

---

# Il tuo design HTML in Contao

---

Versione 1.0 – revisione del 8 agosto 2013

Guida per convertire un qualsiasi design statico html in un tema per Contao

Questo documento è l'aggiornamento di uno precedente che avevo scritto per Typolight circa 4 anni fa. Contao oggi è migliorato sotto vari aspetti e la guida precedente non è più valida. Ho deciso di riscriverla.

## in questa guida...

Il tuo design HTML in Contao.....	1
Premesse sul funzionamento.....	2
Setup del tema in 6 fasi.....	3
Fase 1: Creare i template html in Contao.....	3
Fase 2: Creare un tema .....	5
Fase 3: Importare e sistemare gli stili CSS.....	6
Fase 4: Creare i moduli che formano la pagina .....	9
Fase 5: Assemblare i moduli in un Layout.....	10
Fase 6: Assegnare il Layout alla root, ed alle pagine del sito .....	14

## Premesse sul funzionamento

Prima di tutto iniziamo dicendo che Contao utilizza un **sistema di templates** interno al suo core (/system/modules/core/templates), ogni template si occupa di una particolare parte (es: la pagina generale, il menu, il breadcrumb, la lista short news, la galleria foto... ecc).

Ogni template ha estensione **".html5"** oppure **".xhtml"** per la compatibilità con moduli vecchi, io considero solo la prima versione in quanto ormai faccio solo siti HTML5.

Normalmente il nome del file ha un **prefisso significativo**, che viene riconosciuto da Contao... ad esempio **"fe\_page.html5"** è il template generale di front end, mentre **"nav\_default.html5"** è il template della navigazione.

Quindi, quando creiamo i nostri template personalizzati dobbiamo tenere questo formato, per esempio chiamandoli **"fe\_miapagina.html5"**, **"fe\_miahomepage.html5"**, **"nav\_miomenu.html5"** ecc...

Questi sono tutti file che **generano html come output**, e contengono tags con determinati id e class, e ovviamente parti di php.

Questi due appena menzionati (fe\_ e nav\_) sono templates che in fase di produzione del sito incontriamo immediatamente, e li possiamo personalizzare via pannello amministrazione senza usare alcun client FTP o editor HTML. Contao usa un **sistema override**, cioè fa una copia del file originale presente in core e la mette nella **cartella /templates/**, e inizia ad usarlo con precedenza sul default.

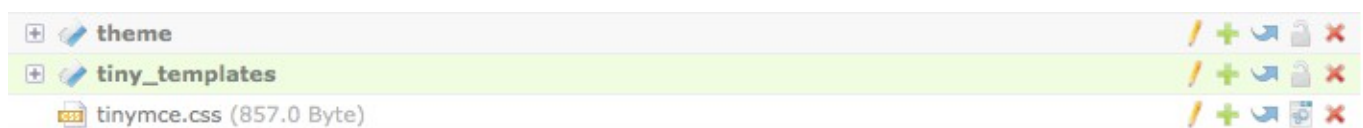
Possiamo lasciare il **nome predefinito** del file, e in questo caso il sistema prenderà per buono il template che abbiamo appena generato, oppure possiamo **rinominare il file** e in questo caso sarà necessario sceglierlo quando ne abbiamo la possibilità, tramite i menu di selezione presenti in moduli, elementi della pagina, o nel layout. Per esempio se lascio **"gallery\_default.html5"** significa che in automatico tutte le gallerie fotografiche del sito prenderanno questo file come template, mentre se lo rinomino in **"gallery\_slideshow.html5"** e poi ne creo un altro con nome **"gallery\_thumbnails.html5"** significa che sto preparandomi due template con codice diverso per specifiche mie esigenze. Poi, quando creo un elemento di tipo gallery, andrò a scegliere il template adeguato tra le tre scelte che mi verranno proposte: **"default"** dal core, **"slideshow"** e **"thumbnails"** dal tema. Non tutti i template sono selezionabili da qualche parte, alcuni hanno solo la versione predefinita, con il tempo si impara a distinguerli, di solito sono template che non necessitano di personalizzazioni.

Nelle versioni più recenti di Contao, è stata introdotta la possibilità di fare vari **temi diversi** sotto la stessa cartella templates (ad esempio **"templates/temaA"**, **"templates/temaB"** ecc...), basta usare il tasto **"Nuova Cartella"**. Mentre per copiare un template dal core bisogna premere il tasto **"Nuovo template"**.

