

# **Pagine e siti web**

Loris Tissino  
© Loris Tissino, 1997–2003

# Indice

<b><u>1 Introduzione</u></b> .....	<b>1</b>
<u>1.1 Obiettivi nella costruzione di pagine e siti web</u> .....	1
<u>1.2 Come realizzare delle pagine web</u> .....	1
<u>1.3 Dove trovare documentazione e riferimenti per conoscere l'HTML</u> .....	2
<b><u>2 La cassetta degli attrezzi</u></b> .....	<b>3</b>
<u>2.1 Editor testuali</u> .....	3
<u>2.2 Programmi di utilità</u> .....	3
<u>2.3 Validatori on-line</u> .....	3
<b><u>3 Quando le pagine sono pronte</u></b> .....	<b>4</b>
<u>3.1 Pubblicare le pagine</u> .....	4
<u>3.2 Servizi aggiuntivi</u> .....	4
<b><u>4 Cenni all'HTTP</u></b> .....	<b>5</b>
<u>4.1 Definizioni di base</u> .....	5
<u>4.2 La comunicazione HTTP</u> .....	5
<u>4.3 La richiesta</u> .....	5
<u>4.4 La risposta</u> .....	6
<u>4.5 Ulteriori informazioni</u> .....	7
<b><u>5 Progettazione delle pagine web</u></b> .....	<b>8</b>
<u>5.1 Documentazione su web</u> .....	8
<u>5.2 Lo spazio dei nomi in un sito web</u> .....	8
<u>5.2.1 Estensione dei nomi dei file</u> .....	8
<u>5.2.2 Nomi dei file e delle cartelle</u> .....	8
<u>5.2.3 File principale di una cartella web</u> .....	9
<u>5.2.4 Mappa del sito e organizzazione delle directory</u> .....	9
<b><u>6 Realizzazione di semplici pagine con editor visuali</u></b> .....	<b>10</b>
<u>6.1 Che programma usare</u> .....	10
<u>6.2 La composizione delle pagine</u> .....	10
<u>6.3 La creazione dei collegamenti ipertestuali</u> .....	11
<u>6.4 Cos'altro si potrebbe fare</u> .....	12
<b><u>7 Concetti di base relativi all'HTML</u></b> .....	<b>13</b>
<u>7.1 Terminologia</u> .....	13
<u>7.2 Gli elementi più importanti</u> .....	14
<b><u>8 Argomenti avanzati</u></b> .....	<b>15</b>
<u>8.1 Le immagini</u> .....	15
<u>8.1.1 Formati di immagine</u> .....	15
<u>8.1.2 Terminologia specifica</u> .....	15
<u>8.1.3 Inserimento delle immagini in una pagina web</u> .....	15
<u>8.1.4 Immagini sensibili (mappe)</u> .....	16
<u>8.2 Meta informazioni</u> .....	16
<u>8.2.1 Attributi specifici di alcuni marcatori</u> .....	16
<u>8.2.2 Meta intestazioni</u> .....	16
<u>8.2.3 Il file robots.txt</u> .....	16
<u>8.3 Elaborazione sul lato server e sul lato client</u> .....	16
<u>8.3.1 Moduli (forms)</u> .....	17
<u>8.3.2 Invio dei dati ed elaborazione sul lato server</u> .....	17
<u>8.3.3 Applicazioni CGI remote</u> .....	17
<u>8.4 Elaborazione sul lato client</u> .....	18

# Indice

<b><a href="#">9 I fogli di stile.....</a></b>	<b>19</b>
<a href="#">9.1 Introduzione.....</a>	19
<a href="#">9.2 I fogli di stile a cascata.....</a>	19
<a href="#">9.3 Fogli di stile: concetti fondamentali.....</a>	19
<a href="#">9.4 Cosa si fa in un foglio di stile.....</a>	20
<a href="#">9.5 Tipi di selettori.....</a>	20
<a href="#">9.6 Proprietà definibili.....</a>	20
<a href="#">9.7 Dove si definiscono gli stili.....</a>	20
<a href="#">9.8 Particolarità di CSS2.....</a>	20
<b><a href="#">10 Accessibilità, usabilità, credibilità.....</a></b>	<b>22</b>
<a href="#">10.1 Accessibilità.....</a>	22
<a href="#">10.2 Usabilità.....</a>	22
<a href="#">10.3 Credibilità.....</a>	23
<b><a href="#">11 Consigli e raccomandazioni.....</a></b>	<b>24</b>
<a href="#">11.1 Cose da fare e da non fare.....</a>	24
<a href="#">11.1.1 In merito alla codifica.....</a>	24
<a href="#">11.1.2 In merito ad usabilità e accessibilità.....</a>	24
<a href="#">11.1.3 In merito all'organizzazione.....</a>	25
<a href="#">11.1.4 In merito allo stile.....</a>	25
<b><a href="#">12 Licenza Open Content.....</a></b>	<b>26</b>

# 1 Introduzione

Questi appunti hanno lo scopo di presentare idee e risorse relative alla creazione e alla gestione di pagine e siti web. Sono il frutto di vaste ricerche su web e della lettura di un discreto quantitativo di testi sull'argomento. Non sono da considerarsi un testo completo, ma solo un punto di partenza. Li ho preparati in occasione di diversi corsi per i quali, negli ultimi anni, mi è stato chiesto di preparare una "dispensa".

## 1.1 Obiettivi nella costruzione di pagine e siti web

Nel momento in cui ci si accinge a costruire una pagina web (o un intero "sito"), ci si possono porre diversi obiettivi, a volte parzialmente incompatibili:

- **usabilità** (chi accede al sito deve trovare facilmente ciò che sta cercando, senza attese inutili per foto, animazioni, ecc. che impediscono o rallentano la normale navigazione; se — esempio ricorrente — vuole acquistare tre confezioni di un prodotto, non deve compilare tre volte un modulo d'ordine; se compila un modulo con dei dati errati o mancanti, deve poter sistemare la cosa senza reinserire da capo tutti i dati...)
- **accessibilità** (deve essere possibile accedere ad un sito anche con calcolatori / browser non dell'ultima generazione, almeno per il recupero delle informazioni essenziali; utenti con impedimenti fisici non devono avere limitazioni nell'uso delle informazioni)
- **manutenibilità** (deve essere possibile modificare facilmente il sito web, aggiornandolo dal punto di vista dei contenuti o dell'aspetto grafico)
- **originalità** (il sito deve avere contenuti originali e una forma originale di presentazione)
- **coerenza nel "look and feel"** (il sito deve essere riconoscibile, diverso rispetto a quelli della "concorrenza", l'utente ci si deve sentire "a casa", ecc.)
- **fruibilità in gruppo** (in contrapposizione con il normale uso, un sito internet deve essere emozionalmente interessante per un gruppo di persone che ci accede insieme, un po' come al cinema)
- **credibilità** (le pagine devono apparire come pagine di una certa organizzazione, non solo esserlo)

## 1.2 Come realizzare delle pagine web

Le pagine web sono documenti di puro testo (*plain text*) e come tali possono essere scritte con qualsiasi editor testuale (come il "Blocco note" di Windows, "SimpleText" del Mac o "Kedit" di Linux). Il fatto che esse siano tali, e che siano scritte usando un formato standard ed aperto (l'**HTML**, *HyperText Markup Language*, ossia "linguaggio di marcatura degli ipertesti") è stato uno dei punti di forza che ha reso possibile l'esplosione del web negli anni '90.

Esistono molti modi per realizzare pagine web:

- utilizzare un editor testuale per scrivere il codice HTML
- utilizzare un editor visuale per "comporre" le pagine in modo simile a quanto si farebbe con un programma di videoscrittura
- comporre le pagine tramite modelli e sistemi di autocomposizione (direttamente su web oppure con un apposito programma)
- affidarsi a sistemi più o meno complessi ed evoluti di *web content management*, solitamente basati sulla memorizzazione dei contenuti in una base di dati

In questi appunti darò per scontato che si voglia approfondire la conoscenza dell'**HTML**, eventualmente al fine di utilizzare in maniera più consapevole le altre modalità indicate.

