

Golive tutorial in italiano

12/11/2005

Copyright 2000-2002, <kasper@typo3.com>

This document is published under the Open Content License
available from <http://www.opencontent.org/opl.shtml>

Traduzione di Virginio Carabba

The content of this document is related to Typo3
- a GNU/GPL CMS/Framework available from www.typo3.com

Table of Contents

Tutorial: Golive Template integration.....	1	Finetuning.....	14
Setting up an ordinary HTML template with Typo3.2		The bottom menu.....	17
Content.....	7	The complete code listing.....	19
The main menu.....	9	The constants-field:.....	19
The submenu.....	12	The setup-field:.....	19

Impostare un template ordinario in HTML

Benvenuti in questo tutorial dedicato all'impostazione di templates ordinari con Typo3.

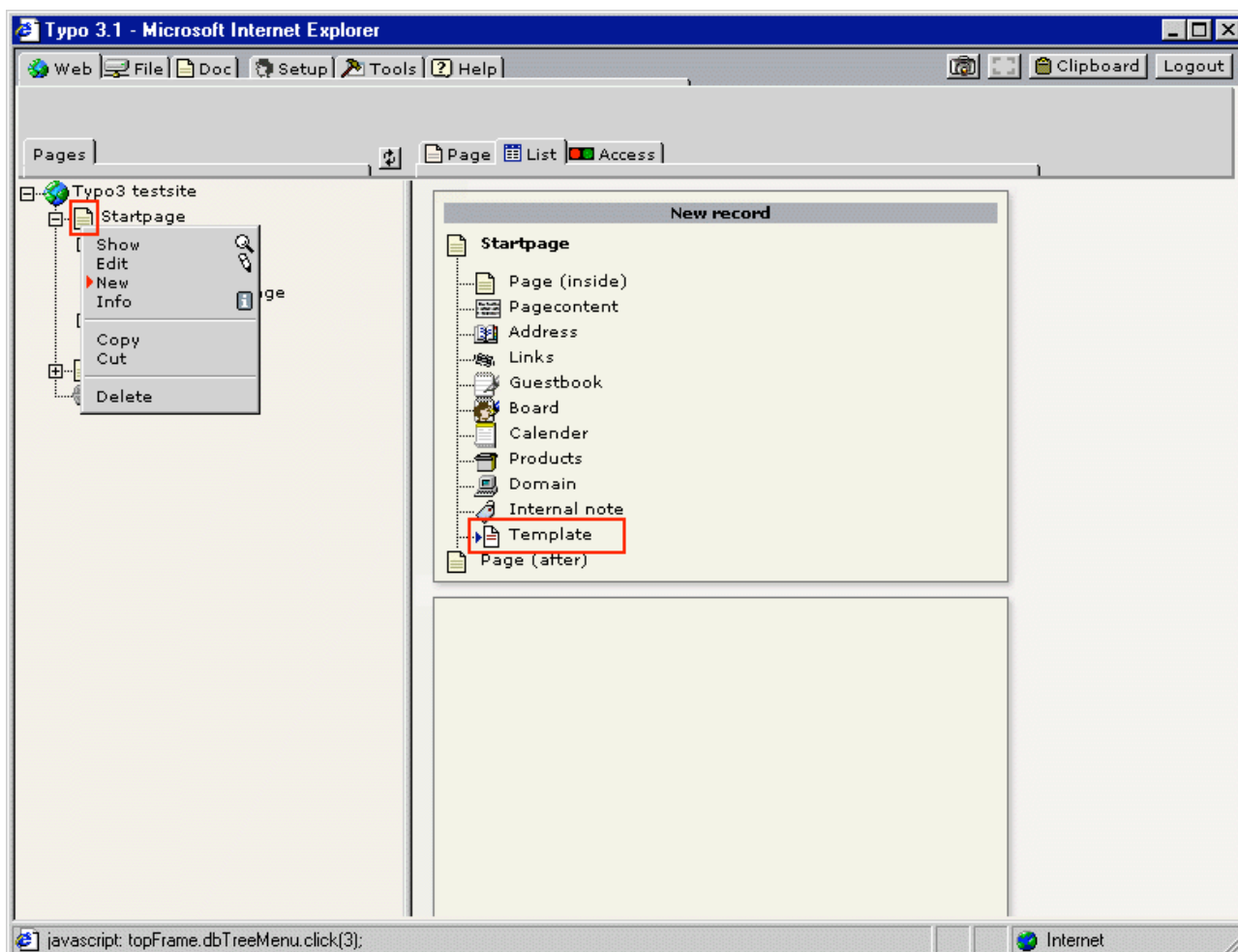
Originariamente la creazione di templates per Typo3 avveniva tramite TypoScript, che permetteva di portare un determinato contenuto dal sistema al codice HTML.

Il Typo Script non è molto semplice da apprendere ma è uno strumento molto potente e costituisce ancora la spina dorsale del sistema Typo3.

Questo tutorial prende ad esempio una pagina di prova costruita con GoLive.

- In primo luogo è necessario installare il sito di prova standard di Typo3
- In secondo luogo copiate i files e le cartelle di questo tutorial nel root del sito di prova, fatta eccezione per "tmplpage.html" e "main.css" che vanno invece copiati nella cartella "fileadmin/". Questo file è il template di base del vostro sito e deve risiedere in questa cartella, poiché Typo3 presume di accedere a quel file.

A questo punto effettuate il login nel Typo3 del sito di prova con l'utente "admin" e create un nuovo record "Template" sulla pagina "Startpage", in questo modo:



Inserite a questo punto il titolo del template. Si tratta di un semplice titolo e non verrà visualizzato altrove. Potete utilizzare il campo di descrizione in fondo per descrivere il vostro template. Inserite quindi il titolo del sito. Riflettete sul titolo più opportuno visto che verrà visualizzato nel campo titolo di ogni pagina (è comunque opzionale).

