



СИСТЕМА УПРАВЛЕНИЯ САЙТОМ  
С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ

# РУКОВОДСТВО

(пошаговое) по созданию шаблонов для  
Joomla 1.5

Уровень подготовленности читателя: начальный

<http://www.joomla.ua.com> - Документация к Joomla на русском языке

Мы рады принять все замечания и предложения по нашим переводам документации.  
Основной сайт: <http://www.joomlaua.com>

Все продукты и названия компаний, зарегистрированные в качестве торговых марок, принадлежат их владельцам.  
Joomla: Copyright © 2007 Joomla Project.

Перевод и публикация: Copyright © 2007 WWW.JOOMLAUA.COM

Информация, предложенная в этом документе, может использоваться только для ознакомления с Joomla и не может рассматриваться как обязательство Проекта. Проект Joomla оставляет за собой право обновления или изменения данно-го документа. Проект не подразумевает несение ответственности за любые ошибки или неточности, которые могут быть найдены в настоящем документе.

## **Создание шаблонов Joomla!**

Для того, чтобы изменить внешний вид сайта и привести его в соответствие и привести его в соответствие с вашими требованиями, необходимо создать новый шаблон или изменить уже имеющийся. В этой статье вы узнаете об основах создания шаблонов в Joomla!

### **Корпоративная индивидуальность**

Корпоративное индивидуальность (Corporate Identity или CI) – это отличительные признаки внешнего представления компании, традиционно сложившиеся за время существования компании или специально разработанные для нее.

Корпоративная индивидуальность включает:

- Корпоративный образ (цены, продукция, рекламная стратегия)
- Корпоративный дизайн (внешнее представление)
- Корпоративное общение
- Корпоративное поведение (отношение сотрудников друг к другу и к клиентам)

Данные аспекты корпоративной индивидуальности следует принять во внимание во время разработки сайта. В этой статье мы рассмотрим Корпоративный дизайн. Как минимум он состоит из логотипа, основного шрифта, и основных цветов компании. Пользователь должен «узнать» компанию впервые зайдя на ее сайт.

# HTML/XHTML, CSS, XML

Аббревиатуры HTML/XHTML, CSS и XML – это Интернет технологии применяемые Joomla! Данные технологии стандартизированы консорциумом W3C (World Wide Web Consortium).

## HTML/XHTML

WWW (World Wide Web или Всемирная паутина) основана на HTML. HTML – это не язык программирования. HTML является языком описания и форматирования текста. Текст при этом состоит из различных структур: заголовков, списков, частей, выделенных жирным или наклонным шрифтом, таблиц и т.д. HTML работает с помощью так называемых Тегов (tags). Тег имеет открывающую и закрывающую формы. Например, заголовок первого уровня может выглядеть следующим образом:

```
<h1> Это заголовок </h1>
```

Теги интерпретируются пользовательскими веб обозревателями (браузерами) и отображаются согласно их значению. HTML прост в изучении, а онлайнные пособия по данному языку широко доступны. Следующим этапом развития HTML стал XHTML версии 1.0

## CSS

Каскадные листы стилей (Cascading Style Sheets или CSS) – это расширение HTML. CSS – это также не язык программирования. CSS можно рассматривать как своеобразный словарь свойств форматирования различных элементов HTML страницы.

С помощью CSS команд можно, например, определить, что заголовки первого уровня будут иметь размер 18 точек, использовать шрифт Arial, и иметь вертикальный отступ от следующей строки в 1.9 см. Внедрение таких опций форматирования не доступно при использовании «чистого» HTML. Необходимости в них во время разработки HTML не существовало. Однако, с растущей коммерциализацией Интернета дополнительные возможности форматирования становились все более важны.

Данные CSS можно интегрировать в HTML следующими путями:

**В центральный HTML файл:** Команды CSS могут быть определены в заглавной секции HTML файла следующим образом:

```
<head>
<title>title of the file</title>
<style type="text/css">
<!--
/* ... тут определяются css команды ... */
-->
</style>
</head>
```

**В отдельном CSS файле:** если команды CSS должны применяться к нескольким HTML страницам, целесообразнее хранить их в отдельном файле, прописывая путь к нему в заглавных секциях HTML файлов. Именно такой подход используется Joomla!:

```
<head>
<title>title of the file</title>
<link rel="stylesheet" type="text/css" href="formate.css">
</head>
```

**Внутри HTML тега:** Команды CSS также могут быть интегрированы в HTML теги:

```
<body>
<h1 style="... CSS command ...">...</h1>
</body>
```

Данные методы могут быть без особых проблем совмещены в одном CSS файле. Например, CSS команды, указанные в заглавной части HTML файла, могут быть заменены CSS командами, записанными внутри тегов, имеющими более высокий приоритет исполнения. Однако, практика совмещения различных методов внедрения CSS как правило приводит к появлению сложных и неудобочитаемых конструкций исходного кода. Поэтому рекомендуется использовать один из вариантов, например, размещать CSS команды только в отдельном файле.

## XML

Расширенный язык разметки (Extended Markup Language или XML) – это обширный и многофункциональный мета-язык, разработанный на основе более сложного SGML (Standardized Generalized Markup Language). Зачастую XML используется для написания различных файлов настроек или обменных файлов для передачи информации между объектами. Применительно к Joomla!, XML используется как язык описания метаданных создаваемых шаблонов. Эти метаданные важны для работы Инсталлятора Шаблонов (Template Installer) и Менеджера Шаблонов (Template Manager).

Метаданные в XML файлах представляют собой текстовые элементы с открывающими и закрывающими тегами. Например:

```
<name>Joomla_book</name>
```

В отличие от HTML в XML нет predefined тегов. Пользователь может самостоятельно организовывать структуру и именовать теги.

## Создание собственных шаблонов

Теперь сфокусируемся на том, как создать собственный шаблон. Перед началом работы над шаблоном необходимо обдумать некоторые ключевые моменты. Рассмотрим их по порядку.

### Концепция

Перед началом работы необходимо определить концепцию. Для графического воплощения концепции целесообразно создать набросок или диаграмму шаблона. При этом вы вольны воспользоваться любыми удобными для вас средствами. Это может быть Adobe Photoshop, Microsoft Paint, любой другой графический редактор или просто листок бумаги.

### Фиксированный или изменяемый размер

Вы можете создать два типа шаблонов. Шаблоны, адаптирующие свой размер под размеры окна веб обозревателя или шаблоны фиксированного размера.

Пример «гибкого» шаблона: если горизонтальное разрешение вашего экрана 2048 пикселей и окно веб обозревателя развернуто, шаблон «растянется» до размеров окна. Однако полученное при этом изображение может выглядеть несколько странно, особенно, если вы используете графические или немасштабируемые элементы, как логотипы или подписи.

Второй вариант – это определиться с разрешением шаблона, и размещать элементы на шаблоне, используя пиксельные координаты. Преимуществом данного метода является то, что вы всегда можете быть уверены, что шаблон отобразится правильно не зависимо от разрешения монитора пользователя. Хотя, если разрешение пользовательского монитора 1400 на 1050 пикселей а размер шаблона 800 на 600, он будет несколько «теряться» в окне веб обозревателя.

Однозначного решения, какой из типов шаблона использовать, нет. Если вы предпочитаете фиксированный размер, стоит выбрать размер удобный для любого монитора (800 на 600 пикселей). Учитывая то, что каждый веб обозреватель имеет полосу прокрутки справа, а само окно обозревателя находится в рамке, ширину шаблона можно сделать еще меньше, например 780 пикселей.

## Структура

Так как вы имеете дело с отображением структурированных данных, необходимо определиться с общим размещением. Обычно Joomla! использует структуру размещения элементов показанную ниже:

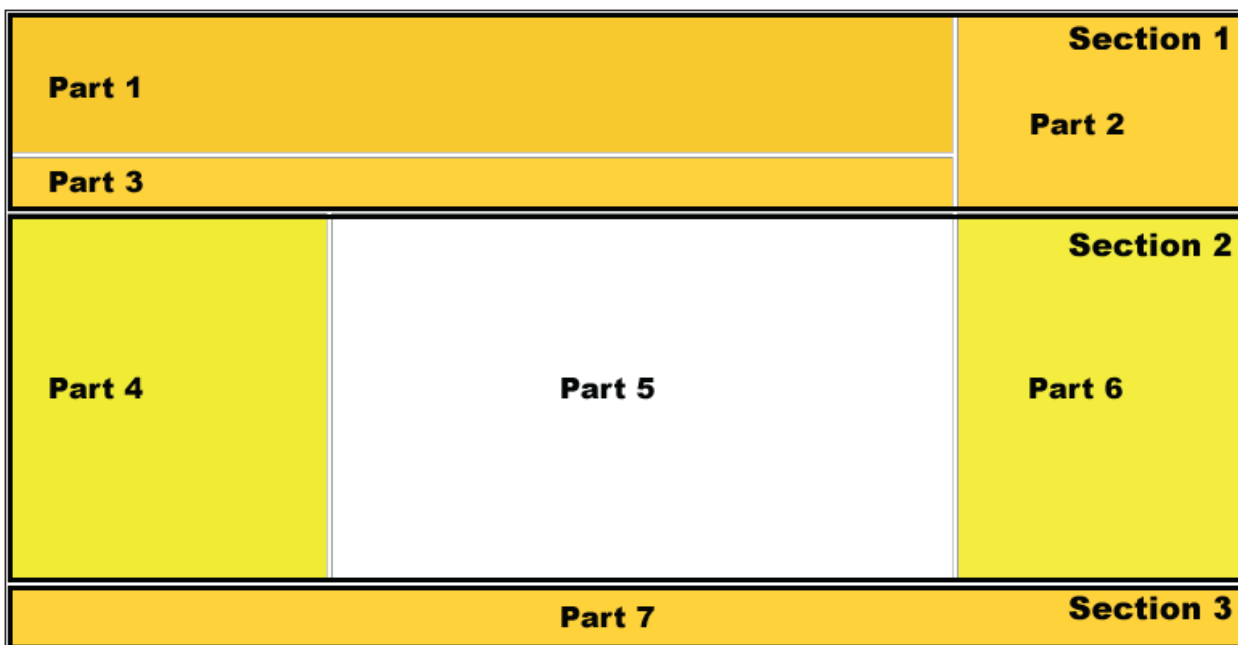


Рис. 1: Структура

Секция 1:

- Часть 1: Тут стоит разместить логотип или название сайта
- Часть 2: Тут стоит разместить поисковое поле
- Часть 3: Тут стоит разместить навигационные ссылки

Секция 2:

- Часть 4: В левой колонке размещаются самые важные меню
- Часть 5: Тут размещается основной контент сайта
- Часть 6: В правой колонке можно разместить дополнительные меню

Секция 3:

- Часть 7: Нижняя часть страницы (footer)

## Переход на HTML

Теперь необходимо превратить нашу концепцию в HTML код. В зависимости от графического редактора, который вы использовали для разработки концепции шаблона, существует вероятность, что изображение может быть конвертировано в HTML автоматически. Это также можно сделать вручную с помощью текстового редактора Notepad или HTML редактора, такого, как Macromedia Dreamweaver (<http://www.macromedia.com/software/dreamweaver>), или одного из многочисленных бесплатных HTML редакторов (<http://www.thefreecountry.com/webmaster/htmleditors.shtml>).