## 1.3 Dove trovare documentazione e riferimenti per conoscere l'HTML

Provate ad utilizzare un motore di ricerca scrivendo "HTML tutorial" o "guida HTML" e troverete moltissimi riferimenti a documenti sul linguaggio. Non tutti sono corretti al 100%, ma si può sempre imparare qualcosa. Quella che segue è una mia selezione delle cose migliori sulla rete in tema di HTML.

- [W3C](http://www.w3.org) [<http://www.w3.org>] (World Wide Web Consortium): sicuramente la fonte più autorevole, anche se spesso molto tecnica; una parte della documentazione è stata [tradotta in italiano](http://www.liberliber.it/biblioteca/w/world_wide_web_consortium/index.htm) [[http://www.liberliber.it/biblioteca/w/world\\_wide\\_web\\_consortium/index.htm](http://www.liberliber.it/biblioteca/w/world_wide_web_consortium/index.htm)] grazie a [Liber Liber](http://www.liberliber.it) [<http://www.liberliber.it>]
- [HTMLHelp](http://www.htmlhelp.org) [<http://www.htmlhelp.org>]: pagine curate da un'associazione di webdesigners, offrono diversi tutorials e guide alle raccomandazioni W3C
- [HTML.it](http://www.html.it) [<http://www.html.it>]: una risorsa italiana che contiene molti riferimenti all'HTML e ad argomenti correlati (Fogli di stile, Javascript, Java, ASP, PHP, ecc.)
- [Manuali.net](http://www.manuali.net) [<http://www.manuali.net>], [GuidaInLinea.com](http://www.guidainlinea.com) [<http://www.guidainlinea.com>] e [Apinformatica](http://www.apinformatica.com) [<http://www.apinformatica.com/biblio/>]: offrono link a documentazione consultabile on-line e, molto spesso, scaricabile per una più comoda lettura off-line
- [Webreference.com](http://www.webreference.com) [<http://www.webreference.com>]: una guida molto valida e sempre aggiornata su tutto ciò che c'è da sapere sul mondo del WWW;
- [WebMonkey](http://hotwired.lycos.com/webmonkey/) [<http://hotwired.lycos.com/webmonkey/>]: la guida di HotWired alla creazione di pagine web
- [Style-Master CSS Guide](http://www.westciv.com/style_master/) [[http://www.westciv.com/style\\_master/](http://www.westciv.com/style_master/)]: una guida fatta molto bene su un argomento un po' complesso: i fogli di stile e le raccomandazioni CSS1 e CSS2

Inoltre, segnalo che nella sezione esempi del mio sito web è presente una sezione dedicata alle [pagine web](http://www.tissino.it/esempi/pagineweb/) [<http://www.tissino.it/esempi/pagineweb/>].

Piccola nota: molto spesso, in documenti su web e in libri dedicati all'argomento, si trovano alcuni errori ed imprecisioni. Ad esempio:

- Riferimenti allo standard HTML 3.2 anziché al più recente e migliore 4.01
- Presentazione degli elementi per la formattazione prima degli elementi per la suddivisione logica della pagina
- Confusione tra elemento e marcitore (concetti distinti anche se parzialmente sovrapponibili)
- Presentazione errata di come devono essere utilizzati i marcatori (ad esempio, "*con <p> si inizia un nuovo paragrafo andando a capo*" piuttosto che "*<p> marca l'inizio di un nuovo paragrafo*")
- Confusione tra funzione logica di un marcitore e rappresentazione fisica normalmente utilizzata dai browser per la resa dell'elemento marcato
- Descrizione di trucchi per ottenere alcuni effetti (su alcuni browser) in contraddizione con il significato logico degli elementi o delle entità utilizzati (ad esempio **&ampnbsp**; descritto come "spazio fisico")
- Descrizione acritica di marcatori fuori standard (ad esempio, **marquee** o **bgsound**) o il cui uso è deprecato per palese illogicità (ad esempio, **center**)
- Assenza di riferimenti alla possibilità di usare stili e fogli di stile CSS
- Descrizione acritica di come si possono ottenere i riquadri (frames)
- Assenza di informazioni in merito alla corretta degradazione delle pagine e agli aspetti di accessibilità

Molte di queste cose probabilmente non saranno chiare adesso. Spero che lo diventino procedendo con la lettura (ammesso che riesca a finire di scrivere tutto ciò che ho in mente di dire).

## 2 La cassetta degli attrezzi

In generale, più un editor nasconde il codice HTML, più facile è realizzare la pagina e peggiori sono i risultati. L'HTML non è difficile, un po' di esperienza, la propria intelligenza ed un editor testuale sono molto meglio degli editor "faccio tutto io" di tipo WYSIWYG (*what you see is what you get*). Tra l'altro, con il web è per definizione impossibile ottenere ciò che si vede, visto che la resa delle pagine avviene su altri calcolatori, con caratteristiche (ad esempio risoluzione, profondità di colore, font disponibili) non note a priori.

Inoltre, per poter ottenere buoni risultati di solito è necessario integrare il codice HTML con codice di programmazione (ad esempio javascript o php), e per farlo è necessario conoscerlo a sufficienza.

### 2.1 Editor testuali

Segnalo alcuni software (per Windows: ma per Linux o per Mac ci sono software analoghi) che possono essere utili:

- [1st page](http://www.evrsoft.com/1stpage/) [<http://www.evrsoft.com/1stpage/>]: buon editor testuale, disponibile gratuitamente, che funziona a diversi livelli di complessità e che incorpora il validatore Tidy
- [Html-Kit](http://www.chami.com/html-kit) [<http://www.chami.com/html-kit>]: editor testuale per Windows che "incorpora" il validatore Tidy
- [Arachnophilia](http://www.arachnoid.com/arachnophilia) [<http://www.arachnoid.com/arachnophilia>]: un buon editor testuale molto facilmente personalizzabile
- [SciTE](http://www.scintilla.org/) [<http://www.scintilla.org/>]: un editor leggero, multiuso, multipiattaforma, rilasciato come software libero

### 2.2 Programmi di utilità

- [Tidy](http://www.w3.org/People/Raggett/tidy/) [<http://www.w3.org/People/Raggett/tidy/>]: un'utility per la validazione e la correzione del codice HTML 4.0
- [Style-Master](http://www.westciv.com/style_master/download/download.html) [[http://www.westciv.com/style\\_master/download/download.html](http://www.westciv.com/style_master/download/download.html)]: un buon editor di fogli di stile (contiene un'ottima guida alla progettazione degli stessi)

### 2.3 Validatori on-line

Anche se una pagina è scorretta, il browser tende a farcela vedere correttamente (ma cosa questo significhi esattamente dipende dal singolo browser). Per controllare la validità di una pagina, non basta vederla con il browser, è meglio affidarsi ad un validatore.

- [Validator](http://validator.w3.org/) [<http://validator.w3.org/>]: il validatore on-line del W3C verifica la correttezza delle proprie pagine e la validità dei link
- [Bobby](http://www.cast.org/bobby/) [<http://www.cast.org/bobby/>]: un validatore che permette di valutare l'accessibilità di un sito Internet
- [Lynx-it](http://www.slcc.edu/webguide/lynxit.html) [<http://www.slcc.edu/webguide/lynxit.html>]: uno strumento utile per vedere come appaiono le proprie (o altrui) pagine con Lynx [link non più valido]
- [See Lynx for yourself](http://www.trill-home.com/lynx/public_lynx.html) [[http://www.trill-home.com/lynx/public\\_lynx.html](http://www.trill-home.com/lynx/public_lynx.html)]: alcuni link a risorse che via telnet permettono di utilizzare Lynx per navigare il WWW: un'esperienza che tutti i disegnatori di pagine web dovrebbero fare

# 3 Quando le pagine sono pronte

## 3.1 Pubblicare le pagine

Questa è la parte più facile. Sono innumerevoli i siti che offrono spazio per le proprie pagine. Basta verificare cosa offrono esattamente, cosa chiedono in cambio, attivare lo spazio e lavorare di FTP (*File Transfer Protocol*, protocollo per la trasmissione di file). La trattazione di questo argomento esula dagli scopi di questo documento: qui basti dire che con un programma client FTP è possibile, previa autenticazione dell'utente mediante coppia nome/password, trasferire file dal proprio calcolatore ad un calcolatore remoto, che normalmente è quello che ospita le pagine web.

Il costo dell'hosting di un sito web è oramai irrisorio (meno di 20 euro all'anno). Ma se si vuole si può anche trovare uno spazio gratuito. Si vedano:

- [Gratis.it](http://www.gratis.it) [<http://www.gratis.it>]: una guida alle risorse gratuite presenti su Internet
- [The Free Site](http://www.thefreesite.com) [<http://www.thefreesite.com>]: un elenco di tutto ciò che serve per metter su un sito internet

## 3.2 Servizi aggiuntivi

Alcune risorse per migliorare il proprio sito con servizi aggiuntivi:

- [ScriptSearch](http://www.scriptsearch.com/) [<http://www.scriptsearch.com/>]: offre script di vario tipo
- [FreeFind](http://www.freefind.com) [<http://www.freefind.com>]: un motore di ricerca personalizzato
- [JavaScriptSource](http://www.javascriptsource.com) [<http://www.javascriptsource.com>]: una ricca collezione di codice JavaScript pronto per il taglia&incolla

# 4 Cenni all'HTTP

## 4.1 Definizioni di base

*connessione*

un circuito virtuale a livello trasporto stabilito tra due programmi al fine di comunicare

*messaggio*

l'unità di base della comunicazione HTTP: è una sequenza di ottetti che segue la sintassi definita

*risorsa*

un oggetto o un servizio identificabile con un'URI, disponibile eventualmente in diverse forme o varianti

*URI*

Uniform Resource Identifier (identificatore uniforme della risorsa): una stringa di testo in grado di specificare in maniera non ambigua dove è locata (URL, uniform resource locator) o quale è (URN, uniform resource name).

*entità*

l'insieme di informazioni trasferite per effettuare una richiesta o dare una risposta; è composta da un'intestazione (header) e da un corpo (body)

## 4.2 La comunicazione HTTP

La comunicazione è iniziata da uno *user agent* (*browser*, *spider*, ecc.) che si rivolge, direttamente o tramite qualche intermediario (*proxy*, *gateway*, *tunnel*) ad un *server*.

## 4.3 La richiesta

HTTP/1.1. prevede i seguenti metodi per le richieste:

- OPTIONS
- GET
- HEAD
- POST
- PUT
- DELETE
- TRACE

Un esempio:

```
GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1
```

Normalmente la richiesta è accompagnata da informazioni aggiuntive:

```
GET /pub/WWW/TheProject.html HTTP/1.1
Host: www.w3.org
```

Informazioni aggiuntive possono essere:

- Accept
- Accept-Charset
- Accept-Encoding
- Accept-Language
- Authorization
- From

- Host
- If-Modified-Since
- If-Match
- If-None-Match
- If-Range
- If-Unmodified-Since
- Max-Forwards
- Proxy-Authorization
- Range
- Referer
- User-Agent

## 4.4 La risposta

La risposta del server contiene:

- una riga di stato
- alcune righe di intestazione con informazioni aggiuntive
- il documento richiesto

Ad esempio:

```
HTTP/1.1 200 Ok
Server: Apache
Content-Type: text/html
Content-Length: 2098
Last-Modified: Wed, 30 Sep 1998 16:23:40 GMT
```

La riga di stato a sua volta contiene:

- Versione del server
- Codice dello stato
- Descrizione dello stato

Gli stati possibili sono codificati come segue:

- **Ixx**: Informazione (es. 100 Continue)
- **2xx**: Success (es. 200 Ok)
- **3xx**: Redirection (es. 301 Moved Permanently)
- **4xx**: Client Error (es. 401 Unauthorized, 402 Payment Required, 403 Forbidden, 404 Not Found)
- **5xx**: Server Error (es. 501 Not Implemented, 505 HTTP Version not supported)

Tra le intestazioni troviamo:

- Content-Base
- Content-Encoding
- Content-Language
- Content-Length
- Content-Location
- Content-MD5
- Content-Range
- Content-Type
- ETag
- Expires

- Last-Modified

## 4.5 Ulteriori informazioni

Si può provare a visualizzare cosa succede dietro le quinte tramite una pagina web da me [appositamente realizzata](#) [<http://www.tissino.it/snoopy/>].

# 5 Progettazione delle pagine web

## 5.1 Documentazione su web

Una volta appresa la tecnica di creazione delle pagine, giunge il momento di pensare a progettare un sito web accessibile, navigabile, comprensibile. Un mix di forma e contenuto a cui val la pena di riflettere leggendo qualche buon documento sul tema.

- [Anybrowser](http://www.anybrowser.org/campaign/) [<http://www.anybrowser.org/campaign/>]: una campagna per evitare che le pagine web siano visibili solo con un determinato tipo di browser; contiene molti link a risorse interessanti
- [Accessible Design Guide](http://www.anybrowser.org/campaign/abdesign.shtml) [<http://www.anybrowser.org/campaign/abdesign.shtml>]: una buona guida, curata da Cari D. Burstein, sulla realizzazione di pagine web facilmente accessibili da tutti
- [UseIt](http://www.useit.com) [<http://www.useit.com>]: le fondamentali pagine di Jacob Nielsen, guru del web design
- [Mestiere di scrivere](http://www.mestieredisscrivere.com/) [<http://www.mestieredisscrivere.com/>]: un sito curato da Luisa Carrada, copywriter, con un ottimo testo sull'arte di scrivere per il web (il quale si concentra sui contenuti più che sugli aspetti tecnici)
- [Good Documents](http://www.gooddocuments.com) [<http://www.gooddocuments.com>]: contiene alcuni buoni consigli (con degli esempi efficaci) sulla pubblicazione di documenti su Internet / Intranet
- [Usabilità dei siti web](http://www.hyperlab.net/ergonomia/dispensa/index.html) [<http://www.hyperlab.net/ergonomia/dispensa/index.html>]: l'applicazione delle regole dell'ergonomia al mondo dei siti web
- [Apogeonline](http://www.apogeonline.com) [<http://www.apogeonline.com>]: una webzine che contiene spesso utili suggerimenti e notizie relative alla produzione di pagine web
- [Documents about the WWW written or recommended by Jukka Korpela](http://www.cs.tut.fi/~jkorpela/www.html) [<http://www.cs.tut.fi/~jkorpela/www.html>]: contiene link a molti interessanti documenti tecnici attinenti il WWW
- [Usabile.it](http://www.usabile.it) [<http://www.usabile.it>]: un sito web tutto italiano dedicato al tema dell'usabilità
- [Siti che funzionano](http://www.sitichefunzionano.it) [<http://www.sitichefunzionano.it>]: il sito di supporto per l'omonimo libro di Sofia Postai

## 5.2 Lo spazio dei nomi in un sito web

### 5.2.1 Estensione dei nomi dei file

Prima considerazione: l'estensione di un file pubblicato su web è significativa per il browser solo nel caso in cui il webserver non gli comunichi il tipo/sottotipo MIME (per le pagine web, text/html). Diverso il caso di file locali, non forniti da un webserver (file aperti direttamente dal disco rigido o da cd-rom).

I file HTML dovrebbero avere estensione **.html**, se è previsto che il browser li apra direttamente (senza richiederli ad un server web tramite http). Sotto Windows, molti editor usano l'estensione **.htm**, che sconsiglio.

Per le immagini vale lo stesso discorso. Normalmente si usano le estensioni **.jpeg**, **.jpg**, **.gif**, **.png** (ovviamente a seconda del formato).

### 5.2.2 Nomi dei file e delle cartelle

I nomi dei file possono contenere solo lettere dell'alfabeto inglese (vietate le lettere accentate dell'italiano), cifre numeriche e il segno di percentuale. Considerati questi nomi:

- mia pagina.html (non corretto, c'è uno spazio)
- attività.html (non corretto, c'è una lettera accentata)
- Miapagina.html (corretto, ma attenzione alle "sensibilità alla differenza maiuscolo/minuscolo")
- miaPagina.html (corretto, ma attenzione alle "sensibilità alla differenza maiuscolo/minuscolo")
- miapagina.html (corretto)

- pagina2.html (corretto)

Le stesse considerazioni valgono per i nomi delle cartelle.

### 5.2.3 File principale di una cartella web

Un sito web è normalmente organizzato in cartelle (directories). Esiste almeno una directory radice dove sono contenuti i suoi file.

I server web sono configurati per offrire una pagina web iniziale in assenza di indicazione esplicita. Solitamente, questa pagina si chiama **index.html** oppure (in ambito Microsoft Internet Information Server), **default.htm**. Converrà utilizzare tale nome per definire la pagina iniziale del proprio spazio web.

I link ad una cartella web (nell'esempio, la cartella **fiori**) potranno essere fatti in uno dei seguenti modi:

- **fiori** (con il nome della cartella, senza barra finale): utile per la stampa
- **fiori/** (con il nome della cartella, con la barra finale): utile per pagine offerte da un server web
- **fiori/index.html** (con il nome della cartella e l'indicazione esplicita del file da prelevare): utile per pagine da aprire direttamente con il browser

### 5.2.4 Mappa del sito e organizzazione delle directory

Supponiamo che l'associazione *Inventata* progetti uno spazio web accessibile all'indirizzo <http://www.qualchespazio.net/inventata>.

L'associazione potrebbe voler organizzare i propri contenuti in diverse sezioni:

- attività svolte
- attività programmate
- spazio soci
- statuto e informazioni

Le quattro sezioni potrebbero essere organizzate in quattro directory...

- **activitasvolte**
- **aktivitaprogrammate**
- **spaziosoci**
- **statuto**

... o in quattro files:

- **activitasvolte.html**
- **aktivitaprogrammate.html**
- **spaziosoci.html**
- **statuto.html**

Ci sono vantaggi e svantaggi nelle due modalità. Qualunque sia la decisione, bisognerebbe cercare di essere coerenti.

# 6 Realizzazione di semplici pagine con editor visuali

## 6.1 Che programma usare

Ci sono moltissimi programmi che permettono la creazione e la modifica visuale di pagine HTML. Tra questi, alcuni blasonati programmi di videoscrittura, programmi dedicati e nati con questo scopo, programmi di grafica e desktop publishing arricchiti di funzionalità di esportazione in HTML, ecc.

In questi esempi farò riferimento all'HTML editor che fa parte della suite [OpenOffice.org](http://www.openoffice.org) [<http://www.openoffice.org>], se non altro per dare il mio modestissimo contributo a supporto degli sforzi di chi realizza software libero di qualità. Le considerazioni qui fatte varranno comunque anche per moltissimi altri programmi (magari solo con qualche variazione terminologica).

## 6.2 La composizione delle pagine

Si supponga di voler creare alcune pagine web con lo scopo di illustrare alcuni termini informatici che si usano nel campo, già che ci siamo, del world wide web.

Avremo una pagina iniziale con lo scopo di illustrare il progetto (chi ha scritto le pagine, cosa ci si può trovare, ecc.). Di solito la pagina iniziale viene chiamata *index.html*, ed è la pagina che viene manda al programma client che non specifica il documento esatto nella richiesta, ma solo il nome della directory (cartella). Fa eccezione un server web fornito da una nota casa di software che produce un noto sistema operativo, il quale si aspetta come nome della pagina iniziale *default.htm* (ma la cosa è comunque configurabile).

Un esempio di contenuti potrebbe essere il seguente:

Il mio glossario dei termini informatici

Questo documento ha lo scopo di illustrare alcuni termini informatici in cui spesso ci si imbatte quando si affronta il mondo del world wide web. Ovviamente, per spiegare un termine molto spesso è necessario usarne degli altri, per cui questo glossario è per sua natura ipertestuale.

I termini da cui suggerisco di partire sono i seguenti:

- \* www
- \* http
- \* html
- \* ipertesto
- \* link
- \* browser

L'esempio qui sopra è stato mantenuto volutamente "grezzo". Il contenuto potrà essere personalizzato come meglio si riterrà opportuno. In pratica, con *OpenOffice.org HTML editor* sarà sufficiente digitare il testo, formattarlo e salvarlo in una cartella appositamente creata (ricordandosi quale e dove essa è) con il nome *index.html*. Non ci si curi, in questa fase, della formattazione delle parole che costituiranno un collegamento ipertestuale: verrà da sé quando quest'ultimo sarà definito.

Si noti che sarebbe opportuno dare un titolo al documento (cosa diversa dal nome del file: il titolo apparirà nella barra del titolo del browser, nel menù dei segnalibri o --- termine errato --- *preferiti*, nei paginoni dei motori di ricerca) tramite menù **File / Proprietà**. Per i documenti tradizionali questo titolo non è molto importante, ma per le pagine web lo è.

La pagina relativa al www potrebbe chiamarsi *www.html*: essa avrà più o meno il seguente contenuto:

WWW - Il mio glossario

Il termine WWW è l'acronimo di World Wide Web (letteralmente, *ragnatela mondiale*), ossia l'insieme di documenti ipertestuali messi a disposizione dai server web tramite il protocollo **HTTP** e in forma di testi **HTML**. (...)

Curiosità: in inglese *worldwide* è una parola unica, ma è stata spezzata per rendere più carina la sigla.

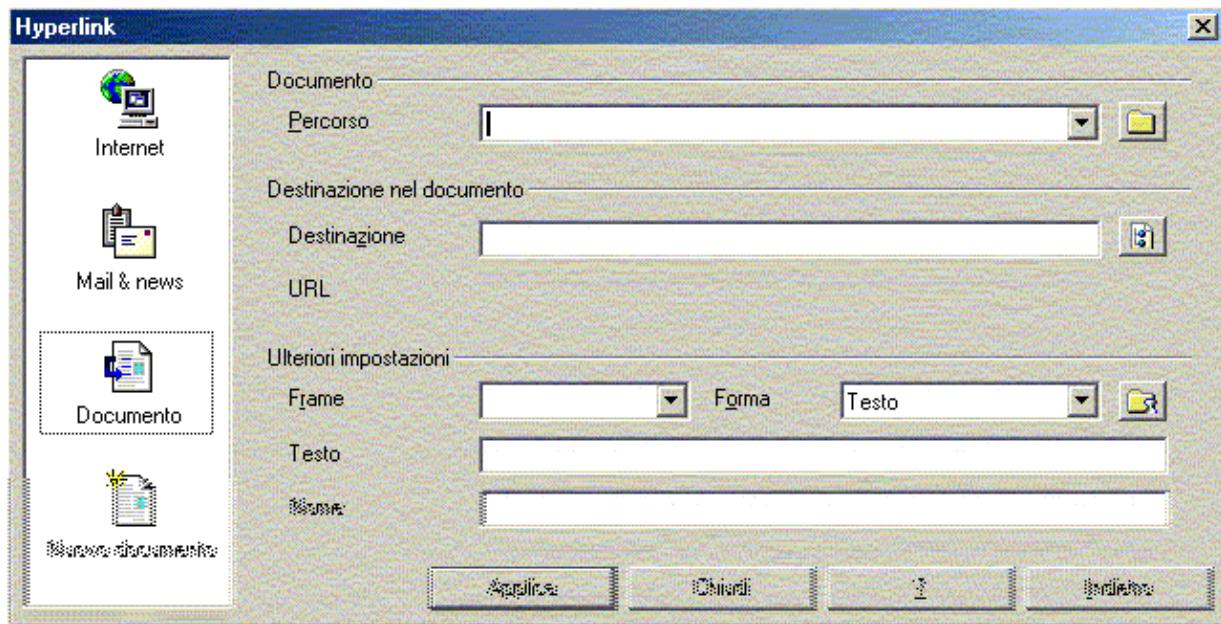
-----

### Indice del glossario

Via di seguito con le altre pagine, create sulla falsa riga di questa, e possibilmente con la medesima formattazione (un po' di coerenza non guasta mai).

## 6.3 La creazione dei collegamenti ipertestuali

Una volta scritte le pagine, si tratta di creare i collegamenti ipertestuali tra uno e l'altro. Si selezioni la parola **HTTP** nella pagina *www.html*; si scelga la voce di menù **Inserisci / Hyperlink**. Ci si aprirà una finestra di dialogo in cui potremo decidere che tipo di collegamento stiamo creando:



Se si vuole creare un collegamento verso qualche risorsa su Internet (pagine web di altri, siti ftp, ecc.), nella colonna di sinistra si sceglierà la voce **Internet**.

Se si vuole un collegamento ipertestuale che faccia aprire il programma che l'utente ha configurato per la gestione della posta o dei gruppi di discussione, si sceglierà la voce **Mail & news**.

Se si vuole un collegamento ad un altro documento locale (come nel nostro caso), si sceglierà la voce **Documento**. Si noti che i collegamenti di questo tipo vengono chiamati relativi se si indica la posizione del file collegato in relazione alla posizione del file su cui è presente il collegamento. I collegamenti relativi funzioneranno correttamente anche spostando (insieme) i file su altri calcolatori.

Se si vuole creare immediatamente un nuovo documento, si sceglierà la voce **Nuovo documento**. (Questo ero indeciso se scriverlo o no, NdA).

La parte di destra della finestra di dialogo andrà poi completata seguendo il proprio fiuto e buon senso.

## 6.4 Cos'altro si potrebbe fare

Una volta realizzati i collegamenti ipertestuali, si provi a navigare l'ipertesto con un *browser* e, magari, a farlo navigare a qualcun altro, segnandosi perplessità e difficoltà incontrate nel passare da una pagina ad un'altra. La difficoltà nella realizzazione degli ipertesti non è tecnica, ma relativa a contenuti e microcontenuti. (Detto in altri termini: è molto più difficile progettare bene un ipertesto e avere chiaro cosa si vuole ottenere che trovare la strada per ottenerlo).

Quando si è raggiunto un risultato soddisfacente, si provi a dare un'occhiata al codice sorgente (HTML) dei propri documenti.

# 7 Concetti di base relativi all'HTML

## 7.1 Terminologia

L'HTML è un linguaggio di **marcatura**. Nel testo, si *marca* l'inizio e la fine (oppure la presenza) di qualcosa (un paragrafo, un collegamento ipertestuale, un titolo, ecc.).

Le "cose" marcate si chiamano **elementi**, i codici scritti per marcarli si chiamano **marcatori**.

Ad esempio, nel codice:

```
<p>Il mio gatto è grigio</p>
```

è presente un paragrafo ("Il mio documento"), marcato da un marcitore di apertura (*<p>*) ed un marcatore di chiusura (*</p>*).

Si dovrebbe distinguere il concetto di marcatore da quello di elemento. Un elemento può esistere anche senza essere marcato esplicitamente (ad esempio, il corpo del documento, *body*).

Alcuni marcatori non prevedono chiusura. Ad esempio:

```

```

che introduce un'immagine.

Alcuni marcatori prevedono la specificazione di **attributi**, che ne precisano il significato (nell'esempio precedente, quale immagine deve essere inserita). Gli attributi sono specificati con la coppia nome/valore, e con il valore preferibilmente scritto tra virgolette.

Per alcuni marcatori è obbligatoria la chiusura, per altri è facoltativa. È preferibile inserire sempre il marcatore di chiusura.

Marcatori e nomi degli attributi non sono sensibili alla differenza maiuscolo / minuscolo (ma ci sono delle eccezioni), mentre i valori degli attributi di solito lo sono.

Un elemento può contenere al proprio interno un altro elemento. Si parla in questo caso di elementi **annidati**. Ad esempio:

```
<p>Proprio un <em>bel</em> gatto</p>
```

Per riprodurre i caratteri accentati della lingua italiana (in generale, quelli diversi dai 26 caratteri dell'alfabeto inglese), è necessario specificare quale **codifica** si intende usare (ISO-8859-1 nel caso dei caratteri dell'Europa occidentale). Questa specificazione va fatta nell'ambito delle meta-intestazioni che vengono lette dal browser dell'utente.

Per riprodurre alcuni caratteri riservati (ad esempio il simbolo ">") si fa uso di specifiche **entità** predefinite: ad esempio, `&gt;`, `&lt;`, `&amp;`.

Gran parte degli elementi si possono dividere in due categorie: elementi **di blocco** ed elementi **in linea**. I primi (almeno nelle lingue occidentali) presuppongono uno spostamento alla riga successiva (ossia l'andare a capo), i secondi invece presuppongono uno spostamento in avanti. Fra gli elementi di blocco, vi sono i titoli (*h1*, *h2*, ecc.), i paragrafi (*p*), le tabelle (*table*). Fra gli elementi in linea, vi sono i testi enfatizzati (*em*), fortemente enfatizzati (*strong*), le immagini (*img*).

## 7.2 Gli elementi più importanti

Gran parte di un corso sull'HTML si basa sul capire quali marcatori e quali attributi vanno usati per ottenere un determinato risultato.

Gli elementi che sicuramente vanno conosciuti sono i seguenti:

- per la definizione della pagina: *html, head, title, meta, body, frameset, frame*
- per l'impostazione dei collegamenti ipertestuali: *a*
- per la struttura del contenuto della pagina: *h1..h6, p, link*
- per la definizione delle tabelle: *table, tr, th, td*
- per la gestione della presentazione grafica: *font* (deprecato), *div, span, b, i, style*
- per le immagini, le animazioni, gli oggetti multimediali: *img, applet, object, script*

# 8 Argomenti avanzati

## 8.1 Le immagini

### 8.1.1 Formati di immagine

Le immagini si dividono in due grandi categorie: **vettoriali** e **bitmap**. Per diversi motivi, su web si usano normalmente immagini bitmap compresse, registrate in uno di questi formati:

- **GIF** (256 colori, compresse per righe, possibile trasparenza)
- **JPEG** (milioni di colori, diversi livelli di compressione)
- **PNG** (milioni di colori, gestione migliore delle trasparenze e delle gamme di colori, supportato solo dai browser di quarta generazione)

### 8.1.2 Terminologia specifica

Dovendo gestire immagini, è bene avere familiarità con termini quali

- **risoluzione** (quanti pixel per pollice)
- **profondità** (quanti bit per memorizzare il colore di un pixel)
- **anti-aliasing** (tecnica per evitare le scalettature tramite la sfumatura di alcuni pixel)
- **dithering** (tecnica per cercare di riprodurre un colore mediante l'accostamento ripetuto di pixel di colore diverso)

### 8.1.3 Inserimento delle immagini in una pagina web

Il codice HTML da utilizzare per inserire un'immagine in una pagina web è il seguente: ``. Il percorso può essere omesso nel caso in cui il file dell'immagine risieda nella stessa cartella del file HTML in cui è richiamato. Per il percorso valgono le normali considerazioni (riferimenti relativi o assoluti, ecc.).

È estremamente consigliabile indicare anche, come attributi del marcatore img, le dimensioni in pixel dell'immagine: ``, in modo da informare il browser sullo spazio da riservare all'immagine stessa nell'ambito della pagina. Il browser ridimensiona l'immagine seguendo le indicazioni degli attributi width e height. Salvo pochissime eccezioni, l'immagine dovrebbe però essere inserita con le sue dimensioni reali; altrimenti:

- nel caso di rimpicciolimento, il browser potrebbe "impastare" l'immagine, e si trasmetterebbe un file più grande del necessario
- nel caso di ingrandimento, il browser mostrerebbe probabilmente dei gradoni annullando l'effetto antialiasing

Un altro attributo importante è **alt**, che serve a specificare un testo descrittivo dell'immagine ad uso di chi non può vedere la stessa (o perché non vedente, oppure perché usa un browser configurato per non caricare le immagini o che non supporta le immagini). Uso: ``. Purtroppo, i browser più recenti, travisando il significato dell'attributo, visualizzano il suo valore in un rettangolino giallo al passaggio del mouse sopra l'immagine.

## 8.1.4 Immagini sensibili (mappe)

È possibile inserire anche delle immagini con la caratteristica di attivare collegamenti ipertestuali diversi a seconda del punto dove, su di esse, si clicca. Si parla in tal caso di **immagini sensibili**. L'elaborazione del punto su cui si è fatto click può essere svolta dal server o dal client. Si vedano un paio di esempi nella [mia raccolta](http://www.tissino.it/esempi/pagineweb/) [<http://www.tissino.it/esempi/pagineweb/>].

## 8.2 Meta informazioni

In un documento HTML possono essere contenute informazioni da non rappresentare all'utente (tipicamente), ma utili per l'indicizzazione da parte dei ragni (spiders) dei motori di ricerca, o per un maggior controllo da parte del browser.

### 8.2.1 Attributi specifici di alcuni marcatori

- attributo ***lang*** (es. <html lang="it">)
- attributo ***summary*** (es. <table summary="questa tabella illustra...">)
- attributo ***title*** (es. <a href="..." title="Appunti Internet di Loris Tissino">)

### 8.2.2 Meta intestazioni

- ***keywords*** (es. <META NAME="keywords" CONTENT="web, cgi, http, php, asp">)
- ***description*** (es. <META NAME="description" CONTENT="Pagina dedicata all'analisi delle tecniche per l'elaborazione di dati sul lato server">)
- ***robots*** (es. <META NAME="robots" CONTENT="noindex,nofollow,noarchive">)
- ***robots*** (es. <META NAME="robots" CONTENT="index,follow,archive">)
- ***tipo di contenuto*** (es. <META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">)
- ***reindirizzamento*** (es. <META HTTP-EQUIV="refresh" CONTENT="5; url=http://someurl.url/page">)

### 8.2.3 Il file robots.txt

Oltre che nelle singole pagine, è possibile dare indicazioni ai motori di ricerca anche tramite un file, *robots.txt*, da porre nella radice dello spazio web. Ulteriori informazioni nelle pagine dedicate all'argomento su [RobotsTxt.org](http://www.robotstxt.org/) [<http://www.robotstxt.org/>].

## 8.3 Elaborazione sul lato server e sul lato client

I termini **statico** e **dinamico** sono usati spesso in maniera poco chiara nella documentazione relativa alle pagine web. Infatti:

Si può considerare "dinamico":

- ciò che si muove (lampeggia, scorre...)
- ciò che si modifica (è diverso, si aggiorna...)

Si può considerare "statico":

- ciò che sul server è già pronto per essere inviato al client
- ciò che giunto al client appare immobile

Per evitare confusione, io preferisco parlare di ***pagine web dinamiche*** (elaborazione sul lato client) e di ***pagine web generate dinamicamente*** (elaborazione sul lato server).

### 8.3.1 Moduli (forms)

Il browser gestisce moduli che l'utente può compilare. Sono utilizzabili alcuni tipi di campi visibili in quest'[esempio](http://www.tissino.it/esempi/pagineweb/moduli_base.php?) [[http://www.tissino.it/esempi/pagineweb/moduli\\_base.php?](http://www.tissino.it/esempi/pagineweb/moduli_base.php?)] (altri tipi di campi sono stati introdotti con HTML 4.0 ma hanno ancora scarso supporto da parte dei browser).

### 8.3.2 Invio dei dati ed elaborazione sul lato server

I dati possono essere inviati al server (o meglio, all'URL indicata con l'attributo *action* del marcitore *form*) in due modi:

- **GET**: i dati vengono codificati nell'URL, nella sezione che segue il punto interrogativo, denominata *query string* (es.  
`http://www.fiabe.it/cgi-bin/elabora?nome=pinocchio&naso=lungo`);  
l'applicazione invocata legge i dati da una variabile d'ambiente impostata dal webserver
- **POST**: i dati vengono codificati come sequenza di caratteri ed inviati dal browser al server senza che di essi compaia traccia nella *query string*; il server li gira all'applicazione **CGI** (*Common Gateway Interface*) nel suo *standard input*

Il metodo GET va utilizzato quando i dati da inviare sono limitati e non contengono informazioni riservate (ad esempio, password); è inoltre l'unico metodo utilizzabile per rendere disponibili documenti tramite un collegamento ipertestuale e non in seguito alla compilazione di un modulo.

Sul lato server, i dati tradizionalmente venivano (e vengono tuttora in molti casi) elaborati da applicazioni CGI scritte in linguaggio PERL (ma qualsiasi linguaggio di programmazione con supporto degli standard input ed output può essere utilizzato): l'applicazione, dopo aver elaborato i dati, genera un documento testuale (ad esempio la pagina web di risposta) oppure un file binario (ad esempio, un'immagine GIF, o un documento PDF) nel suo standard output e lo passa al webserver, il quale lo "gira" al browser.

Per chi ha studiato il linguaggio Pascal e/o il linguaggio C/C++, potrebbe essere istruttivo vedere qualche esempio di applicazioni CGI in questi due linguaggi. Si vedano in proposito: [CGI in Free Pascal](http://www.sciences.univ-nantes.fr/info/enseignement/deug/STPI/stpi1/freepascal/user/node10.html) [<http://www.sciences.univ-nantes.fr/info/enseignement/deug/STPI/stpi1/freepascal/user/node10.html>] e [CGI in C](http://www.cs.tut.fi/~jkorpela/forms/cgic.html) [<http://www.cs.tut.fi/~jkorpela/forms/cgic.html>].

La metodologia CGI ha alcuni problemi di efficienza e di complessità d'uso. Sono state pertanto sviluppate metodologie alternative di elaborazione dei dati sul lato server:

- **FastCGI**: applicazioni CGI che non vengono invocate ed inizializzate ad ogni esecuzione, ma rimangono in attesa come dei demoni (con tempi di risposta molto più ridotti); possono inoltre essere eseguite su macchine diverse rispetto a quella dove risiede il web server
- **Servlet Java**: applicazioni Java eseguite sul server, beneficiano della pulizia del codice Java e del suo essere un linguaggio *Object-Oriented* puro
- Linguaggi di *scripting sul lato server*: tecniche per inserire, con marcatori appositi, istruzioni direttamente all'interno di speciali pagine web, che prima di essere spedite al client vengono elaborate sul lato server; tra le tecnologie più diffuse, citiamo **PHP** (originariamente, *Personal Home Page*, ora *PHP: Hypertext Preprocessor*), **ASP** (*Active Server Pages*), **Allaire Cold Fusion**, **Pervasive Tango**

### 8.3.3 Applicazioni CGI remote

Non sempre con lo spazio web a disposizione viene offerta la possibilità di scrivere od utilizzare applicazioni CGI. Una soluzione in questi casi potrebbe essere quella di utilizzare servizi ospitati da server esterni. Alcune

risorse di questo tipo sono elencate in [CGI–resources.com](http://www.cgi-resources.com/Remotely_Hosted/) [[http://www.cgi-resources.com/Remotely\\_Hosted/](http://www.cgi-resources.com/Remotely_Hosted/)].

### 8.4 Elaborazione sul lato client

La dinamicità sul lato client può essere ottenuta con queste tecniche (a volte combinate tra loro):

- immagini animate
- javascript
- fogli di stile (css)
- html dinamico (dhtml)
- applets java
- animazioni flash
- audio / video

Alcuni esempi sono disponibili nella mia [raccolta](http://www.tissino.it/esempi/pagineweb/) [<http://www.tissino.it/esempi/pagineweb/>].

# 9 I fogli di stile

## 9.1 Introduzione

Molto spesso chi realizza le pagine web desidera un controllo fine su come la pagina apparirà con il browser dell'utente. Un controllo assoluto non è né possibile né desiderabile, ma con un po' di buon senso si possono definire degli stili per ottenere effetti grafici carini e che "degradano bene" (se un browser non supporta gli stili, renderà comunque in maniera decente il testo di base).

Da notare che per un controllo assoluto di come la pagina deve apparire, esistono altri strumenti più appropriati (in particolare, il formato di file PDF, portable document format, leggibile ad esempio con Acrobat Reader).

