

CASCADING STYLE SHEET (CSS)

LE REGOLE CSS

I fogli di stile sono un insieme di **regole**, le quali istruiscono il browser su come rappresentare le diverse **proprietà** dei vari elementi del documento XHTML. Una regola CSS è strutturata come rappresentato seguito :

```
selettore {  
  proprietà: valore;  
  proprietà: valore;  
  proprietà: valore;  
}
```

Esempio: **p { font-size: 1em }**

p	= selettore
{ font-size: 1em }	= dichiarazione
font-size	=proprietà
1em	=valore

La regola sopra riportata è costituita da un **selettore** e una dichiarazione chiusa fra parentesi graffe '{ }'. Il selettore specifica a quale elemento del documento XHTML deve essere applicata la regola. La dichiarazione è costituita da una **proprietà**, la quale specifica quale aspetto del selettore deve essere regolato, e un **valore**, separato dalla proprietà dal carattere ':', che specifica lo stile che deve essere assunto dalla proprietà a cui è associato. Ogni regola può essere associata a uno o più selettori separati da una virgola e può essere costituita da più dichiarazioni separate dal carattere ';'.

Raggruppamento

E' molto utile e di uso comune assegnare una regola a più selettori senza dover replicare il codice css che la riguarda. Basta riportare, prima della graffa aperta, tutti i selettori coinvolti separati da una virgola. Esempio:

```
selettore1, selettore2, selettore3 {  
  proprietà: valore;  
  proprietà: valore;  
  proprietà: valore;  
}
```

Se il raggruppamento totale ha degli indubbi vantaggi, il raggruppamento parziale è da usare con cautela e solo sotto alcune semplici condizioni, cioè nel caso in cui le proprietà in comune siano una maggioranza significativa e le proprietà differenti siano al massimo una o due, e vengano specificate nel foglio di stile immediatamente sotto la regola di raggruppamento. Ecco un esempio in cui il raggruppamento parziale è accettabile:

```
selettore1, selettore2 {  
  proprietà: valore;  
  proprietà: valore;  
  proprietà: valore;  
  proprietà: valore;  
}
```

```
selettore1 {  
  proprietà-x: valore;  
}
```

```
selettore2 {  
  proprietà-x: valore;  
}
```

Selettore di tipo (type selector)

Ridefinizione dei tag XHTML

Le regole specificate tramite selettori di tipo verranno applicate a tutti le istanze di elementi associati ad un dato tag. In sostanza attraverso i selettori di tipo è possibile attribuire una regola di presentazione per i tag XHTML. La regola dell' esempio seguente consentirà di mostrare tutti i paragrafi di una pagina XHTML con sfondo giallo e testo blu.

```
p {  
  background-color: yellow;  
  color: blue;  
}
```

Selettore di classe (class selector)

Serve quando si ha la necessità di associare ad uno stesso tag XHTML differenti stili. Nei CSS i selettori class utilizzano “ . ” (il punto)

```
p.centrato {  
  text-align: center;  
}
```

In questo caso il paragrafo di classe CENTRATO avrà il testo allineato al centro; per richiamarlo nel foglio xhtml si inserirà :

```
<p class="centrato"> son il testo centrato </p>
```

Altro esempio:

```
.contenitore {
    color: rgb(158,0,0);
    border: 1px solid Red;
    padding: 3px;
    margin-top: 05em;
    background: #f9f5de;
}
```

`<p class="contenitore"> sono un testo dentro un contenitore </p>`

Selettore di ID (id selector)

I selettori di ID permettono di assegnare una regola CSS ad un unico elemento nella pagina XHTML che è specificato attraverso l' attributo ID. Comunemente sono utilizzati per applicare uno stile a BOX specifici. Nei CSS i selettori ID utilizzano " # " (il cancelletto)

CODICE CSS

```
#colonna-sx {
float: left;
width: 20%;
}
```

CODICE XHTML

```
<div id="colonna-sx">
  <!-- eccetera -->
</div>
```

Si può specificare un selettore ID nei seguenti modi:

```
#menu {
  proprietà: valore;
  proprietà: valore;
}
```

oppure

```
div#menu {
  proprietà: valore;
  proprietà: valore;
}
```

CSS esterni e interni

E' esterno un foglio di stile definito in un file separato dal documento. Si tratta di semplici documenti di testo editabili anche con il Blocco Note o TextEdit ai quali si assegna l'estensione .css.

