



PARTNER
WWW.JOOMLSPACE.IT

JOOMLA 1,5

GUIDA AI TEMPLATE

Realizzata da

www.joomlapixel.eu

<http://portal.lucazone.net/>



Capire il layout

[Approfondimenti,tutorial e sviluppi disponibili su www.joomlapixel.eu](http://www.joomlapixel.eu)

Capire il layout di Joomla!

Abbiamo sempre apprezzato Joomla! Per la facilità d'uso,per le ampie possibilità di impaginazione e di personalizzazione,per la vastità di elementi aggiuntivi che permettono di finalizzare il sistema per diversi utilizzi.

La nuova release fa un salto qualitativo veramente “corposo” non tanto in riferimento alla quantità di elementi nuovi,quanto semmai alla vastità di operazioni che queste stessa permettono!

Non è di sicuro esagerato affermare che si riesce a raggiungere il controllo totale sia a livello di personalizzazione ed impaginazione,sia a livello di dinamicità e interazione.

Ma in cosa è focalizzata questa nuova ed estesa possibilità di controllo?

- Gestione dell'output dei componenti
- Gestione dell'output dei moduli
- Gestione di tutti i parametri relativi sia alla struttura fisica dei template sia relativa ai selettori css tipografici e non..

Cominciamo analizzando “virtualmente” il file index.php del template per capire innanzitutto quale è il rendering del CMS.

Joomla! È un framework basato sul codice php,ed il file principale del template altro non è che un misto tra html e pezzetti di istruzioni in php.Le parti in html sono utilizzate per creare la struttura,il layout del sito,mentre gli snippet di codice php permettono di richiamare a video i contenuti presenti nel database.

E' un sistema certo largamente ed efficacemente usato nella maggior parte dei CMS,ma di sicuro non in maniera così completa e versatile.

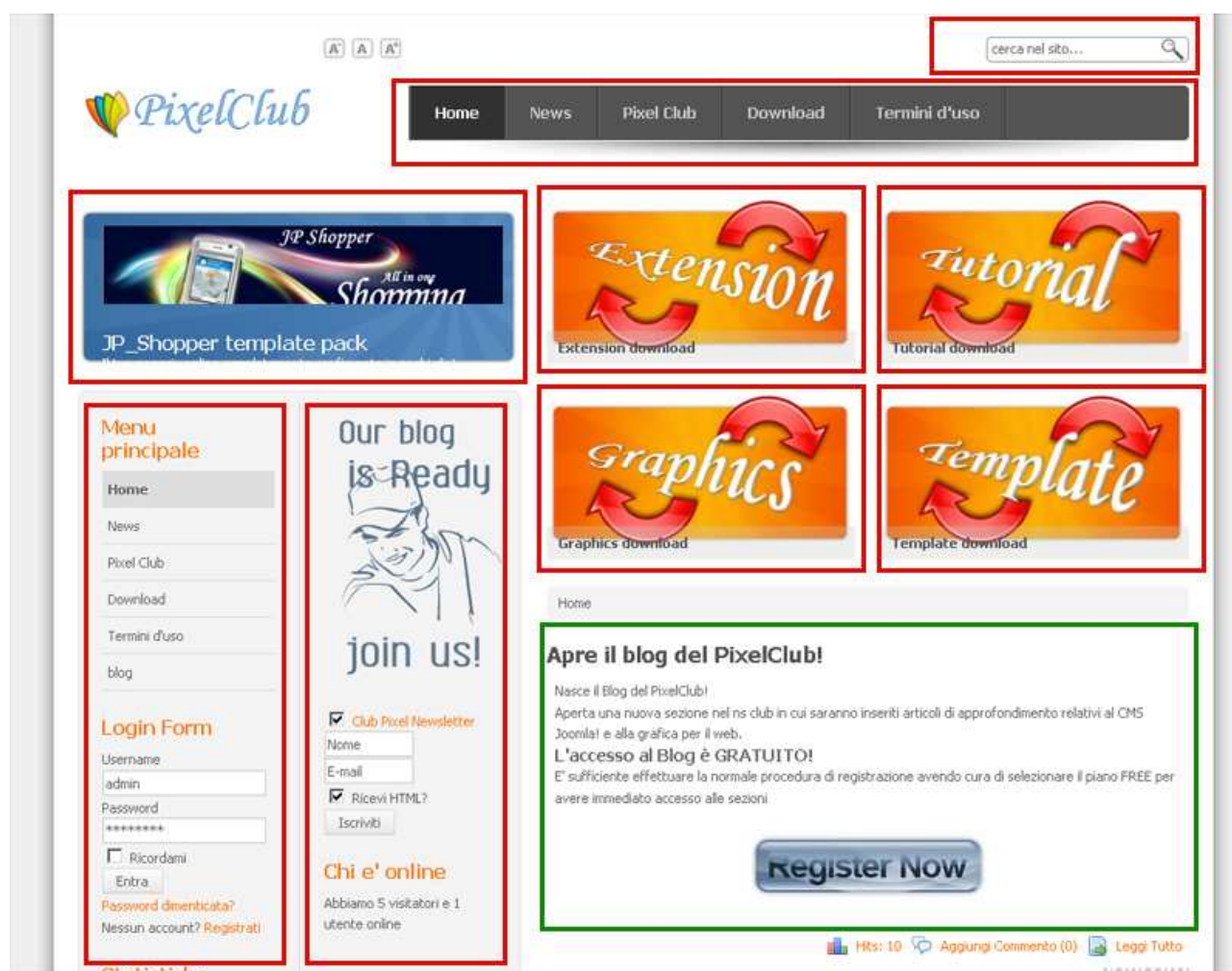
La struttura per riportare un esempio banale può essere paragonata ad un gioco come i blocchi di costruzioni,in un piano base possiamo inserire qualsiasi blocco,colorarlo come meglio ci piace,estenderlo,e non solo.....possiamo farlo per tutte le pagine del nostro sito!

Comprendere questo tipo di struttura,soprattutto per i neofiti è uno dei passi fondamentali nella realizzazione di un template per Joomla! 1,5.và comunque detto che questo discorso non si discosta dalla precedente versione

Nella figura sottostante un esempio che chiarifica il concetto.

Non siamo vincolati ad una struttura CMS classica e ad uno schema basato su due-tre colonne con l'aggiunta di un header e un footer, questo sistema a "blocchi" ci permette la massima libertà nell'atto di realizzazione del nostro template, proprio perché risulta estremamente facile inserire i blocchi nella posizione che desideriamo.....ovviamente tutto ciò è strettamente legato al livello personale di conoscenza della programmazione per il web (nello specifico html e Fogli di stile).

Nell'immagine abbiamo bordato in rosso i "blocchi" modulo ed in verde l'area di output dei contenuti.



LA STRUTTURA DEI TEMPLATE

I templates di Joomla possono contenere diverse cartelle e files ma la struttura di base ed alcuni di essi sono fondamentali per il corretto funzionamento e questa struttura, lo ripetiamo solo a livello base (che non contenga cioè override di componenti e moduli e/o parametri del template stesso) è identica alla precedente:

cartella del template (il nome deve essere il medesimo inserito nel file xml di installazione)

- o File index.php
- o File TemplateDetails.xml
- o File template.css
- o Cartella images
- o Cartella css

Il file index.php

Come già menzionato in precedenza è un misto tra html e codice php ed è sicuramente tra i file più importanti del sistema

Il file TemplateDetails.xml

Questo “raggruppatore” di dati riveste una particolare importanza nella realizzazione dei template per la nuova struttura, infatti oltre alle specifiche che già conoscevamo e che impostavano nome del template, autore, licenza, url ed email dell'autore e dati riferiti a cartelle e file specifici che accompagnavano le installazioni precedenti ha ora funzioni più avanzate gestendo innanzitutto le posizioni dei moduli (in precedenza era sufficiente inserire il codice di richiamo della zona modulo ad es. `<?php mosLoadModules('user1', -2); ?>` perché il sistema lo interpretasse e mostrasse a video l'output relativo) in caso contrario le stesse non saranno disponibili, e gestendo ed elencando anche i parametri di configurazione avanzata che un template può avere, questo un esempio tratto dal template Beez di default nella installazione del pacchetto che considera appunto quanto sopra detto riguardo alle posizioni disponibili nel template:

<positions>

<position>left</position>

<position>right</position>

<position>top</position>

<position>breadcrumb</position>

<position>user1</position>

```
<position>user2</position>

<position>user3</position>

<position>user4</position>

<position>debug</position>

<position>syndicate</position>

</positions>
```

Il file template.css

Ovviamente legato sia alla struttura in html del file index.php ma anche a quella che è la gestione dell'output del framework con dei selettori specifici riconosciuti dal sistema (ad es. la classe componentheading per il titolo dei componenti, la classe author per il nome dell'autore etc etc)

La cartella images

Nulla di cambiato rispetto alla versione precedente, raccoglie tutti le immagini impostate per la struttura.

La cartella css

Come sopra ma contenente i fogli di stile

Questi i file di base necessari per la realizzazione di un nuovo template per il CMS.

In realtà e come vedremo più avanti possono essere presenti diverse altre cartelle e file, per il momento ci siamo soffermati solo all'individuazione dei file necessari.

Step by step il nostro primo template per Joomla!

Realizzare praticamente il nostro layout è forse il miglior esercizio possibile per capire meglio la metodologia di lavoro, e lo faremo partendo da file vuoti che riempiremo e testeremo di volta in volta per osservare i risultati ottenuti.

E' bene premettere però alcuni punti fondamentali,il primo,strettamente personale riguarda gli strumenti e i metodi usati per questo lavoro.

ogni web developer ha i suoi metodi e usa gli strumenti che preferisce ed ognuno di noi ottiene i risultati sperati,ciò non vuol dire che un metodo sia migliore dell'altro, l'importante è il raggiungimento di un risultato soddisfacente nel minor tempo possibile e con strumenti ed operazioni rapide,la mia è una metodologia di lavoro affinata con il tempo e l'esperienza e che sento di consigliare,non di imporre.

Il secondo punto riguarda gli approfondimenti necessari per quel che riguarda la creazione della struttura in html e fogli di stile:

forniremo le basi necessarie,qualche utile consiglio,ma il campo è davvero vasto ed è necessario un approfondimento su queste tematiche ove sia necessario.

[Approfondimenti,tutorial e sviluppi disponibili su www.joomlapixel.eu](http://www.joomlapixel.eu)

Gli strumenti da utilizzare

Gli strumenti che utilizzeremo

- Dreamweaver
- D-Form_Joomla15 – estensione per dreamweaver
- Firefox – web browser
- Aardvark – estensione per firefox
- Cswiever – estensione per firefox
- Top style light – editor css

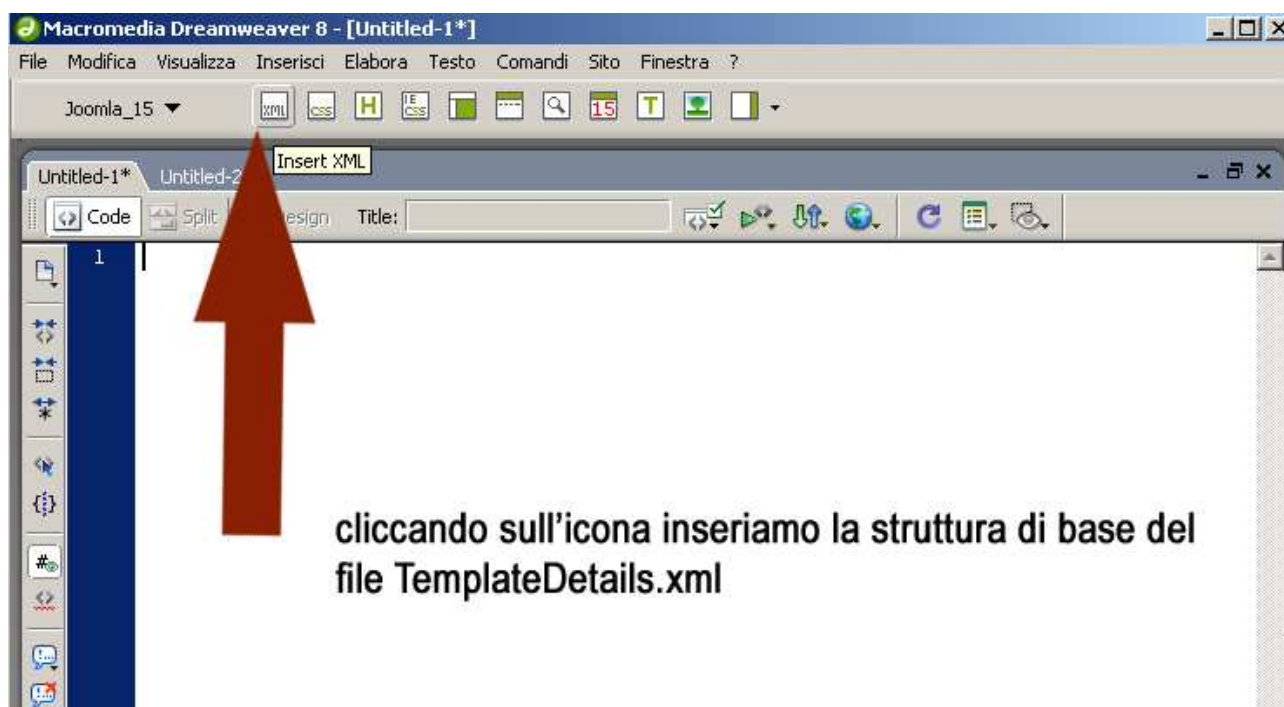
Cominciamo!

Ovviamente lavoriamo per comodità in locale e quindi iniziamo creando la cartella che ospiterà il nostro template all'interno della cartella “templates” di un installazione tipica di Joomla! Versione 1,5.

Il nome del template sarà “test” e quindi creiamo la cartella relativa,aprendola successivamente e inserendo al suo interno altre due cartelle “images” e css.