Spuntate il flag "Clear" (così facendo sarete sicuri che il TypoScript avrà cancellato ogni precedente

template nel page-tree) ed il flag "Rootlevel" (per indicare che volete che questo template costituisca una nuova root di un sito web).

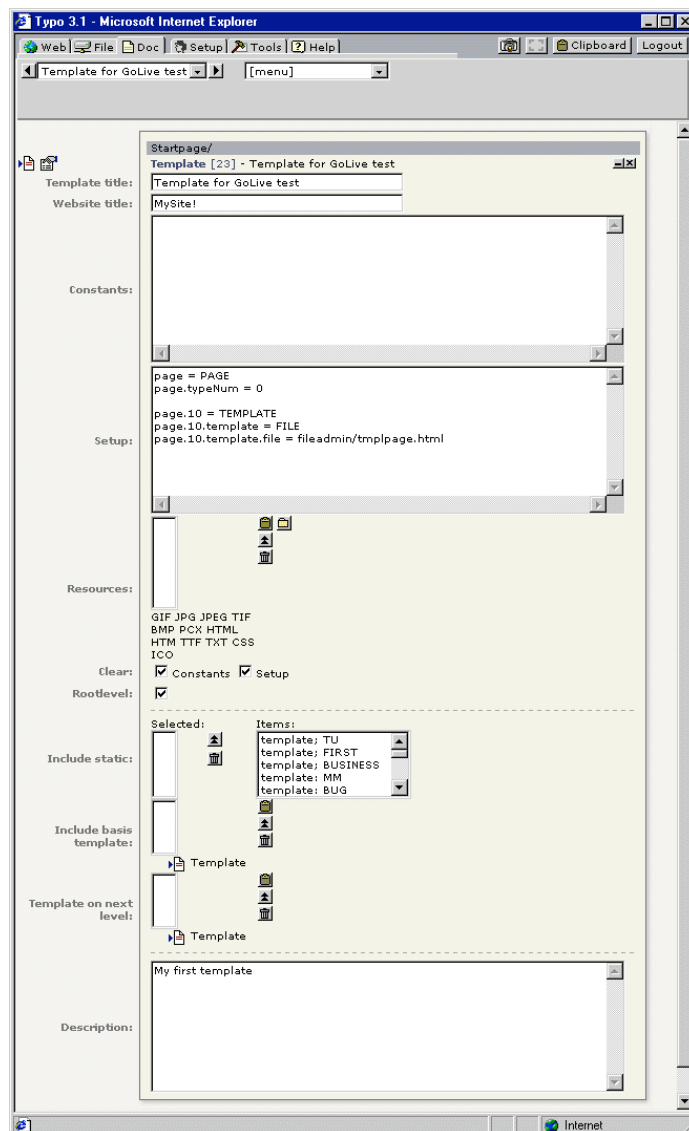
Nel campo „Setup“ inserite i seguenti valori:

```
page = PAGE
page.typeNum = 0

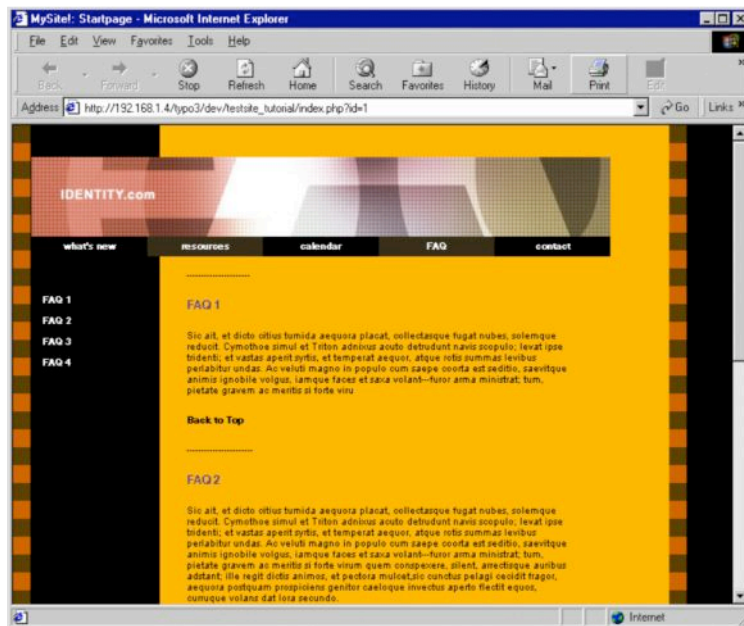
page.10 = TEMPLATE
page.10.template = FILE
page.10.template.file = fileadmin/tmplpage.html
```

Questi comandi TypoScript avviano un nuovo template definendo un oggetto-Pagina (page) ed assegnando ad esso il type-number di default "0". A questo punto il file template è caricato come input per un oggetto contenuto del template posto nella lista cObject della pagina pageobject.