 Nuova cartella  Nuovo template

Altra cosa da sapere su Contao: all'interno del CMS è implementato un vero e proprio **editor di CSS**, il file css statico del nostro sito possiamo importarlo, Contao fa un parsing di tutte le regole che trova e le salva in database, questa funzionalità è molto interessante in quanto possiamo successivamente modificare gli stili del sito con gli ottimi strumenti a disposizione, e non serve più toccare alcun client FTP o editor HTML.

Infine le **immagini ed i file del layout vengono caricati e gestiti nella cartella /files/** (in passato era /tl\_files/). Usando il normale file manager quindi possiamo creare la cartella del tema (ad esempio **"theme"**).



## Setup del tema in 6 fasi

Con le premesse descritte, ora iniziamo con il tutorial vero e proprio, in sole 6 fasi, partiamo da una situazione come la seguente.

1. Contao è installato correttamente e nella nostra lingua, presso un web server.
2. Una struttura del sito è stata abbozzata (**Layout** → **Struttura sito**), ed abbiamo inserito anche qualche contenuto (**Contenuti** → **Articoli**). Questo serve per dare forma al sito una volta impostato il template.
3. Abbiamo la nostra pagina **html originale**, il **css**, ed eventuali **immagini** del layout e **javascript**.

### Fase 1: Creare i template html in Contao

Entrare in amministrazione ed accedere al pannello Layout → Templates.

Premere  Nuova cartella

Quindi scrivere un nome per il tema, esempio "theme". Salvare.

Poi premere  Nuovo template



Seguendo le istruzioni, creare un template copiandolo da quello originale, lo mettiamo dentro la cartella di destinazione che nel nostro caso sarà "theme".






















**Template originali**

Scegliere il file template che desidera adattare. Una copia locale sarà generata nella cartella *templates*.

**Cartella di destinazione**

Una copia del modello selezionato verrà memorizzato nella cartella di destinazione.

Possiamo ripetere questa operazione per una serie infinita di nostri template, premendo sul simbolo  rinominiamo il file, mentre sul simbolo  possiamo modificare il codice sorgente, html e php.

File system	
<b>theme</b>	  
com_blog.html5 (739.0 Byte)	  
fe_default.html5 (3.1 KiB)	  
gallery_bootstrap.html5 (1.1 KiB)	  
gallery_bootstrap_titles.html5 (1.2 KiB)	  
gallery_default.html5 (859.0 Byte)	  
mod_comment_form.html5 (1.7 KiB)	  

Ora consideriamo il nostro "fe\_page.html5", che io ho rinominato "fe\_default.html5". E' il file dell'architettura base della pagina, premendo sull'icona di modifica del codice si apre un'area editabile come la seguente:

Editor sorgente

```
1 <!DOCTYPE html>
2 <html lang="<?php echo $this->language; ?>">
3 <head>
4 <meta charset="<?php echo $this->charset; ?>">
5 <title><?php echo $this->pageTitle; ?> - <?php echo $this->mainTitle; ?></title>
6 <base href="<?php echo $this->base; ?>">
7 <meta name="robots" content="<?php echo $this->robots; ?>">
8 <meta name="description" content="<?php echo $this->description; ?>">
9 <meta name="keywords" content="<?php echo $this->keywords; ?>">
10 <meta name="generator" content="Contao Open Source CMS">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <?php echo $this->framework; ?>
13 <?php echo $this->stylesheets; ?>
14 <?php echo $this->mooScripts; ?>
15 <?php echo $this->head; ?>
16 <!--[if lt IE 9]><script src="<?php echo TL_ASSETS_URL; ?>assets/html5shiv/<?php echo HTML5
17 <!--[if lte IE 9]>
18 <script src="<?php echo TL_ASSETS_URL; ?>assets/zod_js/placeholder-fix.js"></script>
19 <![endif]-->
20 </head>
21 <body id="top" class="{ua::class}<?php if ($this->class): ?> <?php echo $this->class; ?>
22
23 <div id="wrapper">
24
25 <?php if ($this->header): ?>
26 <header id="header">
27 <div class="row fluid">
```

Modifica il contenuto del file

Salva Salva e chiudi

Notiamo che in questo file ci sono i tag di una tipica pagina **HTML5**, con dentro alcune parti di php. In realtà il codice che vedete in figura qui sopra è già stato ritoccato da me, ho messo un **Viewport** come serviva a me, ed ho aggiunto un commento condizionale che carica un javascript per correggere i placeholder in Explorer inferiore a 9. In pratica potete **ritoccare il codice** affinché si strutturi con i pezzi del design che dovete ottenere: attenzione alle **parti php**, dovete mantenerle nei punti che vi servono. Il codice javascript l'ho messo qui ma potevo farlo anche nel layout, negli elementi della pagina, oppure in dei moduli. Di concetto io lo metto in questa sorgente quando sono parti onnipresenti nel sito, per esempio lo script di google analytics alla fine della pagina, prima della chiusura del `</body>` (e faccio anche altre piccole cose, ma le tralascio per rimanere semplice in questa guida).

In particolare vorrei sottolineare alcune caratteristiche importanti: il tag `<base href="<?php echo $this->base; ?>" />`. Contao lo usa per fare i propri percorsi di pagine e file, con le funzioni rewrite che aggiungono la lingua e/o alterano il path, è fondamentale che sia presente per non sbagliare tutti i percorsi.

Per quanto riguarda i css del nostro tema, non dobbiamo inserirli a template, abbiamo detto che li importiamo, il tag che si occuperà di caricarli come specificato nel layout è `<?php echo $this->head; ?>`. Nel pannello del layout c'è anche una textarea per inserire codice aggiuntivo nella porzione `<head>` (quindi potete scegliere come è meglio procedere, Contao è sempre molto flessibile).

Nella parte <body> abbiamo 5 sezioni utilizzabili: **header, left, main, right, footer**. Rimuovendo del tutto l'html aggiuntivo sarebbero in una forma come la seguente.

```
<?php if ($this->header): ?>
<?php echo $this->header; ?>
<?php endif; ?>
```

In queste 5 sezioni vengono caricati i contenuti della pagina, come sono stabiliti dal layout.

Le sezioni personalizzate invece io le elimino senza paura:

```
<?php echo $this->getCustomSections('before'); ?>
<?php echo $this->getCustomSections('main'); ?>
<?php echo $this->getCustomSections('after'); ?>
```

Preferisco inserirle alla maniera delle altre, cioè una per una e con il controllo "if", il codice è come il seguente (non sono mai tante di più rispetto al default di Contao):

```
<?php if ($this->sections['mia-sezione-inventata']): ?>
<?php echo $this->sections['mia-sezione-inventata']; ?>
<?php endif; ?>
```

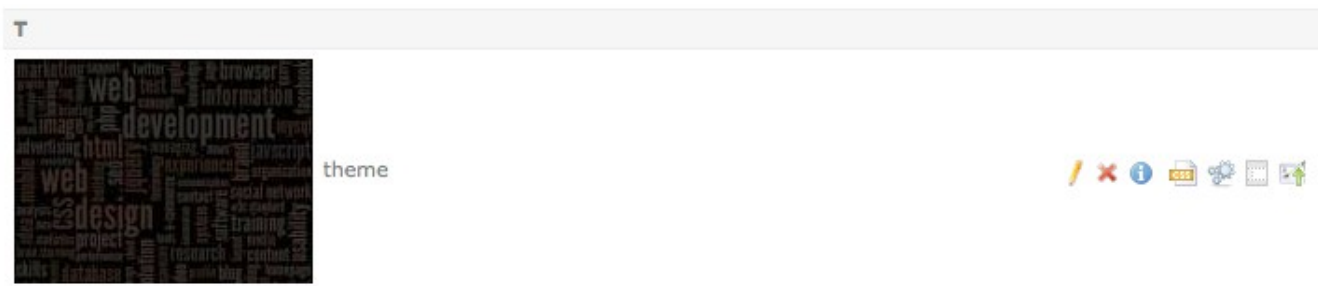
Alla fine della pagina, ci sono altri script, lasciate come vuole Contao, al limite aggiungete script vostri prima della chiusura del </body> come ho accennato prima.


Io consiglio di osservare e mantenere tutto come dal file **fe\_page**, ripulirlo, riordinarlo, e chiedersi sempre se è meglio inserire il codice qui oppure altrove (nel layout, nei moduli o nel contenuto).

## Fase 2: Creare un tema

Accedere al pannello Layout → Temi, questa volta è possibile creare nuovi temi per il sito. Si possono importare ma nel mio caso ne creo uno mio, vuoto, quindi un "Nuovo tema" e lo chiamo theme.

 Nuovo tema  Importa tema  Theme store  Modifica tutti i records



 Modifica i parametri del tema (la figura sotto), quindi posso definire la cartella dei file, quella dei templates, le variabili che si posso usare poi nei css, e la minitura che vedete qui sopra (facoltativa).

 Per gestire i CSS,  per gestire i Moduli,  per gestire i Layout

- ▾ Titolo e autore

**Nome del tema**

theme

Inserisci un nome univoco da assegnare al tema

**Autore**

zod

Inserisci il nome dell'autore del tema

- ▾ Configurazione

**Cartelle**

files/theme

Modifica selezione

Seleziona le cartelle che appartengono al tema

**Screenshot**

files/theme/webwords.jpg

Modifica selezione

Seleziona uno screenshot per il tema

**Cartella template**

theme

Seleziona la cartella del template che verrà esportata con il tema

- ▾ Variabili globali

**Variabili globali**

Chiave	Valore	
\$txt	eeeeee	+ ↑ ↓ ×
\$href	aaaaaa	+ ↑ ↓ ×
\$red	cc0000	+ ↑ ↓ ×

E' possibile definire delle variabili globali che potranno essere utilizzate nei CSS (e.g. \$rosso -> c00 or \$margine -> 12px)

Nota: se non abbiamo creato una cartella del tema, e una del template, non sarà possibile selezionarle.

## Fase 3: Importare e sistemare gli stili CSS

Importare i file è facile, basta andare in **Temi** → **CSS**

In alto abbiamo due bottoni:  **Nuovo foglio di stile**  **Importa CSS**

Con il primo creiamo un foglio stile nuovo, vuoto, e lo prepariamo quindi interamente in Contao.

Altrimenti abbiamo  **Importa CSS** per caricare un file nostro già preparato.

Gli stili possiamo configurarli con i **commenti condizionali**, le **media query**, e possiamo anche impostare delle **variabili** che poi potremmo usare in giro per le proprietà delle regole. Nella figura seguente ci sono tre CSS, quello principale è **screen**, mentre **print** è quello usato per ottimizzare la stampa, e **ie8** per correggere IE.

<b>I</b>	ie8 @media all <!--[if lt IE 9]>	      
<b>P</b>	print @media print	      
<b>S</b>	screen @media all	      

Questa è per esempio la configurazione  dello stile per IE 8 e inferiori.

#### Nome

Inserisci un nome univoco per il foglio di stile.

#### - ▾ Configurazione

##### Incorporare immagini fino a

Specificare la dimensione file in byte da includere direttamente

##### Commento condizionale

I commenti condizionali consentono di creare fogli di stile

#### - ▾ Impostazioni media

##### Media

- Seleziona tutto*
- all
- aural
- braille
- embossed
- handheld
- print
- projection
- screen
- tty
- tv

Scegli il media a cui sarà applicato il foglio di stile (e.g. print/screen/all)


##### Media query



Definire un media usando una media query come *screen and (min-width: 800px)*. Il media definito sopra verrà sovrascritto.

A questo punto abbiamo importato gli stili, ma dobbiamo modificarli per correggere i path delle immagini usate nel design statico con le immagini caricate ora in Contao tramite il file manager. E probabilmente dovremo ritoccare molti altri stili, o aggiungerne di nuovi per adeguare gli elementi di Contao alle nostre esigenze.

Premendo l'icona  in corrispondenza dello stile css, apparirà la lista di regole, come nell'esempio seguente.

```
html {  
  height:100%;  
}  
  
body {  
  margin:0;  
  padding:0;  
  background:#2d2d2d url("files/theme/hexa.jpg") center top repeat-x;  
  font-family:'Spisamira',Arial,Helvetica,sans-serif;  
  line-height:1.5;  
  color:$txt;  
  background-attachment:fixed;  
}  
  
a {  
  color:$aaa;  
}
```

Ogni elemento è separato e modificabile, può essere spostato con il drag & drop, oppure con l'icona 

Si possono anche aggiungere nuovi elementi premendo  **Nuova regola css** oppure  a fianco di un'altra (l'elemento nuovo si posiziona subito sotto).

E' possibile anche accendere e spegnere in velocità certe regole premendo 

Infine si possono modificare un insieme di regole premendo  **Modifica tutti i records**

Le regole sono tutte editabili in modo guidato con dei campi specifici, ed hanno alla fine un ulteriore campo libero dove poter scrivere a mano qualcosa di personalizzato.

**Dimensione**

**Dimensione**  
Modifica la dimensione (width, height, min-width, min-height, max-width e max-height).

**Posizione**

**Posizione**  
Attributi position, float, clear, overflow e display

**Margini ed allineamento**

**Margin, padding e allineamento**  
Modifica il margine, il padding e l'allineamento (margin, padding, align, text-align, vertical-align).

**Margine**

0 0 0 0 -

Inserisci un valore per margini: top, right, bottom, left e

**Padding**

0 0 0 0 -

Inserisci un valore per il padding: top, right, bottom, left e

**Allineamento dell'elemento**

-

Per allineare un elemento i suoi margini left e right saranno

**Allineamento verticale**

-

Definisce l'allineamento verticale.

**Allineamento del testo**

-

Definisce l'allineamento orizzontale del testo.



## Fase 4: Creare i moduli che formano la pagina

In fase 1 avevamo visto nel template le 5 posizioni: **header, left, main, right, footer** (ed eventuali custom). In ognuna di esse ora dobbiamo inserire i moduli della pagina.

Un **modulo** è un **pezzo di pagina**, esempio: un menu, il breadcrumb, una lista di news, il contenuto principale di un articolo, oppure un pezzo html scritto manualmente.

Questi moduli li assembliamo nelle posizioni per formare un layout, oppure vengono inclusi come elementi negli articoli (ma quest'ultimo caso al momento lo tralascio).

Supponiamo di avere una situazione come la seguente, molto semplice:

- Header
  - **Logo del sito e pay off** (è un modulo html personalizzato)
  - **Menu top** (è un modulo di menu per la navigazione principale del sito)
- Left
  - niente (per semplicità non metto moduli)
- Main
  - **Breadcrumb** (è un tipo di navigatore per l'orientamento nel sito)
  - **Articoli** (parte centrale dove viene inserito l'articolo)
- Right
  - niente (per semplicità non metto moduli)
- Footer
  - **Footer** (pezzo html statico dove scrivo i copyright)

Bene, abbiamo 4 moduli da creare, **Articoli** è già predisposto dal CMS quindi non lo devo creare. Quindi accediamo alla gestione Temi → Moduli.

Premiamo  Aggiungi modulo

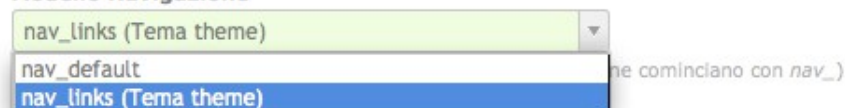
I moduli sono di vario tipo e non posso spiegarli tutti, cmq quelli più semplici sono i **navigatori** e i moduli di **html personalizzato**. Poi importantissimi sono quelli delle liste e dei dettagli delle news.

Nel sito potremmo necessitare di più di un layout: per esempio uno per la **home page** e un altro che per le **pagine interne**. Quindi eventualmente i moduli potrebbero aumentare (ad esempio "lista news in evidenza in home page" e "lista news della pagina archivio", sarebbero due moduli diversi).

Dobbiamo aggiungere quelli che ci servono, configurandoli nel modo giusto, e solo l'esperienza vi può aiutare in questo compito (buon divertimento con l'esplorazione di Contao).

Esempio di campo per selezionare il template in un modulo, questo è per il menu, come accennato in precedenza le opzioni che troviamo sono: **nav\_default** (quello di core) e **nav\_links** (quello nel tema "theme").

### Modello Navigazione



La schermata delle impostazioni del modulo cambia in base al tipo, ce ne sono diversi...

#### - ▾ Titolo e tipologia

##### Nome del modulo

Inserisci un nome univoco per il modulo.

