

# Elementi di web design (parte I): introduzione all'HTML

**Luca Casotti**

E' necessario conoscere:

- Filesystem: cos'è un file e cos'è una cartella.
- Differenze fra tipi di file (sui sistemi windows: le “estensioni”).
- Cos'è un file di testo.

## **Le pagine web**

Due o più computer sono collegati in rete quando possano scambiarsi informazioni. In particolare, se se ne ha l'autorizzazione, sarà possibile leggere da un certo computer il contenuto di file di testo che si trovano su un altro computer.

Ci vuole poco ad immaginarsi come, in molte situazioni (ad esempio una rete locale), sia molto comodo avere un computer con una cartella condivisa (cioè visibile agli altri computer collegati) in cui tenere una serie di documenti contenenti informazioni utili (ad esempio i numeri di telefono dei clienti di un'azienda).

Allargando le dimensioni a quelle della rete internet e prendendo in considerazione file di tipo più complesso rispetto ai semplici documenti di testo si comprende facilmente quale sia stato lo sviluppo del web (cioè la parte “navigabile” di internet).

Un *server* web è un computer abilitato alla condivisione, secondo regole ben definite, di file verso l'esterno. Questo vuol dire che collegandosi alla rete è possibile, con un opportuno programma (il *browser*, cioè il “navigatore”; browse = scorrere, scartabellare), consultare questi file.

Il browser in questo caso svolge la funzione di *client*, è infatti un fruitore delle informazioni offerte dal server.

Un computer, per essere un server web, deve essere collegato in rete e deve essere dotato di un programma apposito (fra i più noti vi sono *Apache* e *IIS*).

Le regole per la comunicazione client/server stabiliscono:

- Il protocollo con cui devono comunicare server e browser.
- Il modo in cui il browser identifica un file sul server.
- Il modo in cui devono essere interpretati i file letti dal browser.

Il protocollo si chiama http; le sue caratteristiche non ci interessano in questo corso.

Il modo di identificare i file sul server si basa sull'indirizzo IP e sul nome di dominio (ai quali abbiamo accennato nella scorsa lezione). In pratica, digitando un indirizzo IP sul browser oppure il nome di dominio associato (se esiste) si possono visualizzare i file contenuti nello spazio relativo a quell'indirizzo, che altro non sono che le pagine del sito associato a quell'indirizzo.

I file letti dal browser vengono interpretati diversamente a seconda del tipo; i file corrispondenti alle pagine web sono di tipo “html” (la loro estensione è htm o html). Ci occuperemo di questo tipo di file in quanto la comprensione di alcuni elementi di base del formato html ci consentirà di fatto la

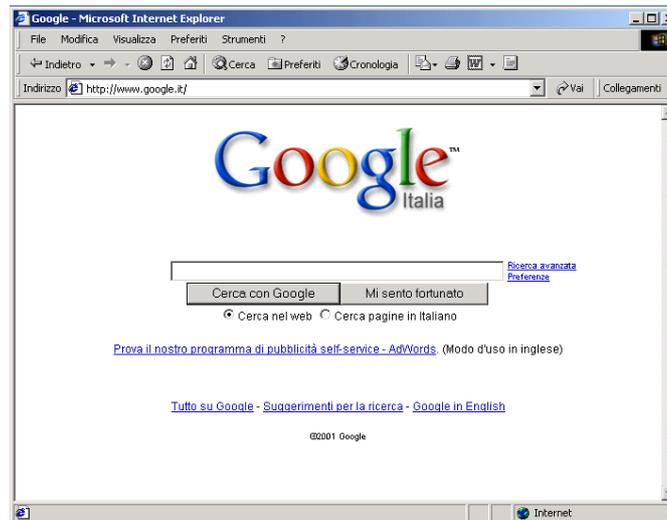
realizzazione di siti internet.

Più semplicemente un file html è una pagina web. Un insieme di pagine web, generalmente collegate fra loro in vario modo, costituisce un sito web.

## **Il ruolo del browser**

Quando si naviga un sito web si fa qualcosa di più che leggere un file da un computer remoto. Limitiamo le nostre considerazioni ai file html.