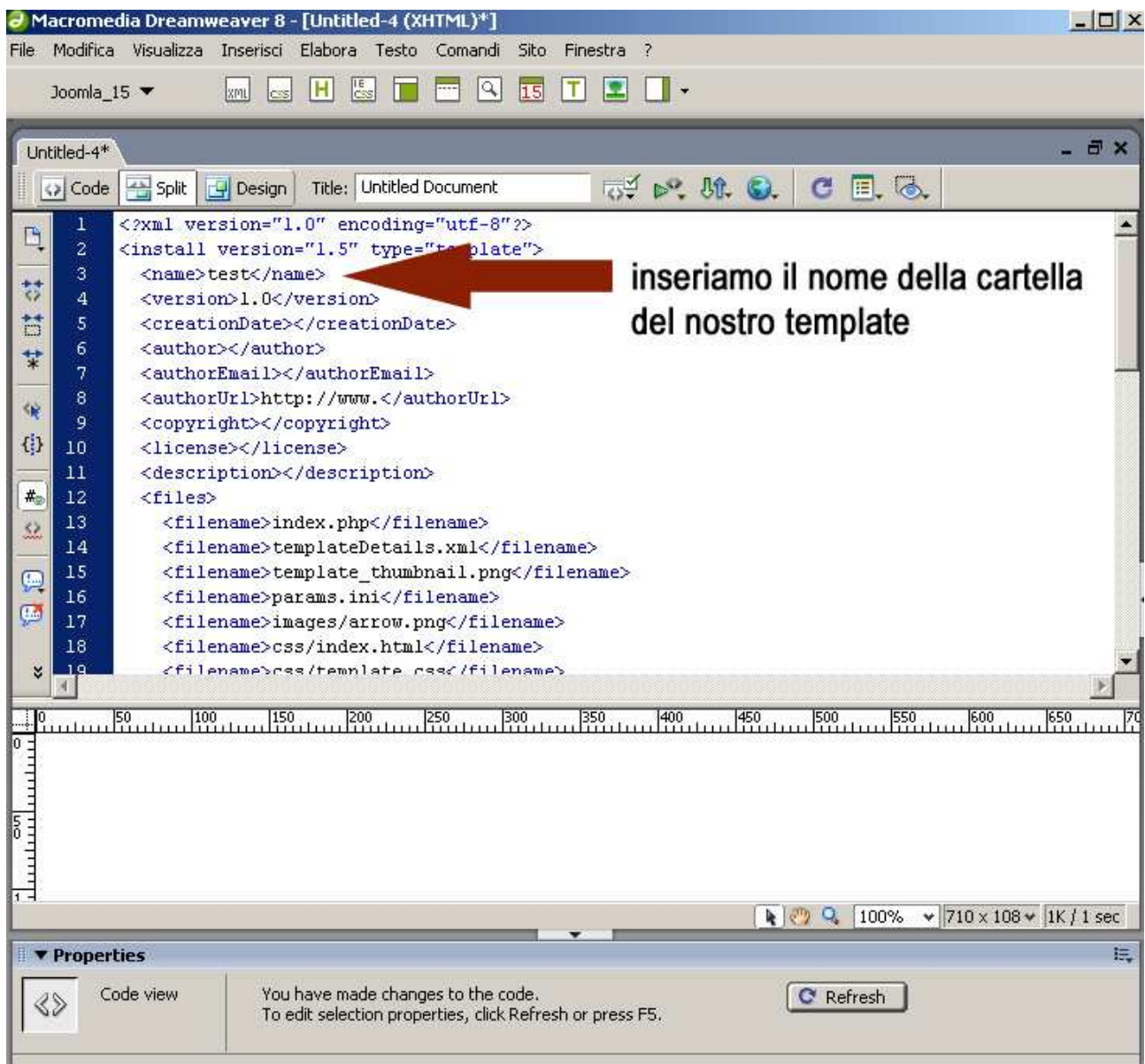
Step 1 – creazione file TemplateDetails.xml

Apriamo un nuovo documento vuoto in dreamweaver,ripulendolo dal codice generato dal programma e abilitiamo gli strumenti dell'estensione D-Form caricata in precedenza



Otterremo così una prima sommaria compilazione del file xml necessario per il sistema.

Non occupiamoci per il momento della esatta compilazione del file limitandoci a inserire il nome della cartella affinché venga riconosciuta dal sistema.



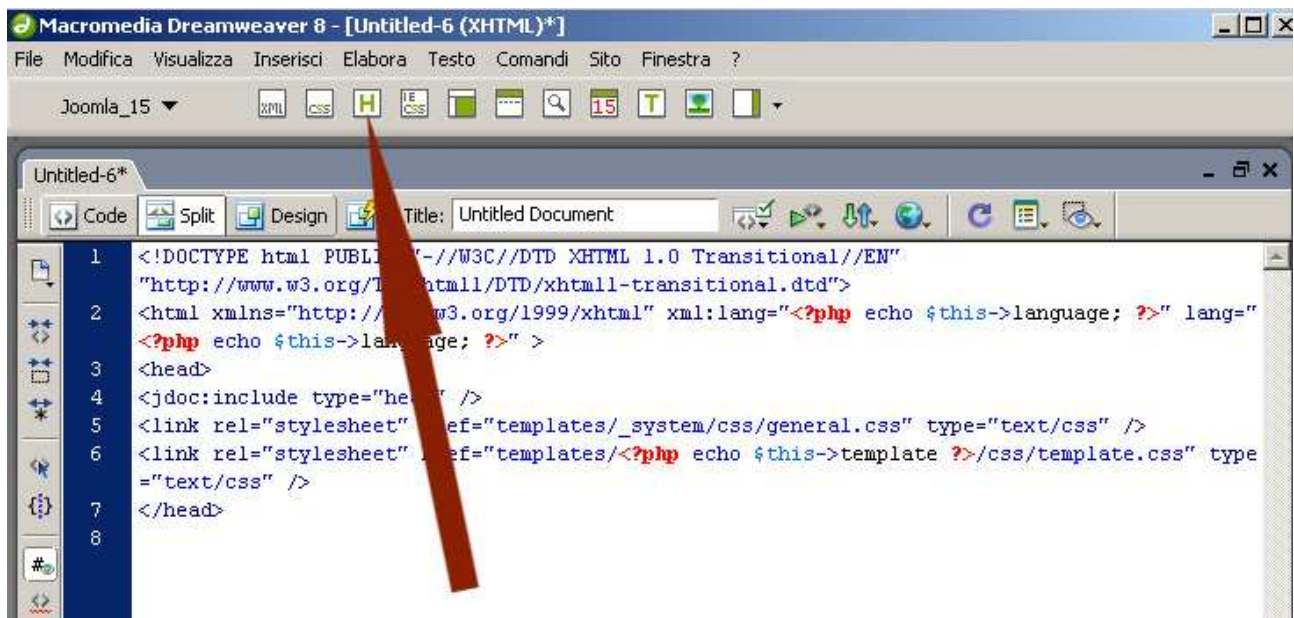
Salviamo nella cartella di destinazione con il nome TemplateDetails e l'estensione xml

Il file index.php

[Approfondimenti,tutorial e sviluppi disponibili su www.joomlapixel.eu](http://www.joomlapixel.eu)

creazione del file index.php del template

Apriamo un nuovo documento php in dreamweaver ripulendo anche in questo caso il codice generato in automatico,ed inserendo la parte iniziale relativa all'header del template



Questa parte iniziale di codice inserisce informazioni specifiche del sito compresi i metadata utilizzati dai motori di ricerca attraverso delle classi impostate nel framework:

- jdoc (jdocument)

esistono fondamentalmente due oggetti distinti nel framework che hanno specifiche funzioni

jdocument e jdocumenthtml

la prima con esempio visibile nell'immagine sopra

<jdoc:include type="head" />

come è facile intuire opera un “include” permettendo così il recupero a video ad esempio di moduli,componenti e/o informazioni,largamente utilizzata proprio nella realizzazione dei template per il sistema.

La seconda `documenthtml` principalmente attraverso l’oggetto variabile `$this` verifica l’esistenza di una o più particolari condizioni e rappresenta una risorsa eccezionale soprattutto nella gestione delle istruzioni condizionali (come ad esempio la gestione della collapsabilità delle zone modulo) delle quali parleremo più avanti nella guida.

Un altro particolare importante che rileviamo in questo primo importantissimo blocco di codice è il richiamo a due differenti fogli di stile,da questa nuova versione (attualmente è disponibile la 1.5.1 di Joomla!) è possibile effettuare modifiche direttamente dal pannello di controllo su tutti i fogli di stile presenti nel template (ovviamente sempre racchiusi nella cartella contenitore “css” ed inseriti nel file `index.php` e `templatedetails.xml`,una novità importante e utilissima.

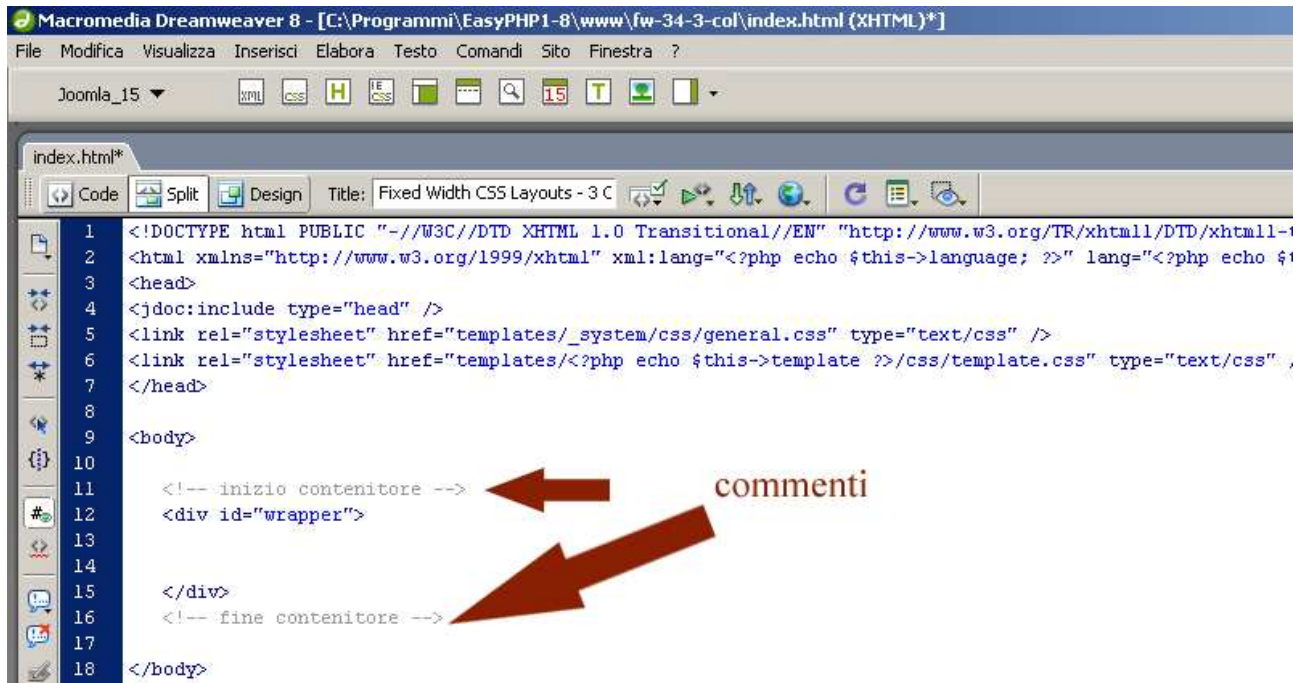
Nelle versioni 1.0.x infatti pur se esiste la possibilità di avere fogli di stile multipli,la gestione e modifica direttamente da pannello era vincolata al solo file `template_css.css`,per agire sugli altri file bisogna disporre di un accesso ftp (se il sito è pubblicato su server remoto.



Continuiamo ora la costruzione della nostra base per il template cominciando a costruire una struttura tableless nella quale,pezzo per pezzo inseriremo moduli e componenti.

Realizziamo una struttura classica su 3 colonne con header e footer e dunque nel file aperto in dreamweaver iniziamo inserendo un div contenitore “wrapper” che avrà il compito di racchiudere la struttura e gestire la centratura nel browser del template.

E' sicuramente molto utile scrivere il codice inserendo anche commenti che facilitino l'individuazione delle zone in previsione di futuri cambiamenti.



Ci è possibile lavorare contemporaneamente sia sulla index.php del template sia sul suo foglio di stile, quindi apriamo topstyle light (versione gratuita) un ottimo programma di editing e inseriamo di volta in volta le proprietà dei selettori che andremo a creare



Assegniamo le proprietà generiche della pagine attraverso il selettore body impostando il font e la sua grandezza, mentre la centratura del contenitore “wrapper” verrà gestita semplicemente dalla proprietà margin impostata su “0 auto” avendo cura di definire anche la sua larghezza, in caso contrario il div non risulterebbe centrato nella pagina.

Continuiamo inserendo header – colonna sinistra – colonna destra – contenuti e footer

```
11 <!-- inizio contenitore -->
12 <div id="wrapper">
13
14 <!-- Header -->
15 <div id="header">
16
17 </div>
18 <!-- fine Header -->
19
20 <!-- colonna sinistra -->
21 <div id="colonnasinistra">
22
23 </div>
24 <!-- fine colonna sinistra -->
25
26 <!-- contenuti -->
27 <div id="contenuti">
28
29
30 </div>
31 <!-- fine contenuti -->
32
33 <!-- colonna destra -->
34 <div id="colonnadestra">
35
36 </div>
37 <!-- fine colonna destra -->
38
39 <!-- Footer -->
40 <div id="footer">
41
42 </div>
43 <!-- fine Footer -->
44
45 </div>
46 <!-- fine contenitore -->
47
```

E nel foglio di stile i rispettivi parametri:

```
#header {
color: #333;
width: 900px;
float: left;
padding: 10px;
border: 1px solid #892105;
height: 100px;
margin: 10px 0px 5px 0px;
}

#colonnasinistra {
color: #333;
```

border: 1px solid #892105;

margin: 0px 5px 5px 0px;

padding: 10px;

height: 350px;

width: 195px;

float: left;

}

#contenuti {

float: left;

color: #333;

border: 1px solid #892105;

margin: 0px 5px 5px 0px;

padding: 10px;

height: 350px;

width: 456px;

display: inline;

}

#colonnadestra {

color: #333;

border: 1px solid #892105;

margin: 0px 0px 5px 0px;

padding: 10px;

height: 350px;

width: 195px;

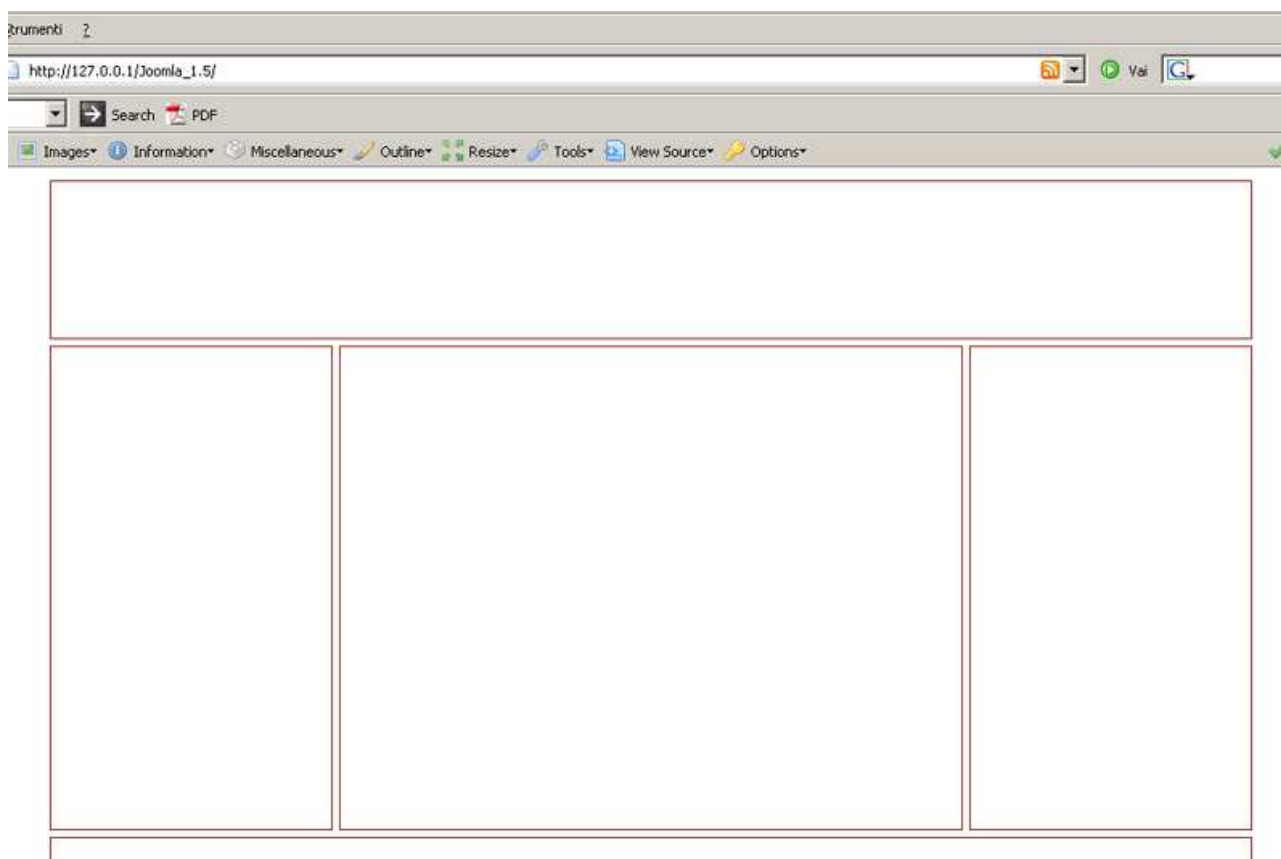
float: left;

```
}  
  
#footer {  
  
width: 900px;  
  
clear: both;  
  
color: #333;  
  
border: 1px solid #892105;  
  
margin: 0px 0px 10px 0px;  
  
padding: 10px;  
  
}
```

Abbiamo inserito delle proprietà per distanziare i “blocchi” usando il margin, impostato un'altezza per le colonne e un bordo perché siano visibili nel browser (successivamente la proprietà height verrà eliminata poiché non necessaria).

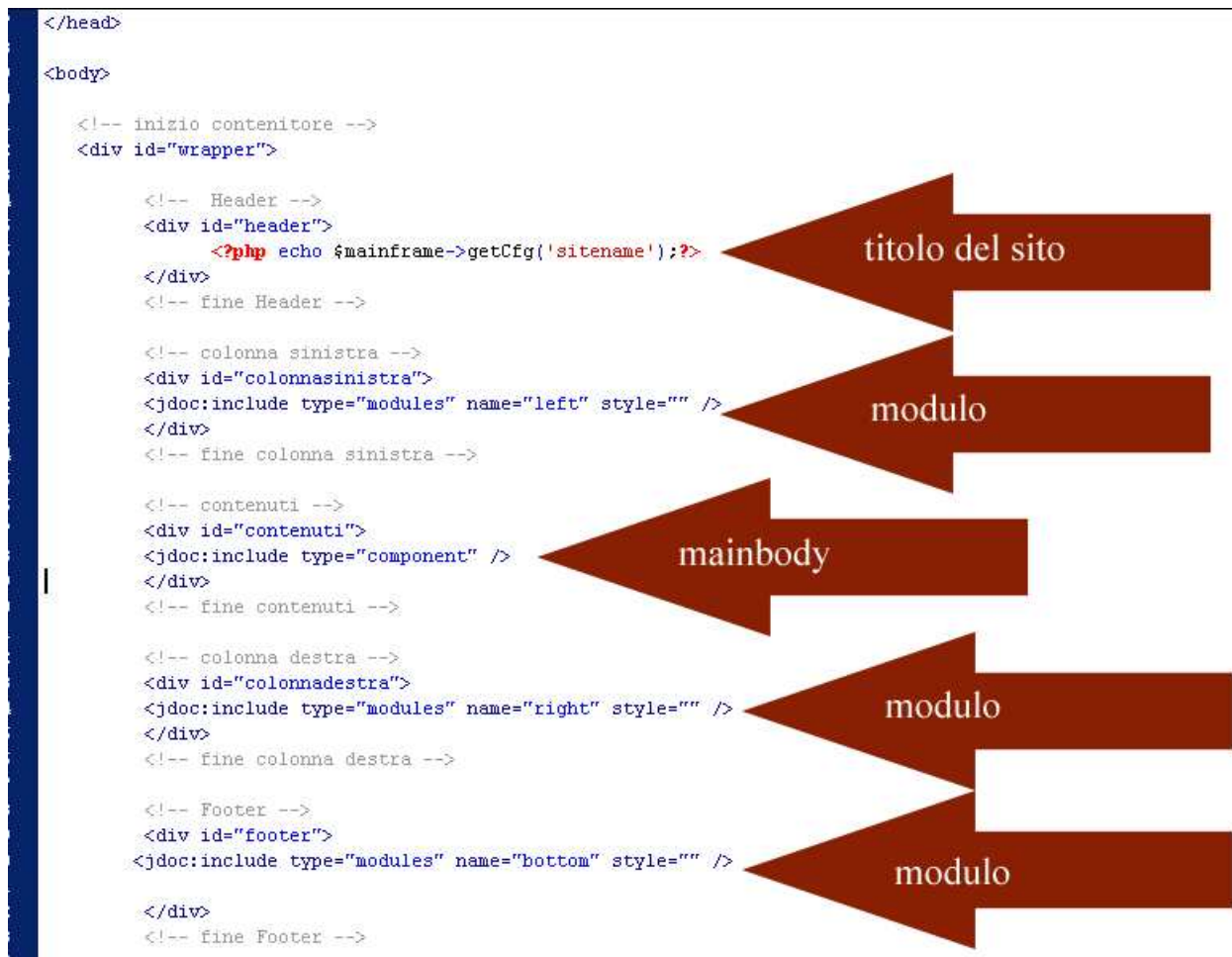
Il posizionamento delle colonne è stato fatto utilizzando la proprietà “float”, il footer con la proprietà “clear both” garantisce stabilità a tutta la zona superiore.

Salviamo il tutto e andiamo a visionare il risultato nel browser.



La struttura risulta centrata perfettamente nella pagina e possiamo notare attraverso il bordo impostato le zone create

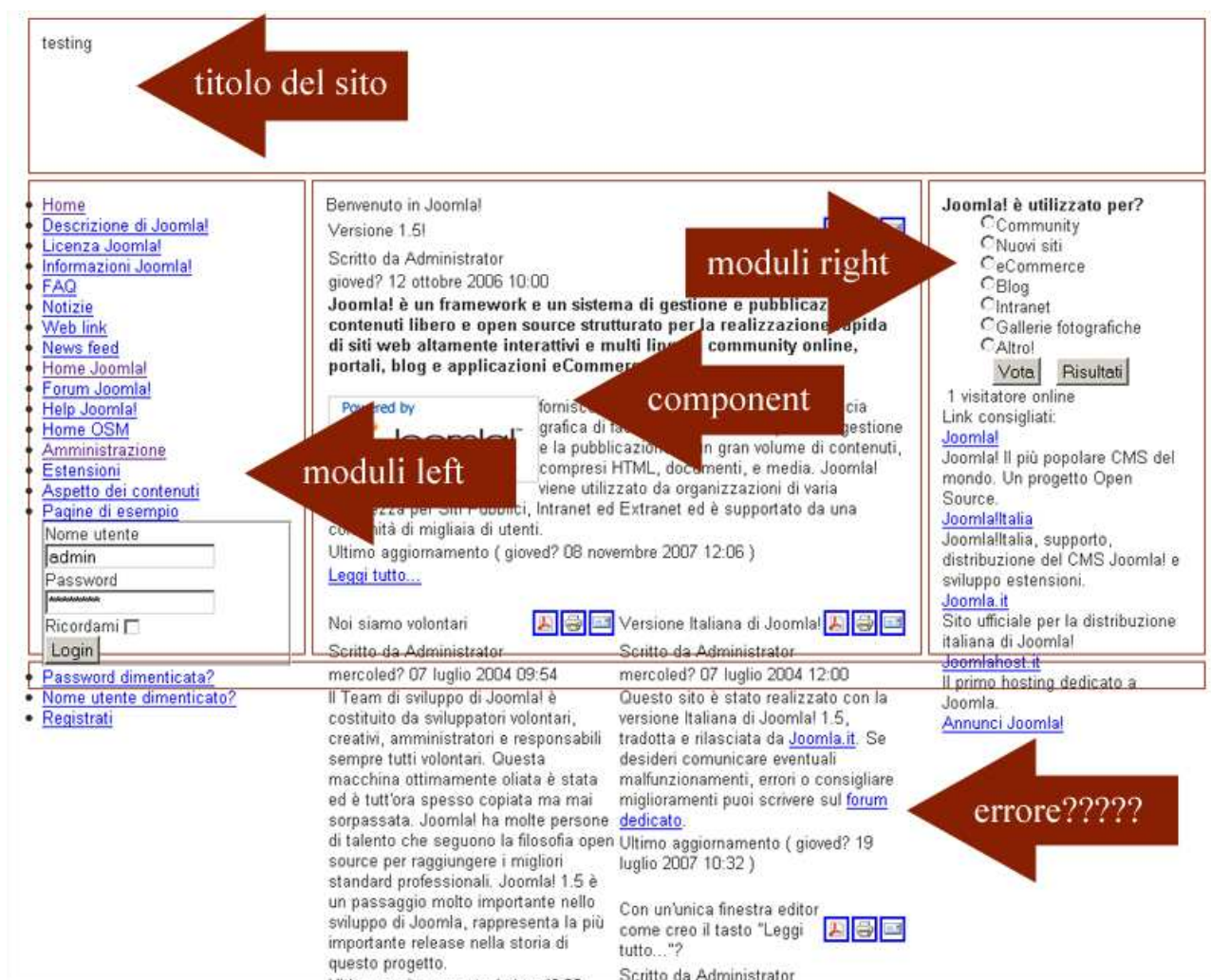
Lo scheletro del nostro primo template è dunque pronto ad accogliere i vari blocchi del framework di Joomla! Che cominceremo subito ad inserire grazie all'estensione D-Form per dreamweaver.



Riconosciamo con facilità gli include che richiamano l'output del mainbody (component) e dei moduli con la direttiva jdoc

Continuiamo la compilazione del nostro template di test, torneremo in seguito ad affrontare il discorso relativo alla gestione grafica dei moduli nel paragrafo dedicato.

Salviamo dunque e controlliamo nel browser



Nelle frecce evidenziate le zone che abbiamo popolato.

Il template è ovviamente molto scarno ma funzionante, la freccia indicante un errore serve solo a ricordarci che abbiamo impostato delle altezze fisse per i div che dovremo a questo punto eliminare.

Un'altra visualizzazione non corretta la ritroviamo nei menù della colonna sinistra, ma anche in questo caso è dovuto alla mancanza di impostazioni sia del menù stesso sia delle "ul" generiche (unordered list) nel file css del template.

Eliminando la proprietà height e impostando le ul per la colonna sinistra

```
#colonnasinistra ul{
```

list-style-type: none;

}

Il risultato



Le colonne si adattano ai contenuti e il menù della colonna sinistra risulta più gradevole.

Se vogliamo controllare la disposizione dei moduli che abbiamo inserito attraverso il pannello (nella versione 1.0.x la voce di menù era “anteprima con posizione blocchi modulo”) è sufficiente portarsi nel pannello amministrativo di joomla cliccare sulla voce di menu “estensioni – gestione template – cliccare sul nome del template (nel nostro caso ricordate si chiama “test”) e una volta aperto cliccare sulla voce “anteprima”



Il template verrà aperto all'interno del pannello e le zone modulo saranno evidenziate (outlined per usare il termine del framework) come da figura



Abbiamo così realizzato una base che è possibile continuare ad implementare rendendola più funzionale, arricchendola con altre zone moduli, inserendo parametri etc etc.

Continuiamo il nostro discorso occupandoci adesso del sistema dei menù in Joomla!

I menù in Joomla!

Il lavoro degli sviluppatori è stato sicuramente imponente, da un lato hanno reso il sistema se possibile ancora più facile ed intuitivo da utilizzare, dall'altro hanno fornito strumenti di personalizzazione avanzata del cms senza trascurare il discorso accessibilità.

Intendiamoci il sistema ricorre ancora ad un output di tipo tabellare (anche se attraverso l'override dei componenti è possibile intervenire con una certa facilità) ma ad esempio per quello che riguarda la creazione dinamica delle voci di menù esso riconosce ed accetta le "ul" e l'assegnazione di classi alle liste.

Questo porta ad una sicura facilitazione nella realizzazione di menù anche dal bell'aspetto grafico senza dover necessariamente ricorrere all'impostazione della classe "mainlevel" come nella versione 1.0.x.

1° caso i selettori specifici

E' possibile impostare gruppi di menù per determinate aree del sito ricorrendo ai selettori specifici.

Poniamo il caso di voler fare in modo che tutte le voci di menù della colonna sinistra abbiano un determinato aspetto differenziato dalle voci di menù che potremmo presentare nella colonna destra, come fare?

Nel nostro template "test" la colonna sinistra ha l'id #colonnasinistra

Esistono nel web centinaia di risorse utili e gratuite che possiamo utilizzare, una lista di queste è inserita in coda alla guida, per realizzare il nostro menu abbiamo scelto:

<http://www.exploding-boy.com/images/EBmenus/menus.html>

in particolare il menù numero 11 che ha il seguente listato:

```
#menu11 li a {
```

```
    height: 32px;
```

```
    voice-family: "\"}\"\"";
```

```
    voice-family: inherit;
```

```
    height: 24px;
```

```
    text-decoration: none;
```

```
}
```

```
#menu11 li a:link, #menu11 li a:visited {
```

```
    color: #889944;
```

```
    display: block;
```

```
    background: url(menu11.gif);
```

```
    padding: 8px 0 0 25px;
```

```
}
```

```
#menu11 li a:hover {
```

```
    color: #889944;
```

```
    background: url(menu11.gif) 0 -32px;
```

```
    padding: 8px 0 0 27px;
```

```
}
```

Per inglobarlo all'interno della nostra struttura sarà sufficiente impostarlo in maniera "selettiva" sostituendo l'id #menu11 con l'id #colonnasinistra e dando il percorso esatto all'immagine di sfondo del menu (risale di una cartella) dunque inseriremo nel foglio di stile:

```
#colonnasinistra li a {
```

```
    height: 32px;
```

```
    voice-family: "\"}\"\"";
```

```
    voice-family: inherit;
```

```
    height: 24px;
```

```
    text-decoration: none;
```

```
}
```

```
#colonnasinistra li a:link, #menu11 li a:visited {  
  
    color: #889944;  
  
    display: block;  
  
    background: url(../images/menu11.gif);  
  
    padding: 8px 0 0 25px;  
  
}
```

```
#colonnasinistra li a:hover {  
  
    color: #889944;  
  
    background: url(../images/menu11.gif) 0 -32px;  
  
    padding: 8px 0 0 27px;  
  
}
```

Salviamo nel foglio di stile,importiamo l'immagine nella cartella e controlliamo il risultato nel browser



Ed ecco il nostro menu relativo solo alla colonna sinistra pubblicato.

Si raccomanda ovviamente di impostare il modulo menu come “lista” per il corretto funzionamento (nella versione 1.0.x è chiamata “flat list”).

Identico discorso è ovviamente possibile con i menù orizzontali.

Copyright © <2008>, <[joomlapixel](#)> rilasciato sotto licenza Creative commons 3.0 non commercial

This help file has been generated by the freeware version of [HelpNDoc](#)

2° caso le classi menù in Joomla! 1,5

2° caso – le classi menù in joomla 1,5

In Joomla! 1,5 è possibile comunque utilizzare un suffisso per il menù principale, bisogna però ricordare che ragioniamo sempre in termini di “liste” e questo ci porta ad affrontare il discorso delle opzioni di output dei moduli in questa versione.