Vi sembra incomprensibile? Non preoccupatevi, non è necessario che comprendiate tutti i dettagli (è per questo che state leggendo questo tutorial, no?). Il template dovrebbe a questo punto apparire così:



...se ora salvate il template e andate al frontend del sito questo è quello che dovrete vedere



Il template GoLive vi apparirà esattamente così, e fin qui tutto bene. Se date un'occhiata al codice HTML della pagina vedrete che il codice di testa del TypoScript è ancora inserito. Ci sono due tag body e di conseguenza dobbiamo concludere che non abbiamo ancora finito. Raccomanderei di mantenere il codice di testa di Typo (anche se potete disabilitare l'output di testa dal TypoScript). Ho già preparato il template GoLive per questo. Date un'occhiata alla posizione subito dopo il tag body del template GoLive. Ho inserito un commento HTML:

```

index[1] - Notepad
File Edit Search Help

    <csimport user="../Site01.data/Components/b
</head>

    <body background="Images/tile_bkgrnd.gif" onload="C
<!--###DOCUMENT_BODY### start-->
    <table cool width="616" height="935" border
    <tr height="17">
        <td width="9" height="934"
        <td width="606" height="17"
        <td width="1" height="17">
    </tr>
    <tr height="113">
        <td width="606" height="113

```

e lo stesso codice si trova anche in fondo al documento stesso. Questo è un "subpart". Modificate il record template all'interno di Typo3 in questo modo:

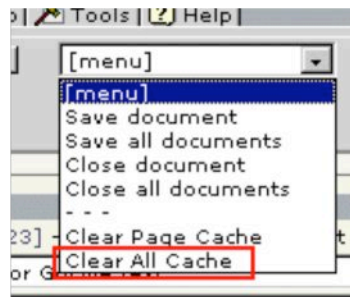
```

page = PAGE
page.typeNum = 0

page.10 = TEMPLATE
page.10 {
    template = FILE
    template.file = fileadmin/tmplpage.html
    workOnSubpart = DOCUMENT_BODY
}

```

La novità qui è l'aggiunta di DOCUMENT-BODY. Il resto è soltanto un modo più intelligente di scrivere codice TypoScript... Salvate il template e ricordate di pulire tutta la cache (dovete farlo tutte le volte che aggiornate un template!).



Una volta ricaricata la pagina questo è quello che vedrete:



Sembra carino, penserete. Lo penso anch'io ma c'è una buona ragione: con l'aggiunta di

```
workOnSubpart = DOCUMENT_BODY
```

solo la parte del template tra i commenti è inserita nella pagina.

In questo modo lasciate il codice di testa con, ad esempio, la definizione del foglio di stile dietro.

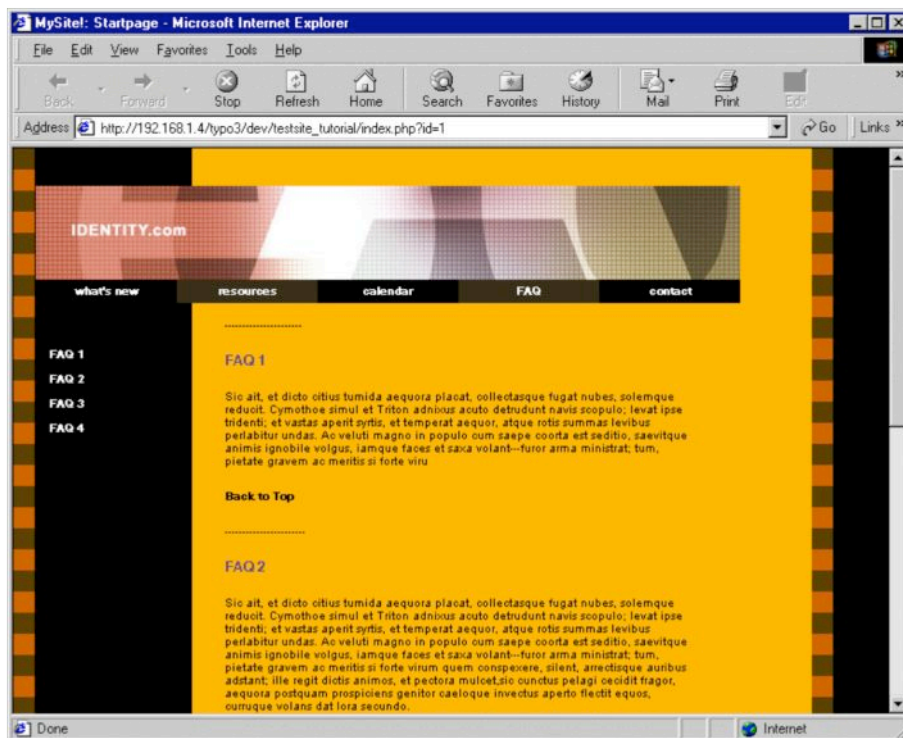
Guardando il codice di testa del file template dal GoLive troviamo queste parti

```
index[1] - Notepad
File Edit Search Help
// --></script>
</csactiondict>
<link href="main.css" rel="stylesheet" type="text/css">
<csimport user="../Site01.data/components/id_bar.html" occur="0"
<csimport user="../Site01.data/components/bottom_nav.html" occur=
</head>
<body background="Images/tile_bkgrnd.gif" onload="CSScriptInit();">
<!--###DOCUMENT_BODY### Start-->
<table cool width="616" height="935" border="0" cellpadding="0"
```

Compensiamo aggiungendo questo TypoScript al record template in Typo3:

```
page.stylesheet = fileadmin/main.css
page.bodyTag = <body background="Images/tile_bkgrnd.gif">
```

Salvate, pulite la cache, guardate la pagina.



Meglio così.