La pagina web [www.google.it](http://www.google.it) è raffigurata di seguito così come ci appare nel browser



Mentre in realtà il file html interpretato come un semplice file di testo (detto in altre parole: andiamo a vedere cosa c'è dentro il file html) rivela il contenuto seguente:

```
<html>
<head>
<META HTTP-EQUIV="content-type" CONTENT="text/html; charset=ISO-8859-1">
<title>Google</title>
<style>
<!--
body,td,a,p,.h{font-family:arial,sans-serif;}
.h{font-size: 20px;}
.h{color:}
.q{text-decoration:none;
color:#0000cc;}
/-->
</style>
<script>
<!--
function sf(){document.f.q.focus();}
function e(x){x =escape(x).replace(/\+/g,"%2b");return x;}
function c(p)
{
if(document.f.q.value.length)location.href=p+'&q='+e(document.f.q.value);
```



```

<a href=/intl/it/help.html>Suggerimenti&nbsp;per&nbsp;la&nbsp;ricerca</a> -
<a href=http://www.google.com/en>Google in English</a>
</font>
<p><font size=-2>&copy;2001 Google</font></center>
</body>
</html>

```

*Nota: per ottenere come risultato visuale la pagina in figura è necessario fornire al browser non solo il file html ma anche le immagini grafiche alle quali il file fa riferimento. Vedremo questo aspetto più in dettaglio nel seguito, l'esempio serve soltanto a mostrare cosa può esserci dietro a una pagina vista dal browser.*

Ciò che deve essere chiaro è che, quando si naviga, non facciamo altro che chiedere al nostro browser di:

- 1 leggere, da un indirizzo specificato, un certo file html;
- 2 interpretare questo file;
- 3 visualizzare graficamente sullo schermo la pagina corrispondente a quel file html.

Se il nostro computer è collegato in rete e la pagina richiesta si trova su un computer remoto il nostro browser effettuerà la richiesta al server associato all'indirizzo da noi specificato, ad esempio [www.trenitalia.it](http://www.trenitalia.it). E' però anche possibile chiedere al browser di visualizzare un file html presente sul nostro computer. In questo caso non dovremo chiamare in causa nessun server.

Questo è in realtà quello che succede ogni volta che salviamo su disco una pagina mentre la visualizziamo nel browser grazie al collegamento alla rete; una volta terminato il collegamento possiamo visualizzare di nuovo la pagina salvata, ad esempio con un doppio click oppure trascinandola su una finestra del browser stesso.

Il browser, quindi, non vede soltanto le pagine reperite tramite internet (a collegamento "acceso"); vede anche quelle salvate sul proprio computer (a collegamento "spento"). In questo caso sul browser non si dovrà digitare l'indirizzo IP né il dominio (non c'è nessun indirizzo IP e non c'è nessun dominio associato) bensì il percorso completo del file, ad esempio "c:\documenti\html\pagina\_prova.html". Questo ci consente di cominciare con i primi esperimenti senza necessitare di una connessione attiva né della disponibilità di un server.

## **La prima pagina HTML**

HTML significa Hypertext Mark-up Language (linguaggio per ipertesti basato su marcatori). Un ipertesto non è altro che un testo all'interno del quale alcune parole o frasi sono evidenziate e cliccabili con il mouse; se vengono cliccate la visuale si sposta su un'altra parte del testo. Le parole cliccabili vengono dette *link* o collegamenti.

I siti internet sono letteralmente basati sui link, che ne caratterizzano profondamente la struttura. L'informazione presente solitamente sui siti segue questa assenza di un unico piano sequenziale, che invece è presente nelle pubblicazioni cartacee.

HTML è un linguaggio. Tramite questo linguaggio è possibile definire una pagina da visualizzare a video tramite browser, quindi l'HTML è il linguaggio con il quale l'autore della pagina "parla" al browser.

Per scrivere utilizzeremo il più semplice editor di testo fornito in dotazione con il sistema operativo; sui sistemi windows questo è il “notepad”, ovvero “blocco note” (sugli altri esistono comunque editor analoghi).

