

**Come Creare e Mantenere
un Sito Web Usabile ed Accessibile
MANUALE D'USO**

Perché questo Manuale

- Per dare un piccolo aiuto a tutte le persone disabili
- Per rendere più facile la vita a chi è “costretto” a cimentarsi con l’accessibilità e l’usabilità
- Per mettere a disposizione la mia esperienza in questo campo, fornendo una serie di consigli pratici
- Per consentire a chi si accinge a creare un nuovo sito a riflettere
- Per ringraziare tutti quelli che avranno la pazienza di leggere tutto il manuale



Flavio Reverditi

frever@libero.it - Novembre 2012

Definizione di **Usabile**: Lo standard [ISO 9241-11:1998](#) definisce l'usabilità come:

Il grado in cui un prodotto può essere usato da particolari utenti per raggiungere certi obiettivi con efficacia, efficienza, soddisfazione in uno specifico contesto d'uso" intendendo

Efficacia come precisione e completezza con cui gli utenti raggiungono specifici obiettivi

Efficienza come risorse impiegate in relazione alla precisione e completezza cui gli utenti raggiungono specifici obiettivi

Soddisfazione come libertà dal disagio e attitudine positiva con cui gli utenti raggiungono specifici obiettivi attraverso l'uso del prodotto.

In questo contesto si parla di **usabilità** come interazione efficace e soddisfacente per l'utente sia in termini di efficienza sia in termini di benessere. L'usabilità ha quindi come obiettivo quello di economizzare lo sforzo cognitivo dell'utente, proponendo prodotti che siano facili da comprendere, da imparare, da usare, da ricordare, che evitino o rendano recuperabili gli errori e che quindi gratifichino l'utilizzatore.¹

Definizione di **Accessibile**: la Legge 4/2004 definisce l'accessibilità come:

La capacità dei sistemi informatici, nelle forme e nei limiti consentiti dalle conoscenze tecnologiche, di erogare servizi e fornire informazioni fruibili, senza discriminazioni, anche da parte di coloro che a causa di disabilità necessitano di tecnologie assistive o configurazioni particolari;²

W3C: Abbreviazione del World Wide Web Consortium: è un consorzio nato nell'ottobre del 1994 per portare il World Wide Web alla sua massima potenzialità definendo protocolli comuni con la finalità di promuoverne l'evoluzione garantendo l'interoperabilità. Attualmente il W3C conta quasi 500 organizzazioni con presenza in tutto il mondo che hanno riconosciuto la funzione del W3C e desiderano contribuire alla crescita del Web. Uno dei progetti portati avanti dal W3C è la Web Accessibility Initiative (**WAI**), specificamente dedicata a fornire linee guida e raccomandazioni per quanto riguarda l'accessibilità

¹ Tratto da WIKIPEDIA

² Definizione della Legge Stanca del 9/1/2004 art. 2

Una volta chiariti i concetti di usabilità ed accessibilità vediamo, in ordine di **importanza** quali sono i presupposti:

L'utente deve essere il centro di tutto: è essenziale conoscere le esigenze dei visitatori, facilitare al massimo la navigazione, prevedere i comportamenti degli utenti.

La Navigazione di conseguenza deve consentire di raggiungere l'informazione desiderata con semplicità, immediatezza e con pochi passaggi.

Il Progetto: deve essere definito con linee di guida ben definite ma anche flessibili, che diano cioè la possibilità di future implementazioni; utilizzare quindi un layout che sia in grado di espandersi, quindi di contenere quell'accumulo di informazione che aumenta nel tempo. Per ottenere ciò l'ideale è ovviamente di collegare il layout ad un database in modo che l'inserimento, la ricerca e la modifica di dati sia facilmente gestibile.

L' **Home-Page** è la vetrina di presentazione del sito: l' home-page è quasi sempre la via di accesso principale al sito: l' utente deve poter trovare immediata soddisfazione alla prima domanda ... "perché sono qui?"... e successivamente alla seconda domanda ... " dove posso andare da qui? Come mi muoverò alla ricerca di ciò che sembra promettente o come troverò ciò che stavo cercando?"....

I **Menù** sono la via principale, e spesso l'unica, per poter accedere alle varie sezioni del sito. Consiglio di utilizzare un menù unico per tutte le pagine del sito, in modo che sempre e comunque l'utente possa accedere **SEMPLICEMENTE** a tutte le sezioni. Quindi meglio utilizzare sottomenù (o menù a cascata), che diano immediatamente una panoramica completa delle sezioni del sito. Un esempio lo trovate navigando sul portale di www.yahoo.it : per trovare gli SPETTACOLI nella home non c'è nessun riferimento; bisogna cliccare sul menù NOTIZIE (a destra) e poi troverete la voce di menù SPETTACOLI (in alto). Semplice ed intuitivo vero? Sarebbe bastato mettere un semplice menù a cascata contenente le sezioni relative a NOTIZIE. Sapete dove è la pagina "Computer" ? Ma cliccate su Shopping!!!

Le **Pagine Web**: ogni singola pagina web è divisa in due parti fondamentali: "head" e "body", vale a dire l'intestazione ed il corpo: cominciamo a vedere quello che può contenere il tag "body"

BODY

Il Testo: ecco gli aspetti da considerare:

- Il **contenuto** del testo deve essere ben compreso dagli utenti che visiteranno il sito, quindi nello scrivere i testi deve necessariamente tener conto del target al quale si rivolge. E' ovvio che rivolgersi a specialisti di un certo argomento e rivolgersi ad un pubblico generico implica linguaggi differenti.
- I **caratteri** (font) non devono dare sfoggio di fantasia estrema: utilizzare 2 o 3 tipi di carattere: quando definite i font, mettete sempre per ultimo un font generico scegliendo tra: serif oppure sans-serif; se sul vostro pc non ci sono installati i font da voi specificati il browser userà il primo font che trova di quella famiglia definita dal font generico sans-serif. Sono consigliati font molto bene leggibili, quali Arial, Verdana, Century Gothic, Tahoma, Bookman Old Style.
Utilizzare max due colori per il testo. Evitare dimensioni troppo piccole e definire tali dimensioni utilizzando la misurazione em, consentendo cioè agli utenti di incrementare o

diminuire la dimensione del testo nel loro browser: l' interfaccia si allargherà o si restringerà mantenendo le esatte proporzioni tra testo e layout. Ad esempio un ipovedente potrà incrementare le dimensioni del carattere predefinito del browser senza che la pagina si "sformi".