Contenuti

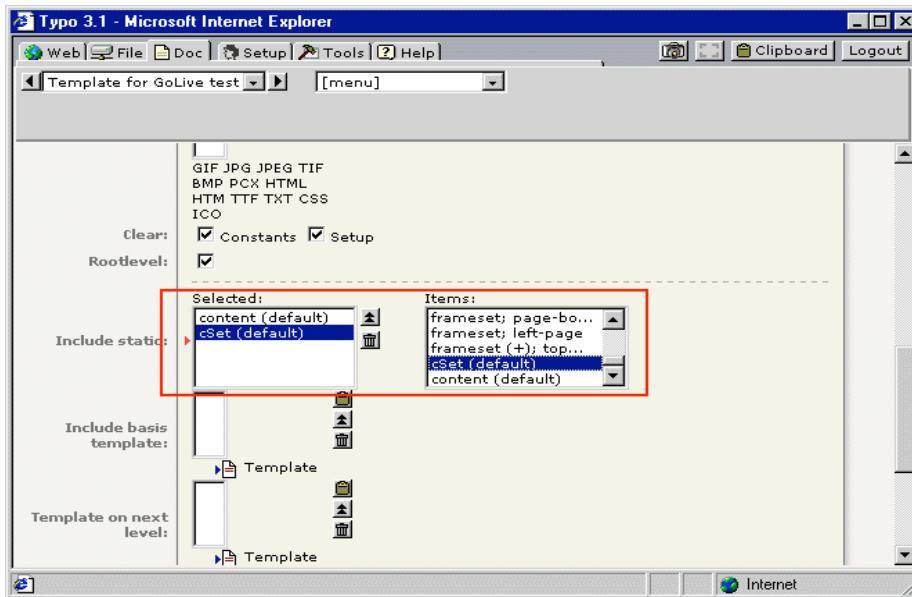
Ovviamente vogliamo che sulla pagina sia presente il contenuto che abbiamo inserito nel backend Typo3. Per fare questo usiamo i commenti

```
###CONTENT###
```

che ho aggiunto al template e sostituiamo queste sottoparti con i contenuti presi da Typo3!

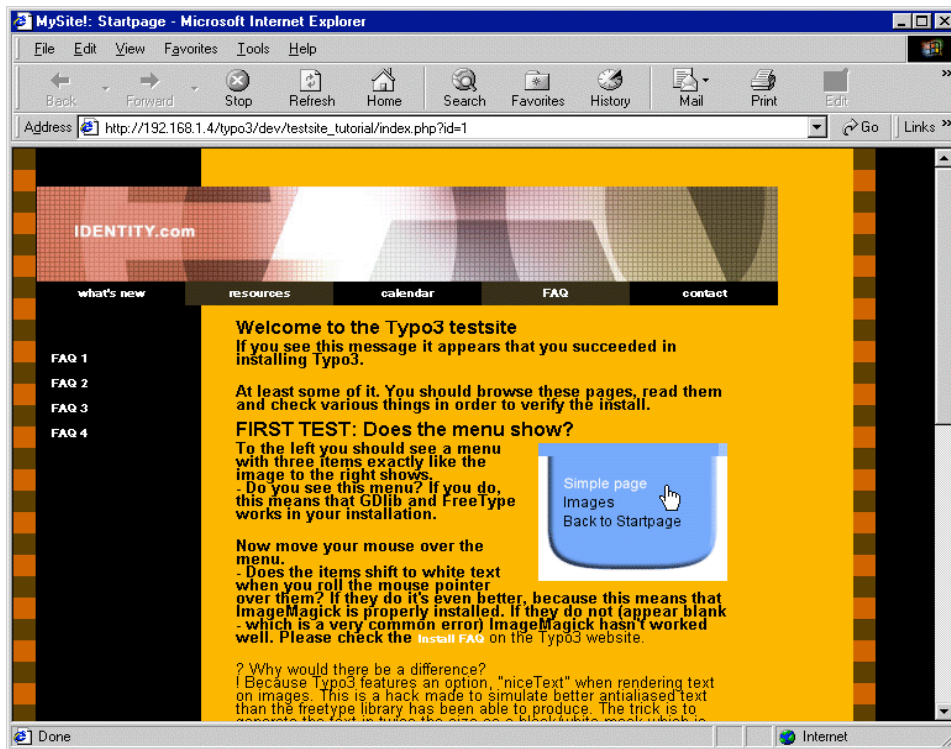
La formattazione del contenuto è troppo complicata da fare (from scratch so here I recommend that you learn how to manipulate the constants regarding content-rendering.)

Il contenuto è reso con il toplevel object "tt_content" del template TypoScript. Ciò significa sostanzialmente che qualunque cosa tu faccia non otterrai niente se non includi il template statico (content (default)". E' inoltre opportuno includere "cSet" per configurare in modo semplice i parametri più basilici del template.



Aggiungete questa riga al template TypoScript:

```
page.10.subparts.CONTENT < styles.content.get
```



Wow!

Navigazione principale

Ora volete creare un menu basato sulle pagine in Typo3. Avrete notato che ho inserito dei commenti nel file template dal GoLive. Questa volta ho pensato di sostituire le cinque voci del menu sotto l'immagine con link alle pagine da Typo3. Devo però assicurarmi che ci saranno solo cinque voci!

Il codice HTML sarà visualizzato in questo modo:

```
<tr height="20">
<!--###MENU### start-->
  <td width="120" height="20" bgcolor="black" nowrap>
    <center>
      <a href="whatsnew.html">what's new</a></center>
    </td>
  <td width="120" height="20" bgcolor="#2f2714" nowrap>
    <center>
      <a href="resources.html">resources</a></center>
    </td>
  <td width="120" height="20" bgcolor="black" nowrap>
    <center>
      <a href="calendar.html">calendar</a></center>
    </td>
  <td width="120" height="20" bgcolor="#2f2714" nowrap>
    <center>
      <a href="faq.html">FAQ</a></center>
    </td>
  <td width="120" height="20" bgcolor="black" nowrap>
    <center>
      <a href="contact.html">contact</a></center>
    </td>
<!--###MENU### stop-->
</tr>
```