Per farlo partire apriamo il menu di avvio, scegliamo il sottomenu programmi, quindi apriamo il sottomenu accessori e scegliamo la voce blocco note.

Nota: è possibile utilizzare qualsiasi editor di testi, anche un word processor come Microsoft Word. Tuttavia ciò è sconsigliato in quanto risulta inutilmente più complesso e molto più gravoso per il calcolatore (utilizza una quantità di risorse molto maggiore). Inoltre per far funzionare il tutto bisogna accertarsi di salvare in formato testo.

Scriviamo il seguente testo:

```
<html>
  <head>
    <title>La mia prima pagina HTML</title>
  </head>
  <body>
  </body>
</html>
```

Questa è la base per qualsiasi pagina HTML. Serve per informare il browser quale linguaggio si usa (l'html in questo caso), qual è il titolo della pagina.

Le parole o frasi racchiuse fra “<” e “>”, parentesi angolate, sono dette *tag* e sono istruzioni per il browser. Tutto ciò che è scritto fra parentesi angolate non verrà quindi visualizzato, ma avrà un effetto su ciò che verrà visualizzato.

Ogni tag ha un inizio (<nome\_tag>) e una fine (</nome\_tag>) e il suo effetto si manifesterà di conseguenza. Ad esempio il tag *title* inizia con <title> e termina con </title>. Tutto ciò che sta nel mezzo sarà interpretato come titolo della pagina. Grazie a questo sistema basato su coppie del tipo <tag> </tag> è possibile individuare porzioni di testo e, in base alla tipologia del tag, specificare una proprietà da applicare a questa porzione.

Alcuni tag (quelli che non devono indicare una porzione di testo bensì un punto del testo) non hanno bisogno di apertura e chiusura separate (vedremo più dettagliatamente in seguito).

A questo punto salviamo il file di testo (scegliendo “File/Salva con nome” dal menu principale). Nel campo “nome file” digitiamo:

```
c:\prove_html\pagina.htm
```

e facciamo clic sul tasto “Salva”.

L'estensione “.htm” (oppure “.html”, che è del tutto analoga) è importante affinché il browser riconosca che il file contiene informazioni in linguaggio HTML. Per evitare che il programma “blocco note” cambi automaticamente l'estensione in “.txt” quando si effettua il salvataggio, può essere necessario (a seconda delle versioni del programma “Blocco note”) scegliere “Tutti i file” nel

menu “Salva come:”.

*Nota: durante questo corso sarà necessario evitare gli spazi e i caratteri speciali (vocali accentate, accenti, segni vari) nei nomi che si danno ai file html. Questo accorgimento eviterà problemi durante la pubblicazione del sito sul web. Ovviamente queste limitazioni riguardano esclusivamente i nomi che decidiamo di assegnare ai file, e non hanno niente a che vedere con il contenuto dei file stessi.*

Proviamo a visualizzare il file nel browser tramite un doppio click sull'icona del file stesso. Si può notare come il titolo nella barra superiore della finestra sia proprio quello scelto da noi (“*la mia prima pagina HTML*”).

Torniamo al blocco note e aggiungiamo una frase all'interno del tag “body”, in questo modo:

```
<html>
  <head>
    <title>La mia prima pagina HTML</title>
  </head>
  <body>
    Questo &egrave; il testo della mia prima pagina.
  </body>
</html>
```

Salviamo e visualizziamo nel browser. Noteremo che nella nostra pagina è comparso il testo desiderato. Per visualizzare la vocale accentata “è” si utilizza il codice “&egrave;” (per la “é” avremmo usato “&eacute;”).

## **Controllo del testo**

Nel linguaggio HTML i caratteri speciali che indicano i punti in cui il testo deve andare a capo (quelli che ad esempio il blocco note inserisce nel testo quando viene premuto il tasto “Invio”) vengono ignorati. Questo vuol dire che, se scriviamo due righe di testo separate all'interno del tag <body>, queste verranno comunque visualizzate sulla stessa riga.