- **Colore del carattere** e colore dello **sfondo**: bisogna stare attenti affinché il testo sia adeguatamente contrastato rispetto allo sfondo: si consiglia un testo scritto in colore bianco o giallo su fondo di colore deciso (nero, blu, verde..) ma anche testo ottenuto con una combinazione di colori più usuale: caratteri in colore nero su fondo bianco.
- **Testo Lampeggiante**: da evitare scritte lampeggianti, ma anche oggetti lampeggianti; le frequenze di intermittenza possano provocare disturbi da epilessia fotosensibile, disturbi della concentrazione o possano causare il malfunzionamento degli screen reader o di altre tecnologie assistive
- **Interlinea**: una interlinea un po' più ampia permette una migliore fruizione del testo e rende la lettura meno faticosa

Le Immagini:

non è superfluo ricordare l'importanza dell'attributo alt nell'indicare la descrizione testuale dell'immagine, ma occorre evidenziare che il testo "alternativo" all'immagine deve essere esaustivo, non deve essere un riassunto breve. Se, per esempio, l'immagine fosse questa i testi alternativi NON dovrebbero essere:

"Quadro"

"Il Quarto Stato"

"Quadro il Quarto Stato"

ma

"Particolare del quadro il Quarto Stato di Pellizza da Volpedo"

Altri aspetti da considerare nella gestione delle immagini sono:

- **Pesantezza**: le immagini non devono avere un peso in byte troppo alto perché, oltre a rallentare il caricamento della pagina, (da non dimenticarsi mai che la banda larga non è disponibile per tutti), spingono l'utente visitatore ad annoiarsi e a perdere interesse per il sito.
- **Sequenza** o **Galleria** di immagini a scorrimento sono possibili, sconsigliate le applet, con javascript, purchè ogni immagine sia ben descritta testualmente e che ci sia una descrizione anche per l'effetto javascript
- **Lampeggio** di immagini: vietatissimo.



I Link: regole generali per qualsiasi tipo di link (testo o immagine) sono ovvie; essere chiari nell'indicazione inserendo nel tag³ title una descrizione completa ed esaustiva di dove porta il link;

³ Tag html: comando o istruzione di formattazione contenuto in un file; i tag sono caratterizzati da contrassegni (< >) che servono per distinguere i tag dal testo del file

evitare link generici e link "duplicati" cioè con lo stesso testo, ad esempio "continua", "clicca qui", "leggi tutto", etc..

- **Testuali:** ottemperare a quanto detto prima riguardo al testo, (font, colori, dimensioni etc...)
- **Immagini:** ottemperare a quanto detto prima riguardo alle immagini (pesantezza, lampeggio)

Le Mappe Sensibili (immagini con link diversi):

- Rispettare sia i requisiti legati a colore, sia i requisiti legati alle immagini (vedi sopra).
- Immagini e aree di dimensioni ridotte le rendono meno accessibili a disabili motori ed ipovedenti.
- Assegnare l'attributo ALT a tutte le aree che compongono la mappa. Il browser testuale o lo screen reader vedranno le aree come una serie di links.

Frame e Iframe

Vietati sia il Frame che Iframe; la legge 04/2004 ⁴ non concede l'uso di frame per la realizzazione di nuovi siti. I motivi principali sono:

- alcuni screen reader non sono in grado di leggerli;
- l'Url di una pagina interna non rispecchia esattamente la pagina presente nel browser, rischiando così di indirizzare l'utente a una pagina diversa;
- i motori di ricerca hanno difficoltà ad indicizzarli;
- non è possibile stampare integralmente una pagina strutturata con i frame, ma soltanto il frame attivo, su cui cioè l'utente ha cliccato per ultimo prima di avviare la procedura di stampa.

Il problema quindi che si pone è non solo come dividere la pagina o le pagine in parti (cornici) ma come non continuare a scrivere lo stesso codice per tutte le pagine del sito (esempio classico il menù).

I **DIV** sono la soluzione al primo problema (dividere la pagina in parti): occorre determinare le dimensioni, ma anche la posizione del div rispetto sia alla pagina ma anche ad eventuali altri div. La possibilità di fissare la posizione ad un valore assoluto rispetto alla distanza in pixel dai 4 margini (alto basso sinistro destro) della pagina, piuttosto che rispetto al div precedente permette un perfetto assemblaggio della pagina. Tante sono le opzioni relative al posizionamento del div, (absolute e relative le più semplici), sta a chi crea la pagina trovare l'esatta collocazione di ogni singolo div. Da tener ben presente che non tutti i browser si comportano nello stesso modo, quindi meglio non esagerare con attributi, quali "float" all'interno del tag div.

A questo punto occorre definire cosa sono le pagine statiche e cosa sono le pagine dinamiche ed in particolare le loro differenze

Le **pagine statiche** non sono altro che le classiche pagine HTML. Praticamente sono pagine standard che contengono testi ed immagini fissi e che non si interfacciano in nessun modo con un database nè offrono alcun tipo di funzionalità.

Una **pagina dinamica** invece è una pagina normalmente ben più complessa e che è composta da una pagina statica standard e dal codice di programmazione che ne definisce e descrive le varie funzionalità.

Questo tipo di pagina offre possibilità immense ed ormai la maggior parte dei siti ne fa uso; considerate infatti che **QUALSIASI** sito che offra, ad esempio, la possibilità di registrarsi, oppure la possibilità di partecipare ad un forum, è un sito dinamico.

⁴ documento successivo alla proposta di Legge Stanca, "Studio sulle linee guida recanti i requisiti tecnici e i diversi livelli per l'accessibilità e le metodologie tecniche per la verifica dell'accessibilità", pubblicato nel maggio 2004
Vedi anche decreto ministeriale dell'1 marzo 2005

Una volta chiarite le due tipologie di pagine web, torniamo a dove eravamo rimasti, cioè al problema di contenuto ripetuto in varie pagine del sito.

Ad esempio: immaginiamo di dover inserire un menù in tutte le pagine del nostro sito. Se in futuro vogliamo modificarlo dovremo modificare tutte le pagine, se al contrario il menù è formato da una singola pagina inclusa in tutte le altre pagine basterà modificare solo una pagina per modificare tutto il sito.

Per poter includere file nelle pagine dinamiche (asp, php, asp-net) si utilizzano i tag include (ovviamente con sintassi diverse in base al linguaggio di scripting⁵).

Il problema si complica quando ho la necessità di inserire contenuto ripetitivo in semplici pagine statiche .htm o .html.

Per includere porzioni di HTML nelle pagine statiche è possibile utilizzare Javascript.

Possiamo utilizzare il tag <script> ed includere un file Javascript che genera la porzione della pagina che ci interessa, utilizzando document.write()

Ecco un esempio relativo ad un semplice menù:

Creare il file menu.js che contiene il seguente codice

```
var menuhtm = '<div id="navigazione_menu">' +  
'<ul>' +  
'<li><a href="home.htm">Home</a></li>' +  
'<li><a href="libri.htm">Libri</a></li>' +  
'<li><a href="riviste.htm">Riviste</a></li>' +  
'<li><a href="enciclopedie.htm">Enciclopedie</a></li>' +  
'<li><a href="contatti.htm">Contatti</a></li>' +  
'</ul>' +  
'</div>';  
document.write(menuhtm);
```

Inserire all'interno dei file htm, nella posizione desiderata la seguente riga di codice:

```
<script type="text/javascript" src="menu.js"></script>
```

Ed il gioco è fatto.

Tablelle

Le tablelle possono essere usate in un solo modo: per raccogliere e mostrare dati; tablelle usate per disporre i contenuti di un layout non si possono usare. Per il layout di contenuti web si utilizzano i div di cui abbiamo parlato precedentemente. Tablelle di impaginazione quindi sono vietate.

Le tablelle di dati servono per visualizzare informazioni di in cui sono definite specifiche relazioni orizzontali e verticali tra i contenuti delle celle.

Tutto semplice per chi non ha problemi di vista, in quanto i dati si presentano identificabili tra colonne e righe; il problema nasce quando l'utente utilizza lo screen-reader. I dati debbono essere correttamente distinti tra righe e colonne; lo screen-reader deve "capire" che deve prima leggere l'intestazione della colonna e poi tutti i dati relativi contenuti nelle righe sottostanti. Per fare questo occorre identificare le celle d'intestazione il tag "<th>" (l'acronimo sta per "table header", cioè "intestazione di tabella") che definisce una cella contenente informazioni di intestazione.

⁵ il linguaggio di scripting è un linguaggio di programmazione interpretato, destinato in genere a compiti di automazione del sistema operativo, delle applicazioni, o a essere usato all'interno delle pagine web.

Se le intestazioni sono lunghe si può utilizzare l'attributo "abbr" per fornire una versione abbreviata in modo da dare agli screen reader qualcosa che possono utilizzare al posto del testo nella stessa intestazione.

Una buona cosa è quella di fornire sempre dei riassunti per le tabelle, mediante l'utilizzo dell'attributo "summary" del tag table; "Tabella contenente il numero dei partecipanti ai corsi di informatica, la tipologia dei corsi e l'anno in cui si è svolto il corso", ad esempio è una buona descrizione del contenuto della tabella.

Questo è il codice relativo ad una tabella di esempio:

```
<table summary=" Tabella contenente il numero dei partecipanti ai corsi di informatica, la
tipologia dei corsi e l'anno in cui si è svolto il corso.">
<caption>Corsi di informatica</caption>
<tr>
  <th abbr="Corso">Tipologia del corso</th>
  <th abbr="Partecipanti">Numero dei partecipanti</th>
  <th abbr="Anno">Anno del corso</th>
</tr>
<tr>
  <td>Rudimenti di Informatica</td>
  <td>22</td>
  <td>2009</td>
</tr>
<tr>
  <td>I fogli di calcolo</td>
  <td>23</td>
  <td>2010</td>
</tr>
</table>
```

Da notare che l'attributo "abbr" consente allo screen-reader di pronunciare la parola "Corso" anziché la frase "Tipologia del Corso"; se i corsi sono 2 la cosa non provoca nessun disturbo, se i corsi fossero 200 ripetere la frase 200 volte sarebbe fastidioso.

Un discorso a parte per le tabelle che io chiamo "complicate", vale a dire quelle tabelle che presentano le intestazioni nelle colonne ma anche nelle righe.

Ecco un esempio:

Articolo	Ritiro Deposito	Ritiro Negozio	Domicilio	Contrassegno
Pen Drive 16 GB	15 €	16€	22€	24€
Hard Disk 1 TB	95€	101€	115€	122€
Stampante Laser	134€	145€	155€	162€

Le intestazioni (in giallo) in questo caso sono sia sulle righe che sulle colonne; penso che non siano il massimo per quanto riguarda l'usabilità intesa come semplicità; sarebbe + opportuno, utilizzare dati dinamici, in questo caso, ad esempio, facendo prima una selezione sugli articoli e poi visualizzare una tabella con i vari prezzi.