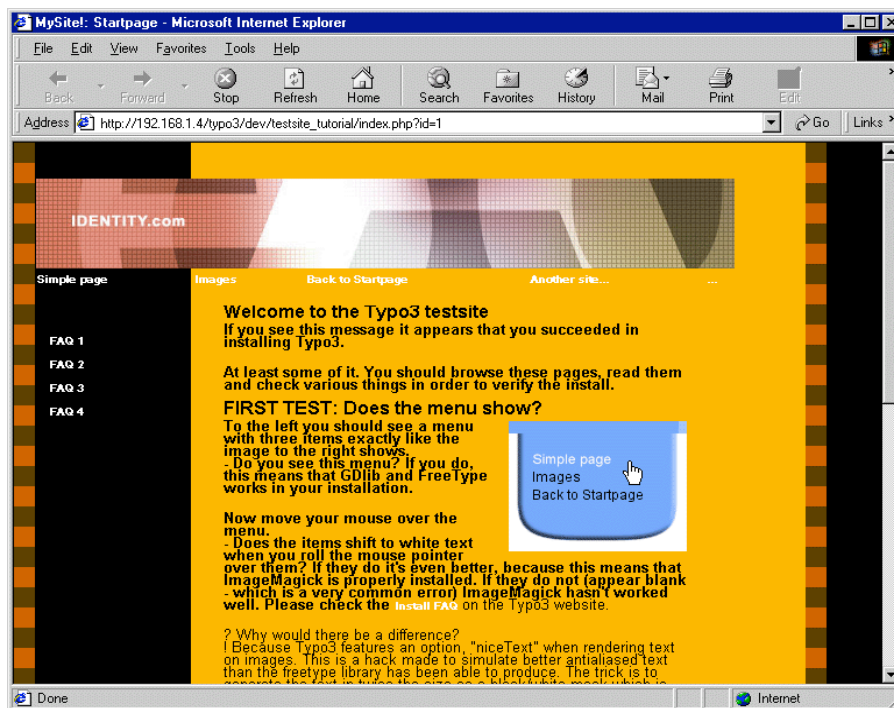
Come potete vedere ogni elemento del menu è inserito in una cella della tabella. Questo sostanzialmente significa che tutte le volte che creiamo un menu è necessario "avvolgere" ogni oggetto nella cella di una tabella. Proviamo ora a modificare il record template di Typo3 in questo modo:

```
page = PAGE
page.typeNum = 0
page.stylesheet = fileadmin/main.css
page.bodyTag = <body background="Images/tile_bkgrnd.gif">

page.10 = TEMPLATE
page.10 {
  template = FILE
  template.file = fileadmin/tmplpage.html
  workOnSubpart = DOCUMENT_BODY
  subparts.CONTENT < styles.content.get

  subparts.MENU = HMENU
  subparts.MENU.minItems = 5
  subparts.MENU.maxItems = 5
  subparts.MENU.1 = TMENU
  subparts.MENU.1.NO {
    allWrap = <td>|</td>
  }
}
```

In questo modo è stato inserito un menu dal primo livello contenente 5 elementi avvolti in altrettante celle di una tabella. Apparirà in questo modo:



Confrontando con il codice HTML iniziale possiamo vedere che gli elementi sono avvolti in un codice più sofisticato. Possiamo quindi, in alternativa, scrivere così:

```
<td width="120" height="20" bgcolor="black" nowrap> | </td>
```

Notiamo però che anche il colore di background dell'oggetto è cambiato! Con Typo3 è possibile fare anche questo! E' una funzionalità chiamata "optionSplit". Sfortunatamente è piuttosto complicata da utilizzare ma funziona e ci consente di definire in che modo debbano, ad esempio, essere avvolti il primo oggetto, gli ultimi due o gli oggetti nel mezzo.

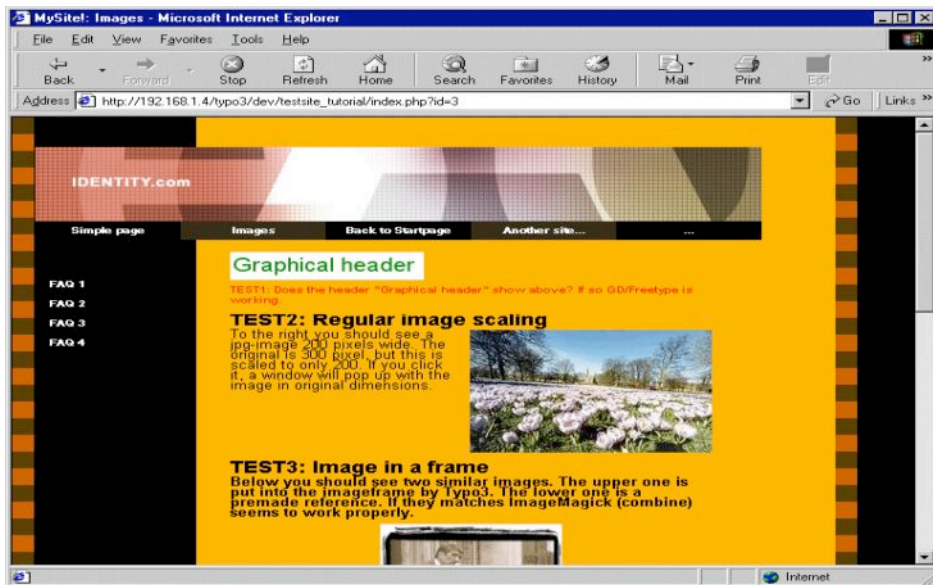
La funzionalità base di optionSplit che ci serve utilizzare è quella che consente all'involucro di cambiare tutte le volte tra due scelte. In questo modo:

```
|*| <td width="120" height="20" bgcolor="black" nowrap> | </td> || <td width="120" height="20" bgcolor="#2f2714" nowrap> | </td> |*|
```

<center>-tags per tutti gli elementi ma in questo caso ho utilizzato un'altra proprietà del TMENUITEM-object (che è quello utilizzato qui. Guarda i riferimenti sul TypoScript).