*Esempio:*

```
<html>
  <head>
    <title>La mia prima pagina HTML</title>
  </head>
  <body>
    Riga 1 della pagina.
    Riga 2 della pagina.
  </body>
</html>
```

Per fare in modo che il browser rappresenti porzioni di testo su righe differenti, dobbiamo inserire un *tag* che indichi al browser di andare a capo di una riga. Il tag in questione è “<br />” (questo tag ha “incorporato” il delimitatore di fine, in quanto indica un punto del testo anziché una regione di testo).

## **Neretto e corsivo**

In HTML si può specificare che una porzione di testo deve essere in corsivo racchiudendola tra i tag “<em>” e </em>. Per il grassetto, o neretto, bisogna racchiudere la porzione di testo interessato tra i tag <strong> e </strong>.

*Nota: per combinare più tag basta metterli l'uno all'interno dell'altro, tenendo presente che si deve sempre chiudere per primo l'ultimo tag che si è aperto.*

### *Esercizio*

Scriviamo un testo in neretto, uno in corsivo e uno in neretto corsivo

*Una soluzione possibile:*

```
<html>
  <head>
    <title>La mia prima pagina HTML</title>
  </head>
  <body>
    Testo normale
    <strong>Testo neretto</strong><br />
    <em>Testo corsivo</em><br />
    <strong><em>Testo neretto corsivo</em></strong>
  </body>
</html>
```

## **Controllare l'aspetto**

E' possibile controllare l'aspetto della pagina (ad esempio il colore e la grandezza dei caratteri) grazie a specifiche indicazioni; questa operazione è detta “definizione degli stili”.

Per realizzare una pagina correttamente strutturata, quindi adatta a far sì che l'informazione che contiene possa essere letta correttamente sia dai motori di ricerca che dai programmi di ausilio per utenti con disabilità, le informazioni di tipo stilistico saranno ben separate da quelle relative al contenuto.

Per farlo utilizzeremo dei comandi appositi (tag e attributi “style”), che potranno essere inseriti in tre diverse modalità: in un file esterno, all'interno della pagina, e addirittura all'interno di un tag.

### **Controlli stilistici: i colori**

Se l'autore non fornisce indicazioni particolari, la pagina viene solitamente rappresentata a sfondo grigio, con il testo scritto in nero. Esiste però la possibilità di controllare questi parametri, introducendo all'interno della nostra pagina web le cosiddette “note stilistiche”.

Torniamo al blocco note e aggiungiamo alcune righe di codice all'interno della porzione delimitata dal tag “<head>”, in questo modo:

```
<html>
  <head>
    <title>La mia prima pagina HTML</title>
    <style type="text/css">
```

```

        .colorSet1{
            background-color: #000000;
            color: #FFFFFF;
        }
    </style>
</head>
<body class="colorSet1">
    Documento con sfondo nero e testo bianco.
</body>
</html>

```

Come è possibile notare, abbiamo aggiunto all'interno del tag STYLE (fra “<style>” e “</style>”) alcune note stilistiche.

## **La codifica dei colori in HTML**

I colori in HTML vengono specificati tramite le loro componenti RGB (Red Green Blue, cioè rosso verde blu). Per ogni componente verrà specificato un valore su due cifre. Si possono ottenere tutti i colori mescolando i contributi dei tre sopra citati.

Il formato numerico è quello esadecimale, quindi i valori ammessi vanno da 00 a FF (da 0 a 255 in decimale) per ogni componente. Senza andare troppo a fondo basterà sapere che nel sistema esadecimale le cifre non sono 10 (da 0, 1, 2, 3, 4, 5, 6, 7,8, 9), come nel sistema decimale, ma 16 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Questo vuol dire che dopo il 9 non ci sarà il numero 10 ma il numero A, poi il B ecc. fino al numero F dopodiché arriverà il 10.

Ad esempio il bianco sarà formato da tutti i colori alla massima intensità quindi avremo FF per il rosso, FF per il verde e FF per il blu. Alla fine il valore sarà quindi FFFFFFF e analogamente il nero sarà 000000, il rosso FF0000, il verde 00FF00, il blu 0000FF.