Detto questo nella tabella esposta sopra vanno evidenziate le celle di intestazione , rendendo così esplicita la relazione tra celle d'intestazione e celle di dati

Per fare ciò si utilizza l'attributo "scope", quindi la cella "Ritiro Deposito" viene creata con il seguente codice

```
<th scope="col" abbr="Deposito">Ritiro Deposito</th>
```

La cella "Pen Drive 16 GB" avrà il seguente codice

```
<td scope="row" abbr="Pen 16">Pen Drive 16 GB</td>
```

Col indica la colonna mentre Row indica la riga; (il tag "th" può indicare solo la colonna).

Esiste un altro sistema per rendere accessibile la tabella esaminata in precedenza usando gli attributi "headers" e "id" in luogo dell'attributo "scope". Il sistema basato su "scope" è più adatto per tabelle semplici, in cui le informazioni di intestazione sono posizionate a inizio riga e/o inizio colonna. Il sistema basato su "headers - id" è invece più adatto a tabelle di struttura complessa e irregolare, nelle quali le relazioni di appartenenza tra dati e intestazioni non sono riconducibili semplicemente a righe, colonne, gruppi di righe o gruppi di colonne; ma tabelle così complesse cozzano, come già detto prima, contro il concetto di usabilità.

Elenchi/Liste

Non c'è molto da dire per la gestione degli elenchi, siano essi puntati (tag "ul") o numerati (tag "ol"). Vietato l'attributo "type" si può utilizzare l'attributo "style" ad esempio...

```
<ul style="list-style-type: circle;">
```

visualizzerà un cerchio come stile dei punti elenco

mentre

```
<ol style="list-style-type: upper-alpha">
```

Visualizzerà un elenco numerato in ordine alfabetico (A B C

Si possono creare elenchi strutturati contenenti altri elenchi.....

```
<ul style="list-style-type: circle;">
```

```
<li>Caffè</li>
```

```
<li>Te
```

```
<ul style="list-style-type: square;">
```

```
<li>Giallo</li>
```

```
<li>Verde
```

```
<ul style="list-style-type: square;">
```

```
<li>Dalla Cina</li>
```

```
<li>Da Taiwan</li>
```

```
</ul>
```

```
</li>
```

```
</ul>
```

```
</li>
```

```
<li>Latte</li>
```

```
</ul>
```

Come si evince dal codice l'importante è aprire e chiudere correttamente i tag "ul" e "li"

Form/Moduli

Grande importanza rivestono oggi i moduli, vengono utilizzati per tantissimi usi: iscrizioni a siti e ai social-network, prenotazioni di voli, commenti nei forum e nelle chat etc....

Occorre dedicare a questo argomento quindi tutta l'attenzione e la cura necessarie per sviscerare per bene l'argomento.

Non dimentichiamoci mai l'importanza del linguaggio, se vogliamo mantenere un profilo di usabilità ed accessibilità. Per cui bisogna sforzarsi per essere chiari, evitando gerghi e tecnicismi, eliminando tutte le possibili ambiguità; ad esempio "Nominativo" è un termine non chiaro, mentre "Cognome e Nome" non dà luogo a nessuna ambiguità, il termine "Nato" va tradotto in "Data di Nascita" e "Comune di Nascita".

Nei moduli posso gestire sia pochi che tanti dati: nel caso debba gestire tanti campi diventa indispensabile dividere la form in sezioni utilizzando il tag "fieldset", che permette di aprire e chiudere una parte del modulo che contiene campi appartenenti ad una determinata sezione.

Ad esempio in un modulo di iscrizione ad un social-network le sezioni potrebbero essere:

- Dati Anagrafici
- Residenza
- Studi effettuati
- Società presso cui ha lavorato
- Interessi specifici

Ogni sezione ha bisogno di una intestazione e per fare questo si utilizza il tag "legend".

Ci sono vari tipi di dati (campi) che si possono utilizzare: tutti i vari tipi di campi hanno comunque in comune

- Il tag "legend" consente di associare logicamente un'etichetta ad un campo
- L' identificativo del campo viene stabilito utilizzando l' attributo "id"
- Nel tag "legend" l'attributo "for" deve far riferimento all' attributo "id" del corrispondente campo
- L' attributo "name" di un campo definisce il nome univoco della variabile che viene usata dallo script che processa il form.
- L' attributo "title" che visualizza un commento con il passaggio del mouse sul campo. Un attributo molto consigliato, quando ad esempio ho necessità di specificare il formato dei dati. Ad esempio "Inserire data di nascita nel formato gg/mm/aaaa" diventa molto utile.

Ecco un esempio di un campo che mi chiede di inserire il Cognome

```
<label for="cognome">Cognome : </label>
```

```
<input type="text" name="cognome" id="cognome" title="Inserire il Cognome"/>6
```

Vediamo adesso quali sono i tipi di campo (controlli) più comuni utilizzati nei moduli.

Input: il tag più comune che, oltre al tipo "text" si può utilizzare per:

- Campo Password (type="password") che contiene un dato di tipo testo ma viene visualizzato con asterischi che nascondono il dato che sto inserendo
- Campo Nascosto (type="hidden") : il campo non viene visualizzato, ma il dato che contiene (attributo "value") viene poi utilizzato dallo script che processa il modulo
- Pulsante di invio dati (type="submit") : non ha bisogno ovviamente di una "label"
- Pulsante di cancellazione dati (type="reset") : consente di riportare il modulo allo stato originario, cancellando tutto quello che è stato scritto
- Pulsante generico (type="button"): è un pulsante generico che può servire per determinate azioni utilizzando un linguaggio di script. Ad esempio se voglio mettere un pulsante che ritorni alla pagina precedente

```
<input type="button" onclick="javascript:history.back()" value="Indietro" Title="Torna alla pagina precedente">
```

- Con le checkbox (type="checkbox") possiamo consentire all'utente di operare delle scelte multiple come ad esempio...

```
<fieldset>
```

```
<legend>Lingue conosciute</legend>
```

```
Italiano<input type="radio" name="lingua" value="italiano"/>
```

```
Inglese <input type="radio" name="lingua" value="inglese"/>
```

```
Francese <input type="radio" name="lingua" value="francese"/>
```

```
</fieldset>7
```

⁶ Come si vede dall'esempio spesso gli attributi "id" e "name" hanno lo stesso valore.