Un foglio di stile si dice invece interno quando il suo codice è compreso in quello del documento. A seconda che si lavori con un CSS esterno o interno variano sintassi e modalità di inserimento. Rispetto a queste diverse modalità si parla di fogli di stile collegati, incorporati o in linea.

Fogli collegati

Per caricare un foglio esterno in un documento esistono due possibilità. La prima e più compatibile è quella che fa uso dell'elemento <LINK>. La dichiarazione va sempre collocata all'interno della sezione <HEAD> del documento (X)HTML:

```
<html>
<head>
<title>Inserire i fogli di stile in un documento</title>
<link rel="stylesheet" type="text/css" href="css/nome-file.css">
</head>
<body>...
```

L'elemento <LINK> presenta una serie di attributi di cui è importante spiegare significato e funzione:

1. **rel:** descrive il tipo di relazione tra il documento e il file collegato. E' obbligatorio. Per i CSS due sono i valori possibili: stylesheet e alternate stylesheet. Approfondimenti nella lezione 5.
2. **href:** serve a definire l'URL assoluto o relativo del foglio di stile. E' obbligatorio.
3. **type:** identifica il tipo di dati da collegare. Per i CSS l'unico valore possibile è text/css. L'attributo è obbligatorio.
4. **media:** con questo attributo si identifica il supporto (schermo, stampa, etc) cui applicare un particolare foglio di stile. Attributo opzionale. L'argomento sarà approfondito nella prossima lezione.

Usare @import

Un altro modo per caricare CSS esterni è usare la direttiva @import all'interno dell'elemento <STYLE>:

```
<style type="text/css" media="screen" title="normale">
  @import url(css/nome-file.css );
</style>
```

Fogli incorporati

I fogli incorporati sono quelli inseriti direttamente nel documento (X)HTML tramite l'elemento <STYLE>. Anche in questo caso la dichiarazione va posta all'interno della sezione <HEAD>:

```
<html>
<head>
<title>Inserire i fogli di stile in un documento</title>
<style type="text/css">
body {
background: #FFFFCC;
}
</style>
</head>
<body>...
```

Come si vede il codice inizia con l'apertura del tag <STYLE>. Esso può avere due attributi:

1. type (obbligatorio)
2. media (opzionale)

per i quali valgono le osservazioni viste in precedenza. Seguono le regole del CSS e la chiusura di </STYLE>.

Fogli in linea

L'ultimo modo per formattare un elemento con un foglio di stile consiste nell'uso dell'attributo style. Esso fa parte della collezione di attributi (X)HTML definita Common: si tratta di quegli attributi applicabili a tutti gli elementi. La dichiarazione avviene a livello dei singoli tag contenuti nella pagina e per questo si parla di fogli di stile in linea. La sintassi generica è la seguente:

```
<elemento style="regole_di_stile">
```

Se, ad esempio, si vuole formattare un titolo H1 in modo che abbia il testo di colore rosso e lo sfondo nero, scriveremo:

```
<h1 style="color: red; background: black;">...</h1>
```

Le cose da osservare nel codice sono due. Come valore di style si possono dichiarare più regole di stile. Esse vanno separate dal punto e virgola. I due punti si usano invece per introdurre il valore della proprietà da impostare.

Sintassi

L'attributo media può accompagnare sia l'elemento <LINK> che l'elemento <STYLE>. Ecco due esempi di sintassi:

```
<link rel="stylesheet" type="text/css" media="print" href="print.css" />  
<style type="text/css" media="screen">...</style>
```

Per sfruttare a fondo questa opzione è fondamentale conoscere i diversi valori possibili per l'attributo:

- all. Il CSS si applica a tutti i dispositivi di visualizzazione.
- screen. Valore usato per la resa sui normali browser web.
- print. Il CSS viene applicato in fase di stampa del documento.
- projection. Usato per presentazioni e proiezioni a tutto schermo.
- aural. Da usare per dispositivi come browser a sintesi vocale.
- braille. Il CSS viene usato per supporti basati sull'uso del braille.
- embossed. Per stampanti braille.
- handheld. Palmari e simili.
- tty. Dispositivi a carattere fisso.
- tv. Web-tv.

Il valore di default è all, usato automaticamente in mancanza di una dichiarazione esplicita (ricordiamo infatti che media è un attributo opzionale). E' possibile usare più di un valore, ma i nomi della lista vanno separati da una virgola:

```
<link rel="stylesheet" type="text/css" media="print, tv, aural" href="print.css" />
```

L'uso più tipico di questa funzionalità consiste nel collegare al documento vari fogli di stile adatti a ciascun supporto. Lo user agent che visualizzerà la pagina sarà in grado, se conforme agli standard, di caricare quello giusto:

BOX MODEL - BORDI E PADDING

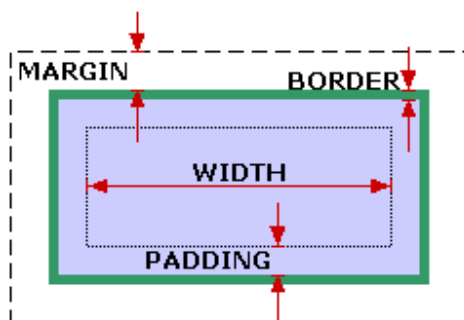
Nel layout da realizzare, le sezioni sono separate anche da bordi e il testo è da questi distanziato impostando il padding.

Ricordatevi che se bordi e padding venissero aggiunti associando le proprietà ai selettori **#principale** e **#colonna-dx** il layout si allargherebbe oltre lo spazio concessogli, poiché, la larghezza del padding e di bordi si aggiungerebbe a quella definita con la proprietà **width**.

.... Il box model dei CSS permette di definire dei blocchi rettangolari con specifici valori per la larghezza e l'altezza della sezione contenuti, del padding, dei bordi, dei margini. Purtroppo, explorer per Windows non ha una interpretazione corretta del box model. E' molto importante conoscere il problema, essendo il box model uno dei migliori strumenti offerti dai CSS ed essendo explorer per Windows ancora il browser più diffuso.

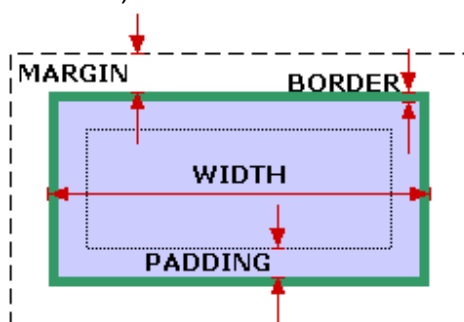
Il box model viene specificato impostando la larghezza e l'altezza dei contenuti, il padding, dimensioni e stile del bordo e il margine. Il padding è lo spazio fra i contenuti e il bordo, mentre il margine è lo spazio fra il bordo e gli altri contenuti della pagina.

Quando, con la proprietà 'width', si specifica la larghezza del box, si specifica la larghezza dell'area che conterrà il testo del box. La larghezza del box fino al bordo è la somma della larghezza specificata con width + ampiezza del padding + spessore del bordo (vedi immagine successiva)..... (CSS hoepli)



Questo secondo gli standard CSS.

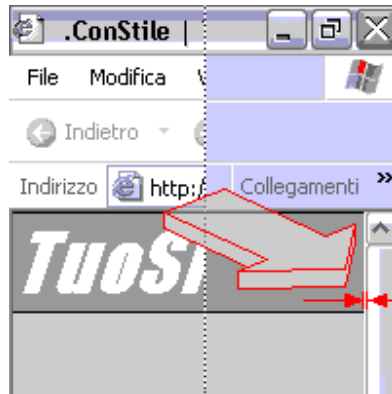
EXPLORER per Windows (non per Mac) considera la larghezza specificata con width, come la larghezza del box fino al bordo. Bordo e padding vengono scalati dalla larghezza specificata (vedi immagine successiva).



La soluzione al problema è semplice: all'interno degli elementi come `<div id="principale">`, `<div id="colonna-dx">` vanno aggiunte delle sezioni contraddistinte dalla classe **box** che serviranno per impostare il padding e i bordi.

In realtà c'è un problema: Internet Explorer lascia, sulla destra, un margine di 1px. Ciò fa sì che la colonna sulla destra non ha a disposizione il 20% della pagina, ma il 20% meno 1px. La colonna di destra, dunque, non affiancherebbe le altre due, ma verrebbe a trovarsi al di sotto della colonna di sinistra. Per ovviare al problema è sufficiente impostare il margine destro della colonna di destra a -1px.

```
#colonna-dx { margin-right:-1px; }
```



Affinché il nostro layout occupi tutto lo spazio a disposizione bisogna impostare a zero il margine e il padding del BODY. Bisogna però sottolineare che, avendo posto a -1px il margine destro della colonna di destra, a causa dei soliti problemi nell'interpretazione del box model di Internet Explorer, apparirà la barra di scorrimento orizzontale. Per ovviare a questo problema è sufficiente impostare il margine destro del BODY a 1px:

```
body  
{ margin:0 1px 0 0; }
```

Nel nostro esempio abbiamo inserito
body
{ margin:0 1px 0 0; }

E subito dopo al super contenitore di tutto
#corpo {
margin: 0px -1px 0px 0px;

Il codice CSS per il link (menu verticali e orizzontali)

LA FORMATTAZIONE DEI LINK

-I link sono quanto di più irritante si possa trovare nel trattamento tipografico sul Web, soprattutto perché (tra le altre cose) cambiano il loro aspetto dinamicamente, cosa di cui la tipografia tradizionale non si era mai dovuta preoccupare. Tim Berners-Lee pensò che i link dovessero essere sottolineati; non fu una cattiva idea: la sottolineatura è rara in tipografia, e quindi non c'era rischio di possibili associazioni logiche. Decise anche che dovessero essere blu il che, ma se ne può discutere, non è stato forse altrettanto saggio. Questo è uno dei rari casi in cui l'uso di due variabili (il colore e la sottolineatura) per differenziare un elemento funziona veramente: una variabile (il colore) vi dice che l'elemento è diverso dal testo circostante e vi da informazioni di stato (visitato, non visitato); l'altra variabile indica che si tratta di un elemento dinamico.

Dal momento che per default il colore di un link cambia in funzione del suo stato, mentre la sottolineatura non cambia, è quest'ultima che ha assunto il significato quasi universale di link. Per questa ragione, di norma, conviene evitare l'uso della sottolineatura per cose diverse dai link. - (CSS hoepli)

PERCHÉ MODIFICARE L'ASPETTO DEI LINK?

Se si usano colori o immagini particolari per lo sfondo della pagina o di una tabella, modificare i colori predefiniti risulta l'unico modo per visualizzare correttamente questi fondamentali elementi. Grazie ai CSS è possibile assegnare colori, decorazioni del testo, font o colori di sfondo diversi a seconda dello stato del link. Nel linguaggio dei CSS le configurazioni possibili dei link sono cinque e rientrano in una speciale categoria, quella delle **pseudoclassi**. Esse definiscono uno stile per un elemento al verificarsi di certe condizioni. La sintassi è: "elemento:condizione".