Modificate quindi il record-template in questo modo:

```
subparts.MENU.1.NO {
    allWrap = |*| <td width="120" height="20" bgcolor="black" nowrap> | </td> || <td width="120" height="20" bgcolor="#2f2714" nowrap> | </td> |*|
    linkWrap = <center>|</center>
}
```



(NO LINEBREAKS!!)

Ecco, perfetto! (Notate però che l'ultimo oggetto menu non è stato trovato fino a quando “...” è stato mostrato).

Sottomenu

Abbiamo ora bisogno di far sì che il menu piccolo sulla destra della pagina funzioni come un nostro sottomenu. Ho preparato il template GoLive per fare questo con l'aggiunta di questi commenti:

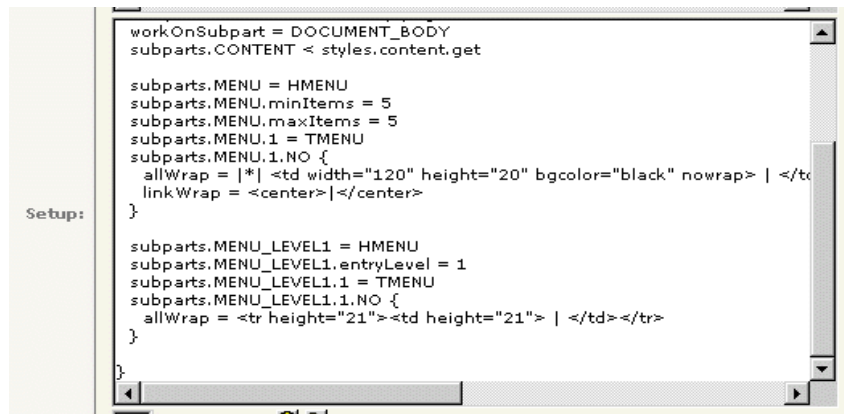
```
<table border="0" cellpadding="0" cellspacing="0" width="150">
<!--###MENU_LEVEL1### start-->
  <tr height="21">
    <td height="21"><a href="#Anchor_faq2">FAQ 1</a></td>
  </tr>
  <tr height="21">
    <td height="21"><a href="#Anchor_faq3">FAQ 2</a></td>
  </tr>
  <tr height="21">
    <td height="21"><a href="#Anchor_faq4">FAQ 3</a></td>
  </tr>
  <tr height="21">
    <td height="21"><a href="whatsnew.html">FAQ 4</a></td>
  </tr>
<!--###MENU_LEVEL1### stop-->
</table>
```

È molto più facile rispetto al menu principale. Solo un problema potrebbe darci noia: come farà Typo3 a selezionare le pagine dal secondo livello delle pagine del sito? La soluzione è la proprietà `.entryLevel` dell'oggetto `HMENU`.

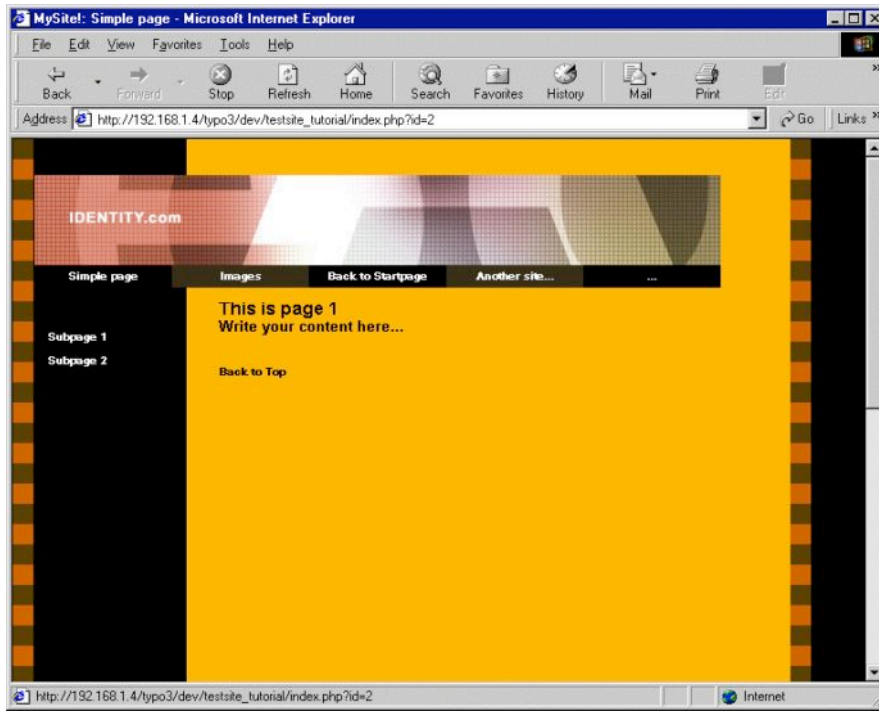
Scrivete così nel vostro template subito dopo il precedente menu:

```
subparts.MENU_LEVEL1 = HMENU
subparts.MENU_LEVEL1.entryLevel = 1
subparts.MENU_LEVEL1.1 = TMENU
subparts.MENU_LEVEL1.1.NO {
  allWrap = <tr height="21"><td height="21"> | </td></tr>
}
```

Ecco come il record template appare ora:



Clickate sul link "Simple Page" sul frontend e vedrete questo:



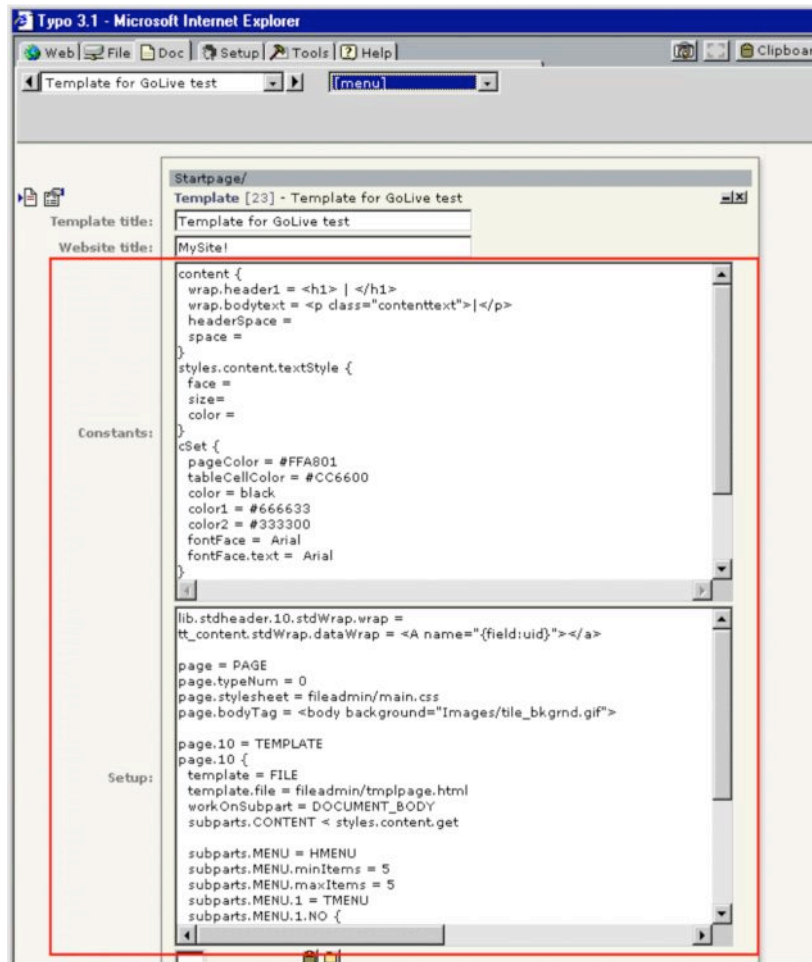
Finetuning

Usando lo static template "cSet" hai l'opzione di definire colori e fonts. Non dimenticare la definizione dell stylesheet. Inserire negli Constants:

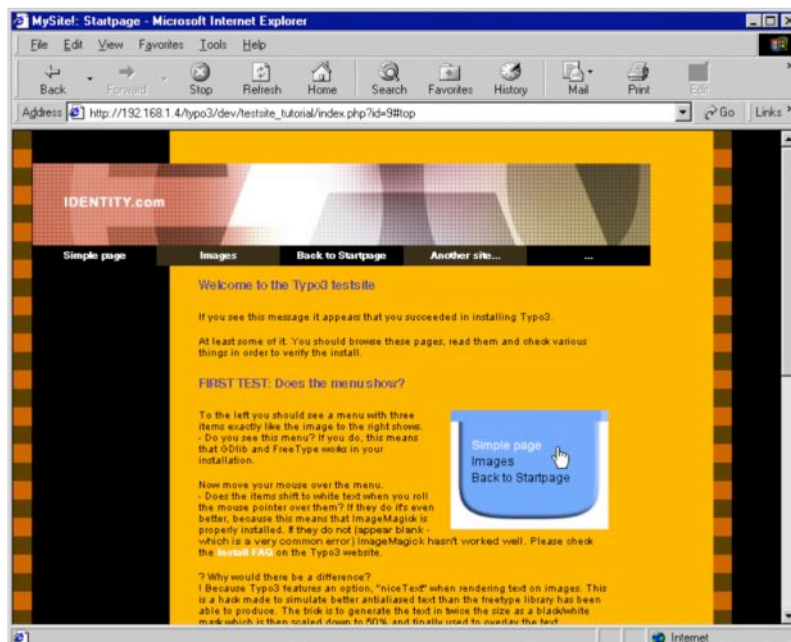
```
content {
  wrap.header1 = <h1> | </h1>
  wrap.bodytext = <p class="contenttext">|</p>
  headerSpace =
  space =
}
styles.content.textStyle {
  face =
  size=
  color =
}
cSet {
  pageColor = #FFA801
  tableCellColor = #CC6600
  color = black
  color1 = #666633
  color2 = #333300
  fontFace = Arial
  fontFace.text = Arial
}
# Image maximum widths, pixels
styles.content.imgtext.maxW = 460
# Targets, empty for no-frames websites
PAGE_TARGET =
content.pageFrameObj =
```

Poi inserire nel Setup:

```
lib.stdheader.10.stdWrap.wrap =
tt_content.stdWrap.dataWrap = <A name="{field:uid}"></a>
```



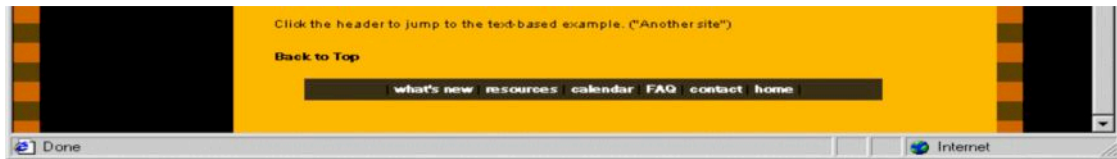
Ora il template si vede così:



Salva il template, svuota il cache e reload.