Prima di farlo abilitiamo lo stile "xhtml" per il modulo

```
<jdoc:include type="modules" name="right" style=" xhtml " />
```

Di seguito un listato più complesso che utilizza la classe `.moduletable` e il suffisso `_menu` che riportiamo nel foglio di stile:

`.moduletable_menu ul, .moduletable_menu ul li`

```
{  
  
    background-image: none;  
  
    display      : block;  
  
    width       : auto !important;  
  
    width       : 100%;  
  
    margin      : 1px 0;  
  
    padding     : 0;  
  
    font-size   : 13px;  
  
    background  : none;  
  
}
```

`.moduletable_menu ul li a, .moduletable_menu ul li a:link, .moduletable_menu ul li a:visited`

```
{  
  
    display     : block;  
  
    font-size   : 13px;  
  
    border-bottom : 1px solid #DCDFE1;  
  
    font-weight  : normal;  
  
    padding     : 3px 0;  
  
    color       : #666666;  
  
    padding-left : 10px;  
  
    text-decoration : none;  
  
}
```

`.moduletable_menu ul li a:hover`

```
{
```

border-bottom : 1px solid #DCDFE1;

font-size : 13px;

font-weight : normal;

color : #666666;

text-decoration : none;

padding-left : 10px;

text-decoration : none;

background : #E5E5E5;

}

.moduletable_menu ul li.active a:link, .moduletable_menu ul li.active a:visited

{

color : #8B1C00;

border-bottom : 1px solid #DCDFE1;

background : #E5E5E5;

text-decoration : none;

font-size : 13px;

font-weight : bold;

}

.moduletable_menu ul li.active ul li a:hover, .moduletable_menu ul li.active ul li ul li a:hover

{ background:none;

font-size : 13px;

font-weight : normal;

color : #8B1C00;

text-decoration : none;

}

.moduletable_menu ul li ul li a:link, .moduletable_menu ul li ul li a:visited, .moduletable_menu ul li ul li ul li a:link, .moduletable_menu ul li ul li ul li a:visited

```
{  
  
    font-size      : 12px;  
  
    font-weight    : bold;  
  
    color          : #999999;  
  
    text-decoration : none;  
  
    margin-left    : 15px;  
  
    border         : none;  
  
}
```

.moduletable_menu ul li ul li a:hover, .moduletable_menu ul li ul li ul li a:hover

```
{ background      : none;  
  
    font-size     : 12px;  
  
    font-weight   : normal;  
  
    color         : #666666;  
  
    text-decoration : none;  
  
}
```

.moduletable_menu ul li.active ul li a:link, .moduletable_menu ul li.active ul li a:visited,
.moduletable_menu ul li.active ul li a:hover, .moduletable_menu ul li.active ul li ul li a:link,
.moduletable_menu ul li.active ul li ul li a:visited, .moduletable_menu ul li.active ul li ul li a:hover

```
{  
  
    background     : none;  
  
    font-size      : 12px;
```

```
font-weight    : normal;

color          : #666666;

text-decoration : none;

}
```

ul.menu li.parent ul li#current a

```
{ background    : none;

font-size      : 12px;

font-weight    : bold;

color          : #000000;

text-decoration : none;

}
```

Il risultato a video per la colonna destra in questo caso sarà il seguente



Per riuscire a farlo recepire dal sistema ho dovuto anticipare i tempi.....

Nell'atto di inserimento dell'include che richiama i moduli abbiamo evidenziato questo codice relativo:

```
<jdoc:include type="modules" name="right" style=" " />
```

Chiedendo al framework di inserire (include) un modulo dal nome right..... E dallo stile NON DEFINITO.

Affinchè il sistema funzioni è bastato settare lo stile in "xhtml"

Affrontiamo ora questo discorso in maniera più approfondita

Il foglio di stile del core di Joomla!

[Approfondimenti,tutorial e sviluppi disponibili su www.joomlapixel.eu](http://www.joomlapixel.eu)

Per riuscire a spingere all'estremo la possibilità di personalizzazione di joomla è stato necessario,da parter dei developers soprattutto suddividere e creare un considerevole numero di classi e ID associate al framework.

Riuscire a tenerle a mente tutte è un'impresa,ma in realtà non è assolutamente necessario grazie ad alcuni semplici strumenti open source per firefox reperibili in rete.

- [aardvark extension](#)
- [csswiever](#)

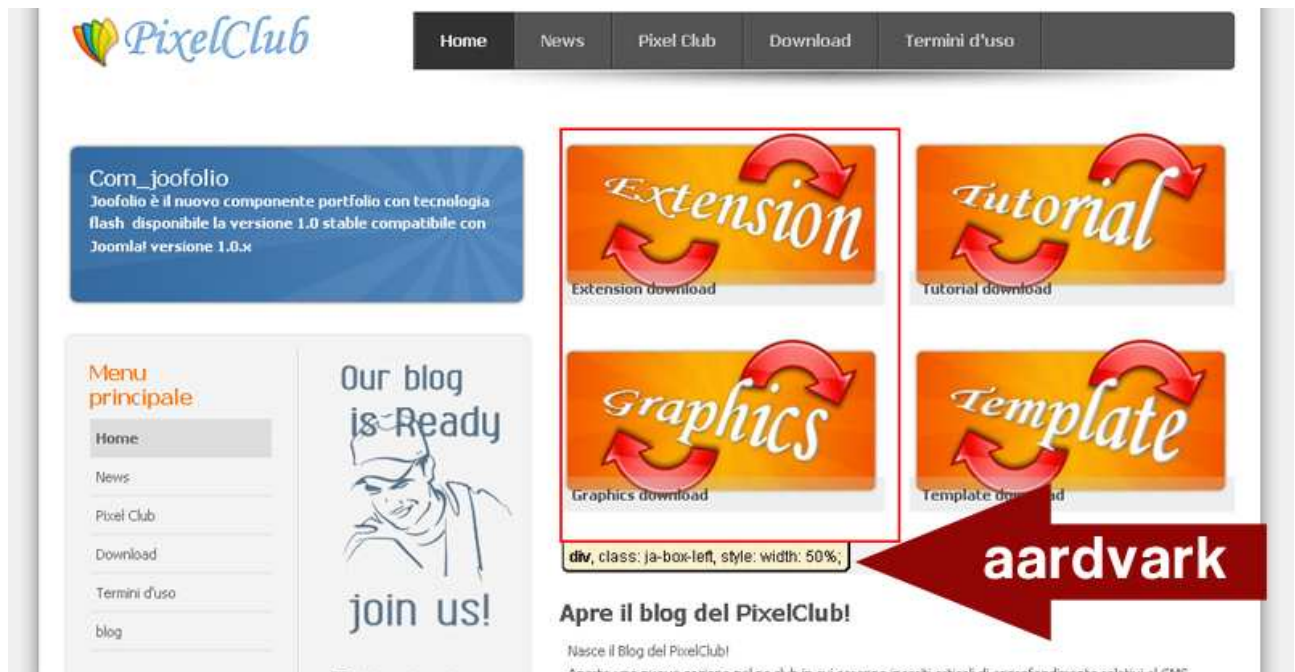
Si tratta di due strumenti a mio avviso poco considerati,di solito si preferisce Firebug o la Webdeveloper toolbar,ottimi strumenti indubbiamente,ma essendo un sostenitore convinto soprattutto della rapidità d'utilizzo e della semplicità ho adottato questi diventati per me irrinunciabili.

Una volta scaricate ed installate le extension per Firefox il loro utilizzo risulta molto semplice,non c'è assolutamente bisogno di perdersi nella miriade di opzioni da settare per gli altri.

E' sufficiente un click col tasto destro in una finestra di Firefox per lanciare aardvark e una volta attivato questi ci fornirà (contornando con un riquadro in rosso le varie zone (div o tabelle) del layout e fornendoci il nome identificativo.

Insuperabile ad esempio quando si ha necessità di individuare con precisione delle aree nel template.

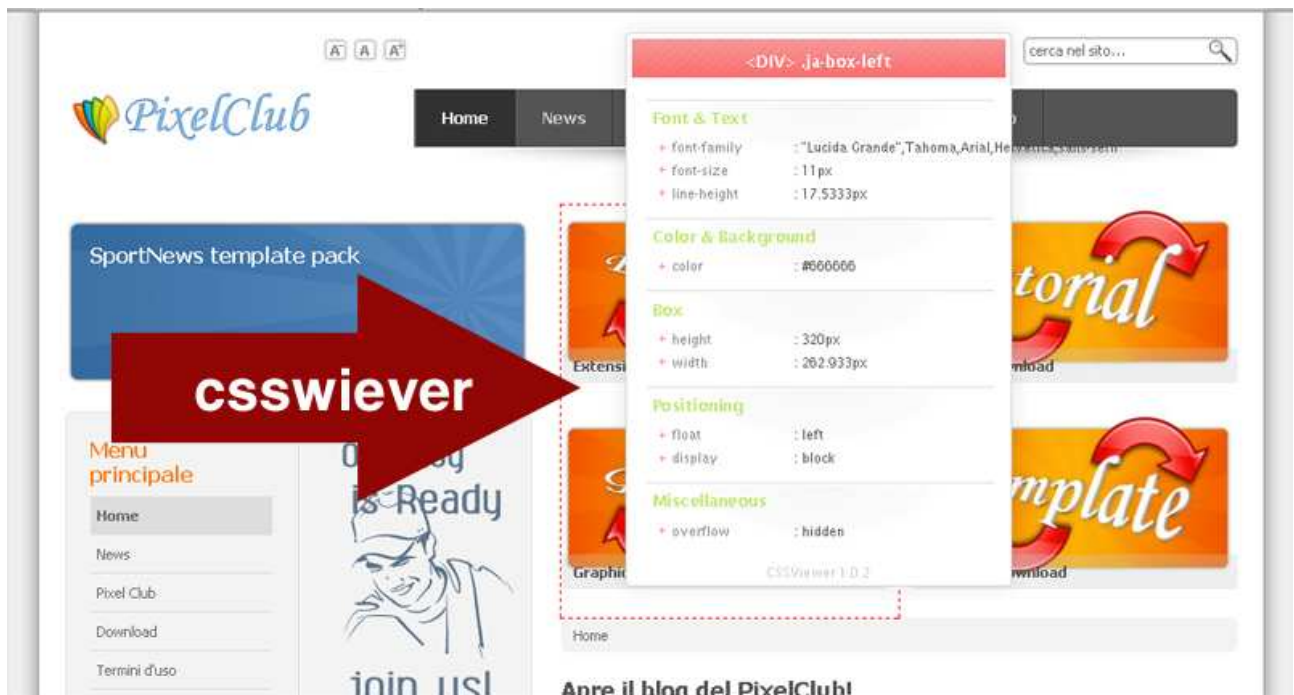
Questo uno screenshot:



continuando a muoversi sulla pagina le altre zone saranno evidenziate allo stesso modo.

Nell'eventualità che sia necessario invece "leggere" tutte le impostazioni css di un area l'aiuto ci viene fornito da csswiewer, in questo caso attivabile dalla barra degli strumenti superiore di firefox.

Screenshot



và da sè che lavorando con il vostro editor css preferito riuscirete facilmente ad intervenire per correggere/impostare il foglio di stile del Vs template con rapidità e senza necessariamente conoscere a memoria tutte le classi di joomla o della struttura che avete creato.

Classi del core di Joomla

Classi CSS	File del core di Joomla! 1.5
adminform	components\com_content\views\article\tmpl\form.php
admintable	plugins\system\legacy\commonhtml.php
article_column	components\com_content\views\category\tmpl\blog.php
	components\com_content\views\frontpagetmpl\default.php
	components\com_content\views\section\tmpl\blog.php
article separator	components\com_content\views\category\tmpl\blog item.php
	components\com_content\views\article\tmpl\default.php
	components\com_content\views\frontpagetmpl\default item.php
	components\com_content\views\section\tmpl\blog item.php
	modules\mod_newsflash\tmpl\vert.php
author	components\com_content\views\archive\tmpl\default items.php
back_button	plugins\system\legacy\html.php
bannerfooter	modules\mod_banners\tmpl\default.php
bannergroup	modules\mod_banners\tmpl\default.php
bannerheader	modules\mod_banners\tmpl\default.php
banneritem	modules\mod_banners\tmpl\default.php
blog	components\com_content\views\category\tmpl\blog.php
	components\com_content\views\frontpagetmpl\default.php
	components\com_content\views\section\tmpl\blog.php
blog more	components\com_content\views\category\tmpl\blog.php
blogsection	components\com_content\views\category\tmpl\blog links.php
	components\com_content\views\frontpage\tmpl\default links.php
	components\com_content\views\section\tmpl\blog_links.php
breadcrumbs pathway	modules\mod_breadcrumbs\tmpl\default.php
button	plugins\content\vote.php
	components\com_content\views\archive\tmpl\default.php
	components\com_content\views\mailto\tmpl\default.php
	components\com_search\views\search\tmpl\default form.php
	components\com_user\views\login\tmpl\default login.php
	components\com_user\views\login\tmpl\default logout.php
	components\com_user\views\register\tmpl\default.php
	components\com_user\views\user\tmpl\form.php
	modules\mod_login\tmpl\default.php
	modules\mod_poll\tmpl\default.php
	modules\mod_search\helper.php
button validate	components\com_contact\views\contact\tmpl\default form.php
buttonheading	components\com_content\views\article\tmpl\default.php
	components\com_content\views\category\tmpl\blog item.php
	components\com_content\views\category\tmpl\blog item.php
	components\com_content\views\frontpage\tmpl\default item.php

	components\com_content\views\section\tmpl\blog item.php
calendar	libraries\Joomla\html\html.php
category	components\com_contact\views\category\tmpl\default items.php
	components\com_content\views\section\tmpl\default.php
	components\com_newsfeeds\views\categories\tmpl\default.php
	components\com_newsfeeds\views\category\tmpl\default items.php
	components\com_web\tnks\views\categories\tmpl\default.php
clr	modules\mod_banners\tmpl\default.php
Componentheading	Components\com_contact\views\category\tmpl\default.php

	components\com_contact\views\contact\tmpl\default.php
	components\com_content\views\archive\tmpl\default.php
	components\com_content\views\category\tmpl\blog.php
	components\com_content\views\category\tmpl\default.php
	components\com_content\views\frontpage\tmpl\default.php
	components\com_content\views\section\tmpl\blog.php
	components\com_content\views\section\tmpl\default.php
	components\com_mailto\views\mailto\tmpl\default.php
	components\com_mailto\views\sent\tmpl\default.php
	components\com_newsfeeds\views\categories\tmpl\default.php
	components\com_newsfeeds\views\category\tmpl\default.php
	components\com_newsfeeds\views\newsfeed\tmpl\default.php
	components\com_poll\views\poll\tmpl\default.php
	components\com_search\views\search\tmpl\default.php
	components\com_user\views\login\tmpl\default login.php
	components\com_user\views\login\tmpl\default logout.php
	components\com_user\views\register\tmpl\default.php
	components\com_user\views\register\tmpl\default message.php
	components\com_user\views\remind\tmpl\default.php
	components\com_user\views\reset\tmpl\complete.php
	components\com_user\views\reset\tmpl\confirm.php
	components\com_user\views\reset\tmpl\default.php
	components\com_user\views\user\tmpl\default.php

	components\com user\views\user\tmpl\form.php
	components\com webCinks\views\categories\tmpl\default.php
	components\com web(inks\views\category\tmpl\default.php
	components\com weblinks\views\weblink\tmpl\form.php
	components\com wrapper\views\wrapper\tmpl\default.php
contact email	components\com contact\views\contact\tmpl\default form.php
content rating	plugins\content\vote.php
content vote	plugins\content\vote.php
contentdescription	components\com contact\views\category\tmpl\default.php
	components\com content\views\category\tmpl\default.php
	components\com content\views\section\tmpl\default.php
	components\com newsfeeds\views\categories\tmpl\default.php
	components\com newsfeeds\views\category\tmpl\default.php
	components\com weblinks\views\categories\tmpl\default.php
	components\com weblinks\views\category\tmpl\default.php
contentheading	components\com contact\views\contact\tmpl\default.php
	components\com content\views\article\tmpl\default.php
	components\com content\views\category\tmpl\blog item.php
	components\com content\views\frontpage\tmpl\default item.php
	components\com content\views\section\tmpl\blog item.php
	components\com newsfeeds\views\newsfeed\tmpl\default.php
	modules\mod newsfeeds\tmpl\ item.php
contentpagetitle	components\com content\views\article\tmpl\default.php
	components\com content\views\category\tmpl\blog item.php
	components\com content\views\frontpage\tmpl\default item.php
	components\com content\views\section\tmpl\blog item.php