Исходный HTML код при этом может выглядеть таким образом:

### Листинг 1: HTML файл

```
<html>
<head>
<link rel="stylesheet"
href="templates/_system/css/general.css" type="text/css" />
<link rel="stylesheet"
href="templates/{TEMPLATE}/css/template.css" type="text/css" />
</head>
<body bgcolor="#FFFFFF" text="#000000">
<table width="780" border="1">
<!-- Section 1 -->
<tr>
<!-- Part 1 -->
<td colspan="2" height="89" bgcolor="#F5C228">&nbsp;&nbsp;&nbsp;</td>
<!-- Part 2 -->
<td width="178" height="120" rowspan="2" bgcolor="#FFCC33">
&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<!-- Part 3 -->
<td colspan="2" height="33" bgcolor="#FFCC33">&nbsp;&nbsp;&nbsp;</td>
</tr>
<!-- Section 2 -->
<tr>
<!-- Part 4 -->
<td width="197" height="233" bgcolor="#F5EE28"> &nbsp;&nbsp;&nbsp;</td>
<!-- Part 5 -->
<td width="389" height="233">&nbsp;&nbsp;&nbsp;</td>
<!-- Part 6 -->
<td width="178" height="233" bgcolor="#FFFF33"> &nbsp;&nbsp;&nbsp;</td>
</tr>
<!-- Section 3 -->
<tr bgcolor="#FFCC33">
<!-- Part 7 -->
<td colspan="3" height="40">&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
</body>
</html>
```



## Листинг 2: CSS файл

```
body {  
  font-family: Arial, Helvetica, Sans Serif;  
}
```

Данный код несколько упрощен, а его заглавная часть (<head>...</head>) не совсем отвечает стандарту XHTML. Также таблица имеет визуальную границу между ячейками (атрибут border=1), позволяющую легче увидеть структуру шаблона. Теперь вы можете максимально воспользоваться своим воображением и креативными способностями в плане подбора цветовой гаммы и логотипов. Этот код должен быть сохранен в файле index.php, именно из файла с таким названием Joomla! проводит чтение команд и скриптов шаблона.

Далее, в заглавную часть кода интегрируются стандартный CSS файл системы Joomla! и индивидуальный CSS файл данного шаблона. На данном этапе CSS файл содержит только одну команду, определяющую тип шрифта.

Первый шаблон вам придется создать полностью вручную в соответствующей директории Joomla! Затем его можно запаковать в архив, которым, с помощью Инсталлятора Joomla! (Joomla! Installer), смогут воспользоваться другие пользователи. Сохраните файл разметки index.php в только что созданной директории [ПутьКJoomla!]/templates/joomla150\_book/. Сохраните файл template.css в [ПутьКJoomla!]/templates/joomla150\_book/css/.

Основная структура вашего шаблона готова. Теперь необходимо определить ее с помощью XHTML файла, который Joomla! сможет отображать в административной секции.

## Структура директорий шаблона

Теперь необходимо позаботиться о кое-каких условиях. Как уже говорилось, шаблон должен иметь определенную структуру директорий:

```
[ПутьКJoomla!]/templates/[НазваниеШаблона]/  
[ПутьКJoomla!]/templates/[НазваниеШаблона]/css/  
[ПутьКJoomla!]/templates/[НазваниеШаблона]/images/
```

Название шаблона не должно содержать пробелов или каких-либо специальных символов. При инсталляции шаблона в виде заархивированного файла, Установщик Шаблонов (Template Installer) использует это название для создания структуры директорий шаблона. Также, при использовании некоторых операционных систем проблемы может вызвать слишком «экзотичное» (с использованием букв нелатинского алфавита) название шаблона. И в завершение, отметим также, что название шаблона должно иметь смысл, например, название нашего тестового шаблона joomla150\_book.

В директориях шаблонов должны присутствовать следующие файлы:

**Файл структуры:** это базовая версия HTML файла, который мы создали ранее: /templates/joomla150\_book/index.php. Данный файл должен иметь расширение .php, так как динамические модули Joomla!, которые мы будем использовать позднее интерпретируются именно языком PHP.

**Картинка предварительного просмотра:** файл /templates/joomla150\_book/template\_thumbnail.png содержит картинку предварительного

просмотра шаблона, отображаемую в Менеджере шаблонов (Extensions | Template Manager) административной секции Joomla! Размер картинке обычно не превышает 227 на 162 пикселя. Данный файл можно создать и позднее, когда работа над шаблоном будет закончена.

**Метаданные шаблона:** файл /templates/joomla150\_book/templateDetails.xml представляет собой инструкцию по автоматической сборке (распаковке) шаблона, используемую Установщиком шаблонов (Template Installer). Здесь можно указать путь для копирования файлов, имя автора и дополнительные метаданные. Во время установки шаблона, Joomla! считывает информацию из этого файла и копирует файлы и данные в соответствующие директории. Для тестового шаблона можно использовать файл из листинга 3, разместив в нем свою информацию. Для любого файла, используемого в шаблоне, необходимо создать соответствующий XML элемент в templateDetails.xml:

```
<files>
<filename> ... укажите название файла в корневой директории шаблона ...
</filename>
<filename> ... и так для каждого файла шаблона
</filename>
</files>
```

Другие элементы XML файла служат для описания шаблона. Вот полный листинг функций такого файла:

### Листинг 3: templateDetails.xml

```
templateDetails.xml
<install version="1.5" type="template">
<name>joomla150_book</name>
<version>0.1</version>
<creationDate>28.07.2006</creationDate>
<author>Hagen Graf</author>
<copyright>GNU/GPL</copyright>
<authorEmail>hagen@cocoate.com</authorEmail>
<authorUrl>http://www.cocoate.com</authorUrl>
<version>0.1</version>
<description>... Описание</description>
<files>
<filename>index.php</filename>
<filename>templateDetails.xml</filename>
<filename>template_thumbnail.png</filename>
<filename>css/template.css</filename>
</files>
</install>
```

**CSS файл:** В одном шаблоне можно использовать несколько CSS файлов. Название CSS файла и метод его создания не имеют особого значения. Хотя существуют стандартные описания различных элементов CSS. Для нашего тестового шаблона мы используем файл /templates/joomla150\_book/css/template.css (Листинг 2).

**Графика, Картинки:** здесь можно разместить графические файлы, необходимые для шаблона. Во время инсталляции, Установщик шаблонов копирует их в директорию /images. Имя и путь к файлу при этом выглядят так: /templates/joomla150\_book/images/[user-defined image files].

## Первый запуск

После того, как мы создали структуру директории [ПутьКJoomla!]/templates/, наш шаблон появится в соответствующей секции административной части Joomla! (Extensions | Template). Теперь его можно использовать как шаблон по умолчанию.

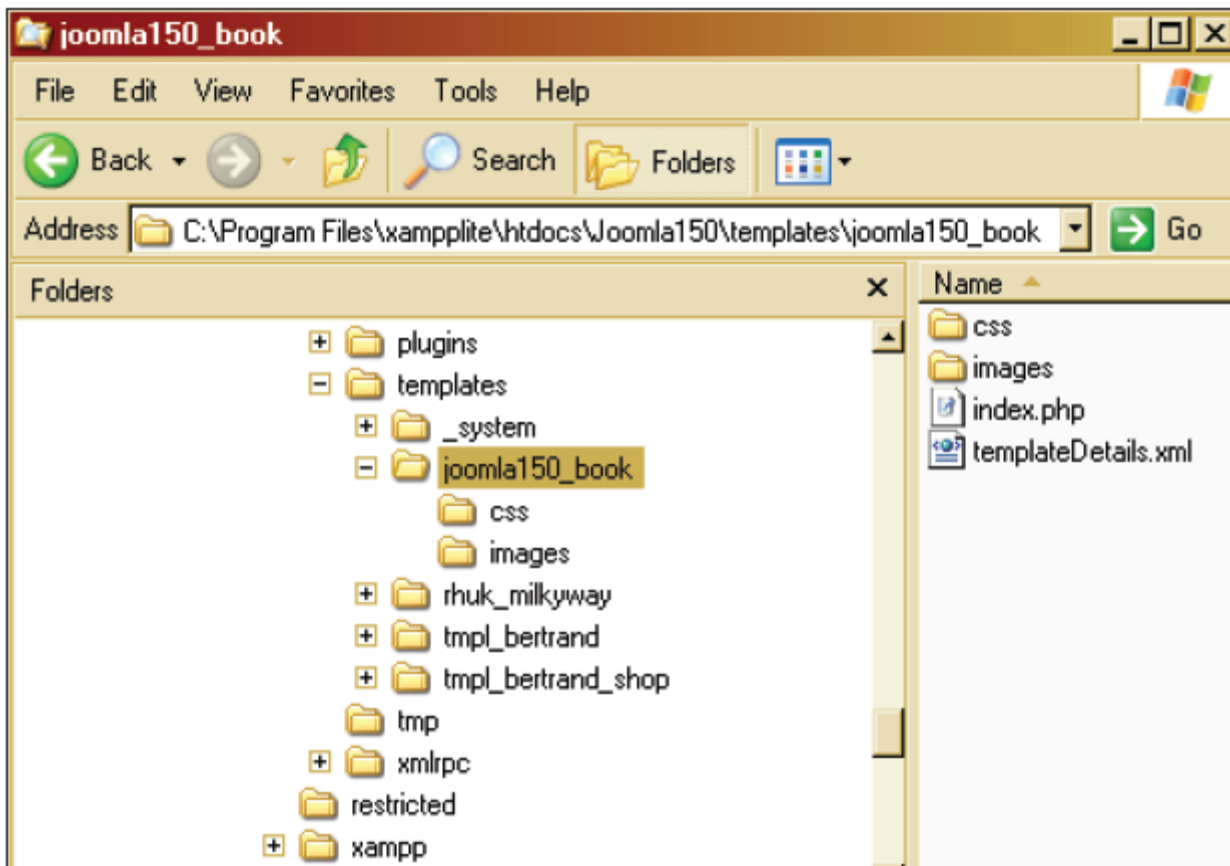


Рис. 2: Структура директории и файлы шаблона Joomla150\_book



Рис. 3: Новый шаблон в менеджере шаблонов

Если вы загрузите свой сайт в веб обозревателе, вы увидите новый шаблон. К сожалению, он пока пуст. Теперь нам предстоит интегрировать контент в шаблон элемент за элементом.

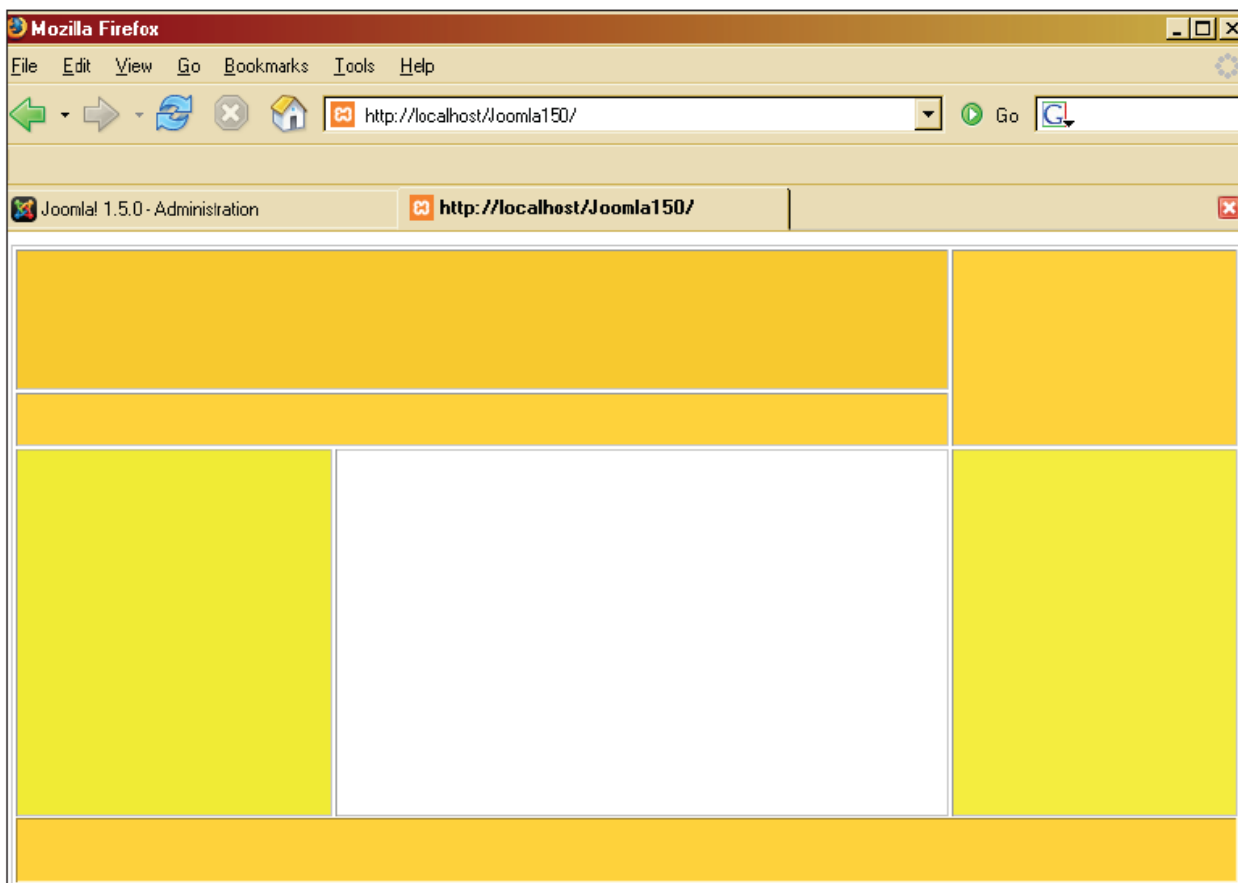


Рис. 4: Вид в веб обозревателе

## Внедрение модуля Joomla!

Внедрение модуля Joomla! проводится с помощью команд, встроенных в HTML код. Это могут быть и старые команды Joomla! 1.0.x или Mambo 4.5.x, или новые команды patTemplate. Класс patTemplate служит для дифференцирования PHP и HTML кода. Для интеграции в шаблон различных элементов, Joomla! использует объект типа jdoc. Если, вместо тега <title>, вы вставите в заголовок файла index.php следующую строку, в окне веб обозревателя при загрузке вашего сайта корректно отобразятся иконка сайта, заголовок страницы и символ источника новостей (Newsfeed symbol):

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

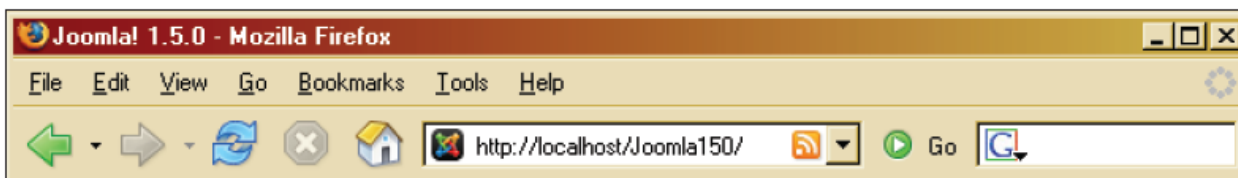


Рис. 5: Иконка сайта и заголовок страницы

Если вы откроете исходный код этой страницы, то заметите, что Joomla! копировала все метаданные, введенные через панель администратора в HTML код. В дополнение к этому источники RSS также были интегрированы в страницу в виде тегов ссылок, отображаемых как символы источников новостей (Newsfeeds) в веб обозревателях, поддерживающих данную технологию (Firefox, Opera и т.д.).

## Листинг 4: Метаданные Joomla!

```
...
<head>
<title>Joomla 1.5.0 - Home</title>
<meta name="generator" content="Joomla! 1.5" />
<meta name="description" content="Joomla! - the dynamic portal
engine and content management system" />
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<meta name="robots" content="index, follow" />
<meta name="keywords" content="joomla, Joomla" />
<link
href="http://localhost/Joomla150/feed.php?option=com_frontpage&
Itemid=1&format=rss" rel="alternate" type="application/rss+xml"
title="RSS 2.0" />
<link
href="http://localhost/Joomla150/feed.php?option=com_frontpage&
Itemid=1&format=atom" rel="alternate"
type="application/atom+xml" title="Atom 1.0" />
<link href="favicon.ico" rel="shortcut ico" rel="shortcut icon"
type="image/x-icon" />
</head>
...
```

Если данный фрагмент кода работает, можно перейти к другим важным объектам `jdoc`. Например, команда `<jdoc:include type="modules" name="top" style="-1" />` получает тип вставляемого объекта в качестве параметра (в данном случае это “modules”). Параметр “name” определяет положение модуля. (top, right, left, user1 и т.д.). Установить положение модуля можно также в административной части Joomla! (Extensions | Module Managers, категория Position). Список всех возможных позиций можно посмотреть в Extensions | Templates | Module Positions. Последний параметр “style” содержит значение, которое определяет тип HTML кода, передаваемый модулем. Например, значение -1 передает «чистый» HTML без использования контейнерных тегов `<div>`.

Параметр	Отображение
1	Горизонтальное меню
-1	«Чистое» отображение без использования начального и замыкающего тегов <code>&lt;div&gt;</code>
-2	Отображение в виде XHTML
-3	Опция форматирования круглых углов

Таблица 1: Параметры модуля

Описание возможных вставок в `index.php` объектов `jdoc` можно увидеть в следующем листинге:

## Листинг 5: `index.php` с `jdoc` вставками

```
<html>
<head>
<jdoc:include type="head" />
<link rel="stylesheet" href="templates/_system/css/general.css"
type="text/css" />
<link rel="stylesheet"
href="templates/{TEMPLATE}/css/template.css" type="text/css" />
```

```

</head>
<body bgcolor="#FFFFFF" text="#000000">
<table width="780" border="1">
<!-- Section 1 -->
<tr>
<!-- Part 1 -->
<td colspan="2" height="89" bgcolor="#F5C228">&nbsp;&nbsp;&nbsp;</td>
<!-- Part 2 -->
<!-- Section 3 -->
<tr bgcolor="#FFCC33">
<!-- Part 7 -->
<td colspan="3" height="40">
<jdoc:include type="modules" name="footer" style="-1" />
</td>
</tr>
</table>
</body>
</html>

```

Если вы загрузите ваш сайт в веб обозревателе теперь, то увидите что благодаря изменениям, внесенным нами в HTML код, сайт наполнился динамическим контентом. Конечно, до визуального совершенства еще далеко, но сайт уже работает:

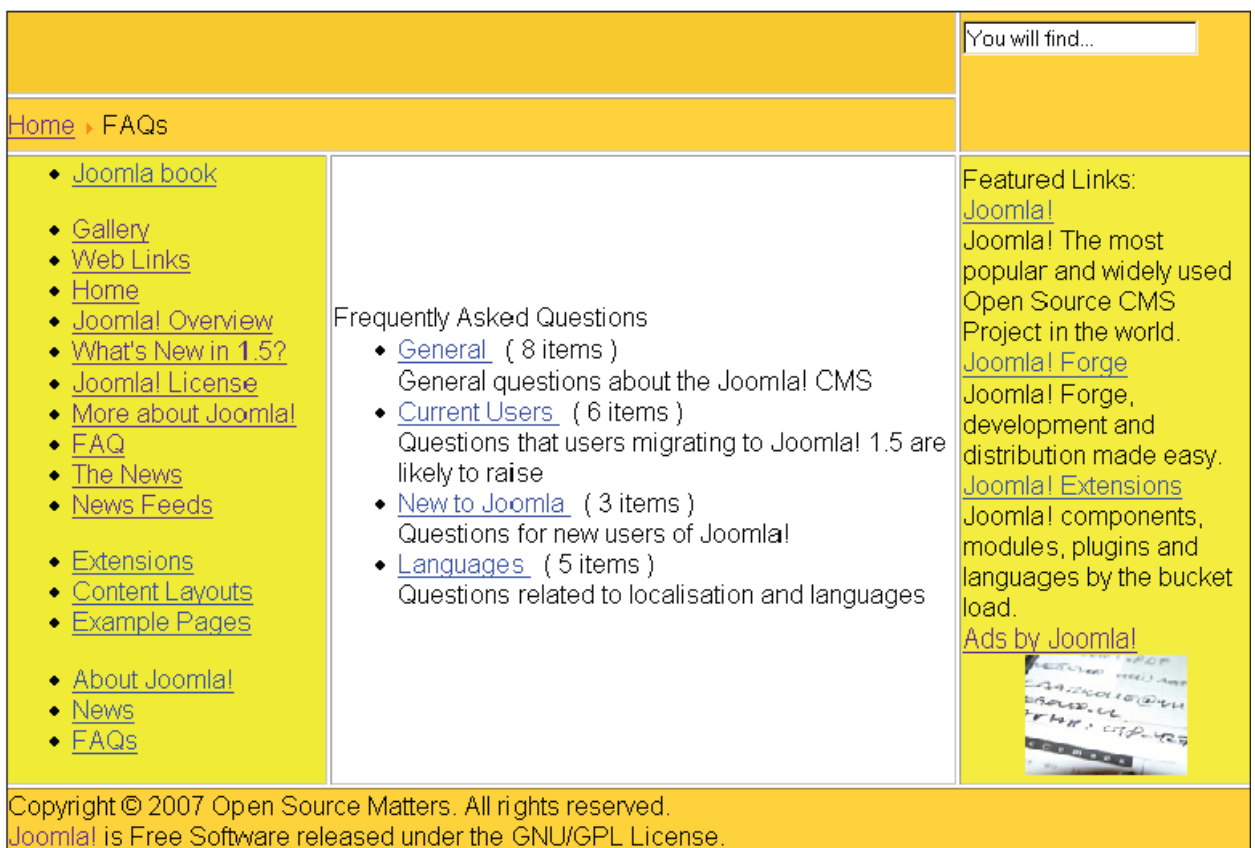


Рис. 6: Шаблон с динамическими данными

Теперь перейдем к CSS форматированию, чтобы улучшить внешний вид нашего сайта. Копируйте следующий код в файл template.css. В данном фрагменте кода мы определяем, что шрифтом по умолчанию будет Arial, ссылки не будут подчеркиваться, а при наведении на них курсора изменят цвет и толщину букв.

## Листинг 6: template.css

```
body {
font-family: Arial, Helvetica, Sans Serif;
}
a:link, a:visited {
color: #ff6600;
text-decoration: none;
font-weight: bold;
font-size: 15px;
}
a:hover {
color: #C43C03;
text-decoration: none;
font-weight: bold;
font-size: 15px;
}
```

Эти изменения значительно улучшат внешний вид нашего сайта. На рисунке внизу курсор мыши указывает на ссылку «More About Joomla!», шрифт при этом меняет цвет и толщину:

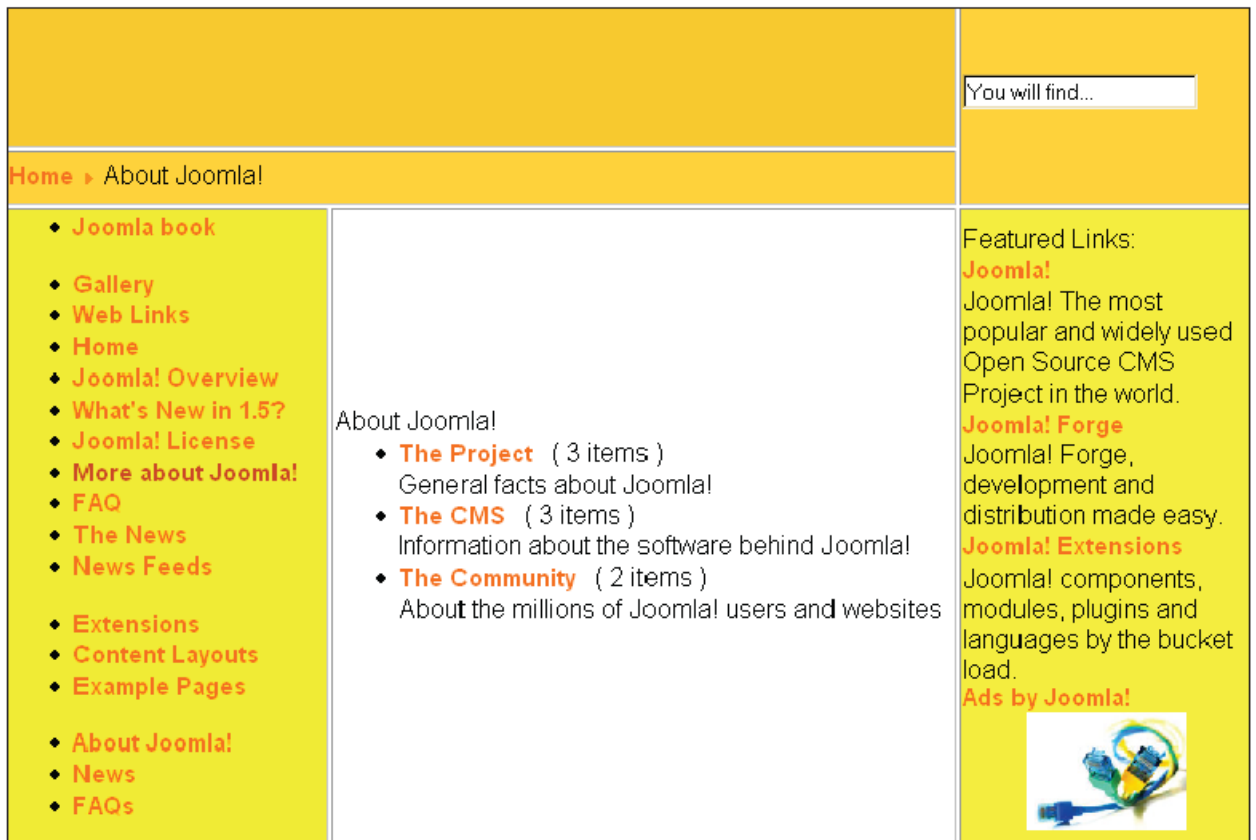


Рис. 7: Шаблон с CSS файлом

Данный шаблон состоит из HTML таблиц. Однако, в настоящее время рекомендуется избегать использование HTML таблиц, так как они ограничивают возможности разметки страниц и размещения на них объектов контента. В дополнение к этому, на нашем сайте пока нет ни одной картинки. Тем не менее, основные вопросы создания шаблонов Joomla! нами решены. Остается только заархивировать шаблон в установочный пакет.

## Создание пакета шаблона

Для того, чтобы вашим шаблоном смогли воспользоваться другие, необходимо запаковать его в zip архив. Перед этим следует создать картинку предварительного просмотра шаблона (template\_thumbnail.png) размером 227 на 162 пикселя. Эта картинка отобразится при наведении курсора мыши на название шаблона в **Менеджере шаблонов (Template Manager)**:



Рис. 8: Картинка предварительного просмотра шаблона joomla150\_book

Теперь запакуйте все файлы шаблона joomla150\_book в zip архив, сохраняя структуру каталогов. Также выберите все файлы и папки директории **[ПутьКJoomla!]/templates/joomla150\_book/** и запакуйте их в архив joomla156\_book.zip. Создайте резервные копии этих архивов.

Теперь можно передавать пакет шаблона другим пользователям или устанавливать его самому. Если вы хотите попробовать установить шаблон из пакета, необходимо сначала удалить уже имеющийся шаблон joomla150\_book, однако перед удалением нужно установить другой шаблон по умолчанию. В главном меню административной части Joomla! выберите **Extensions | Template Manager**. В появившемся окне выберите желаемый шаблон и нажмите кнопку **По умолчанию (Default)**. Теперь можно деинсталлировать шаблон. В главном меню административной части Joomla! выберите **Extensions | Install/Uninstall**. Перейдите на закладку **«Шаблоны» (Templates)**. Выберите шаблон joomla150\_book и нажмите кнопку **Деинсталлировать (Uninstall)**.