- Invece con i radio button (type="radio") devo effettuare una scelta esclusiva.

```

<fieldset>
  <legend>Livello di Studi</legend>
  Elementare<input type="radio" name="studi" value="elementare"/>
  Scuola Media <input type="radio" name="studi" value="media" checked="checked" />
  Diploma <input type="radio" name="studi" value="diploma"/>
  Laureato <input type="radio" name="studi" value="laureato"/>
</fieldset>

```

Da notare l'attributo "checked" che contiene il valore checked: sta ad indicare che quello è il valore di default

Select/Option: può essere considerato un controllo che assembla checkbox e radiobutton: il tag è "select" al cui interno ci sono le "option". Con l'attributo "multiple="multiple" indico che si possono effettuare scelte multiple; in assenza dell'attributo la scelta può essere una sola.

Ecco un esempio

```

<label>Quale browser utilizzi?<br />
<select name="browser" multiple="multiple">
  <option value="explorer">Explorer </option>
  <option value="firefox">Firefox </option>
  <option value="opera">Opera </option>
  <option value="chrome">Chrome </option>
  <option value="safari" >Safari </option>
</select>
</label>

```

Textarea se si ha la necessità di indicare un campo che consenta di inserire una grande quantità di testo conviene utilizzare una "textarea" (area di testo)

```

<textarea name="commento" rows="5" cols="40">
  qui puoi inserire un tuo commento
</textarea>

```

Da notare gli attributi "rows" e "cols" che determinano il numero di righe ed il numero di colonne. In questo caso l'attributo "title" non è indispensabile poiché il commento è inserito direttamente all'interno della textarea.

HEAD e dintorni

Ora che abbiamo visto le cose più importanti che può contenere il corpo della nostra pagina, vediamo come deve essere composta l'intestazione della pagina web.

Il primo elemento che deve essere presente nel codice della nostra pagina web è il "<!DOCTYPE>" (Document Type Declaration) che serve a dichiarare quale versione HTML abbiamo utilizzato per scriverla e quindi a quale DTD (Document Type Definition) facciamo riferimento. Per quanto riguarda le DTD, se dobbiamo costruire un sito nuovo possiamo utilizzare solo "strict" in quanto la "framset" è utilizzata per i siti che supportano i frame (vietati), "transitional" è ritenuta valida solo se si tratta di siti già esistenti.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>

```

7 Da notare come il "name" sia per tutte le scelte lo stesso. In fase poi di gestione dello script la variabile "lingua" conterrà tutti i valori SELEZIONATI

Dopo il doctype scriviamo necessariamente il tag <HTML> che fino alla sua chiusura (</HTML>) deve contenere codice html. Possono essere specificati al suo interno anche gli attributi:

- LANG (linguaggio) il cui valore è espresso con sigle standard di due caratteri ["IT" per Italia]
- DIR (direzione della lettura) che per i caratteri occidentali è LTR [left to right]

Un esempio semplice quindi.....

```
<html lang="it" dir="ltr">
```

Passiamo ora al "titolo" della nostra pagina (<title>): il suo contenuto non solo fornisce informazioni all'utente comparando nella barra in alto dei browser (detta appunto "barra del titolo") ma le fornisce anche ai motori di ricerca essendo uno dei dati che questi prendono in maggior considerazione nell'indicizzare le nostre pagine. Dunque fate molta attenzione nel decidere cosa scriverci. Niente parole troppo scontate tipo "benvenuti nel sito" "informazioni" "vendita", ma parole chiave che identifichino la specificità del sito. Niente preposizioni, articoli che vengono ignorati dai motori di ricerca

Inserire poche parole 4 o 5 al massimo; se c'è necessità di inserire tante parole meglio dividere le parole da inserire nel title nelle altre pagine o meglio nelle altre sezioni del sito.

I meta tag sono metadati presenti nel linguaggio HTML utilizzati per fornire informazioni sulle pagine agli utenti o ai motori di ricerca.⁸

Un meta tag necessario è quello che serve per definire il set di caratteri che preferiamo (codifica)

Due sono le codifiche che generalmente si utilizzano: la codifica UTF-8 che permette, al browser, di interpretare i caratteri particolari, internazionali, accentati ecc. correttamente.

L'altra è iso-8859-1, lo standard per i caratteri occidentali. Consiglio vivamente di utilizzare la codifica UTF-8 se non si vogliono avere problemi con le lettere accentate, per esempio.

Quindi il tag è....

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
```

Il meta tag "keyword" secondo contiene le parole chiave che verranno utilizzate dai motori di ricerca per "trovare" il nostro sito; più le parole sono comuni più perdono efficacia.

```
<meta name="keywords" content="mettere qui le parole chiave separate dalla virgola">
```

Infine "description" ha come contenuto il breve testo (consiglio massimo 300 caratteri) che verrà anch'esso dato in pasto ai motori di ricerca.

```
<meta name="description" content="un breve testo descrittivo">
```

Esistono molti altri meta-tag che servono a gestire i motori di ricerca, fare refresh e reindirizzamenti, fornire informazioni sul sito del tipo: autore, copyright, software usato per generare le pagine, etc.. Personalizzare i meta tag è compito del webmaster⁹.

I fogli di stile - CSS

Capitolo molto importante poiché definisce la formattazione delle pagine web. L'utilizzo dei CSS (*Cascading Style Sheets*) si è reso necessario per separare i contenuti dalla formattazione, permettendo una programmazione più chiara e facile da utilizzare soprattutto per i webmaster.

Le regole per comporre il CSS sono contenute in un insieme di direttive (Recommendations) emanate a partire dal 1996 dal W3C.

Tramite i CSS potrete dare al testo delle vostre pagine un aspetto da word-processor: non solo con il colore o i font che preferite, ma con un sistema di interlinea pratico e funzionale, con le decorazioni che desiderate, riuscendo a spaziare lettere e parole, impostando stili diversi per titoli e paragrafi.

⁸ Tratto da Wikipedia

⁹ Colui che progetta costruisce ed è responsabile di un sito Web

Per quanto riguarda l'accessibilità i CSS sono uno strumento portentoso, anche grazie al fatto di poter essere gestiti con linguaggi di programmazione avanzati in grado di modificare con un solo click l'aspetto di una pagina, ad esempio aumentando o diminuendo la dimensione dei caratteri senza sfalsare la visualizzazione della pagina.

CSS esterni e interni

Un foglio di stile definito in un file separato dal documento si definisce esterno. Si tratta di semplici documenti di testo ai quali si assegna l'estensione .css.

Per caricare un foglio esterno in una pagina web si usa il tag <link> che va collocato all'interno della sezione <HEAD> del documento.

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

Dove stile.css è il file css

Un foglio di stile si dice invece interno quando il suo codice è compreso in quello della pagina web.

```
<style type="text/css">
```

```
.....  
</style>
```

Generalmente nel css esterno vanno messi gli stili e le definizioni che verranno utilizzati in tutto il sito, nel css interno vanno messi gli stili che verranno utilizzati solo in quella pagina. Questa è una considerazione, non è ovviamente una regola fissa.

L' attributo style: l'ultimo modo per formattare un elemento con un foglio di stile consiste nell'uso dell'attributo 'style'. Si tratta di un attributo applicabile a tutti gli elementi. La dichiarazione avviene a livello dei singoli tag contenuti nella pagina.

```
<p style="text-align:center; color:Black;">
```

In questo esempio il paragrafo ha il testo allineato al centro e il colore del carattere è il nero.

In questo caso la formattazione si applica al solo paragrafo interessato.

Ricapitolando, ci sono 3 possibilità di applicare gli stili:

- Con un file .css esterno
- Inserendo lo style all' interno della pagina web
- Utilizzando l' attributo style all' interno di un determinato elemento
-

CSS e i Media (presenti futuri e passati)

Browser classici, browser vocali, browser braille smartphone, ipad, ipod, palmari, cellulari dispositivi del presente e del futuro. Ciascuno di questi supporti presenta caratteristiche diverse in termini di memoria, ampiezza dello schermo e funzionalità. Riuscire ad adattare uno stile unico a tutti è molto difficile, anche se non impossibile. Se rinunciamo a creare un css valido per tutti i dispositivi, possiamo utilizzare l' attributo *media* nel definire un css valido per un particolare supporto.

Questo l'elenco dei media supportati

- screen (Utilizzato per schermi di computer)
- handheld (Usato per dispositivi di piccole dimensioni o palmare)
- print (Stampanti)
- braille (Browser braille)
- embossed (Stampanti braille)
- projection (Proiezioni)
- speech o aural (Sintetizzatori vocali)
- tty (Telescriventi)
- tv (Televisori)
- all (Qualsiasi dispositivo)

Due sono i modi più utilizzati per dichiarare un media:

- Si definisce il media dell'elemento link:
`<link rel="stylesheet" media="screen" href="screen.css" />`
- All'interno di un foglio di stile, poi, si può impostare il media facendo ricorso alla direttiva @media:
`@media screen and (color) {
/* qui vanno le istruzioni CSS */
}`

Più media possono essere raggruppati in una lista separandoli con una virgola:
`media="screen,projection"`

Cosa c'è all'interno di un foglio di stile: regole e commenti: un foglio di stile altro non è che un insieme di regole con la possibilità di inserire note di commento.

Ecco come è fatta una regola:

`p {color: black;}`

Dove **P** è il **selettore** (paragrafo in questo caso)

Dove `{color: black;}` è il **blocco della dichiarazione**

Dove **color** è la **proprietà**

Dove **black** è il **valore**

Con questa regola tutti i paragrafi avranno il colore del carattere nero.

Proprietà e valore relativo sono racchiusi tra parentesi graffe.

Tra proprietà e valore ci vogliono i due punti.

Posso inserire più proprietà separate dal punto e virgola

`p {color: black; font-size:12px;}`

Per inserire parti di commento in un CSS racchiudetelo tra questi segni:

`/*` come segno di apertura

`*/` come segno di chiusura

Selettore

Abbiamo visto che una regola CSS viene applicata ad un selettore. Per selettore si intende una particolare parte o parti di un documento soggette a una specifica regola; ci sono vari tipi di selettore:

- Selettore di **elementi**: è costituito da uno qualunque degli elementi di (X)HTML ad esempio **p, table, div**... Con la seguente sintassi di esempio
`div {background : White; font : 12px;}`
- **Raggruppare** gli elementi : se si vogliono assegnare determinati valori uguali ad elementi diversi possiamo raggrupparli ad esempio
`div, p, table {background : White; font : 12px;}`
In questo caso i div, i paragrafi e le tabelle avranno come sfondo il bianco e come dimensione del testo 12pixel
- Selettore **universale** : utilizzando il carattere jolly ***** possiamo infine selezionare tutti gli elementi contenuti in un documento ed assegnare loro proprietà e valori uguali, ad esempio:
`* {color: black;}`
In questo modo assegno a tutti gli elementi del documento il colore nero del testo

Id e Classi: attributi come selettori

Per essere il più chiari possibili questa volta partiamo dall' esempio:

`<p class="normale_arial">...testo.... </p>`

A questo paragrafo gli ho detto di applicare lo stile denominato "arial". Quindi cercherà, all'interno di una dichiarazione di stile (interna o esterna) quella chiamata "normale_arial" che potrebbe essere questa...

```
.normale_arial {  
font: 12px arial, Helvetica, sans-serif;  
color: white;  
}
```

A questo punto il paragrafo avrà come font arial come dimensione del testo 12px e come colore il nero. La classe avrà effetto anche su un altro tag, ad esempio

```
<div class="normale_arial">
```

Avrà lo stesso stile del paragrafo così come tutti gli elementi a cui si può assegnare uno stile.