The bottom menu

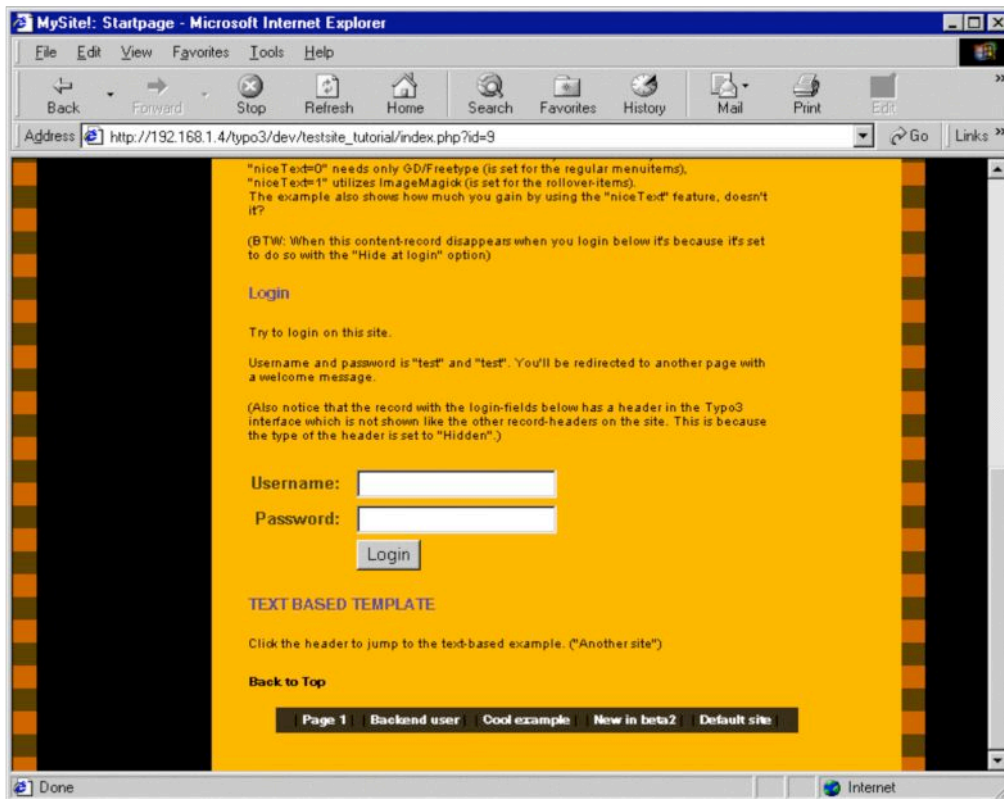
Ooops, abbiamo dimenticato di creare il bottom menu:



Ora aggiungi questo codice (eventualmente modificare il PID (10)):

```
subparts.BOTTOM_MENU = HMENU
subparts.BOTTOM_MENU.special = directory
subparts.BOTTOM_MENU.special.value = 10
subparts.BOTTOM_MENU.1 = TMENU
subparts.BOTTOM_MENU.1.wrap = <center> | </center>
subparts.BOTTOM_MENU.1.NO {
    before = &nbsp;|&nbsp;
    after = |*||*| &nbsp;|&nbsp;
}
```

That's it!



The complete code listing

The constants-field:

```
content {
  wrap.header1 = <h1> | </h1>
  wrap.bodytext = <p class="contenttext">|</p>
  headerSpace =
  space =
}
styles.content.textStyle {
  face =
  size=
  color =
}
cSet {
  pageColor = #FFA801
  tableCellColor = #CC6600
  color = black
  color1 = #666633
  color2 = #333300
  fontFace = Arial
  fontFace.text = Arial
}
# Image maximum widths, pixels
styles.content.imgtext.maxW = 460
# Targets, empty for no-frames websites
PAGE_TARGET =
content.pageFrameObj =
```

The setup-field:

```
lib.stdheader.10.stdWrap.wrap =
tt_content.stdWrap.dataWrap = <A name="{field:uid}"></a>

page = PAGE
page.typeNum = 0
page.stylesheet = fileadmin/main.css
page.bodyTag = <body background="Images/tile_bkgrnd.gif">

page.10 = TEMPLATE
page.10 {
  template = FILE
  template.file = fileadmin/tmplpage.html
  workOnSubpart = DOCUMENT_BODY
  subparts.CONTENT < styles.content.get

  subparts.MENU = HMENU
  subparts.MENU.minItems = 5
  subparts.MENU.maxItems = 5
  subparts.MENU.1 = TMENU
  subparts.MENU.1.NO {
    allWrap = |*| <td width="120" height="20" bgcolor="black" nowrap> | </td> || <td width="120"
height="20" bgcolor="#2f2714" nowrap> | </td> |*|
    linkWrap = <center>|</center>
  }

  subparts.MENU_LEVEL1 = HMENU
  subparts.MENU_LEVEL1.entryLevel = 1
  subparts.MENU_LEVEL1.1 = TMENU
  subparts.MENU_LEVEL1.1.NO {
    allWrap = <tr height="21"><td height="21"> | </td></tr>
  }

  subparts.BOTTOM_MENU = HMENU
  subparts.BOTTOM_MENU.special = directory
  subparts.BOTTOM_MENU.special.value = 10
  subparts.BOTTOM_MENU.1 = TMENU
  subparts.BOTTOM_MENU.1.wrap = <center> | </center>
  subparts.BOTTOM_MENU.1.NO {
    before = &nbsp;|&nbsp;
    after = |*||*| &nbsp;|&nbsp;
  }
}
```

}

Non dimenticare di attivare i flags seguenti:

