



S.I.S. Piemonte

00101010  
**INFORMATICA**

---

Corso abilitante speciale Legge 143/ter D.M. 85/2005  
**A042 - INFORMATICA**

## ***PROGRAMMAZIONE B***

---

*prof. Domenico Valle*

**<HTML>**

*Hyper Text Mark-up Language*

***Erik Amedeo Viotti***  
*matricola 312580*

# *INDICE*

Collocazione curriculare	pag. 3
Prerequisiti richiesti	pag. 3
Obiettivi	pag. 3
Organizzazione delle attività	pag. 3
Materiali e strumenti	pag. 3
<b>1. Premessa</b>	pag. 4
<b>2. Introduzione ad HTML</b>	pag. 4
<b>3. Regole fondamentali</b>	pag. 5
<b>4. Domande frequenti</b>	pag. 5
<b>5. <i>Tag</i> principali</b>	pag. 6
<b>6. Tabelle</b>	pag. 7
<b>7. <i>Form</i></b>	pag. 9
<b>8. Conclusioni</b>	pag. 11
Risorse utilizzate	pag. 12

## COLLOCAZIONE CURRICOLARE

La lezione è rivolta ad una classe quarta (ultimo periodo didattico) di un Istituto Tecnico Commerciale con indirizzo Programmatori. Avendo prerequisiti praticamente nulli l'argomento potrebbe essere tranquillamente affrontato anche in terza; tuttavia, se finalizzato alla programmazione lato *server* che si affronta in quinta, la collocazione scelta non può essere anticipata.

Si presuppone una lezione (anche svolta in tempi successivi) di circa due ore in aula e quattro in laboratorio.

## PREREQUISITI RICHIESTI

Concetti di base sulla gestione dei file e sull'uso di semplici software.

## OBIETTIVI

Essere in grado di costruire una semplice pagina *web* (o un piccolo gruppo di pagine) per creare un modulo per l'inserimento di dati;

conoscere i *tag* fondamentali di HTML in modo da poterli integrare efficacemente in un programma lato *server*.

## ORGANIZZAZIONE DELLE ATTIVITA'

Lezione in aula con l'uso del videoproiettore. In seguito esercitazioni varie svolte individualmente o a coppie in laboratorio.

## MATERIALI E STRUMENTI

In aula: computer con videoproiettore.

In laboratorio: almeno un computer per ogni due studenti.

Software: un qualsiasi editor di testo (es. Notepad) ed un qualsiasi browser (es. Explorer).

## 1. PREMESSA

Questa non è una guida di HTML. Si tratta di una breve relazione su questo linguaggio, prerequisito necessario per affrontare argomenti quali la programmazione lato *server* e l'accesso ai database *online*.

## 2. INTRODUZIONE AD HTML

Il primo concetto da chiarire riguardo ad HTML è questo: non è un linguaggio di programmazione. Si tratta di un linguaggio costituito da “marcatori”, progettato in particolare per definire l'aspetto delle pagine Internet. Infatti, HTML significa *Hyper Text Mark-up Language*.

HTML è stato sviluppato alla fine degli anni '80 da Tim Berners-Lee al CERN di Ginevra. Verso il 1994 ha avuto una forte diffusione, in seguito ai primi utilizzi commerciali del web. HTML è un linguaggio di pubblico dominio la cui sintassi è stabilita dal World Wide Web Consortium (W3C), e che è basato su un altro linguaggio avente scopi più generici, l'SGML. Durante gli anni l'HTML ha subito molte revisioni e miglioramenti, che sono stati indicati secondo la classica numerazione usata per descrivere le versioni dei softwares. Attualmente l'ultima versione disponibile è la versione 4.01, resa pubblica il 24 dicembre 1999. Da allora, da parte del W3C non è stata manifestata alcuna intenzione di apportare ulteriori modifiche all'HTML, poiché verrà presto sostituito dai nuovi linguaggi XHTML ed XML.

Per capire che cosa si intenda per marcatori, osserviamo le frasi seguenti:

- Ieri sera (come faccio spesso) ho letto un libro.
- Giulio mi ha detto: “Sono stanco”.

In queste due semplici frasi sono presenti dei marcatori usati nel linguaggio scritto comune: le parentesi e le virgolette. Leggendo le parole tra le parentesi capiamo che si tratta di un'informazione aggiuntiva e non indispensabile. Trovando una frase tra le virgolette, invece, ci rendiamo conto che si tratta di parole riportate letteralmente, in questo caso pronunciate da qualcuno.