## 9.2 I fogli di stile a cascata

I fogli di stile a cascata (*CSS, Cascading Style Sheets*) sono uno strumento per separare il contenuto dalla forma di presentazione. Esistono due livelli di CSS, definiti in apposite raccomandazioni del W3C:

- **CSS1** (fine 1996) – buon supporto da Netscape Navigator 4.x, Internet Explorer 4.x e 5.0, Opera 3.5
- **CSS2** (maggio 1998) – supporto ancora scarso da parte dei browser

Un foglio di stile è un file di testo che si presenta come in questo esempio:

```
body
{font-family: verdana, "minion web", helvetica, sans-serif;
font-size: 1em;
text-align: justify}

h1
{font-family: "soft hits";
font-size: 1.3em}

code
{font-family: courier, sans-serif;
font-size: 1em}

.more
{background-color: #003333;
border-width: thin;
border-color: black;
color: white;
font-family: verdana, geneva, sans-serif;
font-size: .9em;
vertical-align: text-bottom}
```

## 9.3 Fogli di stile: concetti fondamentali

- **Cascata:** un foglio di stile si può basare su un altro foglio di stile (ad esempio quello preparato per le pagine web di un dipartimento aziendale può basarsi su quello generico aziendale, limitandosi a definire le differenze)
- **Ereditarietà:** ogni elemento in una pagina web è inserito all'interno di un altro elemento (ad esempio, un paragrafo è dentro il corpo) e ne *eredita* le caratteristiche di formattazione, se non diversamente specificato

## 9.4 Cosa si fa in un foglio di stile

1. **selezione**: si identifica un elemento (paragrafo, elemento di lista, cella di tabella, ecc.)
2. **dichiarazione**: si descrive come il browser deve rendere l'elemento, in termini di valori per le varie proprietà possibili

## 9.5 Tipi di selettori

- Selettori di **elementi HTML** (es. un paragrafo, un titolo di livello 1, una cella di tabella)
- Selettori di **classi** (es. un paragrafo di classe "domanda")
- Selettori di **ID** (es. il paragrafo con id="d12")
- Selettori **contestuali / discendenti** (es. un elemento strong dentro un titolo h1)
- Selettori di **pseudo classi relative ai link** (nei diversi stati)
- Selettori di **pseudo elementi** (es. prima linea, primo carattere)
- Selettori di **gruppi**

## 9.6 Proprietà definibili

Nell'ambito della dichiarazione, si possono definire le seguenti categorie di proprietà:

- **Text Style** (color, font-weight, font-family, font-size, font-size-adjust, font-style, text-decoration, text-transform, text-shadow\*, font-stretch\*)
- **Text Layout** (letter-spacing, word-spacing, line-height, vertical-align, text-indent, text-align, direction\*, unicode-bidi\*)
- **Background** (background-color, background-image, background-attachment, background-repeat, background-position )
- **Border** (border-top-width, border-right-width, border-bottom-width, border-left-width, border-width, border-color, border-style, border-top, border-right, border-bottom, border-left border)
- **Margin** (margin-top, margin-left, margin-bottom, margin-right, margin)
- **Padding** (padding-top, padding-left, padding-bottom, padding-right, padding)
- **Page Layout** (position, top, left, bottom, right, width, min-width, max-width, height, z-index, visibility, overflow, float, clear, clip)
- **User Interface** (cursor, outline)

## 9.7 Dove si definiscono gli stili

Gli stili possono essere definiti in tre modi:

- a livello dell'elemento (**CSS in linea**)
- nell'intestazione del documento (**CSS incorporati**)
- in un documento separato e collegato (**CSS esterni**)

## 9.8 Particolarità di CSS2

Con CSS2 sono state introdotte alcune caratteristiche interessanti:

- definizione di regole riferite al mezzo (@**media rules**), che valgono solo per quel mezzo (ad esempio, la stampante, il monitor di un PC, un sintetizzatore vocale, ecc.)
- ammissione di regole riferite alla pagina (@**page rules**), intesa nel senso di foglio stampato
- è stata estesa (e resa personalizzabile) la **generazione di contenuto** (es. virgolette, espressioni tipo "inizio" e "fine", ecc.)

Sono stati aggiunti nuovi tipi di selettori:

- Selettori di ***pseudo classi dinamiche*** (es. *hover, focus*)
- Selettori di ***pseudo classi linguistiche*** (es. virgolette nelle diverse lingue)
- Selettori di "***figli***" (discendenti diretti)
- Selettori di "***primo figlio***" (primo discendente diretto in ordine d'uso)
- Selettori di ***elementi adiacenti***
- Selettori di elementi con specifici attributi (es <a href="...">)
- Selettori di ***righe e colonne*** (di tavelle)

È stato introdotto il concetto di ***webfont***

- specificazione precisa del font da usare
- indicazione per la sintesi di font da parte del client
- scaricabilità dei font
- resa progressiva dei font

# 10 Accessibilità, usabilità, credibilità

## 10.1 Accessibilità

Le pagine web sono **accessibili** quando possono essere "lette" e "comprese" da tutti gli utenti in potenziale difficoltà:

- non vedenti
- ipovedenti
- tecnicamente non "aggiornate" (monitor b/n, browser non dell'ultima generazione, connessione lenta)
- culturalmente inadeguate (es. stranieri)

Per realizzare pagine accessibili, si dovrà far attenzione, ad esempio, a:

- caratteri usati (tipi, scalabilità)
- codici di marcatura (tabelle con descrizione, immagini con testo alternativo)
- colori (profondità, contrasto)
- chiarezza linguistica (acronimi, formato date)
- elementi in movimento
- leggibilità dei grafici
- collegamenti ipertestuali non standard
- formato dei documenti "allegati"
- sommari e mappe
- buona degradazione / approccio incrementale

## 10.2 Usabilità

Un sito web è **usabile** quando l'utente per cui è progettato:

- trova facilmente e velocemente le informazioni che desidera
- compie con esso in maniera efficace, efficiente e soddisfacente le operazioni previste (es. ricerca, consultazione, stampa, conservazione)

Per un sito web usabile si farà attenzione a:

- utilità (serve?)
- facilità di apprendimento (es. con metafore)
- efficienza (quanti click? quanto tempo?)
- soddisfazione (gli utenti tornano?)

Alcuni errori tipici che vengono commessi:

- animazioni inutili
- pesantezza / lentezza
- interfacce misteriose
- validità per un particolare browser
- impossibilità di stampare / conservare / scaricare

Un sito web usabile andrà progettato considerando:

- struttura dei contenuti
- organizzazione dei link e dei percorsi di lettura
- spazio dei nomi e microcontenuti

- strumenti di ricerca e di feedback

... e andrà valutato mediante:

- test di usabilità
- analisi dei file di log

### 10.3 Credibilità

Un sito web è *credibile* quando l'utente è portato a fidarsi dei contenuti proposti, ad esempio in quanto:

- presenta contenuti aggiornati / revisionati
- presenta link corretti verso l'esterno
- "appare" professionale
- ha un URL appropriato

# 11 Consigli e raccomandazioni

## 11.1 Cose da fare e da non fare

Queste raccomandazioni vanno viste come spunti di discussione e di riflessioni piuttosto che come direttive. Si tratta di regole che in alcuni casi possono essere violate, l'importante è farlo consapevolmente...

Le cose che ho scritto qui sono volutamente sintetiche. Spero siano comunque comprensibili.