contentpaneopen	components\com content\views\article\tmpl\default.php
	components\com content\views\category\tmpl\blog item.php
	components\com content\views\frontpage\tmpl\default item.php
	components\com content\views\section\tmpl\blog item.php
	components\com search\views\search\tmpl\default form.php
	components\com search\views\search\tmpl\default results.php
	modules\mod newsfeeds\tmpl\ item.php
ctr	xmlrpc\client.php

description	components\com webiinks\views\category\tmpl\default items.php
editlinktip hasTip	libraries\joomla\html\htmlC.php libraries\joomla\html\html\grid.php
form-validate	components\com con#act\views\contact\tmpl\default form.php components\com user\views\register\tmpl\default.php
hasTip	components\com content\helpers\icon.php components\com user\views\remind\tmpl\default.php components\com user\views\reset\tmpl\complete.php components\com user\viewstreset\tmpl\confirm.php components\com user\views\reset\tmpl\default.php libraries\joomla\html\parameter\element.php
highlight	components\com search\views\search\view.html.php
input	components\com user\views\login\tmpl\default login.php modules\mod_login\tmpl\default.php

inputbox	components\com contact\views\contact\tmpl\default.php components\com contact\views\contact\tmpl\default form.php components\com contact\views\contact\tmpl\default form.php components\com content\views\archive\tmpl\default.php components\com content\views\archive\view.html.php components\com content\views\article\tmpl\form.php components\com content\views\article\view.html.php components\com content\views\article\view.html.php components\com content\views\category\tmpl\default items.php components\com mailto\views\mailto\tmpl\default.php components\com_poll\views\poll\view.html.php components\com search\views\search\tmpl\default form.php components\com search\views\search\view.html.php components\com user\views\login\tmpl\default login.php components\com weblinks\views\weblink\tmpl\form.php components\com weblinks\views\weblink\view.html.php libraries\joomla\html\html\grid.php libraries\joomla\html\html\list.php libraries\joomla\html\html\menu.php libraries\joomla\html\pagination.php libraries\joomla\html\parameter\element\editors.php libraries\joomla\html\parameter\element\filelist.php libraries\joomla\html\parameter\element\folderlist.php libraries\joomla\html\parameter\element\helpsites.php libraries\joomla\html\parameter\element\languages.php libraries\joomla\html\parameter\element\list.php libraries\joomla\html\parameter\element\menu.php libraries\joomla\html\parameter\element\menuitem.php libraries\joomla\html\parameter\element\section.php
----------	--

	libraries\joomla\html\parameter\element\sql.php
	libraries\joomla\html\parameter\element\timezones.php
	modules\mod_login\tmpl\default.php
	modules\mod_search\helper.php
inputbox required	components\com_contact\views\contact\tmpl\default_form.php
	components\com_user\views\register\tmpl\default.php
inputbox required validate-email	components\com_contact\views\contact\tmpl\default_form.php
inputbox required validate-mailbox	components\com_user\views\register\tmpl\default.php
inputbox required validate-passverify	components\com_user\views\register\tmpl\default.php
inputbox required validate-password	components\com_user\views\register\tmpl\default.php
inputbox required validate-username	components\com_user\views\register\tmpl\default.php
int h1	xmlrpc\client.php
intro	components\com_content\views\archive\tmpl\default_items.php
m	plugins\system\legacy\html.php
josForm form-validate	components\com_user\views\remind\tmpl\default.php
	components\com_user\views\reset\tmpl\complete.php
	components\com_user\views\reset\tmpl\confirm.php
	components\com_user\views\reset\tmpl\default.php
jpane-slider content	libraries\joomla\html\pane.php
jpane-toggler title	libraries\joomla\html\pane.php

key	components\com_content\views\article\tmpl\form.php
	components\com_content\views\article\tmpl\pagebreak.php
	components\com_content\view\starticle\tmpl\pagebreak.php
	plugins\system\legacy\common.html.php
latestnews	modules\mod_latestnews\tmpl\default.php
loc link	plugins\content\pagebreak.php
message	components\com_user\views\register\tmpl\default_message.php
metadata	components\com_content\views\archive\tmpl\default_items.php
modal-button	plugins\editors\none.php
	plugins\editors\tinymce.php
	plugins\editors\standard.php
modifydate	components\com_content\views\article\tmpl\default.php
	components\com_content\views\category\tmpl\blog_item.php
	components\com_content\views\frontpage\tmpl\default_item.php
	components\com_content\views\section\tmpl\blog_item.php
moduletable	modules\mod_feed\tmpl\default.php
	modules\mod_newsflash\tmpl\horiz.php
mostread	modules\mod_mostread\tmpl\default.php
newsfeed	modules\mod_feed\tmpl\default.php
page	xmlrpc\client.php
pagecounter	components\com_weblinks\views\category\tmpl\default

	items.php
pagenavbar	plugins\content\pagebreak.php
pagenav	plugins\content\pagenavigation.php
pagenav_next	plugins\content\tpagenavigation.php
pagenav_prev	plugins\content\tpagenavigation.php
pagenavcounter	plugins\content\pagebreak.php
panel	libraries\joomla\html\pane.php
pane-siiders	libraries\joomla\html\pane.php
paramlist	libraries\joomla\html\parameter.php
paramiist admintable	libraries\joomla\html\parameter.php
paramlist description	libraries\joomla\html\parameter.php
paramlist key	libraries\joomla\html\parameter.php
paramlist value	libraries\joomla\html\parameter.php
pathway	modules\mod_breadcrumbs\tmpl\default.php
patTemplate	libraries\pattemplate\patTemplate\Dump\Html.php
patTempfate patTemplateSign	libraries\pattemplate\patTemplate\Dump\Html.php
patTemplateCol	librades\pattempiate\patTempiate\Dump\Htmi.php
	fibraries\pattemplate\patTemplate\Dump\HtmlL.php
	libraries\pattemplate\patTemplate\Dump\Html.php
	libraries\pattemplate\patTemplate\Dump\Html.php
	libraries\pattemplate\patTemplate\Dump\Html.php
	librades\pattempiate\patTempiate\Dump\Htmi.php
	libraries\pattemp!ate\patTemplate\Dump\Html.php
	libraries\pattemplate\patTemplate\Dump\Html.php
patTemplateData	librades\pattempiate\patTempiate\Dump\Htmi.php
patTemplateProp	libraries\pattemplate\patTemplate\Dump\Html.php
patTempiateSection	libraries\pattemplatelpatTemplate\Dump\Html.php
patTemplateSource	librades\pattempiate\patTempiate\Dump\Htmi.php
patTemplateSub	librades\pattempiate\patTempiate\Dump\Htmi.php

patTempiateSubSection	libraries\pattemplate\patTemplate\Dump\Html.php
patTemplatesubCol	libraries\pattemplats\patTemplatelDump\Html.php
patTempiateTpl	libraries\pattempla#e\patTemplate\Dump\Html.php
patTemplateTmpIContent	libraries\pattemplate\patTemplatelDump\Html.php
patTemplateVar	libraries\pattemplate\patTemplate\Dump\Html.php
patTemplteCol	libraries\pattemplatelpatTemplate\Dump\Html.php
poli	modulesMod-poiKtmpikdefault.php
polistabieborder	components\com_pofi\views\poil\tmpl\default_graph.php
	modules\mod_poll\tmpRdefault.php
profiler	piugins\system\debug.php
	plugins\system\legacyadminmenus.php
readon	components\com_content\views\category\tmpl\blog item.php
	components\com
	con#entviews\frontpage\tmpl\default_item.php
	components\com_content\views\section\tmpl\blog item.php
relateditems	modules\mod reated items\tmpl\default.php

required	components\com user\views\reset\tmpl\confirm.php
required validate-email	components\com user\views\remind\tmpl\default.php
required validate-email	components\com user\views\reset\tmpl\default.php
required validate-password	components\com user\views\reset\tmpl\complete.php
required validate-password	components\com user\views\reset\tmpl\complete.php
row	components\com content\views\archive\tmpl\default_items.php
search	modules\mod_search\tmpl\default.php
searchintro	components\com search\views\search\tmpl\default_error.php
	components\com search\views\search\tmpl\default_form.php
section_colour bar	xmlrpc\client.php
sectionableheader	components\com_poll\views\poll\tmpl\default_graph.php
sections	modules\mod_sections\tmpl\default.php
sectiontableentry	components\com_contact\views\category\tmpl\default_items.php
	components\com_content\views\category\tmpl\default_items.php
	components\com_newsfeeds\views\category\tmpl\default_items.php
	components\com_weblinks\views\category\tmpl\default_items.php
sectiontablefooter	components\com_contact\views\category\tmpl\default.php
	components\com_content\views\category\tmpl\default_items.php
	components\com_newsfeeds\views\category\tmpl\default_items.php
	components\com_weblinks\views\category\tmpl\default_items.php
sectiontableheader	components\com_contact\views\category\tmpl\default.php
	components\com_content\views\category\tmpl\default_items.php
	components\com_newsfeeds\views\category\tmpl\default_items.php
	components\com_weblinks\views\category\tmpl\default_items.php
sectiontableentry	components\com_poll\views\poll\tmpl\default_graph.php
	components\com_poll\views\poll\tmpl\default_graph.php
separator	modules\mod_mainmenu\helper.php
smail	components\com_content\views\article\tmpl\default.php
	components\com_content\views\category\tmpl\blog_item.php
	components\com_content\views\frontpage\tmpl\default_item.php
	components\com_content\views\section\tmpl\blog_item.php
	components\com_content\views\section\tmpl\default.php
	components\com_newsfeeds\views\categories\tmpl\default.php
	components\com_search\views\search\tmpl\default_results.php

	components\com_weblinks\views\categories\tmpl\default.php
--	---

	plugins\system\legacy\html.php
smalldark	components\com_poll\views\poll\tmpl\default_graph.php
system-pagebreak	plugins\content\pagebreak.php
system-unpublished	components\com_content\views\category\tmpl\blog_item.php
	components\com_content\views\frontpage\tmpl\default_item.php
	components\com_content\views\section\tmpl\blog_item.php
Table	libraries\joomla\document\error\error.php
tabs	libraries\joomla\html\pane.php
TD	libraries\joomla\document\error\error.php
text area	libraries\joomla\html\parameter\element\hidden.php
	libraries\joomla\html\parameter\element\password.php
	libraries\joomla\html\parameter\element\text.php
	libraries\joomla\html\parameter\element\textarea.php
title	components\com_content\views\archive\tmpl\default_items.php
toolbar	libraries\joomla\html\toolbar.php
	plugins\system\legacy\toolbar.php
validate	components\com_user\views\remind\tmpl\default.php
	components\com_user\views\reset\tmpl\complete.php
	components\com_user\views\reset\tmpl\confirm.php
	components\com_user\views\reset\tmpl\default.php
wrapper	components\com_wrapper\views\wrapper\tmpl\default.php
	modules\mod_wrapper\tmpl\default.php

ID del core di Joomla

CSS ID	File del core di Joomla! 1.5
adminForm	components/com_weblinks/vfews/weblink/tmpl/form.php
alt	components/com_content/views/article/tmpl/pagebreak.php
alternatel	plugins/editors/fxstandard.php
archive-list	components/com_contentJviews/archive/tmpl/default items.php
blockrandom	components/com_wrapperJviews/wrapper/tmpl/default.php modules/mod_wrapper/tmpl/default.php
contact email)	components/com_contact/viewsJcontact/tmpl/default form.php
contact email_copy	components/com_contactJviews/contact/tmpl/default form.php
contact emailmsq	components/com_contactJviewsJcontactJtmpl/default form.php
contact name	components/com_contactJviewsJcontact/tmpl/default form.php
contact subject	components/com_contactJviewsJcontact/tmpl/default form.php
contact text	components/com_contactJviews/contact/tmpl/default form.php
contact textmsg	components/com_contactJviewsJcontactJtmpl/default form.php
created by_alias	components/com_contentJviews/articleJtmpl/form.php
email	components/com_user/viewsJregister/tmpl/default.php components/com_userJviews/remind/tmpl/default.php components/com_user/views/reset/tmpl/default.php components/com_user/views/user/tmpl/form.php
emailForm	components/com_contact/viewsJcontact/tmpl/default form.php
emailmsq	components/com_user/viewsJregisterJtmpl/default.php
explain-sql	libraries/joomla/database/database/mysql.php libraries/joomla/database(database/mysql).php
form-login	components/com_user/views/loqinJtmpl/default login.php modules/mod_login/tmpl/default.php modulesJmod loqinJtmpl/default.php
form-login-password	components/com_userJviewsJMoqin/tmpl/default login.php modules/mod_login/tmpl/default.php
form-login-remember	components/com_user/viewsJloginftmpl/default login.php modules/mod loqinJtmpl/default.php
form-login-username	components/com_user/viewsJloqin/tmpl/default login.php modules/mod loqin/tmpl/default.php
jForm	components/com_contentJviewsfarchive/tmpl/default.php
jformdescription	components/com_weblinksJviewsJweblink/tmplJform.php
jformtitle	components/com_weblinksJviews/weblink/tmpl/form.php
jformurl	components/com_weblinksJviews/weblink/tmpl/farm.php
josForm	components/com_userJviewsJregister/tmpl/default.php
login	components/com_userJviewsJlogmJtmpl/default logout.php
metadesc	components/com_contentJviews/article/tmpl/form.php
metakey	components/com_contentJviews/article(tmpl/form.php
mod_search searchword	modules/mod_searchJhelper.php
modlg	modules/mod_login/tmpl/default.php modulesJmod loqinJtmpl/default.php
modlg_remember	modules/mod_loqin/tmpl/default.php
name	components/com_user/views/register/tmpl/default.php components/com_userJviews/uSerJtmplJform.php

namemsg	components/com_userJviewsJregister/tmplJdefault.php
navigation	components/com_content/views/archive/tmpl/default items.php
password	components/com_user/views/register/tmpl/default.php
	components/com_userJviews/user/tmpl/form.php
passwordl	components/com_userjviews/resetftmplJcomplete.php
password2	components/com_user/views(register/tmpl/default.php
	components/com_userJviews/reset/tmpl/complete.php
	components/com_user/views/user/tmpl/form.php
patTemplateContent	libra ries/pattemplate/patTemp(atef DumpJfitml.php
poli	components/com-poli/views/poli/tmpl/default.php
pw2msg	componentsjcom user/viewsJregisterjtmpl/default.php
pwmsq	components/com user/viewsJregister/tmpl/default.php
search searchword	components/com search/views/searchJtmpl/default form.php
searchForm	components/com search/views/searchJtmpl/default form.php
selectForm	components/com contactJviews/contact/tmpl/default.php
system-debug	piugins/systemJdebug.php
system-pagebreak	piugins/editors/xstandard/directory.php
system-readmore	componentsjcom contentjmodefs(article.php
	pluqins/editors/xstandard/directory.php
titie	components/com contenijwews/article/tmplJform.php
	components/com_contentJviewsJarticle/tmpl/pagebreak.php

token	components/com_user/views/resetJtmptf confirm.php
toolbar	pi uqins/systemllegacy/toolbar.php
username	components/com user/views/login/tmpljdefault login.php
	components/com userJviews/register/tmpl/default.php
usernamemsg	campo nents/com_userJviews/reqFster/tmpl/default.php
xstandard	pluqins/editors/xstandard.php

Personalizzazione dei moduli

[Approfondimenti,tutorial e sviluppi disponibili su www.joomlapixel.eu](http://www.joomlapixel.eu)

Chi ha già maturato una piccola esperienza con il ns CMS ha già sicuramente sentito parlare della possibilità di personalizzare l'output dei moduli,e parlando sempre a livello di codice tableless ricorderà gli stili

-1

-2

-3

Semplici numeri che ci permettono di ottenere risultati differenti e tali da coprire ogni circostanza di pubblicazione.