Questo è proprio il concetto su cui si basano i *mark-up language*. Esiste una serie di marcatori (detti *tag*) che delimitano e caratterizzano ciò che si trova al loro interno. Proprio come le parentesi e le virgolette, praticamente tutti i *tag* hanno una apertura ed una chiusura.

Esempio:

Sto studiando `<b>HTML</b>`.

In questo caso la scritta compresa tra `<b>` e `</b>` apparirà in **grassetto**:

Sto studiando **HTML**.

Tramite questo esempio abbiamo scoperto che i *tag* HTML hanno una sintassi di questo genere:

apertura: `<nometag>`

chiusura: `</nometag>`

In più bisogna sapere che alcuni *tag* possono essere corredati con opportuni attributi (alcuni obbligatori), i quali vanno inseriti all'interno del *tag* di apertura e non ripetuti in quello di chiusura.

### 3. REGOLE FONDAMENTALI

#### SINTASSI GENERALE DEI TAG HTML:

```
apertura:      <nometag attributo1="valore1" attributo2="valore2" >
chiusura:      </nometag>
```

- HTML non è *case-sensitive*, è quindi indifferente scrivere in maiuscolo o in minuscolo;
- non tutti i *tag* possiedono una chiusura (in quanto non racchiudono nulla);
- i valori degli attributi vengono accettati anche senza virgolette, ma è più corretto usarle;
- se si usano *tag* uno dentro l'altro, vanno sempre chiusi in modo simmetrico all'apertura.

#### STRUTTURA GENERALE DI UN FILE HTML

```
<HTML>
<HEAD>
    informazioni sulla pagina
</HEAD>
<BODY>
    contenuto della pagina
</BODY>
</HTML>
```

### 4. DOMANDE FREQUENTI

Prima di cominciare un elenco dei principali *tag* HTML, chiariamo qualche dubbio:

#### FAQ 1 – Quale programma si utilizza per scrivere in HTML?

Un qualsiasi editor di testo andrà benissimo. Esistono comunque numerosi programmi che permettono di creare una pagina web senza conoscere il linguaggio HTML, si tratta di programmi *WYSIWYG* (*what you see is what you get*) che creano in automatico il codice a partire da una pagina creata in modo intuitivo. Tuttavia questi programmi non sono indicati per almeno due categorie di persone:

- chi vuole imparare a scrivere in HTML;
- chi conosce bene questo linguaggio e cerca il “codice pulito e leggero”.

## FAQ 2 – Come si visualizza una pagina web?

Il programma che rappresenta i file HTML si chiama *browser*. Ne esistono parecchi, come per esempio Explorer, Opera, Firefox o Netscape. Tutti sono in grado di leggere un file HTML e di “ricostruire” sullo schermo la pagina web. In genere i browser sono anche in grado di visualizzare altri tipi di file, come per esempio le immagini JPG e i testi TXT.

## FAQ 3 – Quale estensione deve avere un file HTML?

Le estensioni utilizzabili (con lo stesso risultato) sono due: .htm e .html. Per non creare confusione nel proprio lavoro è consigliabile adottarne sempre solo una; per coerenza con il formato generico delle estensioni io suggerisco di usare .htm.

## FAQ 4 – Come si fa a leggere o modificare un codice a partire da un file .htm?

Aperto un file htm tramite doppio click si attiva il *browser*. Tutti i browser offrono la possibilità di vedere il codice tramite Visualizza\Codice. In alternativa è possibile aprire il file direttamente con un editor di testo (tasto destro\apri con...).

## FAQ 5 – Quale è il modo migliore per imparare a scrivere in HTML?

Creare un documento di testo e dargli l'estensione htm, applicare i *tag* che si imparano uno ad uno e osservare il risultato aprendo il file con un browser. Inoltre è molto utile visualizzare il codice di pagine visitate in Internet.

## 5. TAG PRINCIPALI

In questo elenco sono presenti i *tag* necessari per creare pagine web di base. Per ogni *tag* sono indicati gli eventuali attributi più importanti. Dove non specificato diversamente, esiste un corrispondente *tag* di chiusura.

### <HTML>

Comprende l'intero file, e indica che il testo contenuto all'interno è da intendersi scritto in HTML.

### <HEAD>

Comprende informazioni varie, tra cui le parole chiave, l'autore, il titolo, la lingua usata.

### <TITLE>

Si trova all'interno di <HEAD>, e indica il titolo della pagina che apparirà sulla barra della finestra del browser che la visualizza.

### <BODY>

Comprende tutto ciò che deve essere rappresentato sulla pagina.

Attributi:      bgcolor or      colore di sfondo della pagina (in formato RGB)  
                 background      immagine di sfondo della pagina (indicata tramite il percorso relativo  
   o assoluto del file)

Esempio:      <body background="IMG/sfondo1.bmp">

## <FONT>

Indica alcune informazioni sul formato del testo da visualizzare.

Attributi:      face                  font da applicare (es. Verdana o Tahoma)  
                 si ze                  dimensione del font (da 1 a 7)  
                 col or                  colore del testo

Esempio:      <font face="Verdana" si ze=" 4" col or=#8000ff>

## <P>

Definisce un paragrafo. Diversi paragrafi verranno automaticamente separati con un doppio invio.

Attributi:      al i gn                  allineamento del testo: l eft, ri ght, center o j usti fy

## <I MG>

Serve per inserire una immagine, e non ha il *tag* di chiusura.

Attributi:      src                  obbligatorio, indica il file immagine da inserire  
                 al i gn                  allineamento rispetto al testo  
                 al t                  alternativa di testo per *browser* che non visualizzano immagini

Esempio:      <i mg al i gn="ri ght" src="I MG\bl og\spi derman. j pg" >

## FORMATTAZIONE SEMPLICE

<b>      grassetto

<i >      inclinato

<u>      sottolineato

<br>      a capo (senza *tag* di chiusura) – l'invio dato in un editor non viene rilevato

<tt>      carattere *monospaced*

<h1>      formato titolo di livello 1

...

<h6>      formato titolo di livello 6

## <A>

Si tratta del tag concettualmente più importante. Serve infatti a creare i collegamenti ipertestuali a file che si trovano in locale o in rete, cosa indispensabile per un ipertesto.

Attributi:      href                  obbligatorio, indica la destinazione del collegamento  
                 target                  serve a specificare la finestra in cui apparirà il file

Esempio:      <a href="http://www.eri kvi otti . i t" target="bl ank" >

## 6. TABELLE

Le strutture tabellari sono indispensabili per ottenere allineamenti di testi e immagini nelle pagine HTML. Per scoprire i *tag* necessari, vedremo ora un codice commentato relativo ad una semplice tabella e il suo risultato.

```

<html >
<head><title>Orari SIS</title></head>
<body>
<table border="2" bordercolor=#006000 width="500" align="center">
<tr><th>GIORNO</th><th>CORSO</th><th>LUOGO</th></tr>
<tr><td>lunedì</td><td>Pedagogia</td><td>Rosa Luxembourg</td></tr>
<tr><td>martedì</td><td>Programmazione B</td><td>Avogadro</td></tr>
<tr><td>mercoledì</td><td>Didattica delle reti</td><td>Avogadro</td></tr>
<tr><td>giovedì</td><td>Programmazione B</td><td>Avogadro</td></tr>
<tr><td>venerdì</td><td rowspan="2">nulla</td><td rowspan="2">-</td></tr>
<tr><td>sabato</td></tr>
</table>
</body>
</html >

```

### <TABLE>

Racchiude la tabella e ne definisce alcune proprietà.

Attributi:      border                      spessore dei bordi  
                  bordercolor        colore dei bordi  
                  width                 larghezza totale (in pixel o in percentuale della finestra)  
                  align                 allineamento nella pagina

<tr>            riga

<th>           casella evidenziata (il numero di caselle determina il numero delle colonne)

<td>           casella normale (rowspan indica il numero di righe occupate)

GIORNO	CORSO	LUOGO
lunedì	Pedagogia	Rosa Luxembourg
martedì	Programmazione B	Avogadro
mercoledì	Didattica delle reti	Avogadro
giovedì	Programmazione B	Avogadro
venerdì	nulla	-
sabato		



## 7. FORM

Nell'ottica di arrivare a parlare di pagine web dinamiche o interattive, ha molta importanza la possibilità per l'utente di inviare al *server* informazioni o richieste di vario genere.

Questi dati, come si vedrà, saranno utilizzati come input da parte di programmi che girano sul server e generano una pagina web in risposta all'utente.

L'invio di dati è possibile attraverso i *form*, contenitori in cui possono apparire caselle di testo, pulsanti di opzione e altro. Questa parte di HTML va affrontata in maniera più approfondita di altre, in quanto racchiude alcuni importanti concetti teorici.

Innanzitutto ogni *form* è racchiuso da un *tag* `<form>`.

In esso va indicata una informazione importante tramite l'attributo `method`: la modalità con la quale i dati che l'utente immetterà nel *form* verranno inviati al *server*. Con il metodo `GET` i dati vengono inviati concatenati alla stringa che indica la destinazione (URL); esiste un limite alla lunghezza dell'URL gestibile dal *server*. Con il metodo `POST`, invece, i dati vengono inviati con un flusso informativo separato, rendendo possibile l'invio di una quantità indefinita di informazioni. In base al metodo di invio sarà differente il metodo di raccolta dei dati da parte del programma ricevente.