Nelle specifiche dei CSS si distinguono cinque possibili **condizioni** per i link:

1. **a:link** - Definisce un link non ancora visitato.
2. **a:visited** - Definisce un link visitato, ovvero presente nella memoria del browser.
3. **a:hover** - Definisce lo stato del link quando l'utente sposta il cursore su di esso (compare la manina).

Tuttavia, seppur una minoranza, non tutti gli utenti utilizzano il mouse per selezionare i link, ma utilizzano invece il tab della tastiera e dato che la classe 'hover' è legata allo strumento con cui si interagisce (il mouse), con una navigazione da tastiera i link non vengono differenziati come avverrebbe con il mouse.

La soluzione sarebbe utilizzare la pseudo classe 'focus'.

Che però non è supportata da Internet Explorer. Per ottenere lo stesso effetto anche su IE si usa la pseudo classe 'active'.

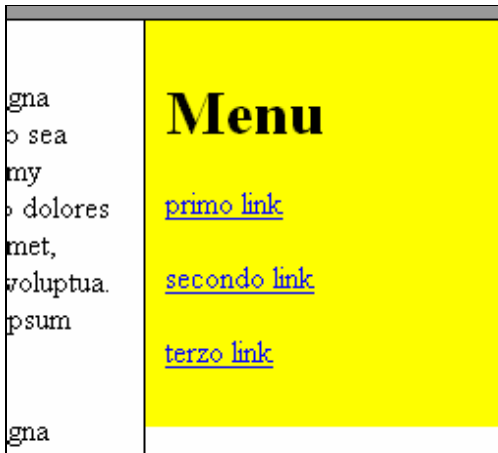
5. **a:focus** - E' importante questa condizione perché è abbinata al link puntato tramite la tastiera con il tasto tab (tabindex) o l'accesskey.

4. **a:active** - Definisce lo stato del link mentre il tasto del mouse è premuto e non rilasciato - o puntato da "focus" -

Inseriamo il seguente codice html nel menu verticale:

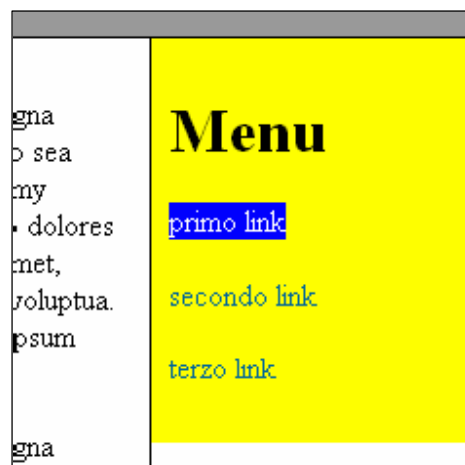
```
<a href="...">primo link</a>  
<a href="...">secondo link</a>  
<a href="...">terzo link</a>
```

Questo sarà il risultato:



Per modificare i colori nelle condizioni definiamo le pseudoclassi nei nel nostro foglio stile:

```
/*---- LINK DA VISITARE ----*/
a.:link {
color: #006699;
text-decoration: none;
}
/*---- LINK VISITATO ----*/
a.:visited {
color: Lime;
text-decoration: none;
}
/*---- LINK PUNTATO DAL MOUSE ----*/
a: hover {
color: White;
background-color: Blue;
text-decoration: none;
}
/*---- LINK ATTIVO ----*/
a.:active {
color: Black;
background-color: Fuchsia;
text-decoration: none;
}
/*---- LINK PUNTATO TRAMITE TASTIERA - TASTO TAB ----*/
a.:focus {
color: Black;
background-color: Aqua;
text-decoration: none;
}
```



MENU GRAFICI CON ROLLOVER

Orientare l'utente nella navigazione

Prima di addentrarci nella realizzazione pratica dei menu, una piccola introduzione sull'usabilità di un menu di navigazione. Ci sono alcuni principi di cui si dovrebbe tenere sempre conto:

- l'utente dovrebbe poter capire facilmente dove può andare: è quindi indispensabile avere una navigazione chiara e ben distinta dal resto della pagina, sia a livello spaziale che di colore;
- è importante distinguere i link al passaggio del mouse: questo per far capire all'utente che il cambiamento di stato del link, grazie al rollover, indica un possibile cambiamento di pagina;
- l'utente dovrebbe poter capire immediatamente dove si trova: è quindi importante distinguere la pagina/sezione corrente tramite il menu.
- Il menu deve essere accessibile anche se il foglio stile non viene caricato (non vedenti)

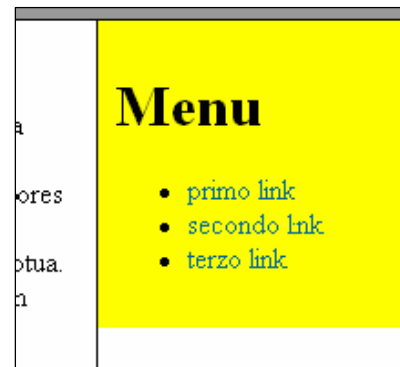
IL MENU COME LISTA

Per rendere i menu verticali compatibili con tutti i browser il codice (X)HTML è stato reso più semplice e chiaro possibile usando le liste. Il codice (X)HTML di un menu sarà, ad esempio, il seguente:

```
<ul >
  <li ><a href="...">primo link</a></li>
  <li><a href="...">secondo link</a></li>
  <li><a href="...">terzo link</a></li>
</ul>
```

Senza le regole CSS il risultato (semplice ma compatibile con qualsiasi browser e/o dispositivo per la navigazione in internet, compresi screen reader e browser testuali come Lynx) sarebbe il seguente:

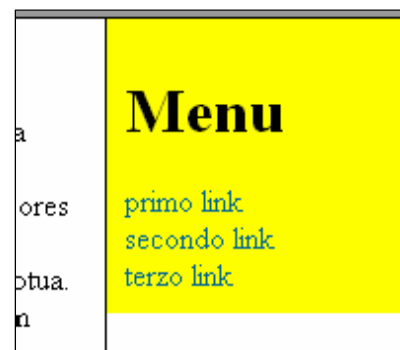
Attraverso le regole CSS trasfo



renderemo una semplice lista di link in un semplice menu verticale. Per fare ciò elimineremo i marcatori di lista e la sottolineatura, aggiungeremo un effetto roll-over realizzato con i soli CSS senza l'uso del JavaScript e renderemo il link attivo non solo sul testo ma su tutta l'area orizzontale associata al link.

Per realizzare tale trasformazione abbiamo adottato le seguenti regole CSS.

```
div#menu {
color: #000;
}
div#menu ul {
list-style: none; /* elimino i punti elenco */
margin: 0px 0px 0px 0px; /* elimino i rientri delle liste */
padding: 0px 0px 0px 0px; /* elimino i rientri delle liste */
```

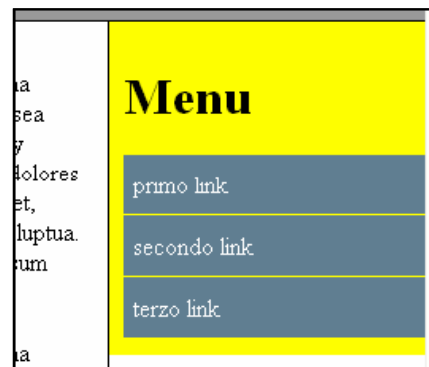


```

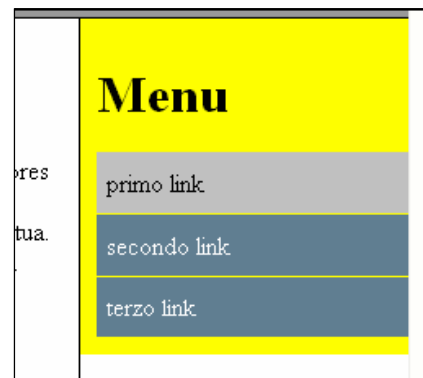
}
Innanzitutto impostiamo il contenitore del menu, e definiamo le caratteristiche del nostro
marcatore:
eliminiamo i marcatori della lista ed annulliamo l'indentazione tipica delle liste.
A questo punto il menu appare con una semplice list di link, senza marcatori né indentazione.
div#menu li{
margin: 0px;
padding: 0px 0px 1px 0px;
}
div#menu li a {
padding: 3px 6px 3px 6px;
display: block; /* il link occupa tutto lo spazio orizzontale */
color: white;
background-color: #607e91;
    text-decoration: none; /* elimino la sottolineatura, che si tratta di un link si capisce dalla
    struttura */
height:2em; line-height:2em; /* altezza del link: height e line-height devono avere il medesimo
valore */
} CSS - Regole pag. - 9

```

Attraverso la dichiarazione `display:block`, il link occuperà tutta l'area orizzontale quindi non sarà cliccabile solo il testo del link ma tutta l'area della riga occupata dal link. Si elimina poi la sottolineatura del link poiché sarà la struttura stessa del menu ad indicare la presenza di collegamenti. Attraverso il `padding` si discostano i link dal bordo del menu. Utilizzando le proprietà `height` e `line-height` è possibile determinare l'altezza dell'area occupata da ogni singolo link. Ciò si dimostra molto utile anche per un piacevole "effetto collaterale" che annulla un fastidioso bug di Internet Explorer. Su quest'ultimo browser, infatti, il link non occupa tutta l'area (nonostante la dichiarazione `display:block`) se è contenuto in un elemento di cui si è fissata la larghezza (come appunto l'elemento `ul.menu1`). Lo svantaggio di specificare l'altezza (necessario per sopperire ad un bug di Internet Explorer) consiste nel fatto che se il testo del link non entra tutto in una sola riga si possono avere spiacevoli sovrapposizioni fra link adiacenti.



Infine si crea l'effetto roll-over, sfruttando la pseudo-classe `a:hover` semplicemente giocando con i colori dello sfondo e del testo dei link.



Gli effetti roll-over sono stati ottenuti, con evidenti vantaggi, senza utilizzare il Javascript. L'usabilità e l'accessibilità dei menu è integra, si noti infatti come la stessa pagina priva dei CSS sia perfettamente utilizzabile con ogni tipo di browser.



ETICHETTE - MENU ORIZZONTALI

Prendiamo spunto dal menu verticale per realizzare il menu orizzontale.

Inseriamo il codice html:

```
<h1>testata</h1>
  <div id="nav">
    <ul >
      <li><a href="index.html" >Home</a></li>
      <li><a href="mappa.html" >Mappa</a></li>
      <li><a href="aiuti.html" >Aiuti</a></li>
    </ul>
  </div>
```

Ed il codice css:

```
/* === NAVIGAZIONE ORIZZONTALE === */
div#nav{
  color: #333366;
  background-color: #efefef;
  height: 20px;
}
div#nav ul {
  list-style: none; /* elimino i punti elenco */
  margin: 0px 0px 0px 0px; /* elimino i rientri delle liste */
  padding: 0px 0px 0px 0px; /* elimino i rientri delle liste */
}
div#nav li {
  display: inline; /* mette gli elementi della lista in linea */

padding: 2px 5px 0px 5px;
margin: 0px 0px 0px 5px;
}
```

```
div#nav li a{
color: #333366;
background-color: #efefef;
text-decoration: none;
}
div#nav li a:hover, div#nav li a:focus, div#nav li a:active {
color: #333366;
background-color: #fff;
border-bottom: 5px solid #A5A5A5;
}
```

L'unica cosa importante e diversa è la linea evidenziata.



Se vogliamo creare delle linguette della stessa lunghezza dobbiamo inserire i seguenti comandi
In

```
div#nav li a
display: block;
width: 65px; (lunghezza linguetta)
```