Questo forse è l'ambito che ha ricevuto cambiamenti minori rispetto al resto,un semplice cambio di sigle può sicuramente essere ritenuto tale.

Di seguito una tabella comparativa che evidenzia i cambiamenti:

versione 1.0.x	Versione 1.5
-1	none
-2	Xhtml
-3	Rounded
	Mod_Chrome

Cambiamenti semplici.....tranne per l'introduzione di una nuova possibilità di personalizzazione evidenziata in rosso e chiamata Mod_Chrome.

Vediamo comunque nel dettaglio e con esempi pratici cosa è possibile ottenere utilizzando gli stili dei moduli.

scelta dello stile

```
<jdoc:include type="modules" name="user2" style=" " />
```

Con I relativi stili

none

xhtml

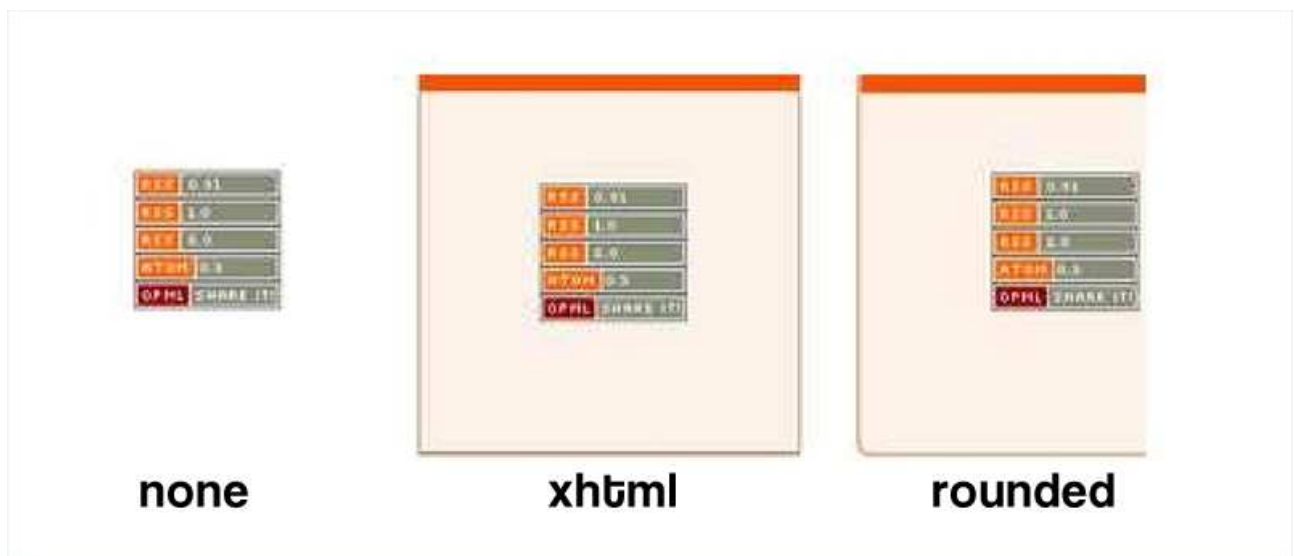
rounded

modChrome

omettiamo volutamente di far riferimenti agli stili che generano un output di tipo tabellare.

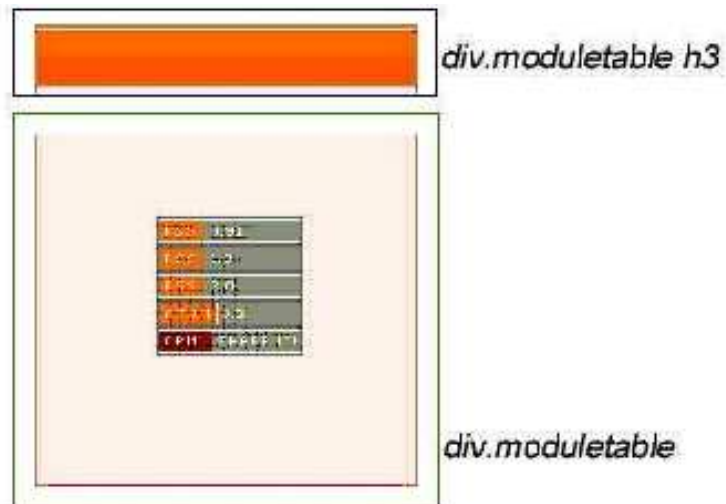
Ma quale è l'esatto significato di questa espressione? o perlomeno quali sono gli effetti tangibilmente visibili delle variazioni?

Il tutto dovrebbe apparire maggiormente chiaro guardando le seguenti immagini:



Lo stile “none” sembra essere graficamente meno accattivante rispetto agli altri due, ma non è del tutto vero come vedremo più avanti.

Stile xhtml



```
div.moduletable {  
padding:  
margin:  
border:  
}
```

```
div.moduletable h3 {  
margin:  
padding:  
color:  
font-size:  
background:  
}
```

Stile xhtml

L'esempio è ovviamente solo un riferimento l'immagine seguente mostra quali sono le aree di pertinenza del modulo:

L'immagine è una forzatura, serve solo a distinguere le due zone, nella realtà dobbiamo tener presente che esiste un contenitore principale `div.moduletable` al cui interno è presente uno spazio, un blocco che mostra il titolo del modulo.

A livello di gestione grafica e di impaginazione a questo punto possiamo:

impostare un'immagine di sfondo, un colore di sfondo, un padding e degli attributi di testo al titolo del nostro modulo come in questo esempio:



```
div.moduletable h3 {  
  
padding: 0;  
  
color: #FFFFFF;  
  
font-size: 1.2em;  
  
text-indent: 8px;  
  
}
```

E nel corpo dell'intero modulo impostare larghezza, inserire bordi, colore/immagine di sfondo, padding e gestire attributi del testo.

```
div.moduletable {  
  
padding: 2px 3px;  
  
margin: 3px 0 22px;  
  
border: 1px solid #C6DBE9;  
  
background: #FFFFFF;  
  
width: 180px;  
  
}
```

La struttura ovviamente si adatta al contenuto in altezza e ciò verrà considerato se decidiamo di utilizzare immagini di sfondo, ad esempio una piccola clipart in basso a destra o uno sfondo ripetuto, utilizzando le adeguate impostazioni nel foglio di stile.

Lo stile “rounded”

La struttura che racchiude e definisce questo stile del modulo è la più complessa, permette di gestire livelli grafici articolati con una regolazione molto fine dell’output.

E’ infatti possibile strutturare moduli adattabili sia in altezza che in larghezza rispetto ai contenuti degli stessi. Per capire meglio il concetto si pensi ad un modulo estensibile per l’intera larghezza disponibile (ovviamente riferita alle specifiche della zona di pubblicazione) come dal seguente esempio

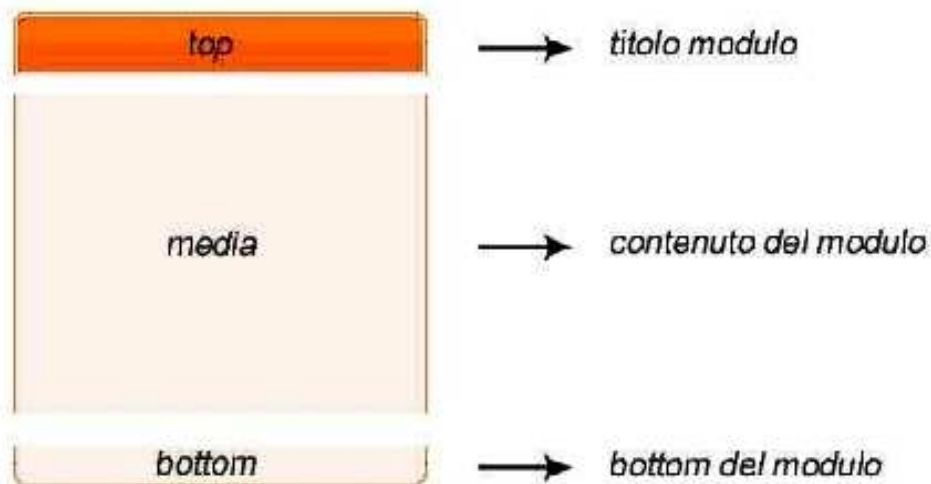


Esistono almeno tre diverse possibilità di strutturazione del layout con strutture di questo tipo, la scelta è ovviamente legata alle necessità finali e va considerata nella prima fase di stesura grafica del layout.

Questa impostazione di stile nei moduli permette soprattutto l’uso di immagini maggiormente rifinite come ad esempio box con angoli arrotondati.

Analizziamo il codice css prendendo come riferimento 3 diversi tipi di strutture:

Modulo a larghezza fissa con angoli arrotondati



Trova la sua applicazione principale in colonne left – right o comunque in zone ad estensione verticale, il listato css utilizzato è il seguente:

```
div.module h3 {  
margin: 0;  
font-size: 1em;  
margin-top: 5px;  
margin-bottom: 20px;  
color: #000;  
}  
  
div.module {  
width: 200px;  
background: url(../images/media.gif) repeat-y;
```

```
margin-bottom: 2px;

margin-right: 15px;

}

div.module div {

background: url(../images/bottom.gif) bottom left no-repeat;

}

div.module div div {

background: url(../images/top.gif) top left no-repeat;

padding: 7px 10px 15px 12px;

}

div.module div div div {

background: none;

padding: 0;

width: auto !important;

width: 100%;

}
```

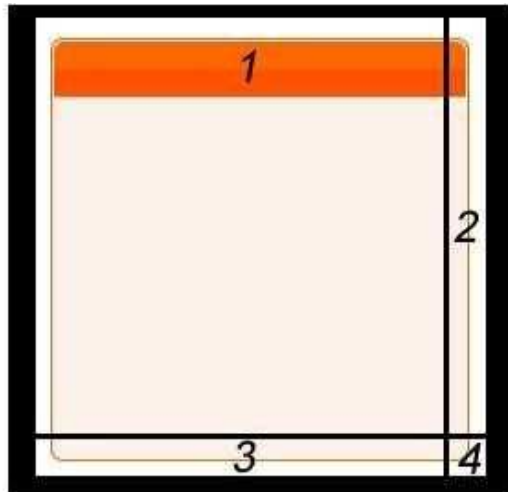
In questa realizzazione inseriamo immagini di background solo in 3 div ed impostando la larghezza del modulo in 200 px.

Realizzare moduli dall'aspetto grafico migliore è abbastanza semplice seguendo le impostazioni del css e l'immagine in alto

Modulo ad altezza e larghezza variabile

Se il nostro intento è invece quello di realizzare un box con grafica particolare e che abbia nel contempo la possibilità di adattarsi in larghezza ed altezza al contenuto (sempre riferito all'area di pubblicazione) è possibile sfruttare in maniera più ampia il

listato css e le sue proprietà di sovrapposizione delle immagini



grafica del modulo esplosa

```
div.module h3 {  
    font-family: Helvetica, Arial, sans-serif;  
    font-size: 14px;  
    font-weight: bold;  
    color: #333;  
    margin: -2px -8px 0 -8px;  
    padding-left: 10px;  
    padding-bottom: 2px;  
}  
  
div.module {  
    background: url(../images/1.gif) 0 0 no-repeat;  
    margin: 0;  
    padding: 0;
```

```
margin-bottom: 20px;

}

div.module div {

background: url(../images/2.gif) 100% 0 no-repeat;

}

div.module div div {

background: url(../images/3.gif) 0 100% no-repeat;

}

div.module div div div {

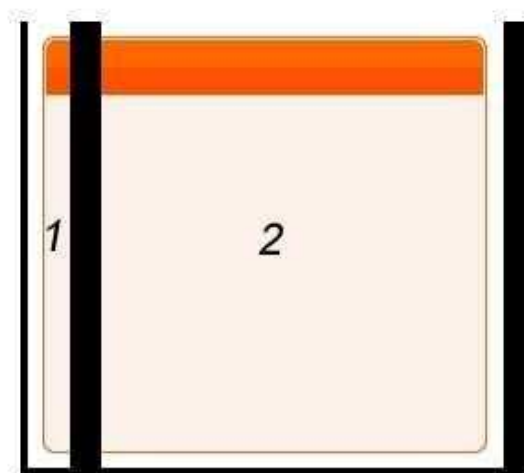
background: url(../images/4.gif) 100% 100% no-repeat;

padding: 8px;

width: auto !important;

width: 100%;}
```

I due metodi descritti risultano utilissimi nella personalizzazione grafica dei moduli ma
è possibile ottenere risultati altrettanto gradevoli utilizzando due sole immagini



grafica del modulo esplosa

E relativo css

```
.module {  
  
background: url(../images/2.gif) top right no-repeat;  
  
padding:0;  
  
margin:0 0 10px 0;  
  
}  
  
.module h3 {  
  
margin:0;  
  
padding:0 0 4px 0;  
  
border-bottom:#ccc 1px solid;  
  
color: #666;  
  
font: bold 1em Tahoma, Arial, Helvetica, sans-serif;  
  
text-align:center;  
  
}  
  
.module div {  
  
background: url(../images/1.gif) top left no-repeat;  
  
margin:0;  
  
padding:6px 0 0 0;  
  
}  
  
.module div div{  
  
background: url(../images/1.gif) bottom left no-repeat;  
  
padding:0 0 0 5px;  
  
}  
  
.module div div div{  
  
background: url(../images/2.gif) bottom right no-repeat;
```

```
padding:0 5px 5px 0;
```

```
height:auto !important;
```

```
height:1%;
```

```
}
```

Giocando dunque sulla sovrapposizione dei livelli è possibile utilizzare 2 sole immagini per abbellire l'output dei nostri moduli.

ModChrome

Il module chrome permette di definire, se le modalità descritte nel paragrafo dei moduli risultassero insoddisfacenti un output dei moduli personalizzato

Il sistema è sempre il medesimo, piccole porzioni di codice in php che possiamo inserire all'interno di una qualsiasi struttura in html.

le porzioni interessate sono evidenziate in rosso, per applicare il nuovo stile del modulo è sufficiente salvare con il nome modules.php all'interno della cartella "html" del template un listato come quello sotto riportato

```
function modChrome_nomedellostile($module, &$params, &$attribs)

{

if (!empty ($module->content)) : ?>

<div class="module">

<div class="module-top"></div>

<div class="module-repeatable">

<?php if($module->showtitle): ?><?php if($module->title): ?><div

class="module-title"><h3><?php echo $module->title; ?></h3></div><?php endif;

?><?php endif; ?>

<div class="module-content"><?php echo $module->content; ?></div>

</div>

<div class="module-bottom"></div>

</div>

<?php endif;

}
```

All'atto pratico possiamo dunque definire ad esempio un contenitore particolare in cui il titolo del modulo si trovi in basso e non in alto come normalmente avviene

Le zone collapsabili

[Approfondimenti,tutorial e sviluppi disponibili su www.joomlapixel.eu](http://www.joomlapixel.eu)

Collassabilità delle zone ovvero implementare una struttura dinamica del layout

Uno degli aspetti più interessanti e ricercati nello sviluppo dei template per Joomla! è la possibilità di avere una qualche forma di dinamismo anche nella struttura dell'impaginazione, riuscire cioè a sfruttare un sistema di pubblicazione non strettamente vincolato e ripetitivo per tutte le pagine del sito.

