

?????????????:???????????? ???? ???????

# Golive

15th of December, 2000

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

Этот документ публикуется в соответствии с Open Content License  
доступной на <http://www.opencontent.org/opl.shtml>

Содержимое этого документа относится к TYPO3  
- GNU/GPL CMS/Framework доступной с [www.typo3.com](http://www.typo3.com)

????????????

<b>Руководство:интеграция шаблонов Golive.....</b>	<b>1</b>	<b>Подменю.....</b>	<b>12</b>
Создание простого HTML шаблона в		Тонкая подстройка.....	14
Туро3.....	2	Нижнее меню.....	17
Контент (Содержимое).....	7	Окончательная распечатка кода.....	19
Основное меню.....	9	Поле constants:.....	19
		Поле setup.....	19

# ????????? ?????????? HTML ???????? ? Typo3

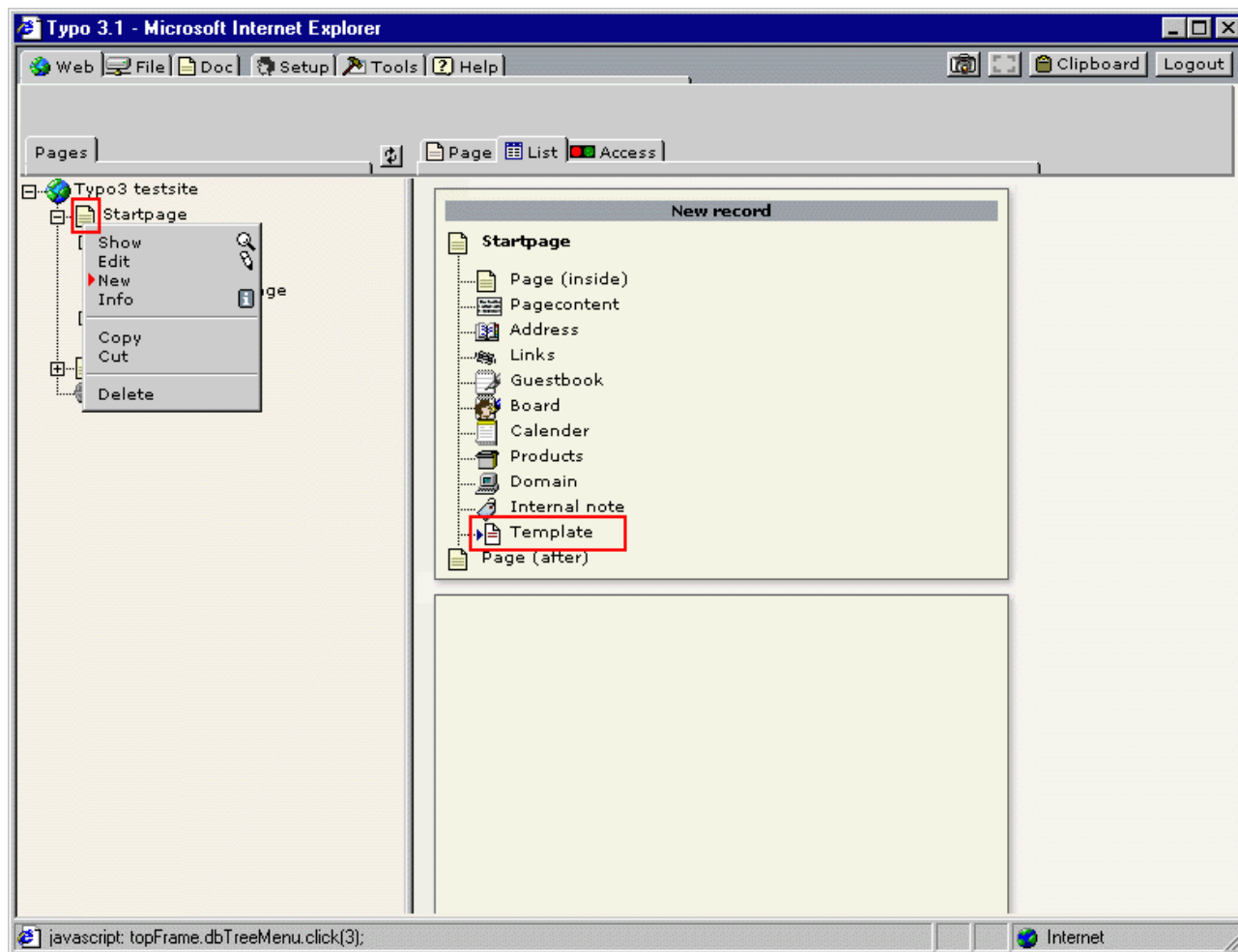
Приветствуем вас в руководстве по созданию простого шаблона в Typo3. Первоначально Typo3 создавался для создания шаблонов с помощью TypoScript, который позволяет упаковывать контент системы в коды HTML. Этим нелегко овладеть, хотя TypoScript – мощное и, еще и ,стержневое средство Typo3 CMS.

В этом руководстве рассматривается простая страница созданная в GoLive. Она распространяется архивированными файлами с этим руководством или может быть загружена отсюда: [http://typo3.sunsite.dk/documentation/tutorial1\\_golive.zip](http://typo3.sunsite.dk/documentation/tutorial1_golive.zip)

- Сначала необходимо установить тестовый сайт Typo3 и заставить его работать.

- Затем – скопировать файлы и каталоги этого руководства в корень тестового сайта, кроме "tmplpage.html" и "main.css", которые следует поместить в каталог "fileadmin/". Этот файл является базовым шаблоном для вашего сайта и должен располагаться в этом каталоге, поскольку Typo3 предполагает найти его там.

Теперь следует зарегистрироваться как администратору в тестовом сайте и создать новую запись типа "Template" на странице "Startpage". Это выглядит так:



Введите заголовок шаблона. Он нигде не показывается, просто заголовок. Вы можете использовать поле описания внизу для описания шаблона. Введите заголовок сайта "Website title". Он показывается перед заголовком страницы поле заголовка <title> каждой страницы, постарайтесь сделать это разумно (если вы введете что-нибудь...дополнительно).

Установите флаг "Clear" (для уверенности очистки установок любых предыдущих шаблонов в дереве страницы) и флаг "Rootlevel" (для указания на то, что вы хотите позволить этому шаблону инициализировать новый корень сайта).

В поле "Setup" введите следующее:

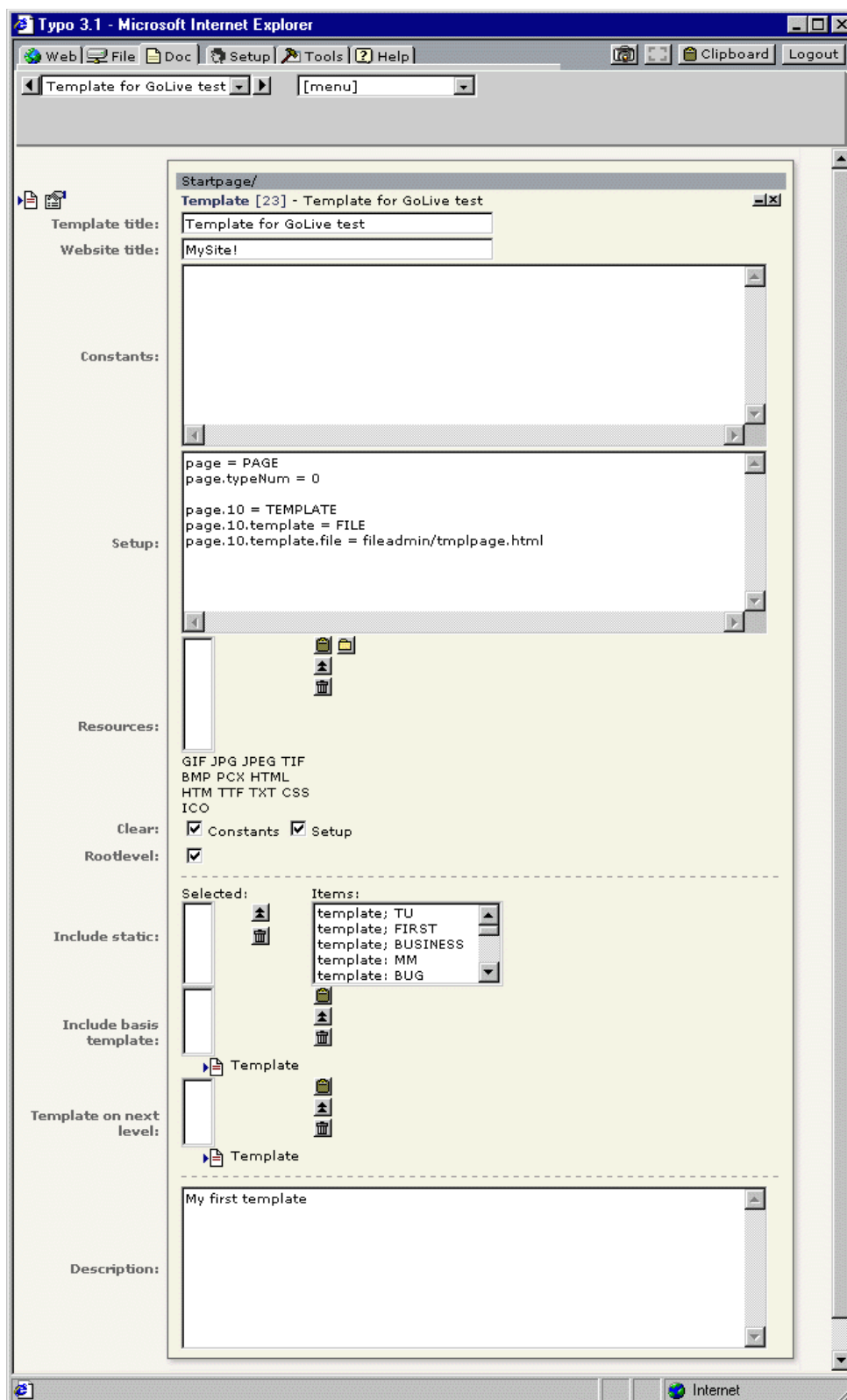
```
page = PAGE
page.typeNum = 0

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

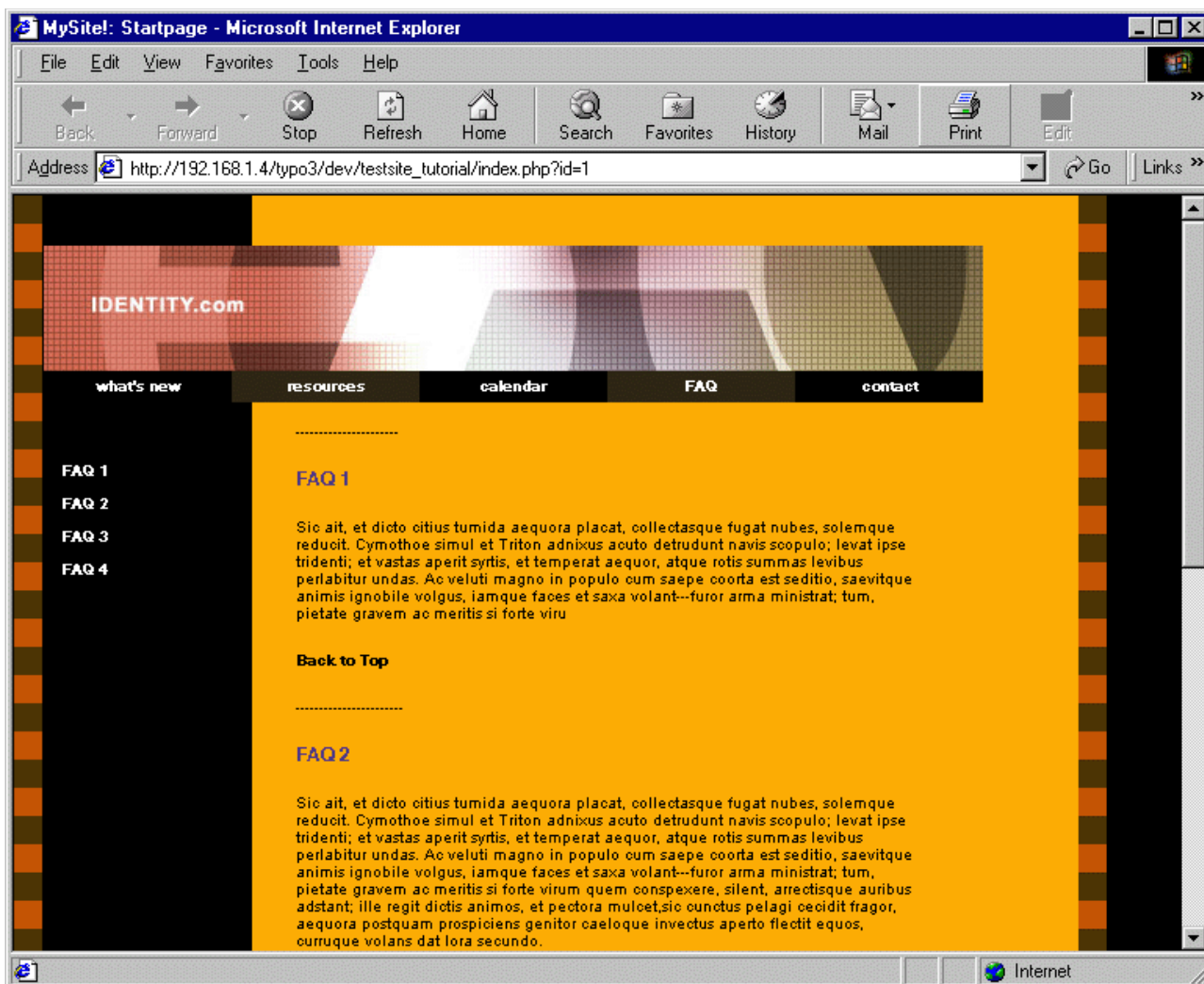
Это TypoScript и он начинает новый шаблон объявлением объекта PAGE (страница) и присвоением ему номера типа по умолчанию - "0". Затем файл шаблона загружается, как исходный для TEMPLATE объекта, помещаемого в массив

сОбъект страничного объекта "page". Тарабарщина? Не беспокойтесь, вам не следует влезать во все детали ( вот почему вы и читаете это руководство, не так ли ?).

Теперь шаблон будет выглядеть так:



... и если вы сохраните шаблон и перейдете во внешний интерфейс , то увидите следующее:



Теперь это точно шаблон созданный в GoLive, что будет верно и далее.

Если взглянуть на код HTML этой страницы, вы увидите, что код заголовка TypoScript все еще вставлен. Здесь присутствуют две надписи `<body ...>`, из чего мы можем заключить, что мы еще не все сделали.

Я настоячиво рекомендую сохранять код заголовка Typo `<head>...</head>` ( хотя можно отключить вывод заголовков из TypoScript). Я уже подготовил шаблон из Golive для этого. Давайте взглянем на позицию правее и ниже `<body>` на шаблоне Golive. Я вставил HTML- комментарий!:

```

index[1] - Notepad
File Edit Search Help

</head>
<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

```

.. и такой же код можно также найти в самом низу документа. Это называется "subpart"-под частью.

Изменим запись шаблона в Typo3 следующим образом:

```

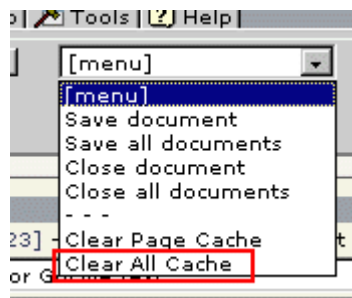
page = PAGE
page.typeNum = 0

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

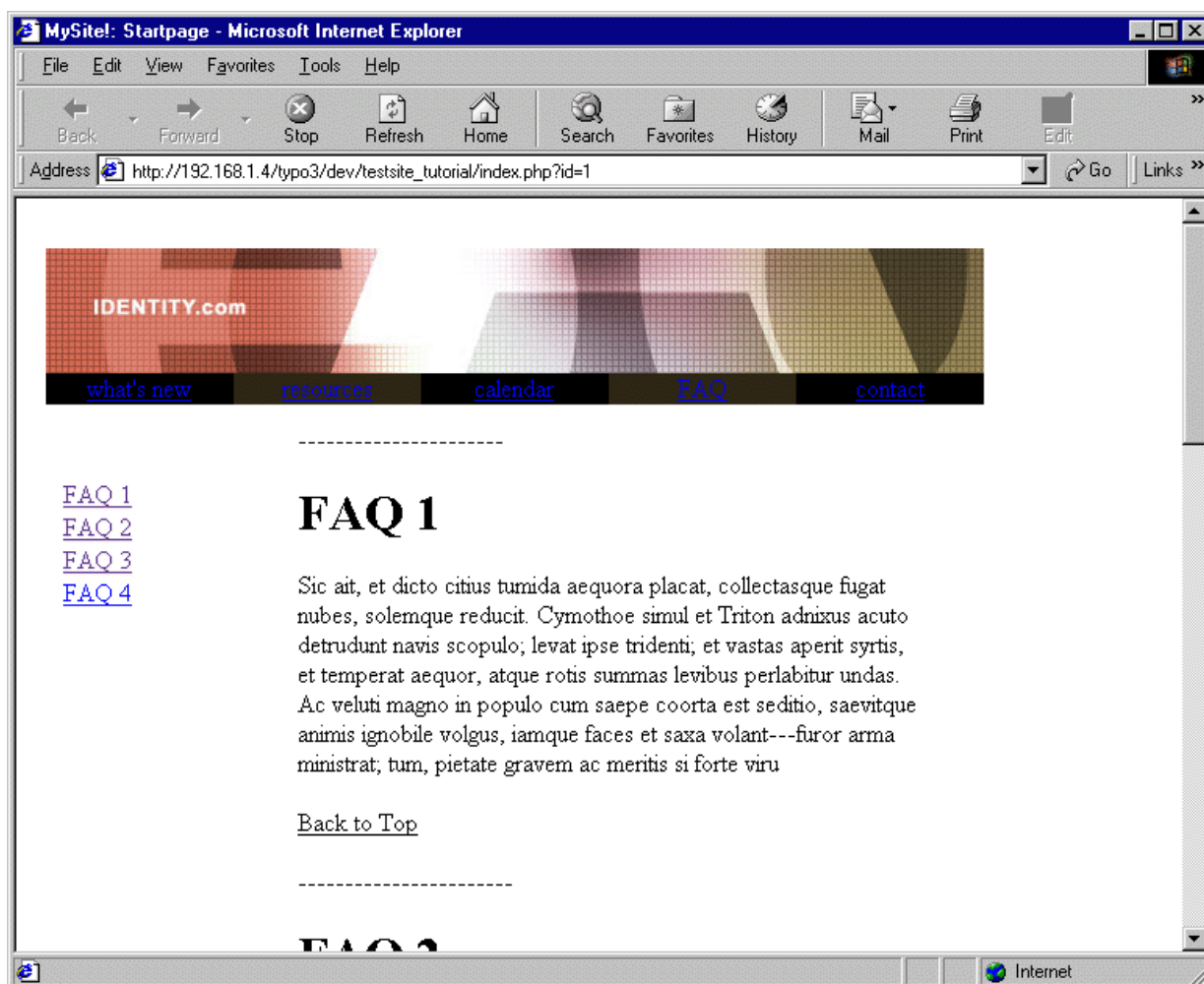
```

Новое здесь – добавление DOCUMENT\_BODY. В остальном - аккуратная запись кода TypoScript...

Сохраните шаблон и не забудьте очистить все кэши ("Clear all cache") , это следует делать всегда при обновлении шаблонов!



Перезагрузите страницу и вы увидите:



Выглядит ужасно, думаете вы. Я - тоже, но есть хорошее объяснение:

```
workOnSubpart = DOCUMENT_BODY
```

... в страницу попадает только часть шаблона между комментариями. Это оставляет код заголовка вместе с определением стиля в стороне.

Глядя на код заголовка шаблона после GoLive мы отмечаем, что он очень сильно нас интересует:

```
index[1] - Notepad
File Edit Search Help

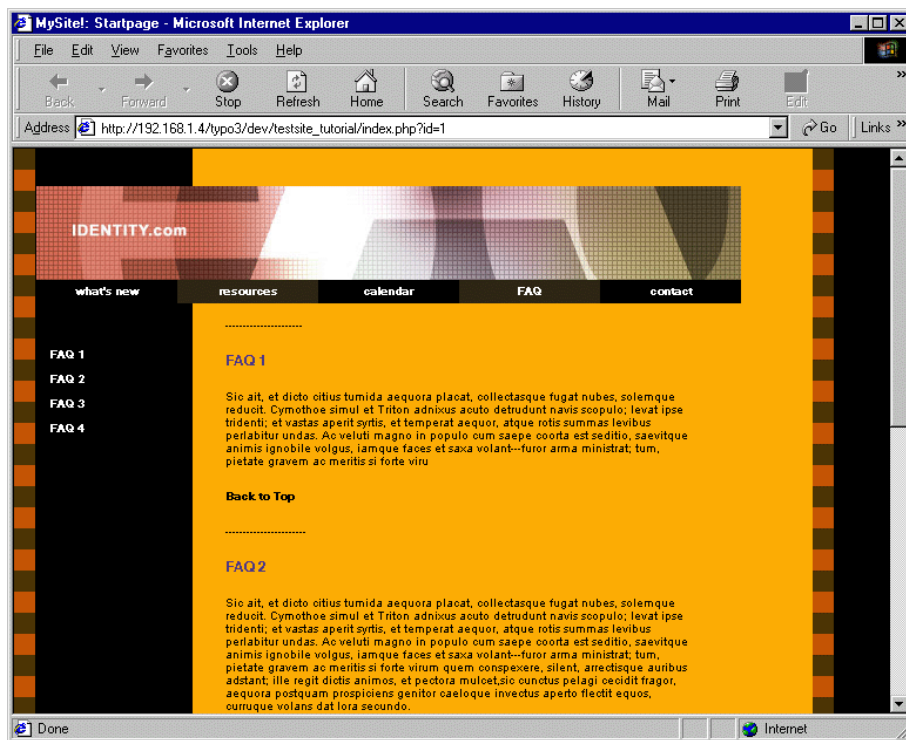
// --></script>
    </csactiondict>
    <link href="main.css" rel="styleSheet" type="text/css">
    <csimport user=" ../Site01.data/components/id_bar.html" occur="0"
    <csimport user=" ../Site01.data/Components/bottom_nav.html" occur=
</head>

    <body background="Images/tile_bkgrnd.gif" onload="CSScriptInit();"
<!--###DOCUMENT_BODY### start-->
    <table cool width="616" height="935" border="0" cellpadding="0"
```

Мы компенсируем это добавлением TurboScript к записи шаблона в Typo3:

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

Сохраняем, очищаем все кэши, смотрим.



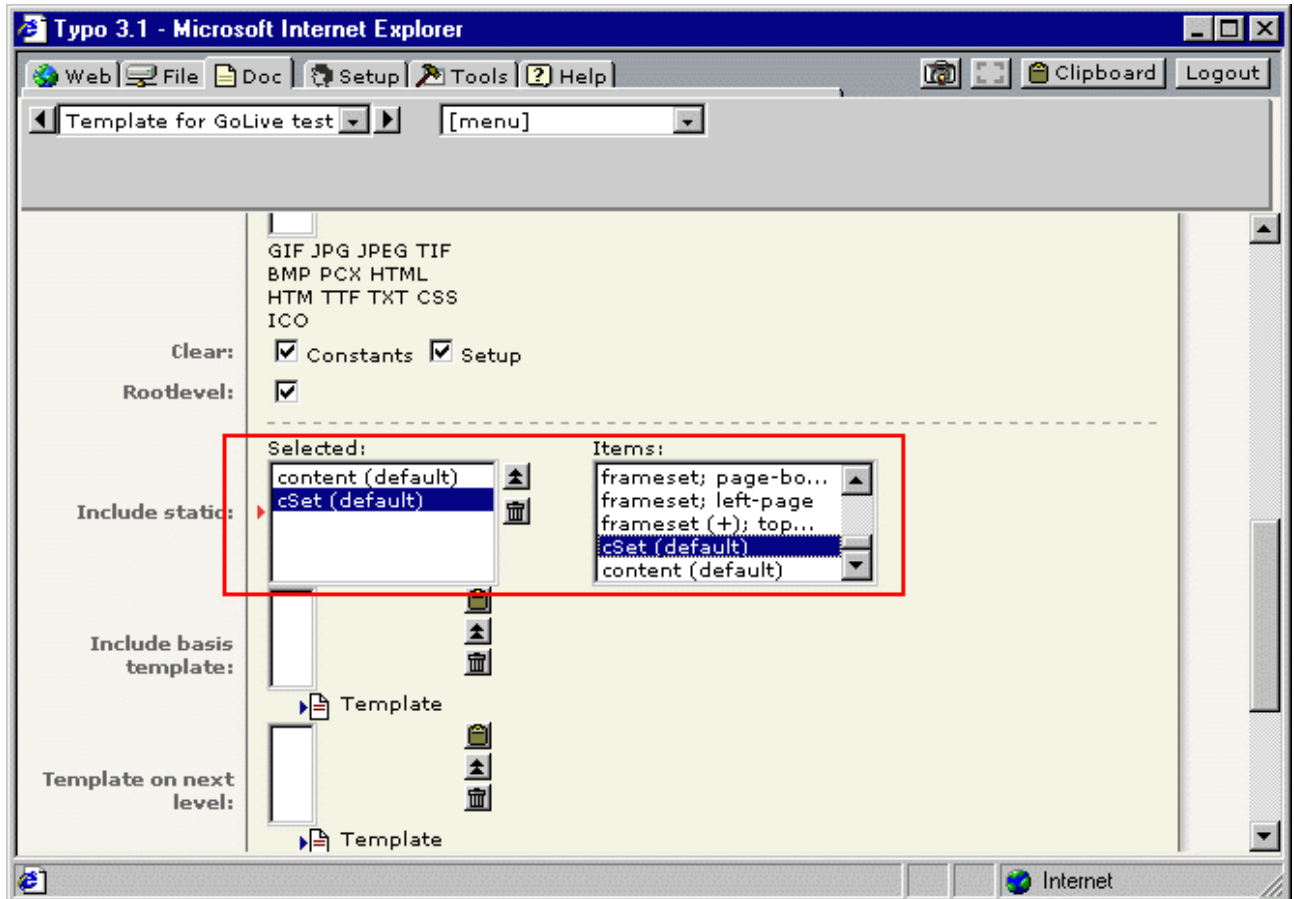
Значительно лучше.

## ????????(С?????????)

Конечно, нам бы хотелось видеть собственный контент на странице, который мы внесли в интерфейсе backend TYPO3. Для этой цели мы собираемся использовать комментарий `###CONTENT###`, уже добавленный к шаблону и наполнить эти подчасти содержимым из Turbo3!

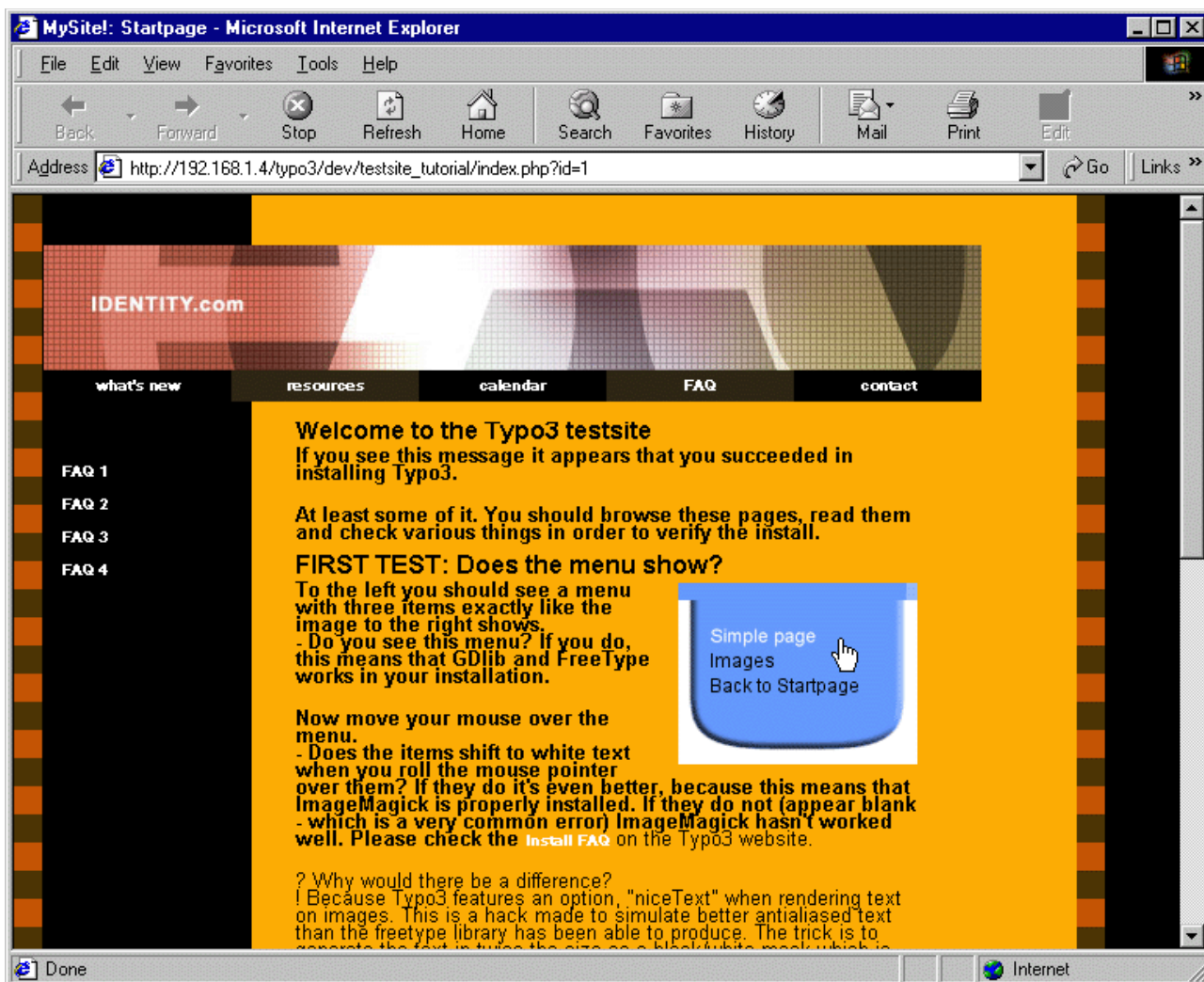
Форматирование контента достаточно сложная задача для выполнения с листа, поэтому здесь я рекомендую вам изучить манипуляцию константами влияющими на его обработку

Контент обрабатывается с помощью объекта верхнего уровня "tt\_content" шаблона TypoScript. В принципе, это означает - что бы вы не делали - ничего не получится , если вы не включили статический шаблон "content (default)". Вы, также, должны включить "cSet" для облегчения конфигурации важнейших параметров шаблона.



Добавьте это в шаблоне TypoScript:

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



Ура!!!



## ???????? ???? ?

Теперь нам хочется создать меню на странице Туро3. Как вы сможете заметить, я вставил комментарии в файл шаблона из GoLive. В данный момент мой замысел сводился к вставке пяти пунктов меню под изображением с ссылками на страницы в Туро3. Но я также хочу быть уверен в том, что в меню будет только 5 пунктов.

Код HTML выглядит следующим образом:

```
<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>
```

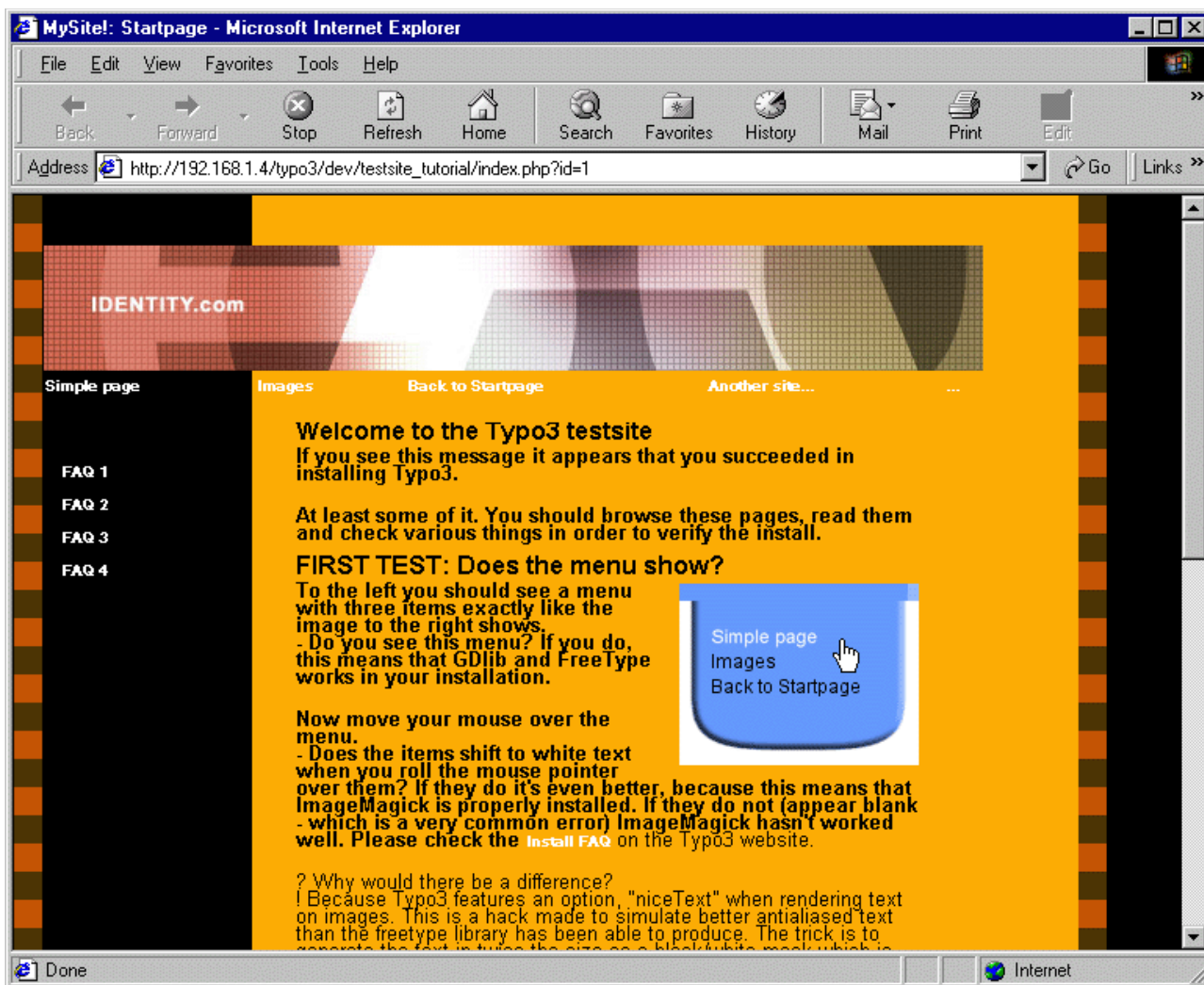
Как можно увидеть, каждый элемент меню является ячейкой таблицы. Итак, нам необходимо поместить каждый пункт в ячейку таблицы при создании меню. Теперь модифицируем запись шаблона в Туро3 следующим образом:

```
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>
  }
}
```

Здесь создано 5-ти элементное меню страниц первого уровня помещенное в ячейки таблицы. А выглядит это так:



Глядя на оригинальный HTML-код мы можем увидеть, что элементы вставлены в более изощренный код. Таким образом, вместо этого мы можем вставить:

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

Но мы также видим, что цвет фона пунктов изменен. Это тоже можно делать в Typo3. Эта операция называется "optionSplit". К сожалению, эта операция очень сложна, но она работает и позволяет определять заполнение. «Первый», «два последних» и средние пункты меню будут помещены в оболочку.

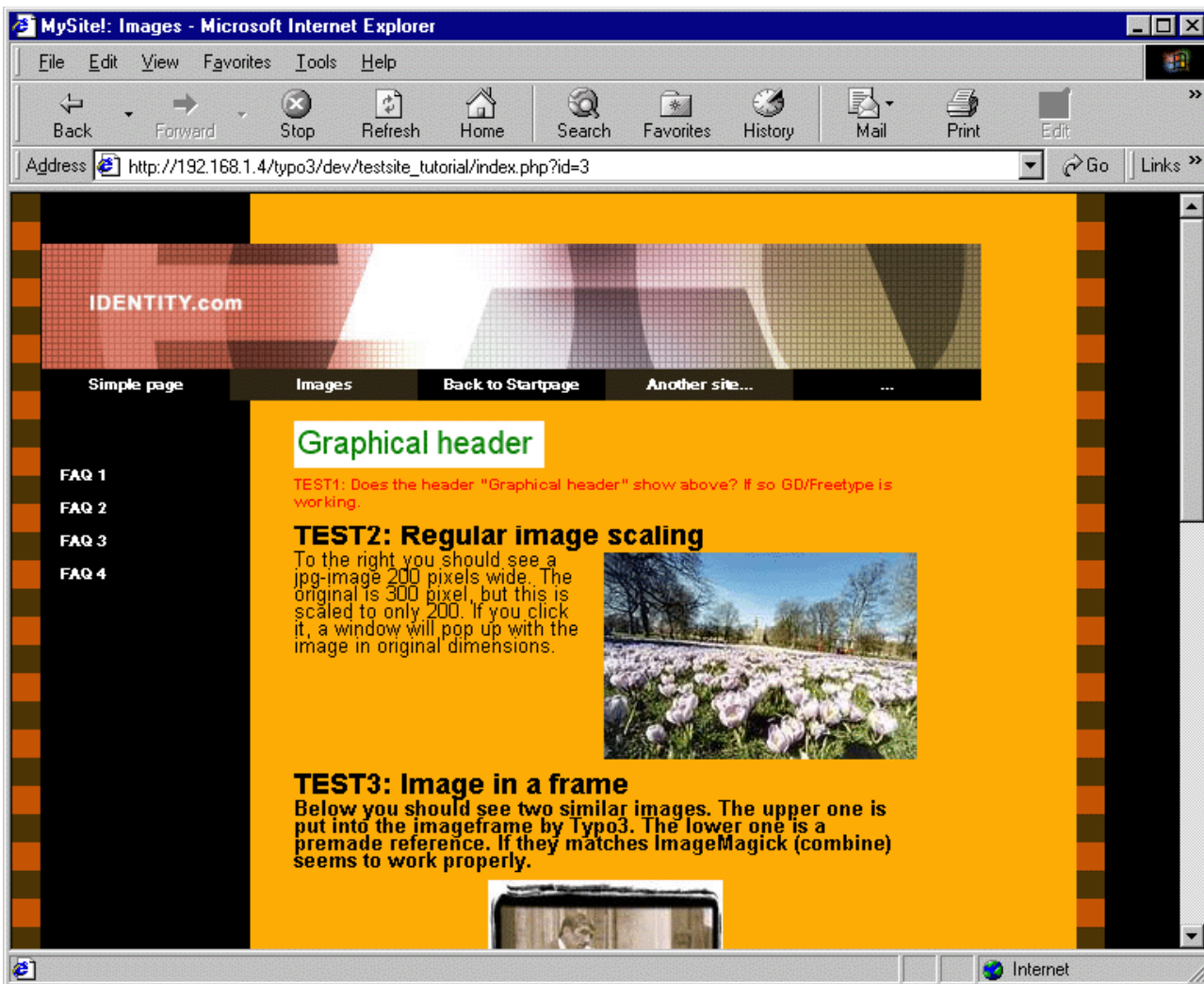
Основная форма optionSplit, в которой мы сейчас нуждаемся, должна позволять нам изменять представление циклично из двух вариантов. Это делается так:

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

Я также ввел метку <center> для всех элементов, но здесь использовалось другое свойство объекта TMENUITEM (который единственный используется здесь. Смотрите TypoScript описание).

Итак, модифицируем запись шаблона:

```
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!! – никаких переводов строки)

Просто замечательно! ( Но отметим, что последний пункт меню не найден, поэтому демонстрируется «...»)

