

Template css per Joomla

di

Carlo Ernesto Campins Candia

Indice generale

Costruire un template per joomla.....	1
CAPITOLO 0.....	3
Introduzione:.....	3
- Perché questo manuale?.....	3
- Chi sono?.....	3
- Licenza.....	3
CAPITOLO 1.....	4
cos'è un template:.....	4
- xhtml VS html.....	4
- css layout(100%)?.....	5
CAPITOLO 2.....	6
Primi passi:.....	6
- Organizziamo il lavoro.....	6
- Creare il template.....	7
CAPITOLO 3.....	15
personalizzare i moduli:.....	15
- Modificare log-in.....	15
- Modificare il menù.....	17
Moduli:.....	19
- La modifica del Modulo Login Form.....	19
- La modifica del modulo Main Menù.....	22

CAPITOLO 0

Introduzione:

- Perché questo manuale?

Mi sono deciso a scrivere questo manualetto per dare la possibilità, a utenti non evoluti, di modificare la parte grafica del cms Joomla. Questo perché in molte situazioni io stesso mi sono trovato in difficoltà nella sua creazione, nonostante il contributo dei diversi forum in primis quello della comunità italiana (<http://www.joomla.it>).

Altro motivo riguarda la mia gratitudine ai programmi open source e free che mi hanno dato la possibilità di imparare osservando il codice e la documentazione.

Sono convinto che dare ad altri il proprio sapere consolida la mia conoscenza dell'argomento, questo perché sono costretto a focalizzare il problema per poterlo spiegare. Inoltre tutto ciò permette un confronto con altre persone che utilizzano tecniche differenti per fare la stessa cosa. Avere un punto di partenza permette di criticare e differenziarsi qualsiasi cosa in modo da evolve, si spera, il sapere.

Spero siano cosa gradita queste poche pagine suscettibili di vostre critiche e cambiamenti.

- Chi sono?

Mi chiamo CarloErnesto, ho 38 anni (non chiamatemi zio :-)) sono un'appassionato d'informatica che si è formato da autodidatta. Conosco xhtml e i css (mi piacciono molto) per quanto riguarda la programmazione recentemente mi sono dato al linguaggio java e php. Utilizzo come database Mysql e come server Apache.

Lavoro su sistema operativo Gnu/Linux utilizzando programmi come Gimp (<http://www.gimp.org/>) per il foto ritocco, Blender (<http://blender.org/cms/Home.2.0.html>) per la grafica 3d, Inkscape (<http://inkscape.org/>) per grafica vettoriale e Scribus (<http://www.scribus.org.uk/>) per l'impaginazione. Per quanto riguarda la suite d'ufficio uso al nuova versione di Openoffice (<http://it.openoffice.org/>). Come avete potuto intuire, cerco di utilizzare solo programmi free e open source, programmi che se usati bene non hanno niente da invidiare a software molto costoso. Naturalmente conosco anche i blasonati Dreamweaver, Flash e Fireworks.

Se volete contattarmi per dare dei suggerimenti, correzioni varie o aggiunte di capitoli, scrivete pure a:

ernesto.c@libero.it

- Licenza

Questo manualetto è scritto sotto la licenza Creative Commons, un copyright flessibile per opere creative (<http://www.creativecommons.it/>) il che vuol dire che potete utilizzarlo, manipolarlo o buttarlo via ;-). In ogni caso vi pregherei di lasciare il nome dell'autore e degli eventuali revisori come da licenza.

CAPITOLO 1

cos'è un template:

- xhtml VS html

Un template è un modello! cioè un qualche cosa che serve per riprodurre tante altre cose in serie. Immaginiamo una formina in metallo che usano le mamme per fare le torte, quindi, possiamo paragonare il template alla formina e le torte alle pagine. Spero che il paragone sia chiaro, con la nostra formina possiamo fare tante torte proprio come con il nostro template possiamo fare tante pagine.

Premetto che non utilizzo applicativi per fare pagine web, anche se conosco bene questi strumenti, mi piace vedere come dal codice si formano le cose. Uso semplicemente degli editor di testo come scite o kate nel mondo Linux o htmlkit per Windows. Credo che sia importante per una persona conoscere veramente quello che i linguaggi fanno in modo da essere "padroni" del risultato. Ho degli amici che usa i famosi tool per fare pagine web ottenendo dei risultati straordinari, ma non conoscono un solo tag html e quando si trovano in difficoltà perché non riescono a risolvere dei problemi, indovinate chi chiamano?

E' anche vero che spesso per essere creativi è meglio non conoscere quello che c'è dietro...ma come in tutte le cose penso che ci voglia equilibrio e anche molta curiosità.

Sostanzialmente un template serve per far convivere parti statiche (il template) e parti dinamiche di un sito (codice php) messi insieme si darà vita al contenuto del sito. Col termine dinamico si fa riferimento a qualcosa che cambia nel corso del tempo, nel nostro caso i cambiamenti possono avvenire anche in pochi secondi, tutto dipende dal sito, pensiamo ai siti d'informazione on-line che in tempo reale devono informare i lettori.

Html o xhtml? Rispondo subito dicendo xhtml!! Semplicemente perché è il formato del futuro basato su standard xml. Sostanzialmente una pagina web può essere scritta in xml con abbinato un foglio di stile e un dtd di seguito un esempi valido di pagina xml:

```
<pagina>
  <titolo>
    venendo andando
  </titolo>
  <sottotitolo>
    storia di un piccione
  </sottotitolo>
  <paragrafo>
    bla bla
    bla bla bla bla bla bla bla bla bla bla bla bla
    bla bla bla bla bla bla bla bla bla bla bla bla
    bla bla bla bla bla bla bla bla bla bla bla bla
  <collegamento>
    www.miapagina.it
  </collegamento>
  bla bla
  bla bla bla bla bla bla bla bla bla bla bla bla
  bla bla bla bla bla bla bla bla bla bla bla bla
  bla bla bla bla bla bla bla bla bla bla bla bla
</paragrafo>
  bla bla
  bla bla bla bla bla bla bla bla bla bla bla bla
  bla bla bla bla bla bla bla bla bla bla bla bla
  bla bla bla bla bla bla bla bla bla bla bla bla
<paragrafo>
</paragrafo>
</pagina>
```

Questa è una pagina xml funzionante e se io aggancio un css posso modificare la grafica (testo, colori, distanze dei paragrafi, ecc..). Posso creare un sito intero con tag personalizzati e tutto funzionerebbe bene.

Ma chi fa gli standard, in questo caso il w3c (<http://www.w3.org/>) si è reso conto che esisteva di già un linguaggio comune per le pagine web che faceva già queste cose, il suo nome è HTML, occorreva solo dargli nuovamente la sua funzione originale, cioè formattare il contenuto con titoli, paragrafi, elenchi puntati, link, ecc... e non permettergli più di occuparsi dell'aspetto grafico, eliminando tutti quei tag (es:) che hanno una funzione di presentazione. Nacque così XHTML che è sostanzialmente un'evoluzione dell'html su standard xml.

Cosa più importate per il web del futuro è la semantica dove ogni tag ha un significato e può contenere solo determinati dati con una propria grammatica(ecco la funzione del dtd o xslt). Rischio di perdere il nostro obiettivo quindi vi do un link (<http://www.html.it/xml/guida/>) per curiosare.

Spero di non aver creato ulteriore confusione. Per approfondire l'XHTML (<http://www.html.it/xhtml/>).

Detto questo nulla vieta l'utilizzo di html e le odiate tabelle che ci accompagneranno ancora per un po' di tempo.

- css layout(100%)?

Premessa dovuta, joomla non possiede ancora un layout css al 100%, con queste parole intendo dire che non è ancora possibile fare un bel layout senza tabelle. Cercherò di spiegarvi meglio, per quanto io mi sforzi di lavorare con i css e costruire un lay-out tableless (senza tabelle), sono destinato a fallire perché al suo interno i componenti di joomla come menù, log-in, banner, ecc... vengono ancora impaginati con <table>, <tr> e <td>.

Scaricatevi e installate l'ottimo template tableless che ha costruito lonely utilizzando i css (http://www.joomla.it/index.php?option=com_smf&Itemid=17&expv=0&topic=1497.from0#new). Potrete notare che nonostante l'eccezionale lavoro, le tabelle non spariscono.

Per dirla tutta, non è difficile modificare il codice con <div>, , e ma il tutto richiede una buona conoscenza di php, xhtml e css, conoscenze che servirebbero per modificare gli array[] dei componenti. Osservate il seguente codice php immerso nei <tr> e <td>:

```
...
<table>
  <tr>
    <td align="center">
      <?php echo _USERNAME; ?>
      <br />
    </td>
    <td align="center">
      <?php echo _PASSWORD; ?>
      <br />
    </td>
  </tr>
  <tr>
  ...
```

Come si può notare un codice così non rientra nei parametri dell'accessibilità di un sito da parte di certe categorie di utenza (<http://www.diodati.org/>).

Per modificar il codice sopra esposto naturalmente occorre tener conto della compatibilità tra i diversi browser ecc... Non ultimo le modifiche renderebbero difficoltoso un eventuale upgrade di versioni future di joomla.

Leggendo la roadmap dei ragazzi del dev-team di joomla questa situazione dovrebbe cambiare al più presto, quindi tutta la parte grafica sarà effettivamente gestibile (layout senza tabelle) dai css. Non vedo l'ora...

CAPITOLO 2

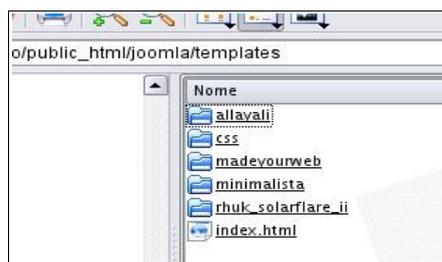
Primi passi:

- Organizziamo il lavoro

Creeremo un semplice template minimalista per concentrare l'attenzione sul procedimento piuttosto che sulla grafica.

I passi da seguire sono:

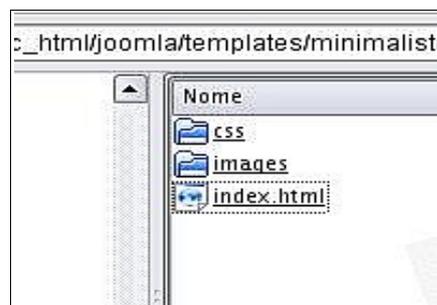
A) Creare una cartella all'interno della directory "templates" la chiameremo per esempio "minimalista".
Portiamoci al suo interno in modo da inserire delle cartelle e dei files.



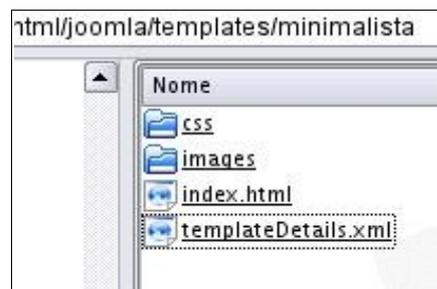
B) Creo un file index.html che non contiene niente (bianco) questo serve per ragioni di sicurezza. Se qualcuno entra nella cartella il server restituisce o la pagina index.php o index.html. Provate a non mettere niente e vedrete cosa succede :-)

C) Create una cartella "images" questa directory conterrà le immagini del nostro template.

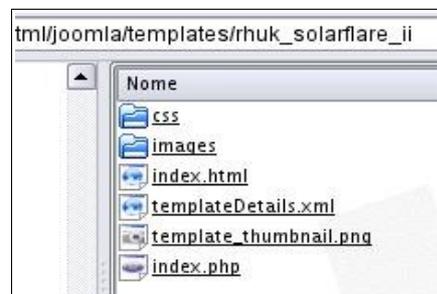
D) Create una cartella "css" questa directory conterrà il css del nostro template.



E) Copiate il file "templateDetails.xml" da un'altro template e incollatelo nella cartella "minimalista". Questo file dovrà essere inseguito editato per adattarlo alle nostre esigenze.



F) Fate un copia e incolla del file "index.php" di un template esistente per esempio "rhuk_solarflare_ii". Da questo file estrarremo la base del nostro template.



Spero sia tutto chiaro fino a questo punto. Qualche dubbio? bene nessuno mi chiede niente :-)
Possiamo proseguire...

- Creare il template

Con un editor di testo apriamo il file "index.php" copiato in precedenza. Commentiamo tutto quello che è presente tra i tag <body> e </body> in questo modo:

```
<body>
<!--
codice html...
-->
</body>
```

Questo viene fatto perché è meglio conservare il codice html e soprattutto i tag php che caricano i vari moduli, in questo modo potremo copiarli e inserirli in seguito.

Torniamo per un momento al nostro file xml "templateDetails.xml". iniziamo a modificarlo inserendo il nome del template, il creatore la data, ecc... Non mi soffermo ulteriormente su questo file perché come potete vedere basta modificare il testo contenuto all'interno dei tag di apertura e chiusura. L'unico consiglio è quello di prestare attenzione ai percorsi relativi delle diverse cartelle e dei file.

La modifica del file xml è necessaria per avere un'anteprima del template.

Ora spostiamoci nella parte amministrativa del sito. e andiamo in:

Site->Template Manager->Site Templates



Ci troviamo in Template Manager. Da qui possiamo scegliere il nostro template "Minimalista" tramite il radiobutton e premere l'icona "Default". In questo modo diciamo a Joomla di inserire il nostro nuovo template. Da notare che in questa sezione ci sono tutti i riferimenti inseriti nel file xml come autore, versione, data e sito internet.



Ora non ci resta che visualizzare il nostro nuovo template. Ci spostiamo nel menù:

Site->Preview->In New Window



Apparirà un nuova finestra che mostrerà che cosa? niente!!!

Perchè? Semplice, il template deve essere ancora costruito :-). Ricordate che abbiamo tolto tutto il contenuto tra i tag "body"? Diciamo che vi ho mostrato un modo per controllare il risultato del nostro lavoro.

Per prima cosa occorre cambiare il collegamento al foglio di stile che, se ricordate, abbiamo copiato da un altro template, quindi occorre modificare il riferimento alla cartella. Nel mio caso il nome "rhuk_solarflare_ii" deve essere sostituito con "minimalista" quindi avremo:

```
<link href="<?php echo  
$mosConfig_live_site;?>/templates/rhuk_solarflare_ii/css/template_css.css"  
rel="stylesheet" type="text/css"/>
```

che diventerà:

```
<link href="<?php echo  
$mosConfig_live_site;?>/templates/minimalista/css/template_css.css"  
rel="stylesheet" type="text/css"/>
```

Ora, proviamo a inserire qualche tag div per fare le nostre pagine. Per questo scopo ho usato dei tag
 e del testo in modo da far vedere le diverse zone del sito. I CSS tendono a contrarsi se non vi è contenuto, dipende dal tipo di layout che si costruisce, in questo caso è fluido. Inoltre con i CSS coloreremo i campi d'azione dei diversi <div> per evidenziare la loro posizione.

```
<div id="contenitore">  
  <div id="contenitore2">  
    <div id="titolo">titolo del sito<br /><br /><br /><br /></div>  
    <div id="sinistra">colonna sinistra<br /><br /><br /><br /></div>  
    <div id="destra">colonna destra<br /><br /><br /><br /></div>  
    <div id="contenuto">contenuto del sito<br /><br /><br /><br /></div>  
    <div id="piede">piede del sito<br /></div>  
  </div>  
</div>
```

Ora proviamo a visualizzare il nostro sito, inserendo il foglio di stile.

Nota: solitamente utilizzo Firefox come browser per la costruzione di siti, perché è lo strumento che si attiene meglio agli standard del W3C, inseguito modifico il codice CSS per Internet Explorer e Opera.

Per questo tutorial usate firefox dato che **non** applicherò trucchi (hack) per IE. Andate nella cartella css del vostro nuovo template, nel nostro caso "minimalista/css" e create un file di nome "template_css.css". All'interno del file ponete questo codice:

```
html,body{
margin: 0;padding:0
}
body{
font-family: arial,sans-serif;
font-size: 76%;
text-align:center
}
div#contenitore{
width:762px;
margin:0 auto;
border-width: 0 1px;
border-style:solid;
border-color: #ccc;
background-color:#0ff;
}
div#contenitore2{
text-align: left;
background-color: #ff0;
}
div#sinistra{
float: left;
width: 120px;
padding: 1em 0 0 10px;
background-color: #ddd;
}
div#destra{
float: right;
width: 190px;
padding: 1em 5px;
background-color: #ddd;
}
div#contenuto{
margin: 0 200px 0 130px;
padding: 1em 10px;
background-color: #ccc;
border:1px solid #ccc;
}
div#piede{
clear:both;
text-align:center;
padding: 5px 0;
background-color: #6cc;
color: #000;
}
```

Non spiegherò i css in quanto esula dall'obbiettivo del manuale, se siete curiosi usate Google. Inoltre qualche cosa per joomla è stato scritto da Ferruccio a questo indirizzo: (http://www.joomla.it/index.php?option=com_content&task=view&id=145&Itemid=28) Comunque ricordate che con i css modificate tutta la grafica del sito menù, form, sfondi e altro.

Premete F5 su firefox dovrebbe comparire una bozza di sito con i diversi campi colorati:

- azzurro per il div contenitore che non si vedrà perché coperto dagli altri div
- giallo per il div contenitore2
- grigio rispettivamente per il div sinistra e destra
- blu-verde per il piede

Questa sarà la nostra base per l'inserimento dei tag php (moduli di Joomla).



Se mi state seguendo ancora vuol dire che il manuale non vi ha annoiato più di tanto oppure non avete ancora trovato altre risorse in rete :-p

Ora se vi ricordate abbiamo commentato il template precedente, spero che nessuno l'abbia cancellato, altrimenti andate a ricopiarvelo dal file originale. Dicevo che dalla parte commentata del codice, possiamo prendere i tag php che ci interessano.

```
<?php mosPathWay(); ?>
```

Questo tag rappresenta le briciole di pane che servono a visualizzare la posizione dell'utente nel sito

```
<?php mosLoadModules ( 'user4', -1 ); ?>
```

Questo tag rappresenta il motore di ricerca

```
<?php
if ( mosCountModules( 'top' ) ) {
    mosLoadModules ( 'top' );
} else {
    ?>
    <span class="error">Top Module Empty</span>
<?php
}
?>
```

Questo codice rappresenta le NewsFlash, come potete vedere si possono fare dei costrutti in php per personalizzare la visione dei tag. In questo caso se non sono presenti news verrebbe visualizzato un messaggio di errore.

```
<?php echo $mosConfig_live_site;?>
```

Questo tag restituisce il percorso del sito `http://nome_sito`

```
<?php mosMainBody(); ?>
```

Questo tag rappresenta il contenuto della parte centrale del sito

```
<?php mosLoadModules ( 'left' ); ?>
```

Questo tag rappresenta il contenuto della parte sinistra del sito

```
<?php mosLoadModules ( 'right' ); ?>
```

Questo tag rappresenta il contenuto della parte destra del sito

```
<?php mosLoadModules ( 'user1' ); ?>
```

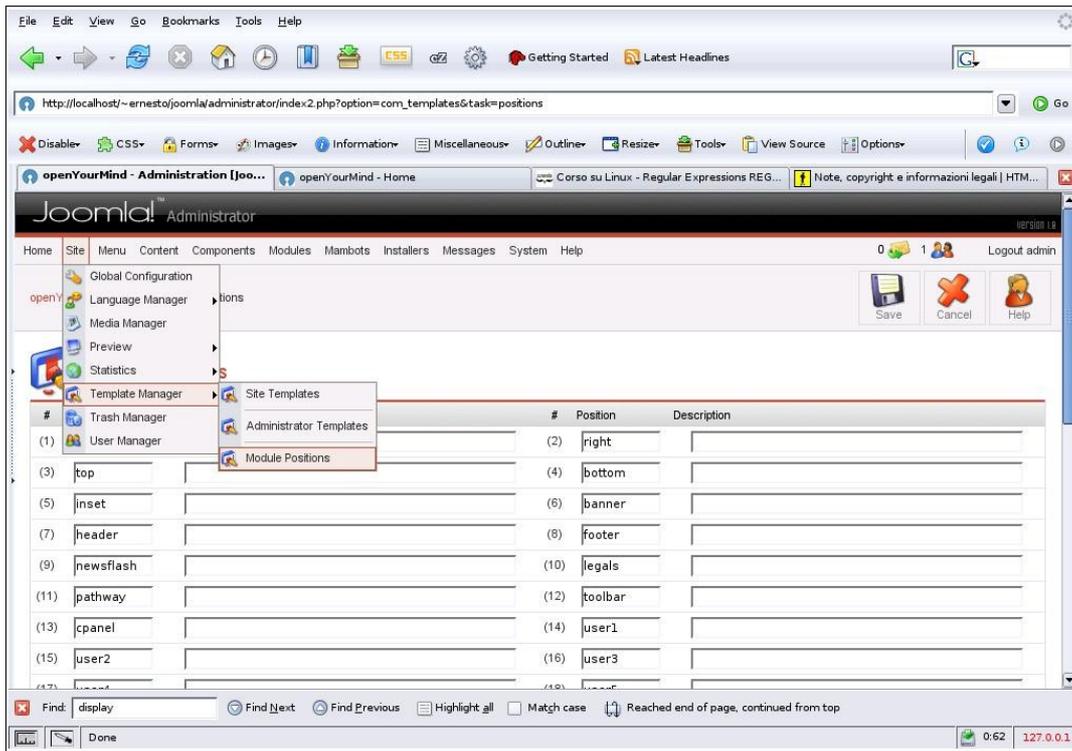
Questo tag rappresenta il contenuto delle Latest News

Gli altri tag ve li scoprite da soli, altrimenti che gusto c'è? Tutto quello che ho detto è vero almeno che non abbiate modificato la posizione dei diversi moduli di Joomla (capirete più avanti).

Da quanto detto si evince che inserendo questi tag php nelle diverse parti dei <div> io cambio la posizione dei diversi moduli. Paradossalmente potrei mettere i moduli di sinistra nella parte destra del sito e i moduli di destra nella parte sinistra, oppure il footer in alto al posto del titolo e viceversa. Tutto dipende da come costruisco il template e la posizione in cui metto i tag php.

Ma da dove vengono questi tag? Facile se voi conoscete Joomla saprete certamente che questi tag li create andando in:

Site->Template Manager->Modules Position



In questa sezione troverete i tag che sono già presenti in Joomla ma cosa più importante potrete crearne uno tutto vostro. Proviamo a creare per esempio "mioModulo", voce che inseriremo in campo vuoto, salviamo il tutto. Semplice no? In questo modo abbiamo creato un tag php per posizionare il nostro modulo, la costruzione è abbastanza semplice basta inserire all'interno delle parentesi tonde del tag php il nome che ho dato al mio modulo:

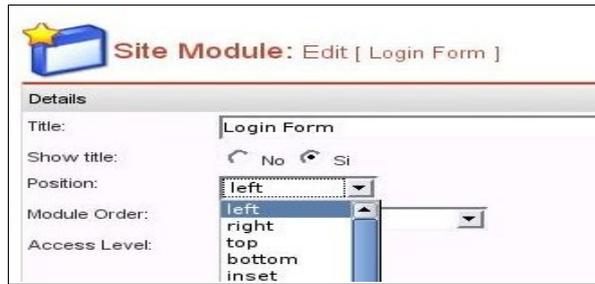
```
<?php mosLoadModules ( 'mioModulo' ); ?>
```

Ora potrò utilizzare questo tag in qualsiasi parte del mio sito e inserire al suo interno i diversi componenti di Joomla attraverso il Module Manager:

Modules->Site Modules



Per esempio entrando nel modulo "Login Form" alla voce "Position" potrò scegliere la voce "mioModulo" o qualsiasi altra per spostare il "Login Form" da una parte all'altra del sito.



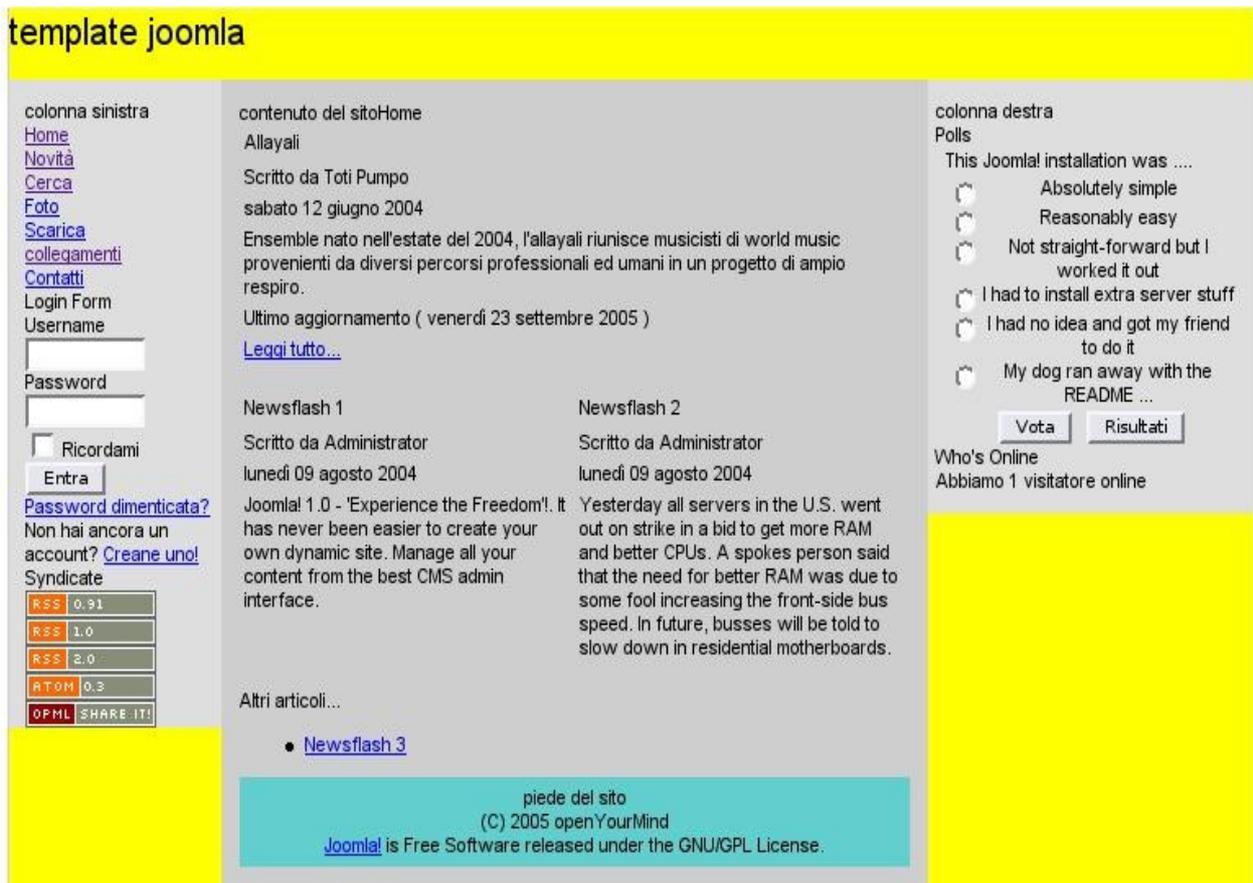
La nuova posizione sarà visibile nel "Module Manager" sotto la colonna "position" che conterrà la voce "mioModulo". Come potete notare dalle immagini io non ho creato la voce "mioModulo", perché è un esercizio che lascio a voi.

#	Module Name	Published	Reorder	Order	Access	Position	Pages	ID	Typ
1	Banners			1	Public	banner	All	18	mod
2	Main Menu			1	Public	left	All	3	mod
3	User Menu			2	Registered	left	All	2	mod
4	Other Menu			3	Public	left	All	31	mod
5	Login Form			4	Public	left	Varies	4	mod

A questo punto possiamo inserire i diversi tag nel nostro codice xhtml e cancellare i testi e i
 precedenti. Questo è il codice inserito:

```
<div id="contenitore">
  <div id="contenitore2">
    <div id="titolo"><h1>template joomla</h1></div>
    <div id="sinistra"><?php mosLoadModules ( 'left' ); ?></div>
    <div id="destra"><?php mosLoadModules ( 'right' ); ?></div>
    <div id="contenuto"><?php mosPathWay(); ?><?php mosMainBody(); ?>
    <div id="piede"><?php include_once( $GLOBALS['mosConfig_absolute_path']
. '/includes/footer.php' ); ?></div>
  </div>
</div>
<?php mosLoadModules( 'debug', -1 );?>
```

il risultato del codice inserito sarà questo:



Non rimane altro che cancellare tutta la parte commentata per ripulire un po il codice xhtml. Nel prossimo capitolo modificheremo tramite i css la grafica dei moduli.

CAPITOLO 3

personalizzare i moduli:

- Modificare log-in

Ora, occorre modificare l'aspetto dei diversi moduli di Joomla. Come posso fare questa cosa? Probabilmente esistono diverse modalità di modifica, ma io conosco quella che vi illustrerò di seguito, spero che qualcuno mi possa insegnare altre tecniche :-)
Proviamo a osservare il codice HTML del nostro sito con attivo il template che abbiamo costruito e i vari moduli. In questa prima parte mostrerò il codice XHTML del modulo "log-in":

```
<table cellpadding="0" cellspacing="0" class="moduletable">
  <tr>
    <th valign="top">Login Form</th>
  </tr>
  <tr><td>
    <form action="index.php" method="post" name="login" >
      <table width="100%" border="0" cellspacing="0" cellpadding="0"
align="center">
        <tr><td>
          <label for="mod_login_username">Username</label><br />
          <input name="username" id="mod_login_username" type="text"
class="inputbox" alt="username" size="10" /><br />
          <label for="mod_login_password">Password</label><br />
          <input type="password" id="mod_login_password" name="passwd"
class="inputbox" size="10" alt="password" /><br />
          <input type="checkbox" name="remember" id="mod_login_remember"
class="inputbox" value="yes" alt="Remember Me" />
          <label for="mod_login_remember">Ricordami</label><br />
          <input type="hidden" name="option" value="login" />
          <input type="submit" name="Submit" class="button" value="Entra" />
        </td></tr>
        <tr><td>
          <a href="index.php?option=com_registration&task=lostPassword">
            Password dimenticata?
          </a>
        </td></tr>
        <tr><td>
          Non hai ancora un account?
          <a href="index.php?option=com_registration&task=register">
            Creane uno!
          </a>
        </td></tr></table>
        <input type="hidden" name="op2" value="login" />
        <input type="hidden" name="lang" value="italian" />
        <input type="hidden" name="return" value="/~ernesto/joomla/" />
        <input type="hidden" name="message" value="0" />
      </form>
    </td></tr>
</table>
```

Come potete notare ho evidenziato in rosso le parti che interessano i CSS. In generale i fogli di stile lavorano su "classi" quando una modifica lavora su gruppi di elementi, in questo caso `class="inputbox"` modifica i diversi campi di testo. Mentre si utilizza "id" quando la modifica lavora su un particolare elemento, in questo caso `id="mod_login_username"` modifica solo il campo di testo "username". I CSS lavorano naturalmente anche sui vari tag. Per comprendere questa cosa leggete in rete qualche manualetto.

Cominciamo la modifica. Aprite il vostro file css con un editor di testo, oppure utilizzate lo stesso Joomla. Spostandosi in:

Site->Template Manager->Site Template



Selezionate il template da modificare e poi premete sull'icona (css) in alto a destra. Come potete vedere è possibile modificare anche il file html. Quando ci troviamo di fronte al foglio di stile che avevamo inserito in precedenza, possiamo immettere nuovi comandi. Infatti è quello che faremo con le seguenti regole css che saranno inserite in fondo alle altre:

```
.inputbox{
color:#f00;           /*modifica il colore dei caratteri del campo*/
background:#ccf;     /*modifica il colore di sfondo del campo*/
border:1px solid #f00; /*modifica il bordo del campo*/
}
```



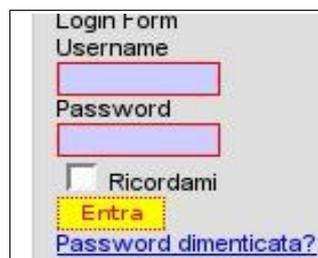
Prima



Dopo

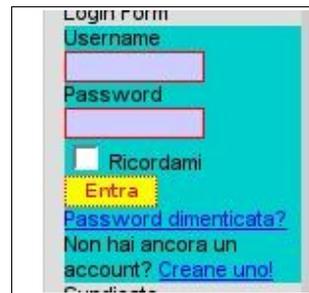
Come si può notare i cambiamenti sono veloci e semplici da fare. Provate a modificare i valori dopo i due punti (:). Nei css i colori possono essere dichiarati con 6 o 3 caratteri alfanumerici (1-2-3-4-5-6-7-8-9-a-b-c-d-f) dopo il cancelletto (#). Modificate anche il valore in "px" di "border", cambiate anche "solid" con "dotted", "dashed", "ridge", ecc.. Insomma divertitevi :-)
Volete modificare il bottone? semplice, cercate la classe o l'id del bottone, che in questo caso è "button", qua invece di usare un valore numerico utilizzo i colori predefiniti date un occhio al link per saper il nome dei colori (http://www.w3schools.com/css/css_colornames.asp) inseriamo il seguente codice:

```
.button{
color:red;
background:yellow;
border:1px dotted red;
}
```



Per modificare lo sfondo del modulo impostate questa regola:

```
#sinistra table form{
Background:#0cc;
}
```



Ricordate che i css funzionano in modo gerarchico, quindi si parte dal <div> di sinistra che contiene la <tabella> che a sua volta contiene il <form> del "login".

Ma perché non continuare a modificare? proviamo anche a cambiare i caratteri e rendiamoli bianchi inserendo questo codice:

```
#sinistra table label{
color:#fff;
}
```



Come si può vedere tutto è abbastanza semplice, naturalmente si può continuare a modificare link, margini, caratteri, margin ,padding ecc... Divertitevi :-)

- Modificare il menù

Bene, proviamo a modificare anche il menù del nostro sito, questo è il risultato che otterremo:



Prima



Dopo

Il risultato sopra esposto l'abbiamo ottenuto inserendo il seguente codice css:

```
#sinistra {
position: relative;
top: 0px;
left: 0px;
background:#d6ce8d;
padding:5px;
}
#sinistra table a:link, #sinistra a:visited, #sinistra a:active {
color:#fff;
text-decoration:none;
}
#sinistra table a:hover {
color:#f00;
}
```

L'ascio a voi il piacere di comprendere quello che ho scritto :-p
Recentemente ho fatto un sito (<http://www.allayali.it>) con la tecnica che vi ho illustrato. Naturalmente non è perfetto, perché non sempre è facile controllare il risultato fra i diversi browser, ma piano, piano migliorerò :-)



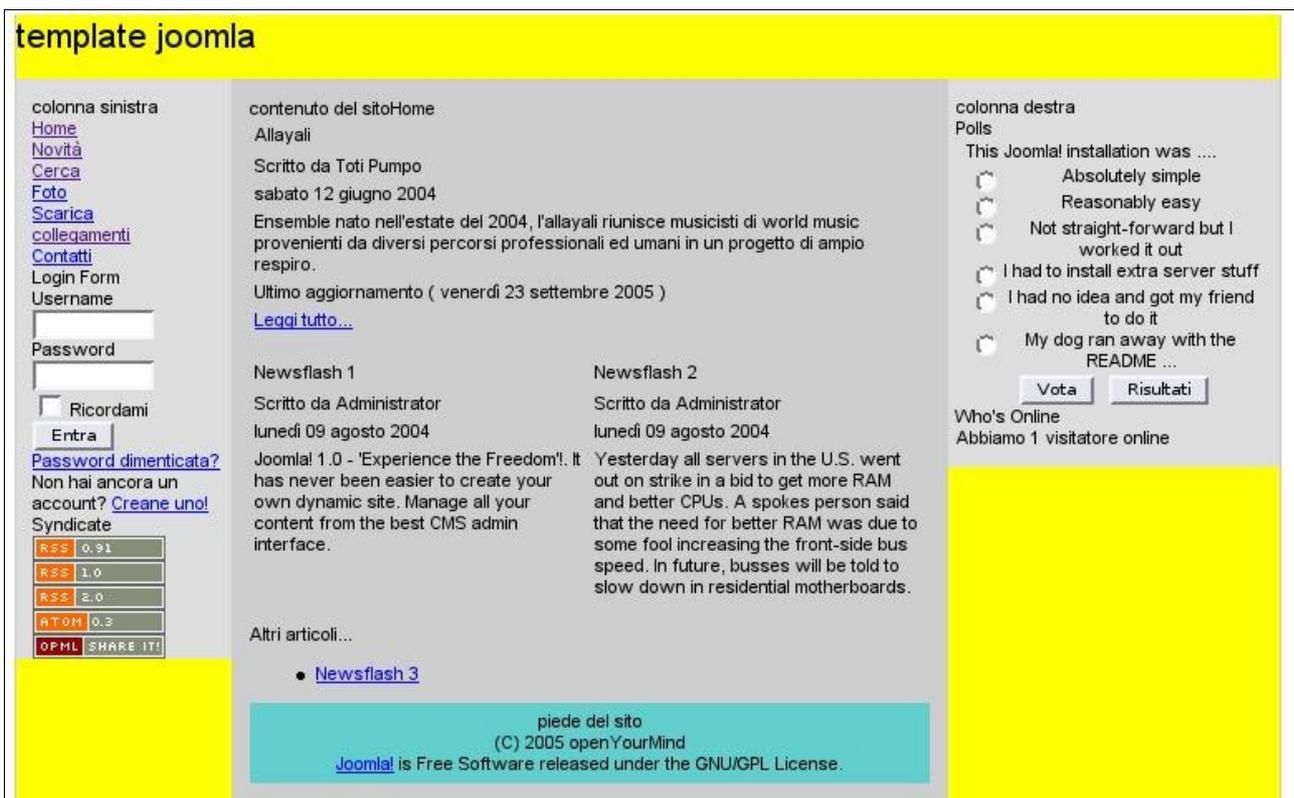
In questo capitolo ho mostrato una tecnica che può servire per chi ama complicarsi la vita :-), infatti, i ragazzi di Joomla desiderano dare ai loro utenti modi semplici per fare le cose. Non arrabbiatevi...è importante conoscere diverse vie per raggiungere lo stesso risultato, spesso per modifiche "estreme" è meglio conoscere queste tecniche. Quindi, se non siete ancora stanchi leggete il prossimo capitolo dove si discuterà la vera modalità di modifica dei moduli in Joomla.

CAPITOLO 4

Moduli:

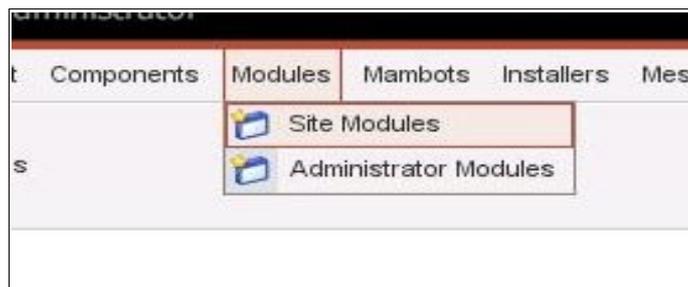
- La modifica del Modulo Login Form

Eccoci nuovamente di fronte al nostro modulo, prima di cominciare eliminiamo o commentiamo il codice che modificava i vari moduli inserito in precedenza in modo da ritornare al nostro template minimalista:

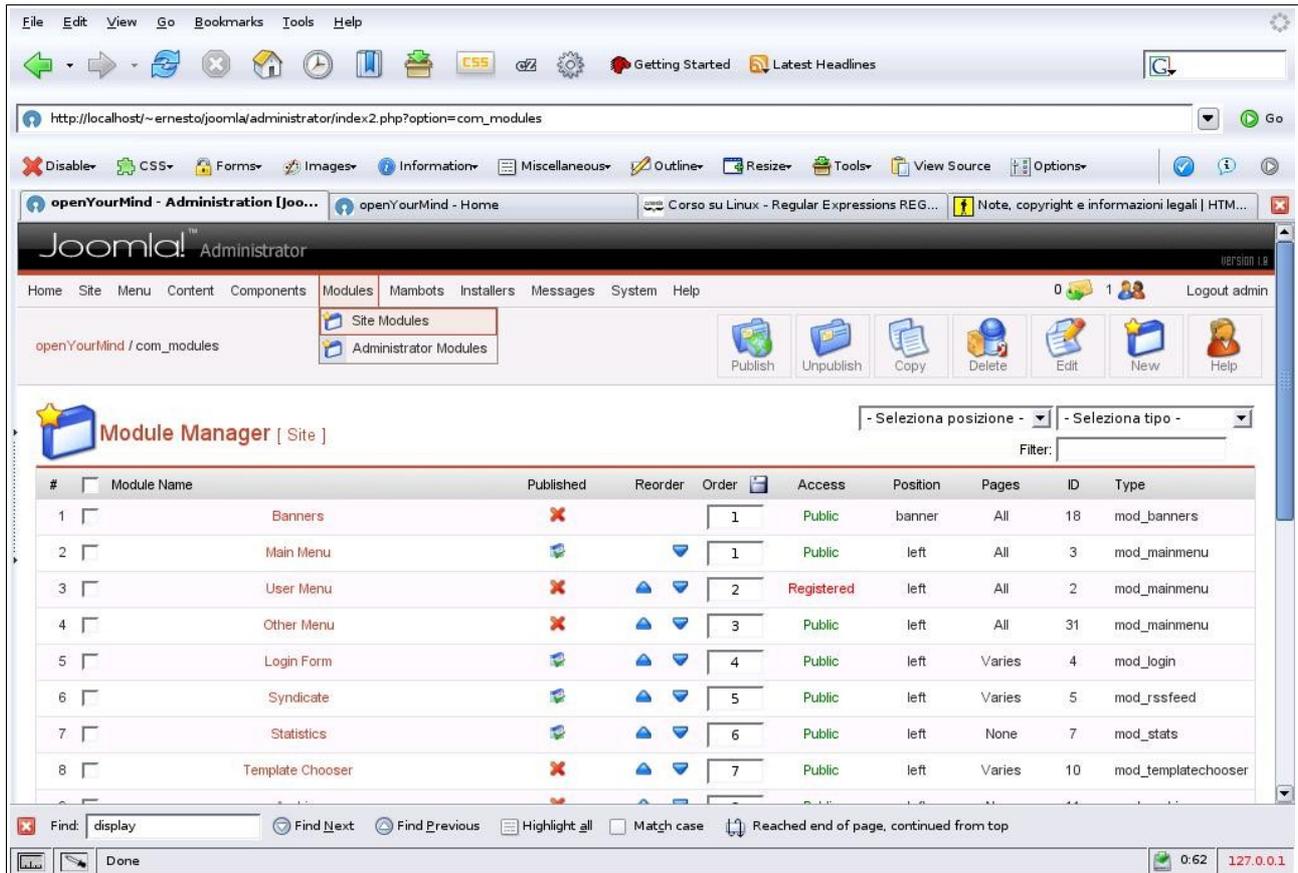


Come accennavo nel capitolo precedente Joomla! offre la possibilità di modificare la parte grafica dei moduli. Portiamoci nella parte amministrativa di Joomla! e spostiamoci nella sezione:

Module->Site Modules



Bene, ci troveremo nuovamente nel Module Manager:



click sul link "login form" ed entriamo nella pagina di personalizzazione del modulo. A noi interessa la parte parameters:



Le altre parti ve le lascio scoprire in autonomia :-P

Prima di procedere è d'obbligo una premessa. Joomla personalizza le proprie tabelle che contengono un modulo con la stessa classe css (`class="moduletable"`). Infatti, se osserviamo il codice html che restituisce il template minimalista, tutti i moduli hanno una grafica simile nel colore, nei link, nei caratteri, ecc... tutto derivato dalla classe sopracitata. Di seguito una porzione di codice html del modulo "login":

```
<table cellpadding="0" cellspacing="0" class="moduletable">
```

A questo punto entra in ballo il nostro parametro "Module Class Suffix" che permette di personalizzare il singolo modulo. Proviamo a inserire un nome che identifica in modo univoco questo modulo, per esempio "accesso". Di seguito il nuovo codice html:

```
<table cellpadding="0" cellspacing="0" class="moduletableaccesso">
```

Qualcuno avrà certo compreso le conseguenze di questo semplice cambiamento. Ora chi ci ferma più? Abbiamo una classe css tutta nostra che possiamo manipolare come desideriamo:

```
.moduletableaccesso{  
width:120px; /*imposto una larghezza del modulo*/  
margin-left:-5px; /*sposta il modulo verso sinistra*/  
margin-top:5px; /*allontana il modulo dalla parte superiore*/  
margin-bottom:5px; /*allontana il modulo dalla parte inferiore*/  
padding-left:5px; /*allontana il contenuto dal bordo sinistro*/  
background-color:red; /*questo indovinatelo voi :-p*/  
border:3px outset #55a; /*particolare effetto dei css sui bordi*/  
color:yellow; /*questo che farà? :-)*/  
}
```

Spero sia tutto chiaro!! :-)

Ora si può continuare a giocare con tutti i componenti del modulo. Ricordate che i css lavorano in cascata, quindi dalla nostra classe possiamo far discendere tutto. Per i link:

```
.moduletableaccesso a:link{  
color:#fff;  
text-decoration:none;  
}  
.moduletableaccesso a:hover{  
color:#0ff;  
text-decoration:underline;  
}
```

Per modificare i campi del form, possiamo inserire questo codice:

```
.moduletableaccesso input{  
border:1px solid #000;  
background-color:orange;  
}
```

Il resto lo lascio alla vostra fantasia.



1° css



2°css



3°css

- La modifica del modulo Main Menù

Le cose spiegate fino a questo dovrebbero essere chiare, altrimenti rileggetele :-P, se ancora sono oscure potete contattarmi ernesto.c@libero.it o chiedere nei vari forum. Questa volta dal "Module Manager" entriamo in "Main Menù", qui troveremo diversi parametri che dovrete esplorare. D'interesse per questo manualetto sono "Module Class Suffix" e "Menu Class Suffix":



Parameters	
Menu Class Suffix	<input type="text"/>
Module Class Suffix	<input type="text"/>

Personalizziamo velocemente il modulo, dovrete avere già le capacità per farlo :-). Nel campo di testo "Module Class Suffix" ho inserito la voce "elenco". Crea la classe la personalizzo:

```
.moduletableelenco{
width:120px;           /*larghezza del modulo*/
margin-left:-5px;     /*sposta il modulo verso sinistra*/
margin-top:5px;       /*allontana il modulo dalla parte superiore*/
margin-bottom:5px;   /*allontana il modulo dalla parte inferiore*/
padding-left:5px;    /*allontana il contenuto dal bordo sinistro*/
background-color:#88a; /* :-) */
border:3px outset #fff; /*particolare effetto dei css sui bordi*/
color:yellow;        /*bhoooooooo :-P*/
}
```

Niente di particolarmente difficile, praticamente è lo stesso codice del modulo "login" :-)
Quello che illustrerò di seguito riguarda invece "Menu Class Suffix". con questo campo si possono personalizzare le voci di menù. Il codice di default in Joomla si presenta così:

```
<tr align="left"><td><a href="index.php?option=com_frontpage&Itemid=1"
class="mainlevel" id="active_menu">Home</a></td></tr>
```

Senza scendere troppo nei dettagli, ho inserito il nome "elenco" nel campo di testo per ottenere la nuova classe:

```
<tr align="left"><td><a href="index.php?option=com_frontpage&Itemid=1"
class="mainlevelelenco" id="active_menuelelenco">Home</a></td></tr>
```

Proviamo a inserire questo codice:

```
.mainlevelelenco {
display:block;
width:95%;
border:1px solid #88a;
color:yellow;
}
```

V'invito a sperimentare... proviamo a migliorare il nostro menù inserendo altro codice per manipolare la tabella:

```
.moduletableelenco a:hover{
display:block;
width:95%;
color:#000;
text-decoration:none;
background-color:orange;
border:1px solid #000;
}
```



.moduletableelenco



.mainlevelelenco



.moduletableelenco a:hover

I più attenti si saranno resi conto di un particolare. A cosa serve l'identificativo `id="active_menuelenco"`? Il nome dovrebbe suggerirlo... se usato permette di identificare il collegamento attivo. Provate a inserire il codice seguente:

```
#active_menuelenco{
display:block;
width:95%;
color:white;
background-color:#000;
border:1px solid #000;
}
```



Bene! ora non avete più scuse per cominciare a creare grafiche per Joomla. Spero che questo breve tutorial sia stato d'aiuto per comprendere meglio la costruzione di un template. Naturalmente tutto quello che è stato detto, può essere fatto anche da chi non conosce l'html; Basta utilizzare uno strumento visuale per costruire il proprio modello di pagina in cui inserire, terminato il lavoro, i vari tag php. Ogni webmaster ha il proprio modo di lavorare :-)
Pregherei le persone che leggeranno queste poche pagine di fornire critiche costruttive per migliorare e ampliare la mia conoscenza. Vi auguro buon lavoro.