### 11.1.1 In merito alla codifica

- **No!** Usare marcatori di *formattazione fisica* per definire titoli ed elementi strutturali
- **No!** Impaginare il documento in una *grande tabella*, che solitamente non viene resa fino a quando non è ricevuta completamente
- **No!** Pubblicare immagini senza la specificazione della *dimensione in pixel* e senza attributo *ALT*
- **Sì...** Per i testi scritti in italiano (o in una lingua dell'Europa occidentale), utilizzare le apposite codifiche per le *lettere accentate* (o non presenti nell'alfabeto inglese) oppure specificare la codifica dei caratteri utilizzata (*charset*) con l'apposito elemento *meta*
- **Sì...** Controllare come appaiono le proprie pagine con *il maggior numero possibile di browser* (non fidandosi di ciò che si ottiene con il più diffuso), a diverse risoluzioni e con diverse impostazioni, e se possibile su diverse piattaforme
- **Sì...** Ricordare che negli *URL* c'è differenza tra maiuscole e minuscole, e che non sono ammessi spazi
- **Sì...** Usare i *fogli di stile* se si desidera formattazione del testo
- **Sì...** Linkare le directories con lo slash ("/") finale
- **Sì...** Fare attenzione alla scelta dei colori
- **Sì...** Avere *titoli* corretti per tutte le pagine
- **Sì...** Inserire *meta-informazioni* per rendere migliore e più comprensibile il lavoro dei motori di ricerca
- **Sì...** Per documenti e file grossi (indicativamente oltre i 100 kb) specificare la *dimensione*
- **Sì...** Inserire correttamente eventuale codice *JavaScript* in modo da farlo ignorare dai browser che non lo supportano
- **Sì...** Testare il codice JavaScript su più browser nelle diverse versioni e piattaforme
- **Sì...** *Validare* il codice HTML e CSS prodotto

### 11.1.2 In merito ad usabilità e accessibilità

- **No!** Usare i *frames* (cornici)
- **No!** Usare gli ultimi *effetti speciali* (es. VRML, o IFRAMES) quando non ce n'è bisogno, e senza fornire alternative
- **No!** Usare le *tabelle* per l'impaginazione (per quanto possibile)
- **Sì...** Fornire una *versione stampabile* delle pagine web organizzate in tabelle
- **No!** Usare in un form oggetti che non seguono le *regole standard* relative all'interfaccia utente
- **No!** Usare form con *action=mailto*:
- **No!** Usare *link di tipo mailto*:
- **No!** Inserire *audio come sottofondo* all'apertura di una pagina
- **No!** Inserire link a documenti in *formato diverso* dall'HTML (es. PDF, .DOC, .PS, ecc.) senza che la cosa sia esplicitata
- **Sì...** Nel caso si utilizzino i frames, fornire alternative per una *navigazione "frame-free"*
- **Sì...** Mantenere piccole le dimensioni delle pagine (testo + immagini + altro): indicativamente sotto i 50 kb totali
- **Sì...** Effettuare *test di "usabilità"* chiedendo ad alcune persone di cercare alcune informazioni nel sito o di effettuare altre operazioni
- **Sì...** Fornire *link normali* oltre ad eventuali mappe cliccabili o menu gestiti con Javascript

### 11.1.3 In merito all'organizzazione

- **No!** Cambiare gli indirizzi all'interno del proprio sito
- **No!** Generare documenti dinamicamente anche quando non è strettamente necessario
- **Sì...** Analizzare i *file di log* del web server per verificare gli errori, i link errati, le richieste al proprio motore di ricerca, ecc.
- **Sì...** Avere *URL* comprensibili, memorizzabili e facili da scrivere
- **Sì...** Avere *link alla home page* e indicazioni tipo "*ti trovi qui*" in tutte le pagine
- **Sì...** Fornire una *mappa del sito* ed un *motore di ricerca* interno
- **Sì...** Aggiornare le vecchie pagine *aggiungendo note e link* a nuovi documenti
- **Sì...** Rendere accessibili le *vecchie pagine* (creando un archivio storico di informazioni)
- **Sì...** Dare la possibilità al lettore di *contattare l'autore* del sito per maggiori informazioni, segnalazione di errori, ecc. (preferibilmente sia da un modulo web sia fornendo un indirizzo di posta elettronica)
- **Sì...** Favorire il *download* di documentazione in forma compressa, in modo che possa essere consultata comodamente *off-line*
- **Sì...** Per la distribuzione di documenti, preferire *formati ad ampia compatibilità* e con grande disponibilità di lettori (ad esempio, RTF o PDF per documenti, CSV per tabelle, ecc.) piuttosto che formati leggibili solo con specifici software proprietari

### 11.1.4 In merito allo stile

- **No!** Usare *scritte lampeggianti o in movimento costante*, oppure immagini animate, che molto spesso distraggono il lettore più che attirare la sua attenzione
- **No!** Usare link che aprono documenti in una *nuova finestra* a tutto schermo
- **No!** Trasformare la propria home page in un portale dimenticandosi il motivo per cui è stata realizzata e per cui probabilmente viene visitata
- **No!** Predisporre banner pubblicitari che fingono di essere finestre di dialogo o forms
- **Sì...** Informare sulla *non disponibilità* di informazioni circa un prodotto, o sulla non disponibilità di un certo documento
- **Sì...** Mantenere le informazioni più importanti nella parte superiore della pagina
- **Sì...** Inserire informazioni sull'*autore* di un documento (con un collegamento ad una pagina specifica)
- **Sì...** Fare in modo che i *link* siano *chiari*
- **Sì...** Utilizzare liste e coppie titolo/descrizione
- **Sì...** Scrivere *frasi brevi* evidenziando alcuni termini (ma senza esagerare)
- **Sì...** Scrivere le pagine in *forma giornalistica*, partendo dalle informazioni più rilevanti
- **Sì...** Tenere presente che su monitor si legge in maniera diversa che su carta
- **Sì...** Tenere in considerazione che chi legge il documento può non comprendere abbreviazioni, sigle, ecc., che le date si esprimono diversamente nelle diverse lingue, che le valute possono essere ambigue

## 12 Licenza Open Content

Questa documentazione è rilasciata in accordo con la [licenza Open Content](http://opencontent.org/opl.shtml) [<http://opencontent.org/opl.shtml>], a cui si fa riferimento per gli aspetti ufficiali e legali. In sintesi e ufficiosamente:

1. puoi copiare e distribuire con ogni mezzo questa documentazione, a patto di lasciare inalterata la dicitura di copyright, l'indicazione relativa alla mancanza di garanzie e il riferimento alla licenza Open Content;
2. non puoi farti pagare per questa documentazione, se non per il costo dei mezzi di supporto eventualmente impiegati;
3. puoi modificare questa documentazione o una sua parte, a patto di:
  - a. far sì che risultino evidenti le parti modificate rispetto all'originale, nonché la natura e la data dei cambiamenti;
  - b. applicare alla documentazione derivata questa stessa licenza.
4. se non accetti i termini di questa licenza, non hai il diritto di copiare, distribuire o modificare la documentazione.
5. **assenza di garanzia: considerato che la documentazione è rilasciata gratuitamente, non esiste nessuna garanzia. A meno che non sia diversamente specificato, la documentazione è fornita "così come è" senza nessuna garanzia di alcun tipo, né espressa né implicita; ciò comprende, senza limitarsi a questo, la garanzia implicita di commerciabilità e utilizzabilità per un particolare scopo. L'intero rischio concernente l'utilizzo di questa documentazione è a carico di chi lo usa, che si assume il costo di ogni riparazione o correzione. In nessun caso, il detentore del copyright o altre parti che possono modificare o distribuire la documentazione come da accordo di licenza, sono responsabili per i danni (generali, speciali, incidentali o consequenziali che siano) che possono derivare dall'uso o dalla non possibilità d'uso di questa documentazione.**

I marchi registrati citati nella documentazione sono di proprietà delle rispettive case. I software ed i siti internet citati vengono proposti come esempi senza che ciò implichi una qualche forma di garanzia sulla loro validità o di raccomandazione.

L'autore sarà grato a chi vorrà esprimergli consigli, suggerimenti, critiche costruttive su di essa. Fatta salva la validità della Licenza Open Content, l'autore gradirebbe essere contattato nel caso di utilizzo della documentazione, al fine di poter fornire indicazioni sugli aggiornamenti della stessa.

---

© Loris Tissino, 1997–2003

---

Appunti Internet – Copyright © 1997–2003 Loris Tissino  
Versione 6.0 (gennaio 2003) [Licenza Open Content](#)

Questo documento è stato generato il 01/01/2003

Per segnalazione di imprecisioni ed errori o per quesiti si prega di usare l'apposito [modulo](#) [<http://www.tissino.it/loristissino/contatto.php>]