## ???????

Следующее, чего нам надо сделать – маленькое подменю слева. Я также подготовил шаблон GoLive для этого, вставив комментарий:

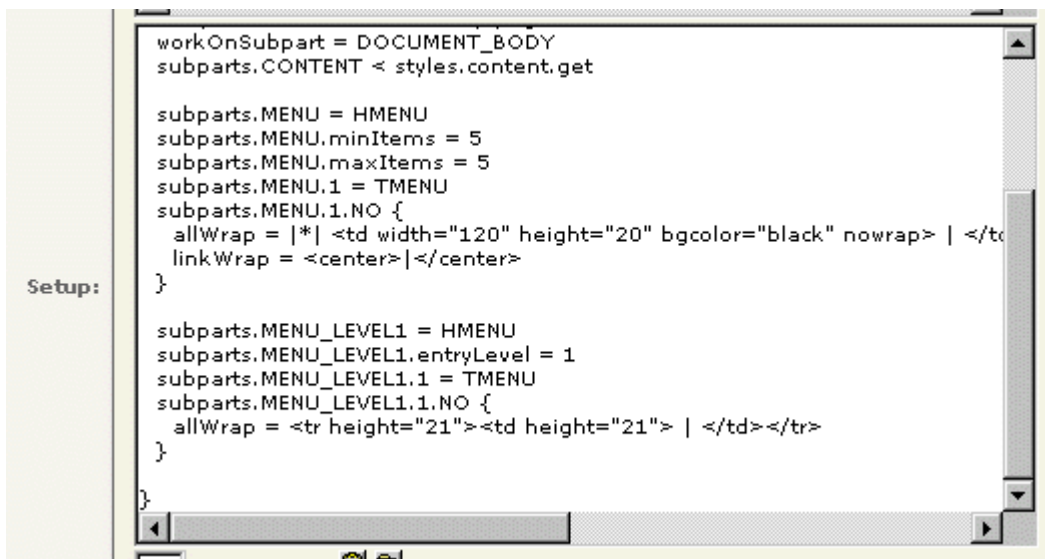
```
<table border="0" cellpadding="0" cellspacing="0" width="150">
<!--###MENU_LEVEL1### start-->
  <tr height="21">
    <td height="21"><a href="#Anchor_faq2">FAQ 1</a></td>
  </tr>
  <tr height="21">
    <td height="21"><a href="#Anchor_faq3">FAQ 2</a></td>
  </tr>
  <tr height="21">
    <td height="21">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

Это значительно проще, чем основное меню. Только одна вещь вызывает сомнения: каким образом Туро3 выберет страницы со второго уровня сайта? Это делается свойством entryLevel объекта HMENU.

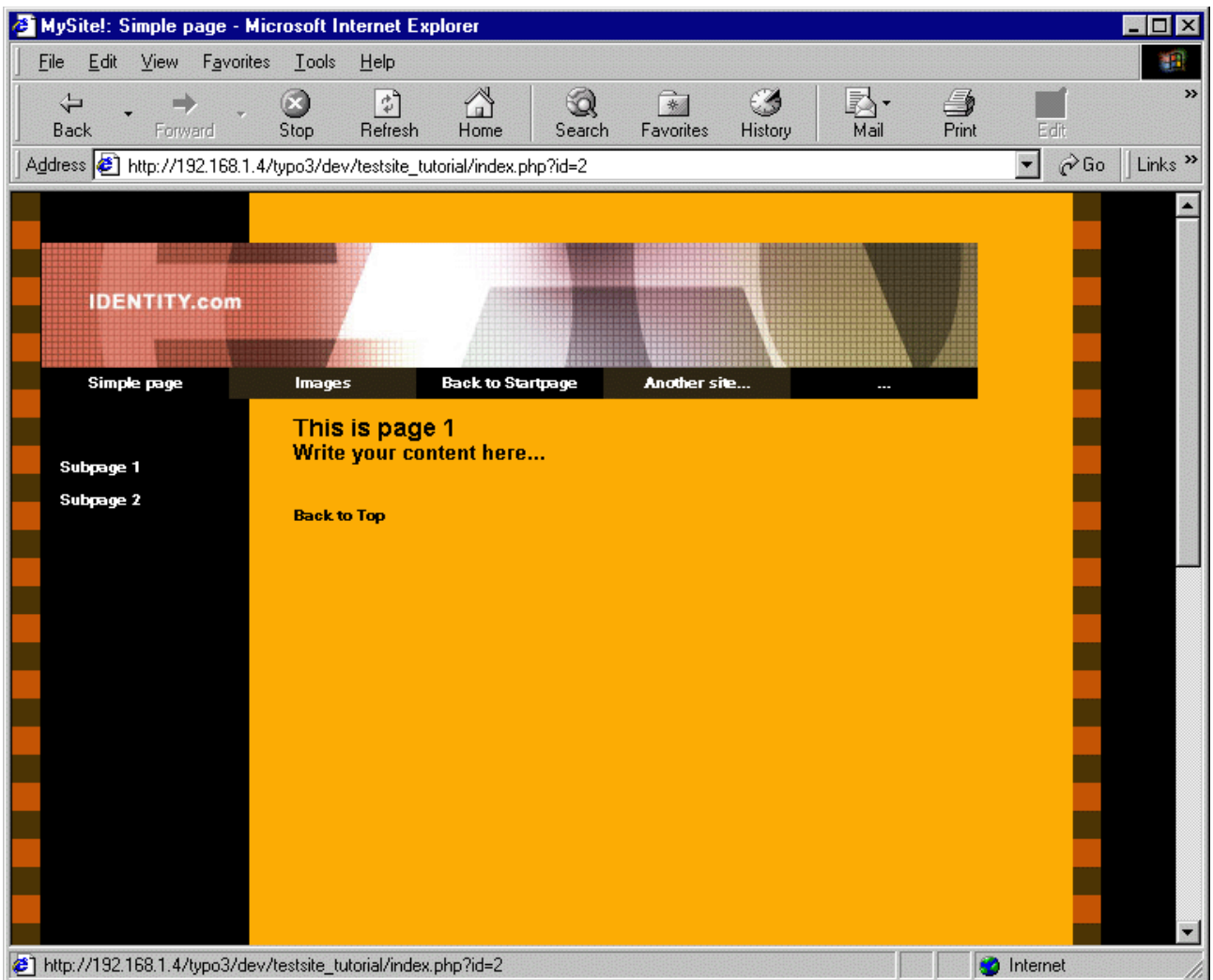
Просто вставим это описание сразу за предыдущим меню:

```
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>
}
```

Теперь запись шаблона выглядит так:



Кликните по ссылке "Simple page" на отображении сайта и вы увидите:



## ?????? ????????????

Используя шаблон "cSet" вы получаете доступ к некоторым константам, которые могут определять параметры вроде шрифтов и цвета. Вы также можете управлять некоторыми параметрами отображения используя таблицы стилей.

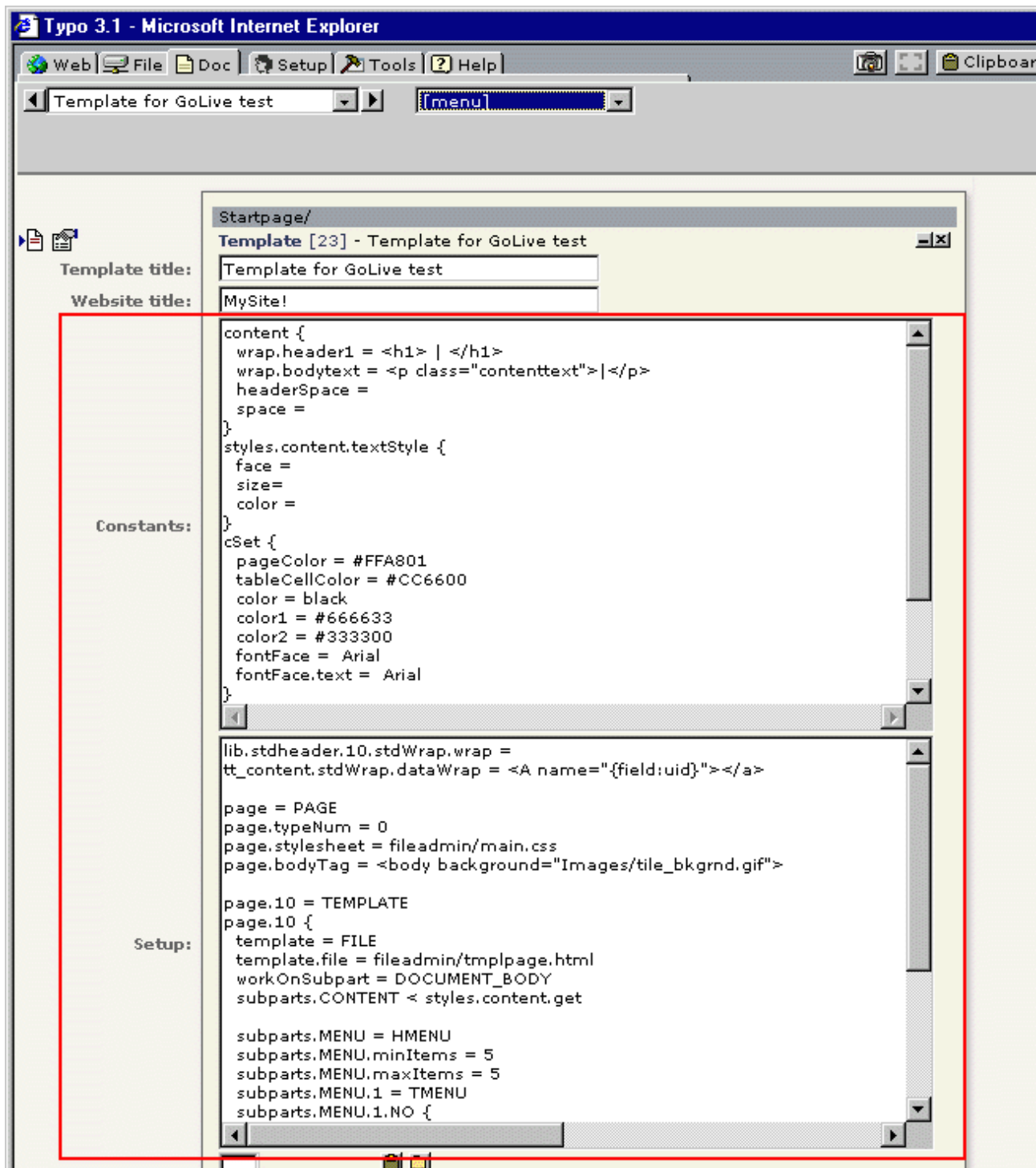
Вставьте это в поле **констант**:

```
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 =
```

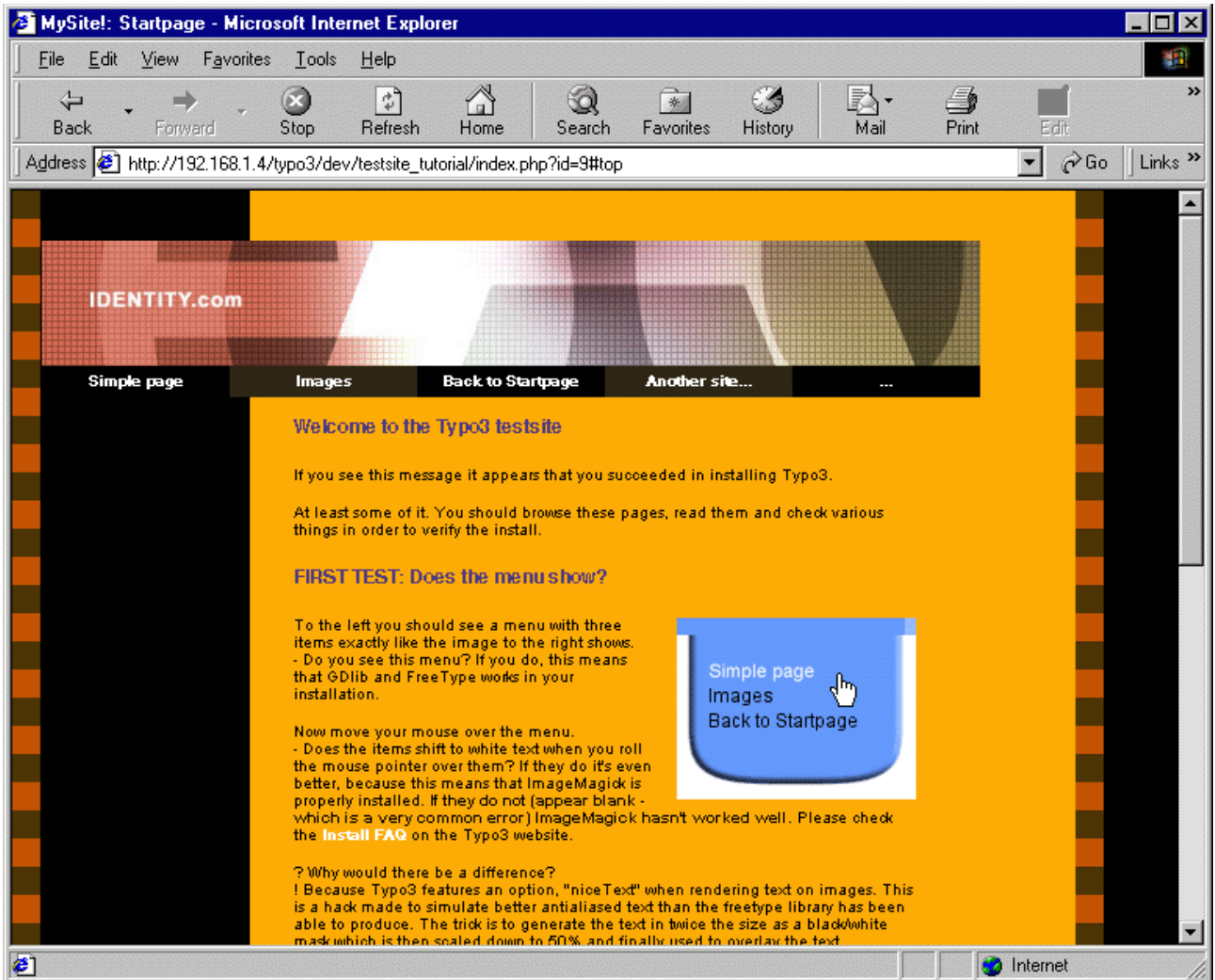
Вставьте это в поле **Setup**:

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

Теперь шаблон выглядит так:



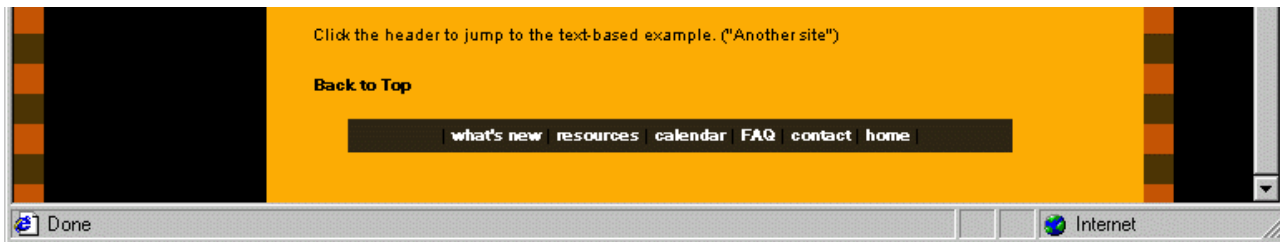
Сохраните шаблон, очистите все кэши ( как обычно ) и перезагрузите внешнее представление.



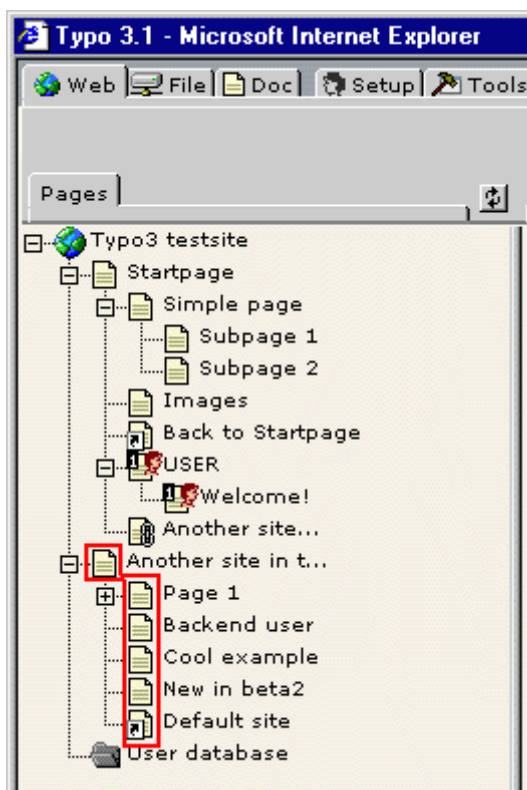


## ?????? ?????

Я просто забыл о нижнем меню. Это меню появляется на всех страницах и связывает страницы, своего рода «внешнее» дерево сайта.



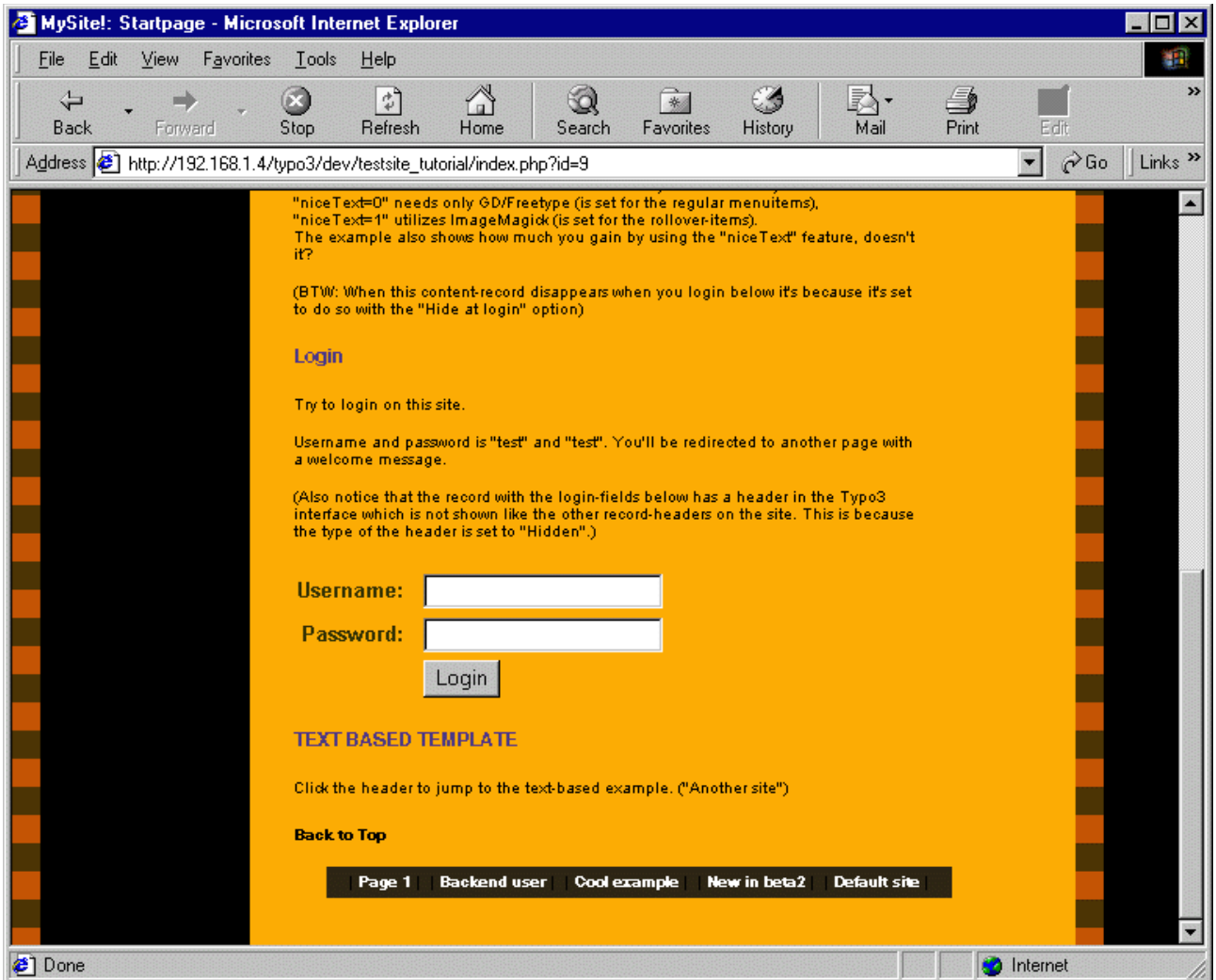
Но не стоит беспокоиться, мы также в состоянии создать эти меню. Посмотрев в шаблон Golive, мы обнаружим HTML комментарий `###БОТТОМ_МЕНЮ###`, который должен быть заменен. Единственная вещь выпавшая из поля зрения – это спланировать какие страницы должны здесь появиться. Для выполнения этого, нам необходимо придумать страницу которая будет их «содержать». Анализируя структуру, мы можем использовать страницы «Another site in ...» в testsite. Для этого нам необходимо найти uid-номер этой страницы, который показывается при наведении указателя мыши на иконку страницы. Итак, "id=10".



Теперь добавим этот код к другим меню:

```
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;
}
```

Вот так!



# ??

## ???? 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 =
```

## ???? setup:

```
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">

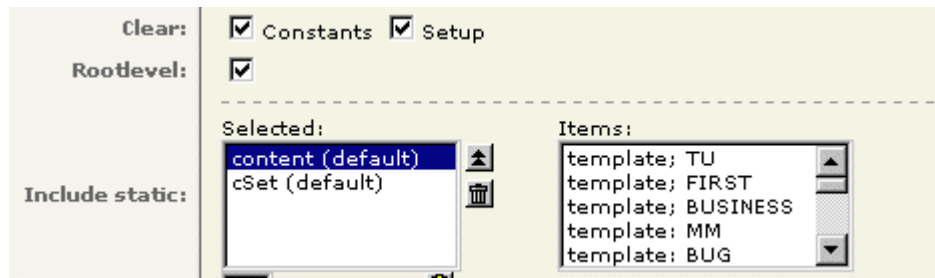
page.10 = TEMPLATE
page.10 {
  template = FILE
  template.file = fileadmin/tmplpage.html
  workOnSubpart = DOCUMENT_BODY
  subparts.CONTENT < styles.content.get

  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;
  }
}
```

Флаги, которые должны быть установлены и подключенные статические шаблоны:



Список записей для страницы "Startpage". Отметим, что мы сделали новый шаблон!