##### Intestazione

 h1 ▾

Se inserisci un'intestazione, verrà visualizzata all'inizio del

#### - ▾ Impostazioni di navigazione

##### Livello di partenza

Definisci da quale livello le voci del menu sono visibili. Scegli 0 o

##### Imponi limite

Non mostrare nessun elemento (neanche quelli attivi) oltre il

#### - ▾ Pagina di riferimento

##### Imposta pagina di riferimento

Imposta una pagina di riferimento da utilizzare come sorgente o pagina di destinazione del modulo.

##### Pagina di riferimento

 Formule (formule.html)

##### Tipo modulo ⚠

Menu navigazione

##### Navigazione

**Menu navigazione**

Navigazione personalizzata

Navigazione breadcrumb

Navigazione rapida

Link rapido

Navigazione libro

Navigatore articoli

Mappa del sito

##### Utente

Login form

Logout automatico

Dati personali

Registrazione utenti

Quindi tornando all'esempio pratico, dovrò costruire un modulo "menu navigazione" oppure "navigazione personalizzata" per fare il mio "Menu Top". Il primo tipo genera una lista in automatico con le voci in base ai nodi figli di una pagina del sito, in un caso banale potrebbe essere la root o la home. Mentre il secondo tipo permette di selezionare le pagine che si desiderano a menu tramite una serie di checkbox.

Poi serve una "navigazione breadcrumb" per fare ovviamente il "Breadcrumb", e due moduli di "html personalizzato" per scrivere i pezzetti per "Logo e payoff", e "Footer".

## Fase 5: Assemblare i moduli in un Layout

Una volta creati tutti i moduli necessari dobbiamo a questo punto accedere a Temi → Layout. Dobbiamo creare sicuramente un **layout default da assegnare alla root del sito**. Poi possiamo crearne altri di specifici, che per esempio variano nelle colonne left e right, inserendo un menu contestuale, o inseriamo moduli differenti nelle varie posizioni.

Premere  Nuovo layout

Qui avviene la magia che raccoglie quanto fatto finora, spiego i punti più importanti di seguito, tralascio le cose di dubbia utilità (come le dimensioni delle colonne che invece setto via css e non qui).

Scrivo il nome del Layout, "default", e imposto come lo voglio configurare per colonne e righe. Attivo, se ne ho, le sezioni personalizzate.





- ▾ Titolo ed impostazioni predefinite

**Nome del layout**

Inserisci un nome per il layout.

- ▾ Impostazione colonne





**Colonne** ⚠

Seleziona il numero di colonne.

- ▾ Intestazione e piè pagina

**Righe** ⚠

Si prega di scegliere il numero di righe.

**Sezioni layout personalizzate**

Seleziona tutto

banner

Poi includo i css necessari, Contao propone già vari CSS del suo Framework, ma io faccio sempre a modo mio, tengo solo il file del TinyMCE che eventualmente configuro con stili custom. Di recente è possibile includere stili anche da file manager, io per esempio includo Bootstrap, in quanto il mio tema "Theme" è basato su di esso, potrei anche inserire i CSS delle plugin JQuery che utilizzo.

- ▾ Fogli di stile

**Framework CSS** ⚠

Seleziona tutto

 Foglio di stile TinyMCE

 Generatore layout

 Responsive grid

 Reset CSS

 Form di base

Qui è possibile attivare i componenti del framework CSS di Contao.

**Fogli di stile** /

Seleziona tutto

 screen

 print

 ie8

Seleziona i fogli di stile che vuoi includere in questo layout

**Fogli di stile aggiuntivi** ⚠

 files/theme/bootstrap/bootstrap.css

Modifica selezione

Poi abbiamo la selezione dei moduli, nel mio caso ho aggiunto **Articoli** anche in colonna sinistra e destra, ed in sezione personalizzata "Banner". Possiamo poi riordinarli, duplicarli, eliminarli.

