hr>	Inserisce una linea divisoria orizzontale. E' un elemento vuoto.	STF
<isindex>	Inserisce un elemento simile alle caselle di testo. Sconsigliato in HTML 4.0	TF
<menu>	Definisce una lista di tipo menu. Sconsigliato in HTML 4.0	TF
<noframes>	Inserisce contenuto alternativo per i browser che non supportano i frames.	STF

<noscript>	Inserisce contenuto alternativo per i browser che non supportano gli script.	STF
	Lista ordinata. può contenere solo l'elemento	STF
<p>	Definisce un paragrafo. Non può contenere altri elementi blocco, ma solo testo o elementi inline.	STF
<pre>	Definisce testo preformattato che mantiene le impostazioni dello spazio bianco.	STF
<table>	Definisce una tabella per l'inserimento di dati tabulari.	STF
<dd>	Descrizione di un termine in una lista di definizione.	STF
<dt>	Definizione di un termine in una lista di definizione.	STF
<frameset>	Definisce un frameset.	F
	Elemento di una lista ordinata o non ordinata.	STF
<tbody>	Definisce il corpo di una tabella. Con <thead> e <tfoot> serve a raggruppare le righe di una tabella.	STF
<td>	Cella di tabella.	STF
<tfoot>	Definisce il "piede" di una tabella.	STF
<th>	Intestazione di cella.	STF
<thead>	Definisce la testata di una tabella.	STF
<tr>	Riga di tabella.	STF

ELEMENTI INLINE

- Quando sono inseriti non danno origine a una nuova riga e possono contenere solo testo o altri elementi inline.

```
<p>Questo tasto è<b>grassetto</b></p>
```

- La parte delimitata dai tag ... non sarà posta su una nuova riga.
- Esempi come questo:

```
<b><p>Testo in grassetto</p></b>
```

- sono tollerati dai browser, ma non reggono al giudizio della validazione in quanto un elemento inline non può contenerne uno di tipo blocco.

ATTRIBUTI DI BODY

- Gli attributi per il testo, i link, il colore di sfondo e i margini dell'elemento <body> sono espressamente vietati solo nella DTD Strict, ma erano già considerati sconsigliati in HTML 4.0.

alink
background
bgcolor
link
text
vlink

VALIDARE LE PAGINE

- L'utilizzo delle dichiarazioni doctype consente l'uso dei validatori, applicazioni web che consentono di verificare se le pagine costruite soddisfano lo standard (DTD) dichiarato.
- Si può usare il validatore del W3C
- O il validatore alternativo del sito htmlhelp.com.
- Una volta validati i siti possono esporre l'icona che certifica la validazione.