Se siamo poco entusiasti dall'idea di dover ricorrere ad una calcolatrice per calcolare il colore, possiamo comunque far ricorso ai principali colori semplicemente indicandone il nome: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white e yellow.

### *Esercizio*

Proviamo a impostare per lo sfondo un grigio molto chiaro e per il testo il colore blu.

### *Una possibile soluzione*

```

<html>
  <head>
    <title>La mia prima pagina HTML</title>
    <style type="text/css">
      .colorSet1{
        background-color: #DDDDDD;
        color: #0000FF;
      }
    </style>
  </head>
  <body class="colorSet1">

```

```
    Questo testo serve a mostrare il colore.
  </body>
</html>
```

## ***I paragrafi in HTML***

Un testo piuttosto lungo può essere suddiviso in paragrafi. Il tag utilizzato per questo è *p*. Vediamone un esempio:

```
<html>
  <head>
    <title>La mia prima pagina HTML</title>
    <style type="text/css">
      .colorSet1{
        background-color: #DDDDDD;
        color: #0000FF;
      }
    </style>
  </head>
  <body class="colorSet1">
    <p>
      Paragrafo 1
    </p>
    <p>
      Paragrafo 2
    </p>
  </body>
</html>
```

I paragrafi, come solitamente viene fatto in un word processor (come Microsoft Word), possono essere allineati al centro, a destra e a sinistra.

Per controllare questa proprietà possiamo ricorrere allo stile “text-align” assegnandogli i valori “center”, “right” e “left” rispettivamente.

### ***Esempio***

```
<html>
  <head>
    <title>La mia prima pagina HTML</title>
    <style type="text/css">
      .colorSet1{
        background-color: #DDDDDD;
        color: #0000FF;
      }

      .leftAling{
        text-align: left;
      }
    </style>
  </head>
  <body class="colorSet1">
    <p class="leftAling">
      Paragrafo 1
    </p>
  </body>
</html>
```

```

        .rightAlign{
            text-align: right;
        }

        .centerAlign{
            text-align: center;
        }

    </style>
</head>
<body class="colorSet1">
<p class="leftAlign">
    Testo allineato a sinistra
</p>
<p class="rightAlign">
    Testo allineato a destra
</p>
<p class="centerAlign">
    Testo allineato al centro
</p>
</body>
</html>

```

## ***I caratteri in HTML***

E' possibile controllare la dimensione e la famiglia dei caratteri utilizzando gli stili nel seguente modo:

```

<html>
  <head>
    <title>La mia prima pagina HTML</title>
    <style type="text/css">
      .colorSet1{
        background-color: #DDDDDD;
        color: #0000FF;
      }

      .style1{
        font-family: "Times New Roman", Times, serif;
        font-size: 14px;
      }

      .style2{
        font-family: Arial, Helvetica, sans-serif;
        font-size: 16px;
      }

      .style3{
        font-family: Verdana, Arial, Helvetica, sans-serif;

```

```

        font-size: 20px;
    }

</style>
</head>
<body class="colorSet1">
    <p class="style1">
        Carattere Times a grandezza 14px.
    </p>
    <p class="style2">
        Carattere Arial a grandezza 16px.
    </p>
    <p class="style3">
        Carattere Verdana a grandezza 20px.
    </p>
</body>
</html>

```

Nota: scegliendo il carattere a piacimento non si può essere sicuri che chi navigherà il sito vedrà correttamente il testo. Se il carattere scelto non è installato sul computer di chi naviga, il sito non potrà essere visualizzato come ci si aspetterebbe. Si consigliano le seguenti combinazioni di caratteri:

- **Arial, Helvetica, sans-serif**
- **Times New Roman, Times, serif**
- **Verdana, Arial, Helvetica, sans-serif**

## ***I link in HTML***

I “link” o collegamenti” hanno lo scopo far sì che una porzione di testo diventi un oggetto cliccabile, che rimanda a un'altra pagina HTML (oppure ad una specifica zona all'interno di una pagina). Per definire un link si utilizza il tag “*a*” e gli attributi principali sono “*href*” (specifica l’indirizzo dell’oggetto linkato) e “*target*” (se associato al valore “*\_blank*” specifica che il collegamento deve aprirsi in una nuova finestra).