#### Moduli inclusi

Modulo	Colonna	
n-header [Codice HTML personalizzato]	Intestazione	+ ↑ ↓ ×
Articoli [Articoli]	Colonna sinistra	+ ↑ ↓ ×
Articoli [Articoli]	Colonna destra	+ ↑ ↓ ×
Articoli [Articoli]	Spazio principale	+ ↑ ↓ ×
n-footer [Codice HTML personalizzato]	Pie' pagina	+ ↑ ↓ ×
Articoli [Articoli]	banner	+ ↑ ↓ ×

Assegna ogni modulo ad un elemento della pagina. Utilizza i pulsanti aggiungi, sposta o elimina un modulo. Se non hai Javascript attivo,

Poi ci sono le impostazioni della pagina, assegno il template che ho prodotto in fase 1 "**fe\_default.html5**", Google web fonts non lo uso, io preferisco includere le mie regole "@font-face" come spiego nel mio blog. Il tag <body> lo lascio stare, lascio vuoto anche il codice personalizzato in <head> perché se voglio mettere dei css ho le opzioni dette prima, e se ho javascript ho le opzioni che dirò in seguito.

#### Modello layout

fe\_default (Tema theme)

Seleziona un modello di layout. Il modello predefinito si chiama

#### Formato di output

HTML5

Qui è possibile selezionare il formato di output

#### Google web fonts

Qui puoi aggiungere fonts web Google al tuo sito web. Specifica

#### Body class

Puoi inserire un attributo class che verrà aggiunto al body della

#### Body onload

Nel caso in cui uno dei vostri script personalizzati richieda l'evento *body onload*, puoi inserire il codice qui.

#### Tag aggiuntivi ▶

Qui puoi aggiungere dei tag alla sezione della pagina generata

Procediamo con l'inclusione dei javascript, Contao propone **Mootools** oppure **Jquery**, io scelgo ormai sempre il secondo. Il select mi permette di scegliere se caricare la libreria dal repository di Google, o dal mio sito, oppure fare entrambe le cose con un fallback.

- ▾ jQuery

**Includi jQuery**  
Includi la libreria jQuery nel layout.



**Sorgente jQuery**



jQuery - googleapis.com con fallback locale ▾



Qui è possibile selezionare da dove caricare lo script jQuery.

**Templates jQuery**

*Seleziona tutto*

  j\_colorbox

  j\_mediaelement

  j\_accordion

Qui è possibile selezionare uno o più template jQuery.

- ▾ MooTools

**Includi MooTools**  
Includi la libreria MooTools nel layout.

Anche in questo caso poi trovo un campo libero per includere altri javascript prima della chiusura del </body>

#### Codice Javascript personalizzato

Il codice javascript inserito qui, verrà aggiunto alla fine della pagina

Questo si rivela estremamente utile quando voglio mettere degli script nelle pagine che usano questo preciso layout, ma non voglio caricarli negli altri layout (per esempio, in home page ho uno slideshow mentre nelle pagine interne no).

Per inserire un js posso scrivere così:

```
<script src="files/theme/js/mioscript.js"></script>
```

Mentre per scrivere un pezzetto JQuery metto così:

```
<script>  
$(document).ready(function() {  
    // mio script qui...  
});  
</script>
```

## Fase 6: Assegnare il Layout alla root, ed alle pagine del sito

Ora dobbiamo solo accedere a **Layout** → **Struttura sito**, per dire alla root quale layout usare. Premiamo la matita in corrispondenza della root.



Nella selezione del layout dico di usare quello appena creato, cioè "default".

### Impostazione layout

#### Assegna un layout

Ogni pagina utilizza lo stesso layout della sua pagina genitore. Selezionando quest'opzione sarà possibile assegnare un layout diverso a

#### Layout pagina

Seleziona un layout. E' possibile gestire i layout utilizzando il

#### Layout per versione mobile

Questo layout sarà utilizzato per gli utenti che utilizzano un

Notiamo che esiste anche una scelta per il layout mobile, in effetti sarebbe possibile crearne uno, e Contao in base allo **User Agent** dell'utente poi fornisce il primo o il secondo layout. Tuttavia io evito di usarlo quando ho siti molto semplici o non richiedono particolari distinzioni tra le due versioni. Preferisco un approccio **responsive** usando Bootstrap e/o piccole media query.

In caso dovessimo assegnare layout differenti in determinate aree del sito l'operazione è simile, basta editare la pagina e cambiare layout, quel nodo e tutti i suoi "children" adotteranno il layout scelto.

Il tutorial è finito, per molti altri trucchi, fix, ed altro riguardo a Contao potete consultare il mio blog: <http://www.zod.it/it/blog/tag/Contao.html>

Buon divertimento.