Un altro importante attributo del *form* è `name`, in cui si nomina il *form* stesso in modo tale che il programma ricevente sia in grado di distinguere tra dati provenienti da differenti *form*.

Infine l'attributo che definisce il destinatario dei dati: `action`. Con esso si indica l'URL o l'indirizzo *e-mail* a cui ciò che viene scritto dall'utente verrà inviato.

I possibili oggetti presenti in un *form* sono ben conosciuti da chiunque abbia un minimo di familiarità con il PC. Ora ne vedremo i relativi *tag*, e successivamente analizzeremo un esempio.

I seguenti *tag* hanno un attributo in comune: `name`. Esso indica il nome dell'oggetto ed è indispensabile al programma che raccoglierà i dati. Questo attributo verrà dato per sottinteso nell'elenco seguente.

### **<INPUT>**

Questo *tag* è molto importante perchè racchiude numerosi tipi di oggetti. Il tipo viene indicato tramite l'attributo `type`. Non ha *tag* di chiusura.

#### **<INPUT type="text">**

Si tratta della classica casella di testo in cui l'utente può scrivere tramite tastiera.

Attributi:     `value`            il contenuto della casella al caricamento della pagina  
              `size`             dimensione in caratteri della casella  
              `maxlength`    lunghezza massima della stringa inseribile

#### **<INPUT type="password">**

E' una casella di testo in cui i caratteri inseriti vengono visualizzati come asterischi o pallini.

Attributi:     quelli della normale casella di testo

#### **<INPUT type="submit">**

Disegna sulla pagina un pulsante cliccabile che ha la funzione di inviare i dati inseriti.

Attributi:     `value`            il testo visualizzato nel pulsante

#### **<INPUT type="reset">**

Disegna sulla pagina un pulsante cliccabile che ha la funzione di azzerare i dati inseriti.

Attributi:     `value`            il testo visualizzato nel pulsante

<INPUT type=radio>

Si tratta dei pulsanti di opzione, che creati in gruppo sono autoescludentesi. Per creare un gruppo di radio, è necessario che abbiano uguale valore dell'attributo name.

Attributi: value il valore che verrà rilevato se l'opzione è selezionata

<INPUT type=checkbox>

E' la casella di selezione, corrispondente ad una variabile logica.

Attributi: checked e' un attributo speciale al quale non si assegna nessun valore; se il tag lo contiene, al caricamento della pagina risulterà selezionato.

<INPUT type=hidden>

Questo campo è nascosto ed è inutilizzabile da parte dell'utente. Il server ne riceverà comunque il valore.

Attributi: value il valore che verrà rilevato dal server

<TEXTAREA>

Crea un campo in cui è possibile inserire un testo su più righe. Esso possiede un tag di chiusura, e ciò che si trova all'interno apparirà al caricamento della pagina.

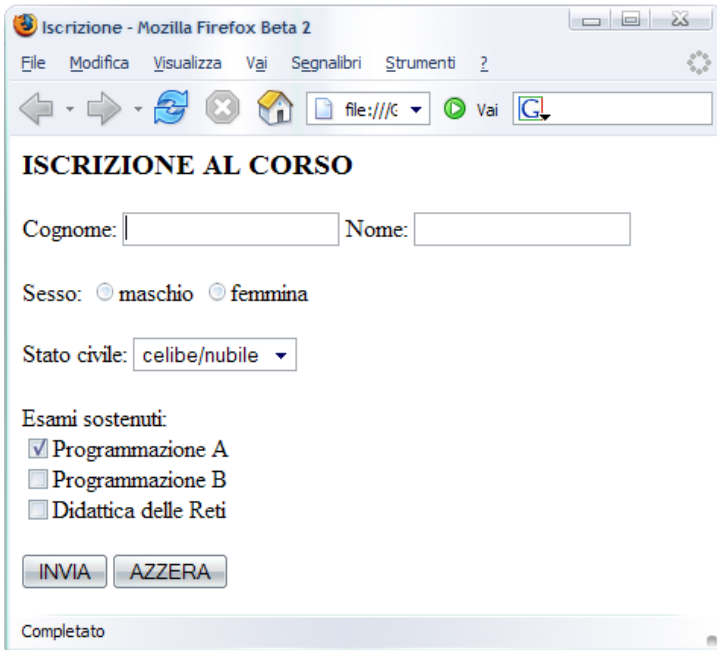
Attributi: rows numero di righe (non limita l'inserimento)  
cols numero di caratteri per ogni riga

<SELECT>

Si tratta del menu "a tendina". All'interno di questo tag si elencano le varie opzioni tramite il tag senza chiusura <OPTION>.

Quello che segue è un semplice esempio contenente alcuni dei tag visti qui sopra.

```
<html ><head><ti tle>I scri zi one</ti tle></head><body>
<h3>I SCRIZIONE AL CORSO</h3>
<form name="f1" method="post" action="http://www.eri kvi otti . i t/reg. asp">
Cognome: <i nput type=text name="cgn" si ze=20>
Nome: <i nput type=text name="nme" si ze=20><br><br>
Sesso: <i nput type=radi o name="s" val ue="m">maschi o
<i nput type=radi o name="s" val ue="f">femmi na<br> <br>
Stato ci vile: <sel ect>
<opti on>cel i be/nubi le<opti on>coni ugato/a<opti on>di vorzi ato/a
<opti on>vedovo/a</sel ect><br><br>
Esami sostenuti : <br>
<i nput type=checkbox name="pa" checked>Programmazi one A<br>
<i nput type=checkbox name="pb">Programmazi one B<br>
<i nput type=checkbox name="dr">Di datti ca del le Reti <br><br>
<i nput type=submi t val ue="I NVIA"> <i nput type=reset val ue="AZZERA">
</form>
</body></html >
```



Si può notare che:

- i campi di testo sono vuoti in quanto non è stato specificato l'attributo `value`;
- nel menu a tendina viene subito visualizzata la prima opzione;
- la prima casella di selezione è già selezionata, avendo inserito l'attributo `checked`.

Ovviamente, grazie agli altri *tag* imparati in precedenza, è possibile ottenere un prodotto graficamente molto migliore. Non va però dimenticato che maggior complessità grafica equivale spesso a maggior "peso" del file.

## 8. CONCLUSIONI

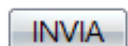
Terminata questa breve esposizione sui concetti di base del linguaggio HTML, ora si è in grado di proseguire da soli nella scoperta di nuovi *tag*. L'argomento non è stato trattato nella sua interezza, data la finalità di questa relazione puramente propedeutica alle successive. A chi fosse interessato consiglio di approfondire tramite le centinaia di risorse disponibili in Internet, senza dimenticare il buon metodo di visualizzare il codice di pagine altrui.

Si è accennato al fatto che XHTML (unione di HTML e XML) sia il più accreditato successore di HTML.

L'XHTML *Transitional* si distingue dall'HTML principalmente perché è più compatibile con le specifiche dell'XML. La differenza più importante è che tutti i tag devono essere ben strutturati, cioè obbedire ad una serie di regole che ne assicurino la coerenza reciproca. Inoltre i tag devono essere sempre scritti in lettere minuscole, convenzione in contrasto con l'abitudine invalsa a partire dalla versione 2.0 di HTML, quando la maggior parte dei programmatori preferiva le maiuscole. Nell'XHTML tutti gli attributi (compresi quelli numerici) devono essere scritti fra virgolette, cosa facoltativa in SGML e HTML, in cui le virgolette possono essere omesse se il contenuto è una stringa alfanumerica o comprende alcuni altri caratteri speciali riservati. Tutti gli elementi del linguaggio devono inoltre essere *terminati*, compresi quelli *vuoti* (ad esempio `img` e `br`). Per eseguire la terminazione in modo implicito si può aggiungere una `"/` di chiusura al tag di apertura (es: `<img ... />` e `<br />`). Altri tipi di abbreviazione non sono invece permessi (es: `<option selected>`).

Come si può dedurre da questo breve approfondimento, sarebbe utile incominciare ad abituarsi a scrivere in HTML in modo abbastanza rigoroso per essere pronti al passaggio.

Ma quello che interessa ora è comprendere ciò che succede nel momento in cui viene premuto il pulsante...



## RISORSE UTILIZZATE

### Siti Internet:

- <http://it.wikipedia.org/wiki/html> - consultato il 12/02/2007
- <http://it.wikipedia.org/wiki/xhtml> - consultato il 12/02/2007

### Libri di testo:

- *Piero Gallo, Fabio Salerno* - "HTML, CSS, Javascript" - ed. Minerva Italica

### Software:

- Microsoft Word 2000
- Adobe Acrobat 6.0 Standard
- Notepad++ 3.9
- Mozilla Firefox 1.5 beta 2

### Copia del presente documento reperibile su:

- <http://www.erikviotti.it/doc/html.pdf>