### *Esempio*

```
<a href="chi_siamo.htm">Chi siamo</a>
```

In questo caso abbiamo creato un collegamento alla pagina associata al file “chi\_siamo.htm”. Ovviamente, affinché il collegamento sia funzionante, dovrà essere presente il file specificato dall'attributo “href”.

Se la pagina che vogliamo collegare non appartiene al nostro sito (ad esempio se volessimo collegare l'homepage di Google) il collegamento si dice “esterno”; in questo caso dovremo specificare tutto l'indirizzo (comprensivo del prefisso “http://”).

Se vogliamo che la pagina di destinazione si apra in una nuova finestra del browser, dobbiamo aggiungere l'attributo "target" in questo modo:

```
target="_blank"
```

Per provare il funzionamento dei link dovremo ovviamente creare almeno due pagine, altrimenti la prima pagina non potrà essere collegata a nessuna altra pagina (collegamenti esterni a parte).

### *Esercizio*

Creiamo un mini-sito di due sole pagine: una home-page (pagina iniziale) e una pagina chiamata contatti.

### *Una possibile soluzione*

Creiamo una pagina contenente il seguente codice e salviamola con il nome "index.htm":

```
<html>
  <head>
    <title>Il mio sito</title>
  </head>
<body>
  <p>
    Home page
  </p>
  <p>
    <a href="index.htm">Home</a> -
    <a href="contatti.htm">Contatti</a> -
    <a href="http://www.google.it" target="_blank">Cerca con Google</a>
  </p>
</body>
</html>
```

Dopodiché creiamo e salviamo il seguente file con il nome "contatti.htm":

```
<html>
  <head>
    <title>Il mio sito</title>
  </head>
<body>
  <p>
    Contatti
  </p>
  <p>
    <a href="index.htm">Home</a> -
    <a href="contatti.htm">Contatti</a> -
    <a href="http://www.google.it" target="_blank">Cerca con Google</a>
  </p>
</body>
</html>
```

A questo punto abbiamo realizzato un primo sito navigabile molto semplice.

## **Stili CSS su file esterno**

Nella realizzazione di un sito web ci troveremo inevitabilmente a dover utilizzare stili identici su più pagine, come ad esempio è capitato anche nell'esercizio precedente (in cui abbiamo realizzato due pagine contenenti la stessa definizione dello stile del tag `<body>`).

In questi casi è utile inserire le definizioni di stile in un file esterno, e richiamare poi questo file nelle pagine in cui intenderemo utilizzare gli stili che questo contiene. In questo modo risparmiamo:

- tempo (le definizioni vengono scritte una volta sola)
- banda (dato che il file esterno è sempre lo stesso per tutte le pagine, questo verrà scaricato una sola volta anziché una volta per ogni pagina)
- errori (saremo sempre sicuri che gli stili saranno definiti allo stesso modo in tutte le pagine; inoltre, in caso di modifiche, non dovremo ricordarci di ricopiarle in tutti i file, processo ad altissima probabilità di errore)

Riprendiamo l'esempio dell'esercizio precedente e creiamo un nuovo file col nome "stili.css"; al suo interno scriviamo le definizioni di stile che avevamo utilizzato all'interno delle pagine. Trattandosi di un file riservato agli stili css, non sarà necessario utilizzare il tag `<style>`.

```
.colorSet1{
    background-color: #DDDDDD;
    color: #0000FF;
}
```

A questo punto non rimane che collegare le nostre due pagine al file appena creato (eliminando le dichiarazioni di stile, che ormai non ci servono più) utilizzando la sintassi che vediamo nella quarta riga:

```
<html>
  <head>
    <title>Il mio sito</title>
    <link href="stili.css" rel="stylesheet" type="text/css" />
  </head>
<body class="colorSet1">
  <p>
    Home page
  </p>
  <p>
    <a href="index.htm">Home</a> -
    <a href="contatti.htm">Contatti</a> -
    <a href="http://www.google.it" target="_blank">Cerca con Google</a>
  </p>
</body>
</html>
```