Lo stesso meccanismo è valido per i selettori di tipo **ID**, con una sola enorme differenza. In un documento HTML l'attributo id è usato per identificare in modo univoco un elemento. In pratica, se assegno ad un paragrafo l'id "normale_arial", non potrò più usare questo valore nel resto della pagina.

Sintassi

Per definire una classe, come si può notare nella precedente definizione, si deve mettere un punto prima del nome della classe

```
.nome_della_classe
```

Per definire un id bisogna far precedere il nome dell' id dal simbolo del cancelletto

```
#nome_dell_id
```

Proprietà e valori vanno, come al solito, inseriti tra parentesi graffe.

Valori ed Unità di Misura

Abbiamo visto che ad ogni proprietà corrisponde un valore; proprietà e valori sono separati dai due punti

```
font-size:12px
```

Nei valori numerici con unità di misura, non bisogna lasciare spazio tra numero e sigla dell'unità.

Si deve scrivere **15px** oppure **5em**. E' invece sbagliato **15 px** o **5 em**.

Ma cosa significa px o em? Semplicemente sono due delle tante unità di misura usate per definire dimensioni, spazi o distanze. Ecco l'elenco completo:

- **in (inches - pollici)**: classica misura del sistema metrico americano. Praticamente nullo il suo valore su un supporto come un browser web viste le variabili relative a risoluzione e ampiezza dei monitor.
- **cm (centimetri)**: stesso discorso visto per i pollici, la difficoltà maggiore sta nella resa su monitor, che è altra cosa rispetto alla carta stampata.
- **mm (millimetri)**: vedi centimetri.
- **pt (points - punti)**: unità di misura tipografica destinata essenzialmente a definire la dimensione dei font.
- **pc (picas)**: unità poco usata. 1 pica equivale a 12 punti.
- **em (em-height)**: unità di misura spesso usata dagli autori CSS. 1 em equivale all'altezza media di un **carattere per un dato font**. E' un unità di misura relativa.
- **ex (ex-height)**: poco usata. 1 ex equivale all'altezza del carattere x minuscolo del font scelto.
- **px (pixels)**: unità di misura ideale su monitor. E' quella più usata e facile da comprendere.

Percentuale: Un valore espresso in percentuale è da considerare sempre relativo rispetto ad un altro valore. Nell' esempio che segue...


```
<div style="font-size:12px;">
<p style="font-size:75%;">CIAO A TUTTI</p>
</div>
```

La frase CIAO A TUTTI risulterà con una dimensione del 75% rispetto ai 12px, cioè 9px
Ma quali misure usare tenendo sempre ben presente il discorso sull'**accessibilità**?

Secondo il W3C (ed anche secondo me) 3 sono le unità di misura da adottare: **px**, **em**, **%**

Una delle necessità, ad esempio, degli ipovedenti è quella di poter ingrandire i testi all'interno di una pagina web, semplicemente aumentando lo zoom della pagina. Per poter mantenere la giusta proporzione tra tutti i contenuti (testo, immagini etc..) all'interno di una pagina bisogna quindi impostare correttamente le unità di misura. Ecco un esempio di codice....

```
<div style="border-width: 2em; width:30em;height:5em;">
<p style="border-style: solid; border-width: 0.3em; font-family:Arial; font-size:1em;">Lorem
ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et
dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.</p>
</div>
```

Utilizzando gli em tutti i contenuti della pagina (in questo caso *div* e *p* ma lo stesso vale per esempio per le immagini) si ridimensionano utilizzando lo strumento ZOOM, del browser. Da notare che anche la dimensione del bordo si ridimensiona automaticamente.

Colore e colori

La prima considerazione da fare è quella di non dimenticarsi dei principi riguardanti l'accessibilità e già richiamati precedentemente

- il testo deve adeguatamente contrastato rispetto allo sfondo
- non affidare le informazioni esclusivamente al colore
- non usare i colori per fornire risposte (tipo una serie di colori da cliccare per stabilire il gradimento di una determinata cosa)

Avvertenza Importante: il contrasto non deve essere il risultato di una semplice differenza di tono dei colori (ovvero di frequenza), come nel caso per esempio di un rosso e di un verde ugualmente saturi, ma sia piuttosto il risultato di una differenza di luminosità. Le differenze di luminosità sono infatti percepibili anche da chi soffre di cecità ai colori (sia pure con delle variazioni sensibili rispetto ai tricromati), mentre le differenze di tonalità possono risultare per loro, in certi casi, del tutto invisibili.

Un'interessante tabella per la scelta dei migliori accoppiamenti di colore per soggetti ipovedenti è stata sviluppata da Franco Frascolla, dell' Associazione Nazionale Subvedenti.

(http://www.diodati.org/scritti/2002/ipovisione/ipo_02.asp#a06)

Per definire i colori non usate mai la proprietà *color*, in quanto definisce esclusivamente il colore di primo piano, ovvero quello del testo e il colore del bordo dell' elemento cui si applica lo stile.

Ciò potrebbe dare luogo a spiacevoli effetti, quindi meglio utilizzare le seguenti proprietà

- il colore di primo piano (*color*)
- il colore di sfondo (*background color*)
- il colore del bordo (*border color*)

in modo da definire esattamente quali sono i colori da utilizzare per l'elemento in questione.

I colori possono essere definiti in due modi

- in formato RGB (Red Green Blue)
- indicandone il nome

Purtroppo la seconda soluzione non è valida per tutti i browser quindi consiglio vivamente di utilizzare la prima soluzione, quindi, ad esempio

```
<div style=" background-color:#00ff00; border-color:#ff0000; color:#000000;">.....</div>
```

Per quanto riguarda lo sfondo esiste anche la possibilità di utilizzare una immagine di sfondo (background-image): questa può essere singola o "moltiplicata" (background-repeat) nel senso che può essere ripetuta fino ad occupare tutto lo spazio a disposizione, ad esempio:

```
<div style="background-image: url(sfondo.gif); background-repeat: repeat;">...</div>
```

In questo caso l'immagine sfondo.gif viene ripetuta sia in orizzontale che in verticale fino a riempire tutto il div.

Testo

Eliminato il tag col css si può dire che abbiamo a disposizione un vero e proprio word-processor per gestire il testo. Le principali proprietà sono:

- il font da usare (**font-family**)

Quando si imposta la proprietà font-family si possono usare tutti i font che si vuole, l'importante è mettere sempre una delle due font comuni a tutti i browser che sono **serif** e **sans-serif** che corrispondono rispettivamente al **times new roman** e all' **arial**

```
<p style="font-family: Arial, Helvetica, sans-serif;"> I font sono separate da virgole; se non è supportata la prima font, viene visualizzata la seconda e via di seguito; per questo è necessario mettere come ultima una delle due font comuni.
```

- la sua dimensione (**font-size**)

Per i valori delle dimensioni vedere il relativo a "Valori ed Unità di misura"

- la sua consistenza (**font-weight**)

Il "peso" visivo di un carattere può essere espresso con una scala numerica o con parole chiave

- **valori numerici** 100 - 200 - 300 - 400 - 500 - 600 - 700 - 800 - 900 ordinati in senso crescente (dal leggero al pesante)
- **normal**. Valore di default. E' l'aspetto normale del font ed equivale al valore 400
- **bold**. Il carattere acquista l'aspetto che definiamo in genere grassetto. Equivale a 700
- **bolder**. Misura relativa. Serve a specificare che una determinata porzione di testo dovrà apparire più pesante rispetto al testo dell'elemento parente
- **lighter**. Misura relativa. Il testo sarà più leggero di quello dell'elemento parente

```
<p style="font-weight:bold;">
```

- l'interlinea tra i paragrafi (**line-height**)

In pratica fissa lo spazio tra le righe all'interno di un elemento (p, div.. etc); possiamo utilizzare sia un valore numerico che un valore con unità di misura.

```
<p style="line-height: 2;">
```

```
<p style="line-height: 14px;">
```

- l'allineamento del testo (**text-align**)

- la sua decorazione (sottolineature e simili) (**text-decoration**)

Box-Model : gestire le Dimensioni

Un box-model è praticamente un elemento dimensionabile come un paragrafo, un titolo ed, esempio più calzante, un div. Ogni box comprende un certo numero di componenti di base, ciascuno modificabile con proprietà dei CSS: ecco i principali componenti:

- Area del Contenuto: è la zona in cui trova spazio il contenuto vero e proprio vale a dire il testo, le immagini, i video etc.... Le dimensioni orizzontali dell'area possono essere modificate con la proprietà width. Quelle verticali con height.
- Padding: è lo spazio tra il bordo dell'elemento e l'area del contenuto

- Bordo: è una linea di dimensione, stile e colore variabile che circonda l'elemento. La sua dimensione viene definita con l'attributo margin.

Vediamo ora un esempio chiarificatore:

```
<div style="margin : 20px; border : 10px solid #CCCCCC; padding : 10px; width : 150px; height : 150px;">
```

Per calcolare la larghezza del box-model quindi bisogna sommare:

margine sinistro (**margin**) 20

padding sinistro (**padding**) 10

bordo sinistro (**border**) 10

larghezza (**width**) 150

margine destro (**margin**) 20

padding destro (**padding**) 10

bordo destro (**border**) 10

La larghezza totale quindi è di 230 pixel. E l'altezza....calcolatela voi.....

Ereditarietà nei CSS - Padri figli e nipoti....

Questa penso sia la parte più "tecnica" e "ostica" di tutto il manuale, ma comunque va affrontata. L'ereditarietà fa sì che le impostazioni dello stile di un elemento vengano applicate anche ai suoi discendenti, fino a quando, per un elemento discendente, non si imposti esplicitamente un valore diverso per quella proprietà.

Subito un esempio: impostando il colore nero per il testo (`color: black;`) a livello dell'elemento BODY tutti gli altri elementi suoi discendenti, cioè tutti gli elementi contenuti nel BODY, ereditano questa impostazione. Se ad un certo punto definiamo nel codice del CSS un elemento (ad esempio in un paragrafo) con la proprietà (`color: red;`) l'ereditarietà viene spezzata.

Conflitti : Nel caso di istruzioni che entrino in conflitto tra di loro (foglio di stile e istruzioni di stile all'interno della pagina html per esempio) l'ordine di importanza è il seguente:

- Foglio di stile (css)
- Stile all'interno della pagina (`<style..... >Istruzioni </style>`)
- Stile del browser

Quest'ultimo entra in funzione in caso di assenza di istruzioni specifiche all'interno del foglio di stile ed all'interno della pagina. In Explorer, ad esempio lo stile è definito all'interno di Opzioni Internet- Generale - Aspetto.

CONCLUSIONI

Questo non è un manuale didattico, ma ho voluto focalizzare l'attenzione di chi si accinge a "creare" un sito web su alcuni aspetti della Accessibilità ed Usabilità.

Se avete appunti da farmi, volete chiarire qualche concetto, segnalare errori od inesattezze, questa è l'email: frever@libero.it