Da una parte con la gestione di pubblicazione dei moduli questo viene già realizzato (ci riferiamo ad esempio alla possibilità di assegnare i moduli agganciandoli a specifiche pagine del sito, ad esempio un modulo login solo in home, un modulo sondaggi visibile in un'altra o più pagine etc).

Esiste una funzionalità aggiuntiva che possiamo considerare ed impostare nell'atto di realizzazione del file index.php, la possibilità cioè di fare in modo che il sistema si renda conto della pubblicazione o meno dei moduli variando di conseguenza la struttura del layout.

Abbiamo visto nel template test che abbiamo creato e del quale stiamo implementando le funzioni che i moduli sono presenti all'interno di zone (DIV) ben definite:

abbiamo cioè per esempio considerando la colonna sinistra scritto

```
<!-- colonna sinistra -->
```

```
<div id="colonnasinistra">
```

```
<jdoc:include type="modules" name="left" style="" />
```

```
</div>
```

```
<!-- fine colonna sinistra -->
```

Con l'include della direttiva jdoc abbiamo richiamato a video i moduli left

Questa è una soluzione "statica" nel senso che una struttura di questo tipo prevede comunque la pubblicazione dei moduli in quell'area, qualora non pubblicassimo nessun modulo nella colonna

sinistra ci ritroveremmo con un impaginazione sgradevole, avremmo cioè comunque visibile ma vuota la zona relativa.

Possiamo in realtà gestire anche questo aspetto facendo ricorso questa volta ad entrambe le direttive `jdocument` e `jdocumenthtml`.

Nelle pagine precedenti abbiamo dato chiarimenti in merito a queste due funzioni dicendo che `jdoc` opera un `include` mentre l'altra attraverso la variabile `$this` verifica l'esistenza o meno di una condizione.

Il fattore importante da considerare è che questo funziona perfettamente se collegato ad una struttura `html` definita!

Basterà cambiare il listato sopra descritto in questo modo:

```
<!-- colonna sinistra -->

<?php if($this->countModules('left')) : ?>

    <div id="colonnasinistra">

<jdoc:include type="modules" name="left" style="" />

    </div>

<?php endif; ?>

<!-- fine colonna sinistra -->
```

In sostanza viene chiesto al sistema di verificare l'esistenza di un modulo (pubblicazione) nella zona `div colonnasinistra` e se il risultato è uguale a zero di collassare dando così la possibilità alle zone limitrofe di espandersi per riempire lo spazio ora vuoto.

Ovviamente bisognerà anche effettuare degli aggiustamenti nel foglio di stile che lo ricordiamo ha parametri in larghezza ben definiti.

L'implementazione del nostro template test sarà totale, riferita cioè a tutte e tre le colonne sinistra – contenuti – destra creiamo cioè una struttura flessibile che si adatti utilizzando tutto lo spazio disponibile ed aumentando le possibilità di differenziare l'impaginazione finale.

Per far questo dobbiamo effettuare delle modifiche su entrambi i file interessati:

- `index.php`

- template.css

innanzitutto inseriamo l'istruzione condizionale su tutte le zone in cui sono inseriti/richiamati gli include compresa la zona del componente.

```
<!-- colonna sinistra -->
```

```
<?php if($this->countModules('left')) : ?>
```

```
<div id="colonnasinistra">
```

```
<jdoc:include type="modules" name="left" style="" />
```

```
</div>
```

```
<?php endif; ?>
```

```
<!-- fine colonna sinistra -->
```

```
<!-- contenuti -->
```

```
<div id="contenuti"<?php echo $contentwidth; ?>">
```

```
<jdoc:include type="component" />
```

```
</div>
```

```
<!-- fine contenuti -->
```

```
<!-- colonna destra -->
```

```
<?php if($this->countModules('right')) : ?>
```

```
<div id="colonnadestra">
```

```
<jdoc:include type="modules" name="right" style="" />
```

</div>

<?php endif; ?>

<!-- fine colonna destra -->

Ho evidenziato una parte del codice in colore verde e dal font più marcato per riportare una considerazione che è alla base di tutta l'implementazione:

vogliamo creare una struttura dinamica che regoli il layout in base all'esistenza o meno dei moduli pubblicati dunque le opzioni che vogliamo arrivare a gestire saranno:

- template a 3 colonne sinistra + contenuti-destra
- template a 2 colonne sinistra + contenuti
- template a 2 colonne contenuti + destra
- template a 1 colonna contenuti

Joomla! è un framework basato sul php un linguaggio open molto potente e versatile, con il quale è possibile effettuare anche operazioni di conteggio e a seconda del risultato a video (pubblicazione) gestire delle variabili.

Nel nostro caso imposteremo delle variabili che andranno a leggere delle classi css distinte a seconda del caso.

<?php

```
if($this->countModules('left and right') == 0) $contentwidth = "intero";
```

```
if($this->countModules('left or right') == 1) $contentwidth = "mezzo";
```

```
if($this->countModules('left and right') == 2) $contentwidth = "tutte";
```

```
?>
```

Basandoci sull'esistenza dei moduli diciamo che

Se left e right è uguale a zero leggi la classe contenutiintero

Se è presente uno solo dei moduli leggi la classe contenutimezzo

Se i moduli left e right sono presenti entrambi leggi la classe contenutitutte

Apriamo dunque il file index.php del nostro template test e inseriamolo appena prima del tag </head> di chiusura:

il listato sarà dunque il seguente:

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language; ?>"
lang="<?php echo $this->language; ?>" >

<head>

<jdoc:include type="head" />

<link rel="stylesheet" href="templates/system/css/general.css" type="text/css" />

<link rel="stylesheet" href="templates/<?php echo $this->template ?>/css/template.css"
type="text/css" />

<link rel="shortcut icon" href="templates/<?php echo $this->template ?>/favicon.ico" />

<?php
if($this->countModules('left and right') == 0) $contentwidth = "intero";
if($this->countModules('left or right') == 1) $contentwidth = "mezzo";
if($this->countModules('left and right') == 2) $contentwidth = "tutte";
?>

</head>

<body>

<!-- inizio contenitore -->

<div id="wrapper">

    <!-- Header -->

    <div id="header">
```

```
<?php echo $mainframe->getCfg('sitename');?>
```

```
</div>
```

```
<!-- fine Header -->
```

```
<!-- colonna sinistra --> <?php if($this->countModules('left')) : ?>
```

```
<div id="colonnasinistra">
```

```
<jdoc:include type="modules" name="left" style="" />
```

```
</div><?php endif; ?>
```

```
<!-- fine colonna sinistra -->
```

```
<!-- contenuti -->
```

```
<div id="contenuti"<?php echo $contentwidth; ?>">
```

```
<jdoc:include type="component" />
```

```
</div>
```

```
<!-- fine contenuti -->
```

```
<!-- colonna destra --><?php if($this->countModules('right')) : ?>
```

```
<div id="colonnadestra">
```

```
<jdoc:include type="modules" name="right" style="" />
```

```
</div><?php endif; ?>
```

```
<!-- fine colonna destra -->
```

```
<!-- Footer -->
```

```
<div id="footer">
```

```
<jdoc:include type="modules" name="bottom" style="" />
```

```
</div>
```

```
<!-- fine Footer -->
```

```
</div>
```

```
<!-- fine contenitore -->
```

```
</body>
```

```
</html>
```

Salviamo il file index e ora andiamo ad impostare le 3 diverse classi css:

```
#contenutiintero {
```

```
float: left;
```

```
color: #333;
```

```
border: 1px solid #892105;
```

```
margin: 0px 5px 5px 5px;
```

```
padding: 10px;
```

```
width:920px;
```

```
}
```

```
#contenutimezzo {  
  
float: left;  
  
color: #333;  
  
border: 1px solid #892105;  
  
margin: 0px 5px 5px 5px;  
  
padding: 10px;  
  
width:715px;  
  
}
```

```
#contenutitutte{  
  
float: left;  
  
color: #333;  
  
border: 1px solid #892105;  
  
margin: 0px 5px 5px 5px;  
  
padding: 10px;  
  
width:485px;  
  
}
```

È sufficiente creare un clone della classe esistente aggiungendo i vari suffissi ed impostando i valori in larghezza evidenziati in rosso, in modo che le varie zone occupino lo spazio relativo.

Salviamo e testiamo il nostro template per vedere il risultato nel browser pubblicando di volta in volta i moduli nelle colonne.

Colonna sinistra + contenuti



Colonna destra + contenuti



Solo contenuti



L'esempio che abbiamo realizzato non è vincolato solo alle colonne centrali del nostro template, la stessa implementazione è possibile per tutte le zone del sito che andremo a creare basta solo ripetere il concetto e utilizzare i parametri e gli operatori corretti a seconda del caso.

Gli operatori php disponibili

La lista degli operatori logici, aritmetici e di comparazione che è possibile utilizzare è la seguente:

reperibile dal sito http://docs.joomla.org/Operators_for_use_with_the_countModules_function

Operator	Example	Description
+	user1 + user2	Total number of Modules in user1 and user2 positions.
-	user1 - user2	The number of Modules in the user1 position minus the number in the user2 position.
*	user1 * user2	The number of Modules in the user1 position multiplied by the number in the user2 position.
/	user1 / user2	The number of Modules in the user1 position divided by the number in the user2 position.
==	user1 == user2	Returns true if user1 and user2 have the same number of Modules enabled; otherwise returns false.
!=	user1 != user2	Returns true if user1 and user2 do not have the same number of Modules enabled; otherwise returns false.
<>	user1 <> user2	Same as !=.
<	user1 < user2	Returns true if user1 has strictly less Modules enabled than user2; otherwise returns false.
>	user1 > user2	Returns true if user2 has strictly more Modules enabled than user1; otherwise returns false.
<=	user1 <= user2	Returns true if user1 has the same or less Modules enabled than user2; otherwise returns false.
>=	user1 >= user2	Returns true if user2 has the same or more Modules enabled than user1; otherwise returns false.
and	user1 and user2	Returns true if user1 and user2 both have at least 1 enabled Module; otherwise returns false.
or	user1 or user2	Returns true if user1 or user2 or both have at least 1 enabled Module; otherwise returns false.
xor	user1 xor user2	Returns true if user1 or user2 but not both have at least 1 enabled Module; otherwise returns false.

Un altro esempio che ritroviamo molto spesso utilizzato riguarda gruppi di moduli, di solito da 1 a 5 e che sfruttano appunto questa proprietà di adattamento:

```
<?php
```

```
if ($this->countModules('user1 and user2') == 2) $divwidth = "49%";
```

```
if ($this->countModules('user1 and user2') == 1) $divwidth = "98%";
```

```
?>
```

Questa volta utilizzando valori percentuali che variano la larghezza dei moduli pubblicati

Il singolo modulo pubblicato prenderà il 98% dello spazio, mentre se sono presenti entrambi occuperanno il 49% dello spazio totale. La dimensione può anche essere impostata direttamente in pixel.

Copyright © <2008>, <[joomlapixel](http://www.joomlapixel.eu)> rilasciato sotto licenza Creative commons 3.0 non commercial

This help file has been generated by the freeware version of [HelpNDoc](http://www.helpndoc.com)

I parametri dei template

I parametri del template

[Approfondimenti, tutorial e sviluppi disponibili su www.joomlapixel.eu](http://www.joomlapixel.eu)

Una delle novità più interessanti che i developer hanno implementato nella nuova 1,5 è la possibilità di configurare dei parametri gestibili dal pannello amministrativo.

Queste impostazioni possono davvero coprire tutti gli aspetti a livello di struttura, impaginazione, css tipografici e non.

Esistono numerosi template nativi per Joomla! 1,5 che fanno abbondante uso di questa nuova opzione che aumenta considerevolmente le possibilità di personalizzazione dei template.

Il tutto viene gestito dal backend operando delle scelte di “pubblicazione” o di “switch” (lettura ad esempio di altre classi o fogli di stile diversi) tra le varie possibilità che è possibile configurare.

Un esempio pratico:

```
<?php if($this->params->get('showComponent')) : ?>
```

```
<jdoc:include type="component" />
```

```
<?php endif; ?>
```

Analizzando questa semplice espressione notiamo che anche in questo caso si richiama la variabile legata a `jdocumenthtml` (operazione di verifica di una condizione illustrata all’inizio della guida) con riferimento al parametro `showcomponent` e attraverso un semplice switch tra 0 e 1 si decide se mostrare o meno l’output dei componenti.

Ma in pratica come è possibile inserire i parametri e renderli attivabili dal backend?

I file principali chiamati in causa nell’impostazione dei parametri sono al minimo 3

templateDetails.xml	<pre><params> <param name="showComponent" type="radio" default="1" label="Show Component" description="mostra/nascondi i componenti"> <option value="0">No</option> <option value="1">Yes</option> </param> </params></pre>
Index.php	<pre><?php if(\$this->params->get('showComponent')) : ?> <jdoc:include type="component" /> <?php endif; ?></pre>
params.ini	File aggiuntivo necessario per il salvataggio della configurazione scelta nel backend (deve necessariamente essere scrivibile!)

Si ribadisce l’importanza del file xml, uno dei cambiamenti principali della nuova versione riguarda proprio questo file.

Oltre alle posizioni modulo dovremo qui dichiarare per il nostro template anche i parametri

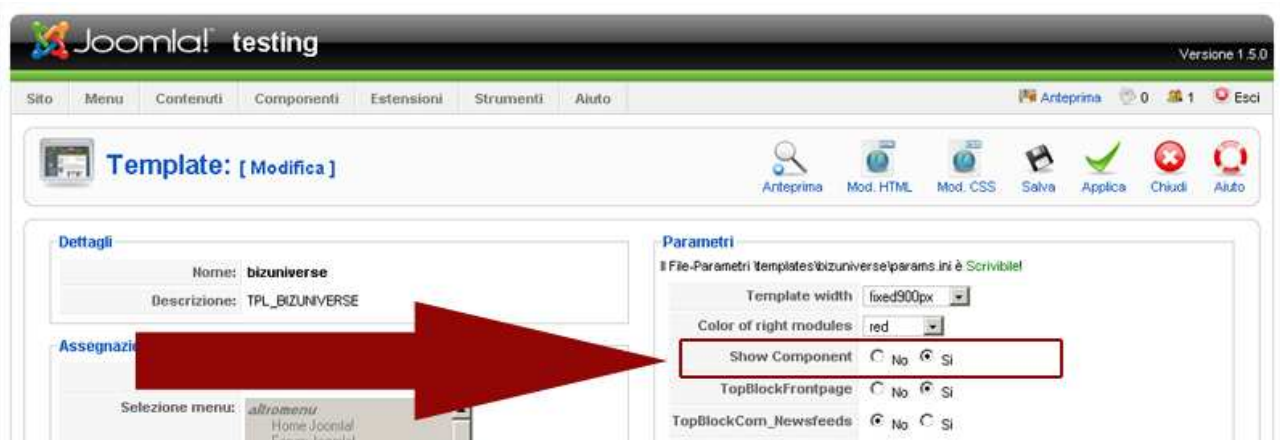
Il contenitore principale è `<params> </params>` al cui interno troviamo il nome del parametro (param name) il tipo (radio – lista etc) un valore di default, l’etichetta e la descrizione.

a seguire i tag di apertura e chiusura delle opzioni

<option value="0">No</option>

<option value="1">Yes</option>

E come è visibile nel pannello di amministrazione



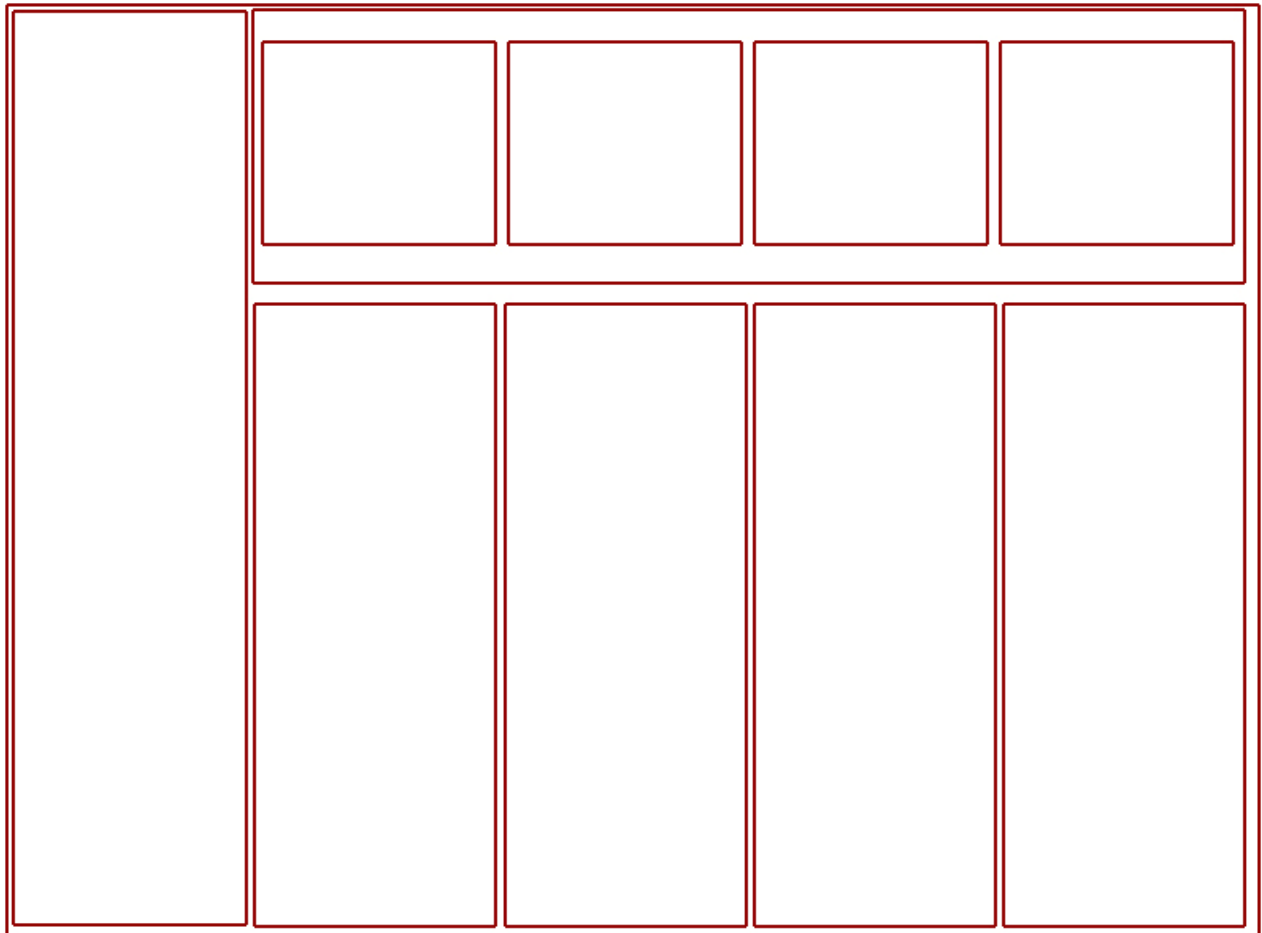
l'amministratore utilizzando questa opzione può dunque decidere o meno se pubblicare i componenti all'interno del suo sito.

Come esempio sembra semplicistico, in realtà fornisce ottimi spunti di riflessione e di studio:

con i parametri possiamo infatti arrivare a gestire un po' tutti gli aspetti del template ed anche dei contenuti!

- Layout - è possibile settare parametri relativi alla visualizzazione o meno di div contenitori (ad es. colonne o box)
- Layout – è possibile settare la lettura di classi preimpostate per regolare ad esempio la larghezza del template
- Layout – è possibile settare la lettura di css per cambiare i colori di base/immagini del template
- Css tipografici – è possibile settare la lettura di classi css preimpostate per modificare al volo le classi del core del CMS

Ci fermiamo a questo punto (peraltro sostanzioso) lasciandovi libertà di pensiero, di inventiva e di innovazione perché di sicuro questa nuova opzione offre spunti molto interessanti, uno tra tutti e il primo che mi viene in mente è relativo alla realizzazione di una struttura template completa e che offra il maggior numero possibile di opzioni di impaginazione regolate per l'appunto semplicemente dai parametri.



Una struttura di questo tipo necessita ovviamente di una buona pianificazione, è necessario considerare sia le "istruzioni condizionali" che abbiamo visto in precedenza per gestire gli spazi occupati sia dai moduli che dalle colonne e sia le relative classi css da abbinare per l'impostazione delle stesse.

Un lavoraccio..... ma con Joomla! (e un pò di pazienza e impegno) si può fare!

Override di componenti e moduli

Template override

Un'altra significativa innovazione in Joomla! 1.5 è la possibilità di “sovrascrivere” le impostazioni di visualizzazione per tutti i componenti e i moduli, sia a livello di struttura che a livello tipografico.

Questo garantisce agli sviluppatori di template un controllo e una definizione maggiore (nelle versioni 1.0.x si è costretti al lavorare direttamente sui file del core).

Osservando la struttura ad esempio del componente “content” notiamo che le strutture di visualizzazione sono raccolte sotto questo percorso:

`components\com_content\views\frontpage\tmpl`

e precisamente ritroviamo:

- `default.php`
- `default_item.php`
- `default_links.php`

relative alla struttura dell'intera pagina, del singolo elemento e della paginazione.

Lo stesso discorso è valido per i moduli

`\modules\mod_footer\tmpl\default.php`

Questa separazione strutturale permette all'atto pratico la sostituzione dei singoli elementi senza dover necessariamente agire sui file del core di Joomla!, basterà inserire in una nuova cartella “html” all'interno della cartella del template (rispettando i percorsi relativi) i file che intendiamo sostituire per far sì che il sistema li riconosca in automatico ed utilizzi questi ultimi per mostrare a video l'output relativo.

Prendiamo ad esempio il listato iniziale del componente content presente nel core del CMS

```
<?php // no direct access
```

```
defined('_JEXEC') or die('Restricted access'); ?>
```

```
<?php if ($this->params->get('show_page_title')) : ?>
```

```
<div class="componentheading"<?php echo $this->params->get('pageclass_sfx') ?>">
```

```
<?php echo $this->escape($this->params->get('page_title')); ?>
```

```
</div>

<?php endif; ?>

<table class="blog<?php echo $this->params->get('pageclass_sfx') ?>" cellpadding="0"
cellspacing="0">

<?php if ($this->params->def('num_leading_articles', 1)) : ?>

<tr>

    <td valign="top">

        <?php for ($i = $this->pagination->limitstart; $i < ($this->pagination->limitstart + $this-
>params->get('num_leading_articles')); $i++) : ?>

            <?php if ($i >= $this->total) : break; endif; ?>

            <div>

                <?php

                    $this->item =& $this->getItem($i, $this->params);

                    echo $this->loadTemplate('item');

                ?>

            </div>

        <?php endforeach; ?>

    </td>

</tr>

<?php else : $i = $this->pagination->limitstart; endif; ?>

<?php
```

Abbiamo evidenziato in rosso la classe css attraverso cui impostare i parametri relativi al titolo della notizia, ora per far capire le possibilità di personalizzazione prendiamo come riferimento l'override impostato nel template “beez” e relativo sempre a questo componente:

```
<?php // @version $Id: default.php 9722 2007-12-21 16:55:15Z mtk $  
defined('_JEXEC') or die('Restricted access');  
?>
```

```
<?php if ($this->params->get('show_page_title')) : ?>  
  
<h1 class="componentheading<?php echo $this->params->get('pageclass_sfx'); ?>">  
  
    <?php echo $this->escape($this->params->get('page_title')); ?>  
  
</h1>  
  
<?php endif; ?>
```

```
<div class="blog<?php echo $this->params->get('pageclass_sfx'); ?>">  
  
    <?php $i = $this->pagination->limitstart;  
  
    $rowcount = $this->params->def('num_leading_articles', 1);  
  
    for ($y = 0; $y < $rowcount && $i < $this->total; $y++, $i++) : ?>  
  
        <div class="leading<?php echo $this->params->get('pageclass_sfx'); ?>">  
  
            <?php $this->item =& $this->getItem($i, $this->params);  
  
            echo $this->loadTemplate('item'); ?>  
  
        </div>  
  
        <span class="leading_separator<?php echo $this->params->get('pageclass_sfx');  
?>">&nbsp;</span>  
  
    <?php endfor; ?>
```

Evidenziate in rosso le modifiche effettuate. La classe “componentheading” è stata impostata come h1, nel layout generale sono state eliminate tutte le tabelle e sostituite dal div contenitore con classe “leading”.

Il risultato è perfettamente visibile utilizzando firefox e le sue estensioni e confrontando l'output dei due template di default.

Beez

Benvenuto in Joomla!



VERSIONE 1.5!



Ultimo aggiornamento (giovedì 08 novembre 2007 12:06)
Scritto da Administrator
giovedì 12 ottobre 2006 10:00

Joomla! è un framework e un sistema di gestione e pubblicazione contenuti libero e open source strutturato per la realizzazione rapida di siti web altamente interattivi e multi lingua, community online, portali, blog e applicazioni eCommerce.

Powered by
 **Joomla!**
version 1.5

fornisce una gradevole ed efficace interfaccia grafica di facile utilizzo che semplifica la gestione e la pubblicazione di un gran volume di contenuti, compresi HTML, documenti, e media. Joomla! viene utilizzato da organizzazioni di varia grandezza per Siti Pubblici, Intranet ed Extranet ed è supportato da una comunità di utenti.

→ Leggi tutto: Versione 1.5!

div, class: leading

Notizie

Questo s
con la ve
Joomla!
Joomla.it
desideri
malfunzi
consiglia
scrivere

rhuk_milkyway



La personalizzazione come detto in precedenza può essere anche strutturale al punto di fornire implementazioni utili, partendo appunto da questo override del com_content del template Beez si può riuscire ad esempio ad impostare classi css alternate per le intro degli articoli pubblicati, creando cioè un'opzione che renda possibile specificare background, immagini, font diversi ad ognuno di essi.

Modifica del file default.php

```
.leading0 {
background: #edf5de;
border: 1px solid #dcf6ac;
}
.leading1 {
background: #c7ced4;
border: 1px solid #5f6f7c;
}
```

L'intento è quello di riuscire ad ottenere layout diversi per le intro delle news pubblicate sul sito, il tutto in maniera automatica e gestibile dal foglio di stile. Ci limitiamo ad impostare 3 stili, dei quali uno predefinito, ovviamente l'opzione può essere estesa con facilità. Abbiamo già visto nelle pagine immediatamente precedenti l'override del Com_content nel template "Beez", riprendiamo dunque il file interessato, default.php:

Riga inizio modifica: 18

Inseriamo la funzione array

- che cosa è un array: Gli array non sono altro che contenitori dove poter stipare dati che possiamo in seguito richiamare ed utilizzare.

Immaginiamo di avere a disposizione innumerevoli variabili che descrivono un comune oggetto tutte numerate partendo dallo zero.

Il codice array inserito è questo **<? \$css = array('0','1','2');?>**

Alla riga 21 **<div class="leading<?php echo \$this->params->get('pageclass_sfx'); ?>**

abbiamo modificato, inserendo una istruzione condizionale ed la variabile di recupero del numero dello stile, ricavandolo dall' array impostato precedentemente.

Codice della riga 21, modificata con la istruzione **IF**

la l'istruzione If. quando vogliamo eseguire un blocco di codice se una condizione è vera.

Recupero del numero della intro, che effettua il conteggio della notizia introduttiva : **\$y**
Per recuperare il numero dello stile, abbiamo inserito la variabile **\$css[\$y]**;

che in automatico, preleva il numero della intro, esempio

```
$css[0];  
$css[1];  
$css[2];
```

```
<div class="leading<?php echo $this->params->get('pageclass_sfx'); ?><?php if ( $y != 3){
    echo $css[$y];
}
else
{
    echo $css[0];
}

?>">
```

In pratica, se il numero dello stile non è superiore al numero 2 , prende il numero dell'array, per creare il nuovo stile, se è superiore al 2, la funzione, preleva come predefinito l'array 0, \$css[0]; Nel foglio di stile:

```
.leading0 {
background: #edf5de;
border: 1px solid #dcf6ac;
}
.leading1 {
background: #c7ced4;
border: 1px solid #5f6f7c;
}
```

e questo il risultato ottenuto con questo semplice hack

The screenshot shows the Joomla! 1.5! homepage layout. On the left is a sidebar with a purple header and menu. The main content area has a light green header. Three red arrows point to specific sections, labeled 'css 2', 'css 1', and 'css 0' from top to bottom.

- css 2** points to the 'Benvenuto in Joomla!' section, which has a light green background and a thin border.
- css 1** points to the 'Dove sono finiti i mambot?' section, which has a light blue background and a thin border.
- css 0** points to the 'Versione 1.5!' section, which has a light green background and a thin border.

The sidebar contains the following sections:

- Ultime notizie**: Aspetto dei contenuti, La Community Joomla!, Versione 1.5!, Notizie flash, grafica in pochi clic.
- Menu principale**: Home, Descrizione di Joomla!, Licenza Joomla!, Informazioni Joomla!, FAQ, Notizie, Web link, News feed, Download.
- Concetti chiave**: Estensioni, Aspetto dei contenuti, Pagine di esempio.
- Pagine di esempio**

The main content area also includes a 'Powered by Joomla! version 1.5' logo and a link to 'Leggi tutto: Versione 1.5!'.

L'esempio è banale,ma serve solo come riferimento,ci siamo limitati ad impostare dei background diversi per 3 news della intro,ma effettivamente le possibilità di intervenire per modificare il layout delle intro degli articoli sono parecchie.

Ringraziamenti

[Joomla.org](#)

e tutto il gruppo dei developers per essere riusciti a regalarci questo meraviglioso strumento

[Joomlaitalia.com](#)

il "padre" del verbo (in senso metaforico ovviamente ma la sua e quella di alexred sono le identità che riconosco da tempi immemorabili)

alexred.ste,luscarpa,vamba,maorinz,matmat,bigham,giorgioblu,surfbit,sara82, andrea s.,giak,bettinz,filpi5481,lucacuratole,jk4nik,zalexo,taolo,vinset,simoegio, Noris, elpaso66,wosky,carloernesto,wolfabrizio,ferraresi_4ever

della comunità dei joomlers italiani presente e attiva su [joomla.it](#)

un ringraziamento particolare all'amico [lucazone](#)

e al servizio hosting dedicato a joomla [joomlaspace](#)