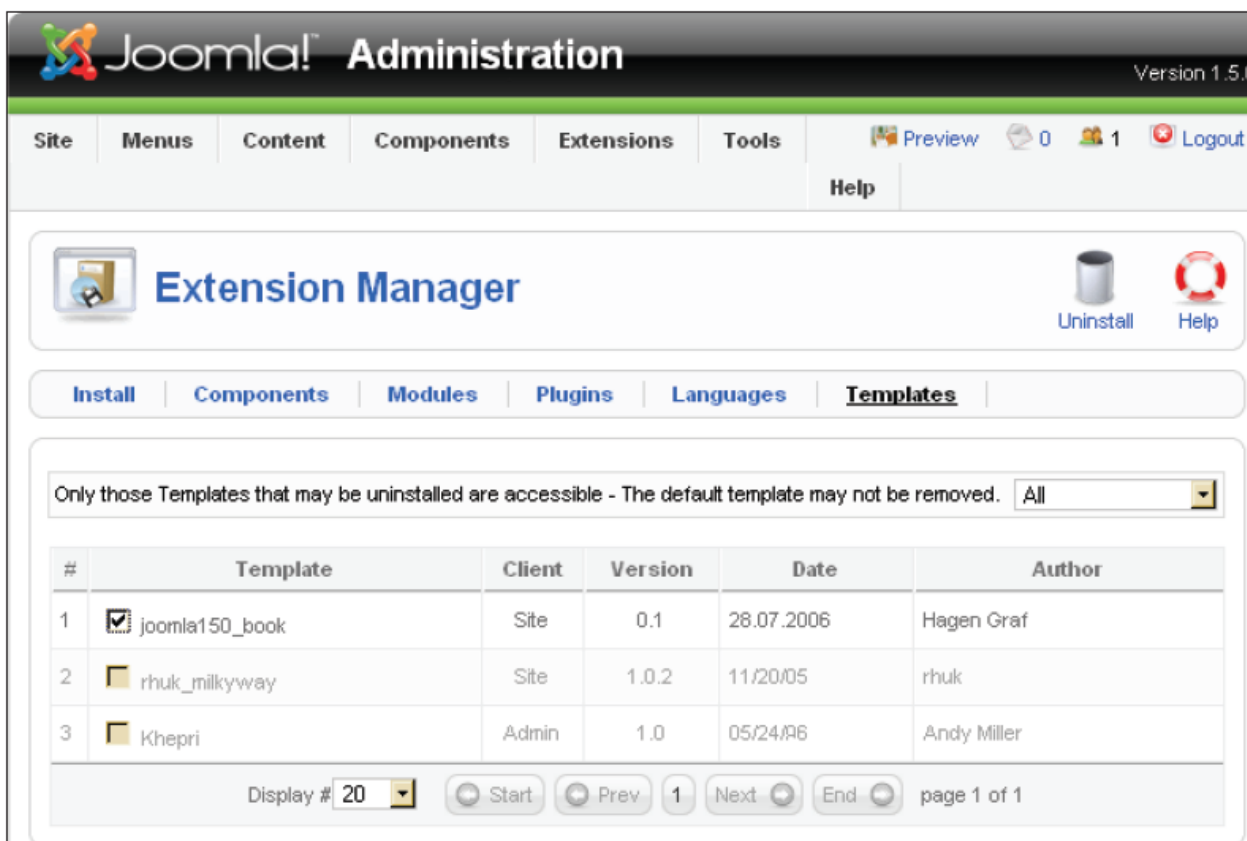


Рис. 9: Деинсталляция шаблона

Также для удаления шаблона достаточно просто удалить директорию [ПутьКJoomla!]/templates/joomla150\_book/.

## Инсталляция с помощью Установщика шаблонов (Template Installer) Joomla!

После создания пакета шаблона и удаления всех его предыдущих версий можно перейти к инсталляции нового шаблона (в главном меню админчасти выберите Extensions | Install/Uninstall).

Здесь можно установить zip пакет с вашим шаблоном. Выберите файл joomla150\_book.zip и нажмите кнопку Загрузить файл и установить (Upload File & Install). Инсталлятор сообщит об успешной установке, также на экране отобразится описание XML файла шаблона.

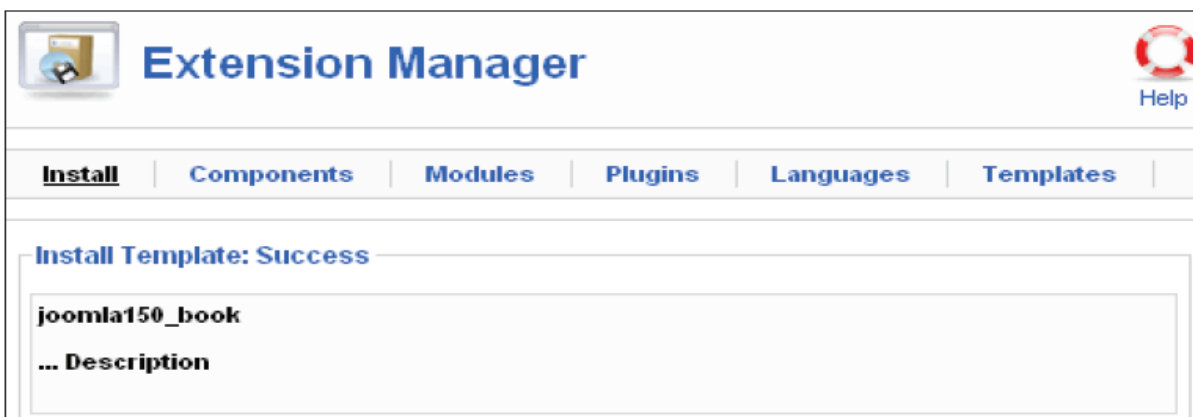


Рис. 10: Установка пакетного файла шаблона

Если вы перейдете к Менеджеру шаблонов, то увидите, что новый шаблон появился в списке и доступен для использования.

## Создание шаблоном с расширением Dreamweaver

HTML редактор Dreamweaver позволяет создавать файлы с определенным расширением. Система Joomla! способна «понимать» такие файлы, что позволяет создавать шаблон полностью в Dreamweaver. Для работы необходима версия Dreamweaver MX 2004 или выше, в котором также нужно установить Template Builder Extension.

## Установка

Загрузите файл joomlasolutions1.0.mxp

([http://developer.joomla.org/sf/frs/do/viewRelease/projects.joomlasolutions/frs.dreamweaver\\_joomla\\_template\\_exte\\_joomla\\_dreamweaver\\_ext\\_1\\_0](http://developer.joomla.org/sf/frs/do/viewRelease/projects.joomlasolutions/frs.dreamweaver_joomla_template_exte_joomla_dreamweaver_ext_1_0)). Запустите Dreamweaver, в главном меню программы выберите Команды | Управление расширениями (Commands | Manage Extensions). На экране появится Менеджер расширений Макромедиа (Macromedia Extension Manager). Выберите необходимый файл и нажмите Установить (Install):

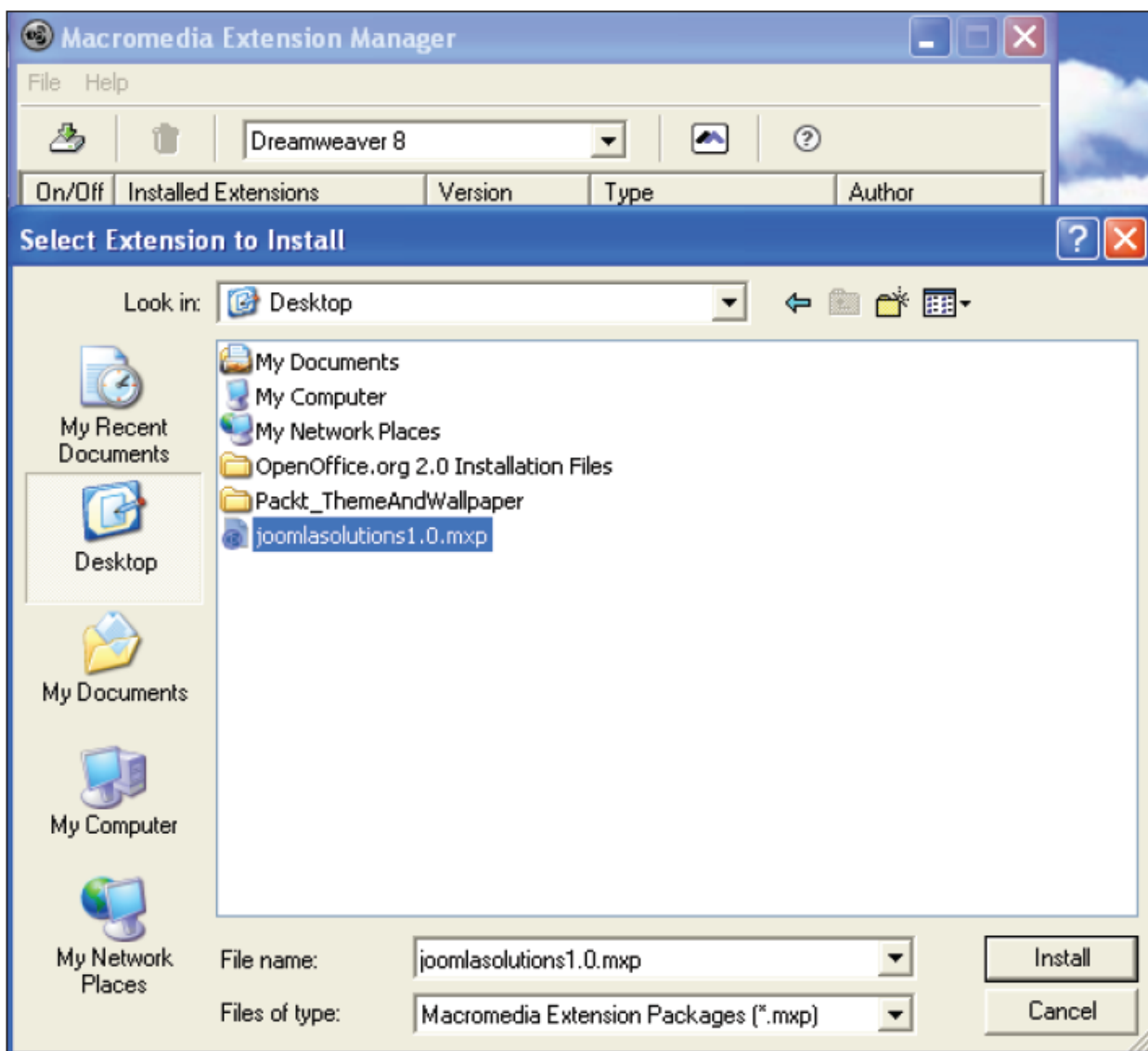


Рис. 11: Установка нового расширения в Dreamweaver

## Создание нового файла шаблона

Для активации расширения необходимо перезапустить Dreamweaver. Создайте новый документ динамического типа (Dynamic page type). Для этого в главном меню выберите **Файл | Новый | Динамическая страница | PHP** (File | New | Dynamic Page | PHP)

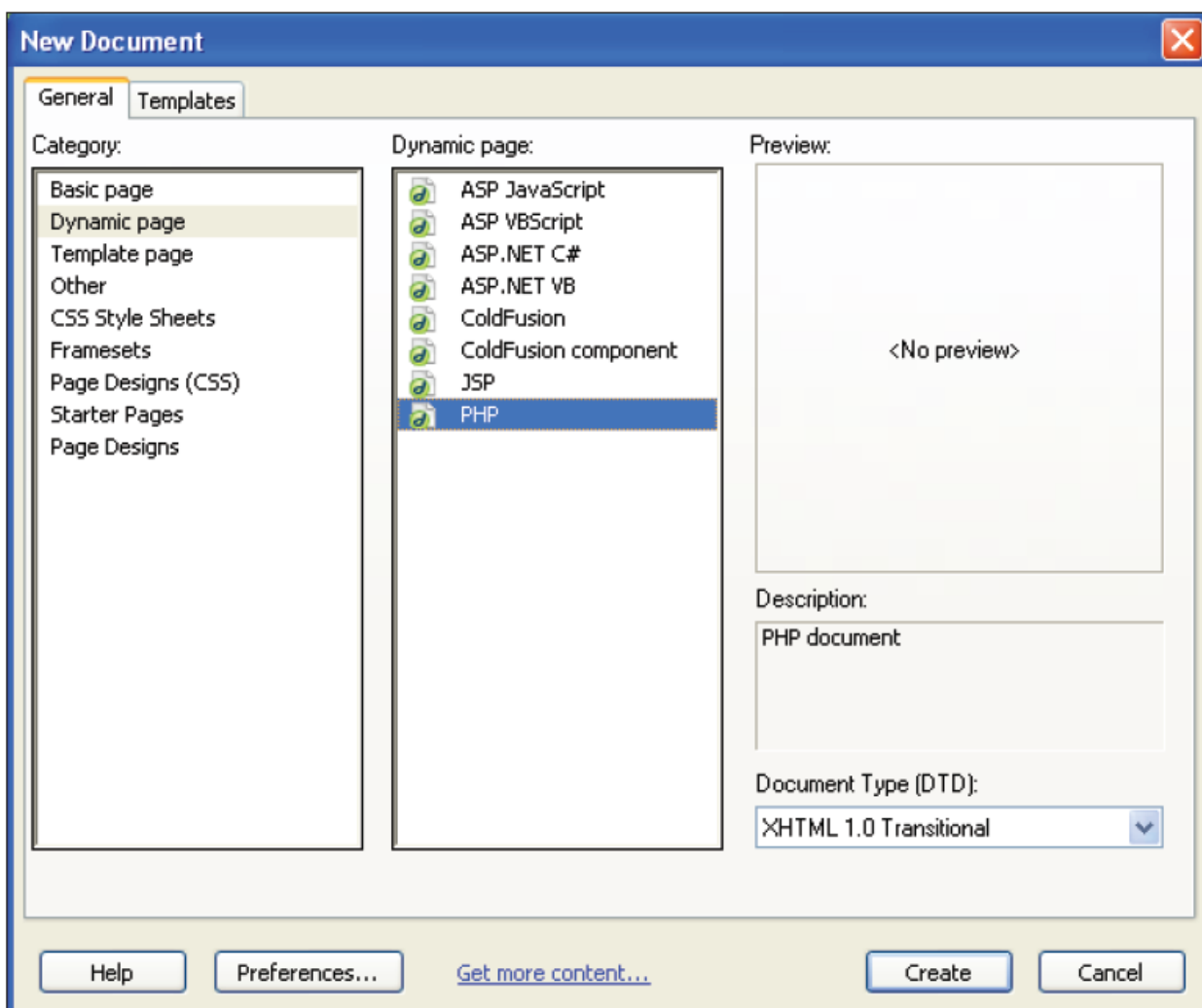


Рис. 12: Настройка нового документа Dreamweaver

После того, как файл был создан, в верхней части экрана вы увидите всплывающее меню со списком наборов инструментов для работы с различными типами файлов. Выберите элемент **Joomlasolutions1**:

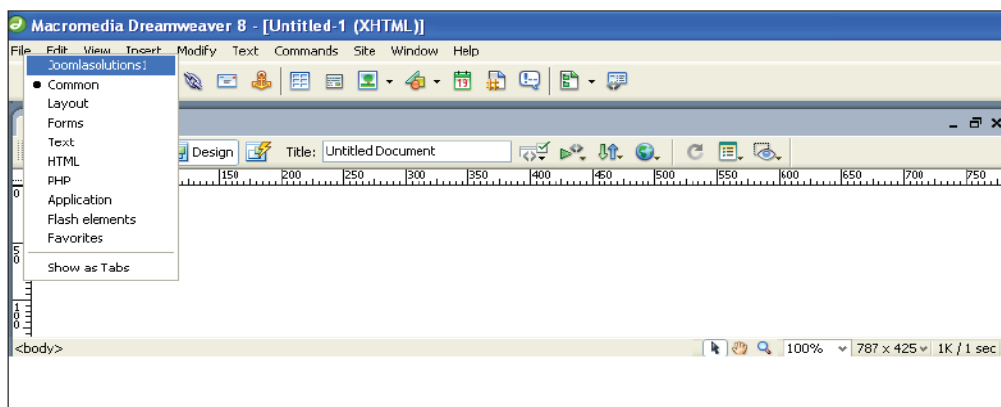


Рис. 13: Выбор набора инструментов для работы с шаблонами

Данный набор инструментов содержит кнопки, позволяющие добавлять различные элементы шаблонов. Первым делом необходимо перейти на закладку Код (Code), удалить заголовок, сгенерированный Dreamweaver, и заменить его заголовком Joomla!

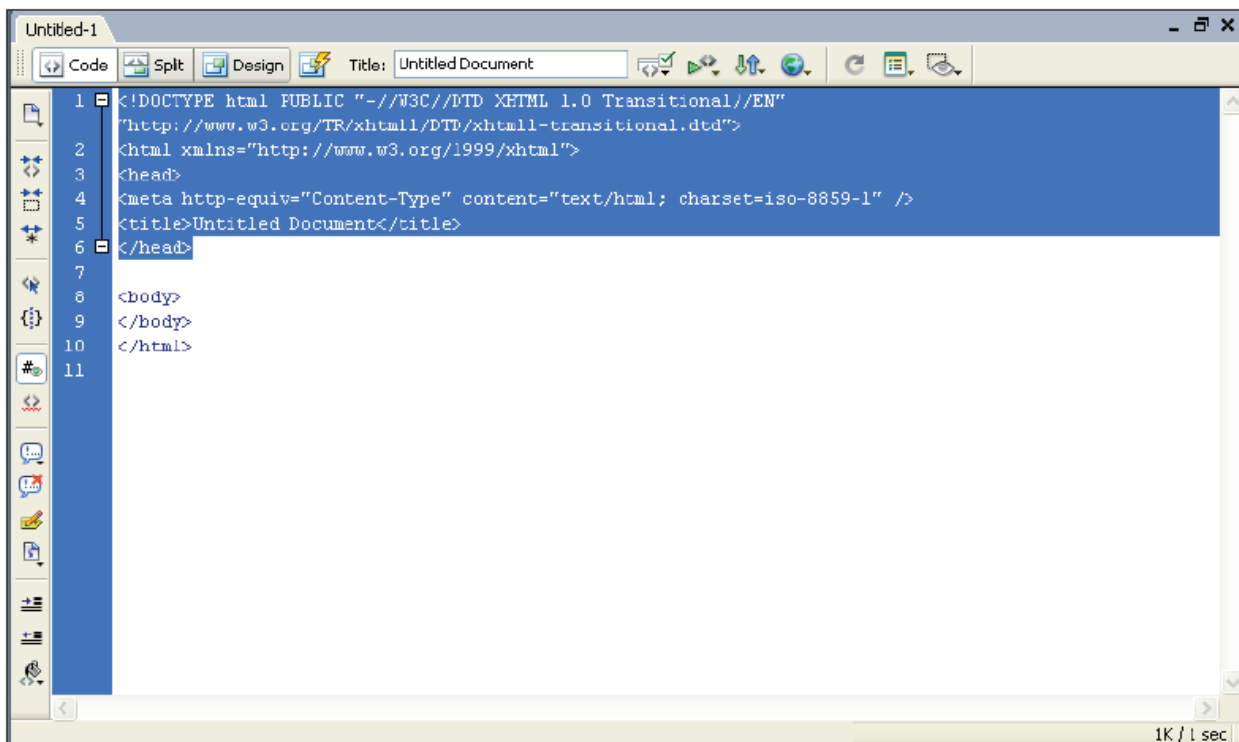


Рис. 14: Удаление заголовка

После этого нажмите на кнопку **Вставить заголовочный код (Insert Header Code)**. При этом на странице появится заголовок, типичный для шаблонов Joomla!

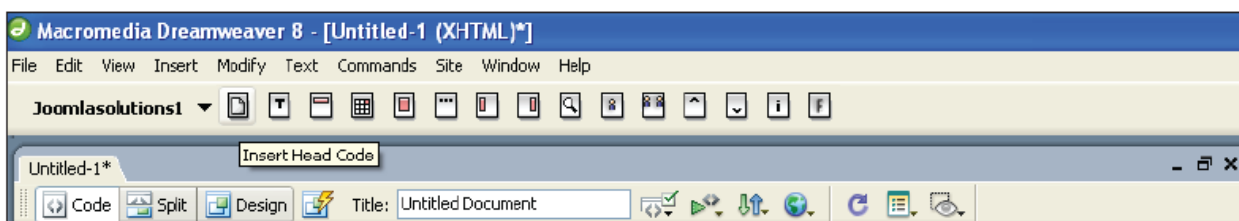


Рис. 15: Вставка заголовка Joomla!

Теперь в нашем файле имеется код заголовка, необходимый при создании шаблонов Joomla!, который ранее мы вносили вручную.

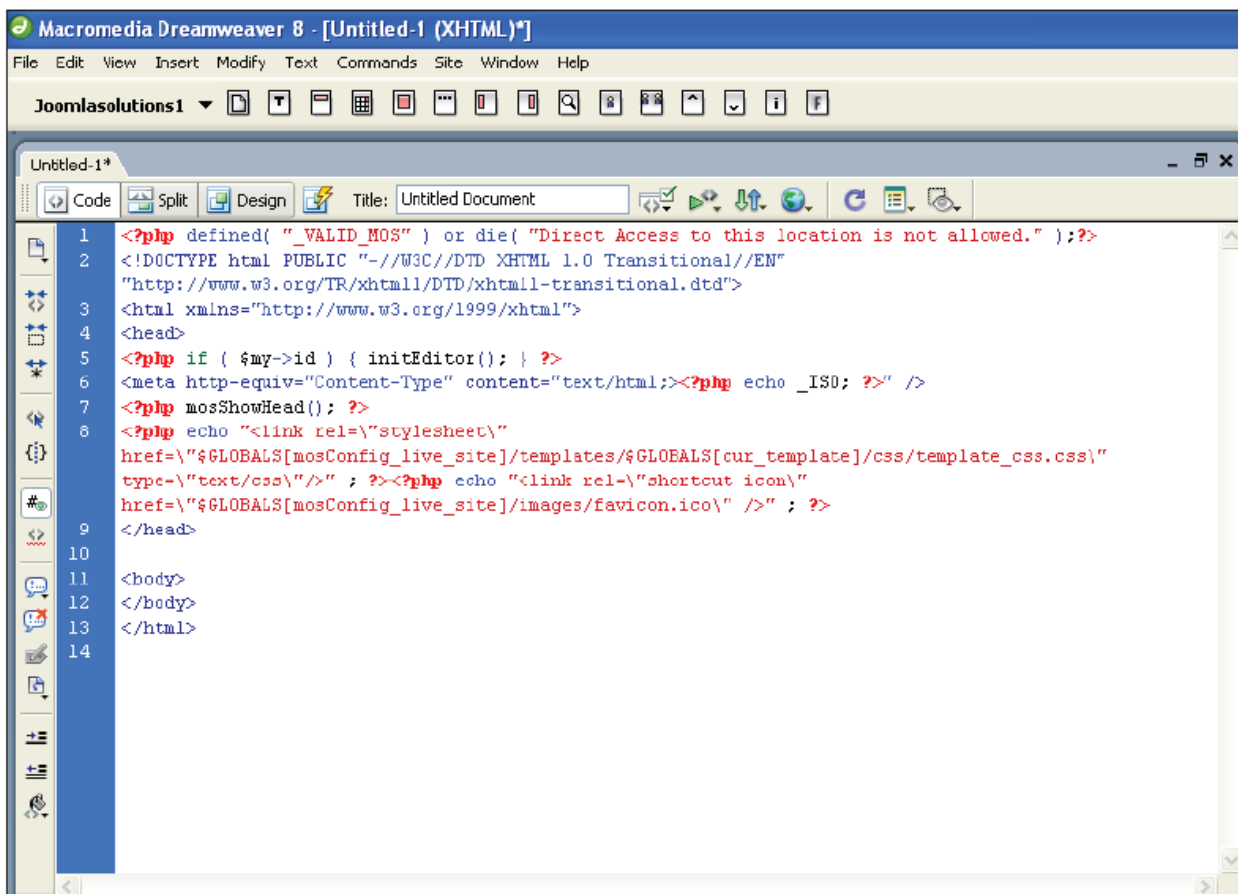


Рис. 16: Код заголовка, необходимый для Joomla!

## Структура шаблона

Помимо специального заголовка, для шаблона необходима структура. Создать структуру можно при помощи таблиц или тегов <div>. Далее мы опишем создание табличной версии структуры. Если в Dreamweaver все еще активирован режим разметки (layout mode), закройте его.

## Таблица

Чтобы создать таблицу можно воспользоваться главным меню Dreamweaver, выбрав в нем Вставить | Таблица (Insert | Table). Опции Dreamweaver позволяют создавать таблицы любой сложности и структуры. Для примера мы продемонстрируем создание простой таблицы. Выберите цвет для таблицы по своему вкусу.

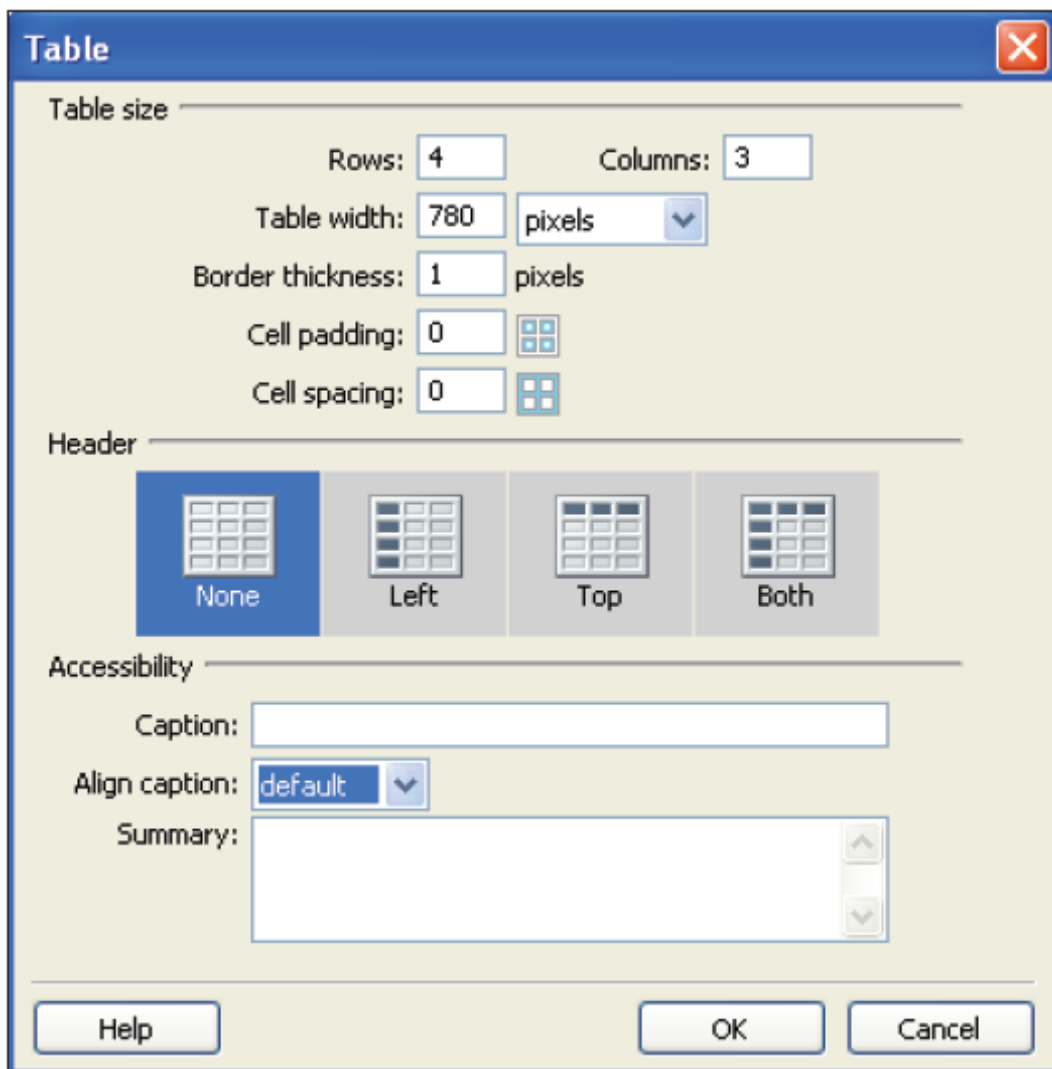


Рис. 17: Создание простой таблицы

Внеся некоторые добавления в таблицу (с.м. Листинг 5), вы получите код, аналогичный этому:

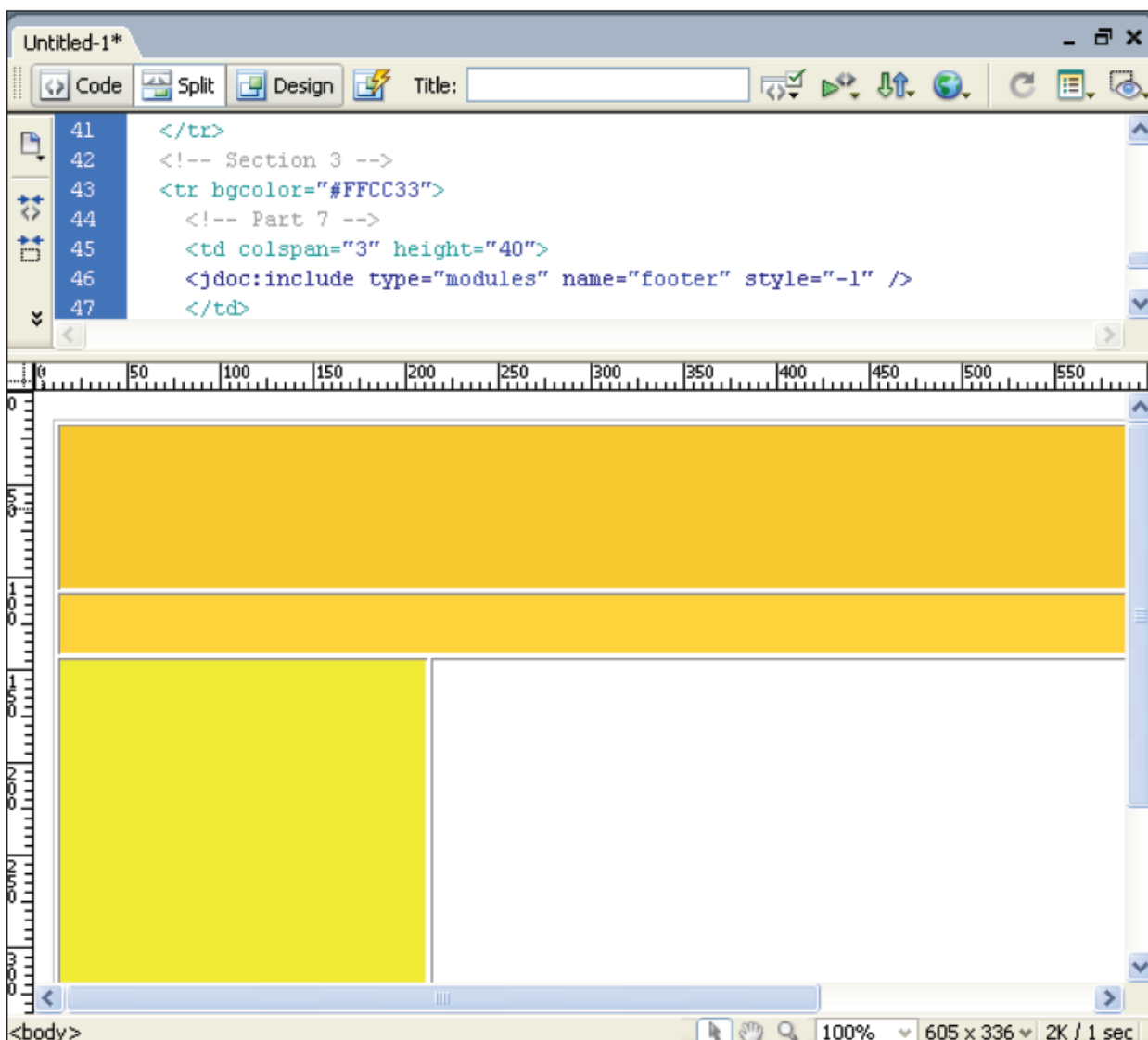


Рис. 18 Структура таблицы

Теперь необходимо сохранить файл. Создайте новую директорию для шаблона Dremweaver: [ПутьКJoomla!]/templates/joomla150\_book\_dw. Сохраните файл под названием index.php. Там же создайте поддиректории css и images.

Разместите все графические файлы, которые вы планируете использовать в данном шаблоне, в папке images. Для создания и редактирования графических объектов можно воспользоваться коммерческой программой Photoshop или бесплатной Gimp.

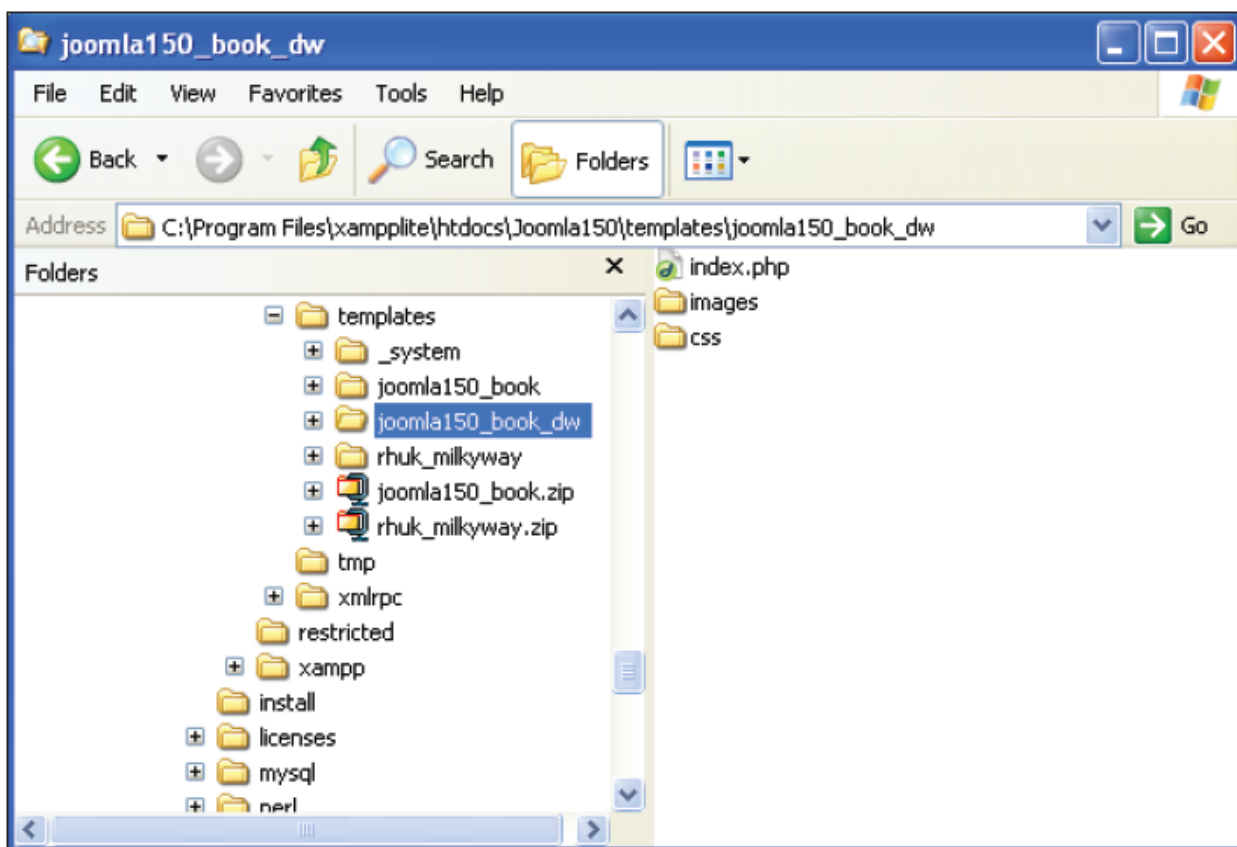


Рис. 19: Структура директорий шаблона

CSS файл необходимо разместить в директории CSS. Он должен называться `template_css`. CSS файл можно создать с помощью Dreamweaver или копировать из другого шаблона. При создании css файла в Dreamweaver, в заголовке вашего шаблона автоматически появится ссылка на этот файл:

```
<link href="css/template_css.css" rel="stylesheet" type="text/css" />
```

Однако, для правильной работы шаблона необходимо указать абсолютный путь к css файлу начиная с директории `templates`, как мы делали ранее (с.м. Листинг 5). Для простоты работы с тестовым шаблоном также рекомендуется использовать существующий CSS файл с уже объявленными классами Joomla! Для этого можно копировать CSS файл или его содержимое из шаблона `rhuk_milkyway` в соответствующую директорию нашего шаблона.



## Вставка модулей Joomla!

Теперь у нас есть основа для вставки модулей. Переместите курсор к той ячейке таблицы шаблона, в которой вы хотите разместить, например, поле поиска и нажмите кнопку Вставить поиск (Insert Search).



Рис. 20: Вставка поля поиска

Поле поиска появится в соответствующей части шаблона, а необходимый для его работы HTML код и JavaScript будут добавлены к нашему исходному коду:

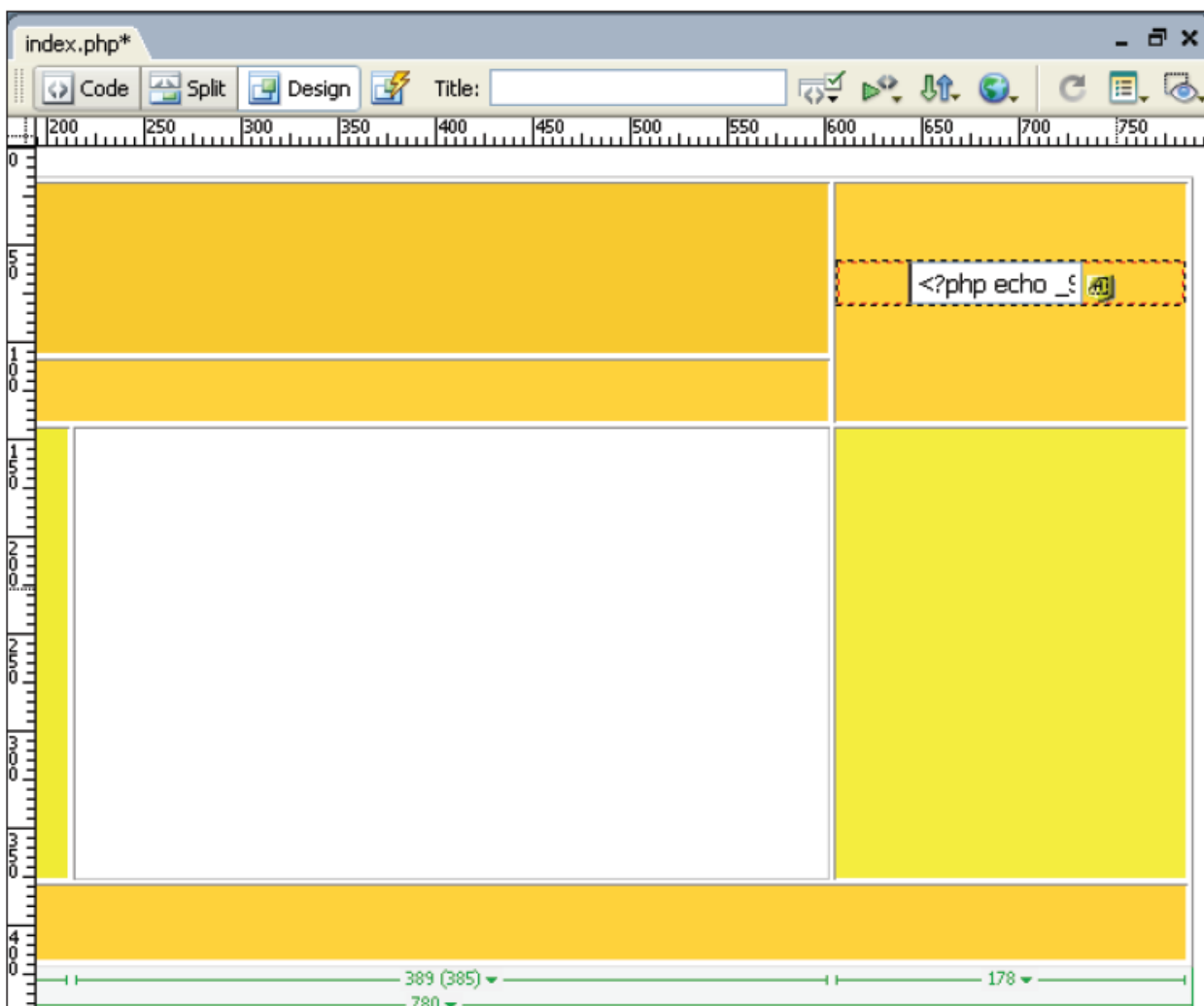


Рис. 21: Поле поиска в шаблоне

Таким же методом вы можете вставить в шаблон любой динамический элемент Joomla!

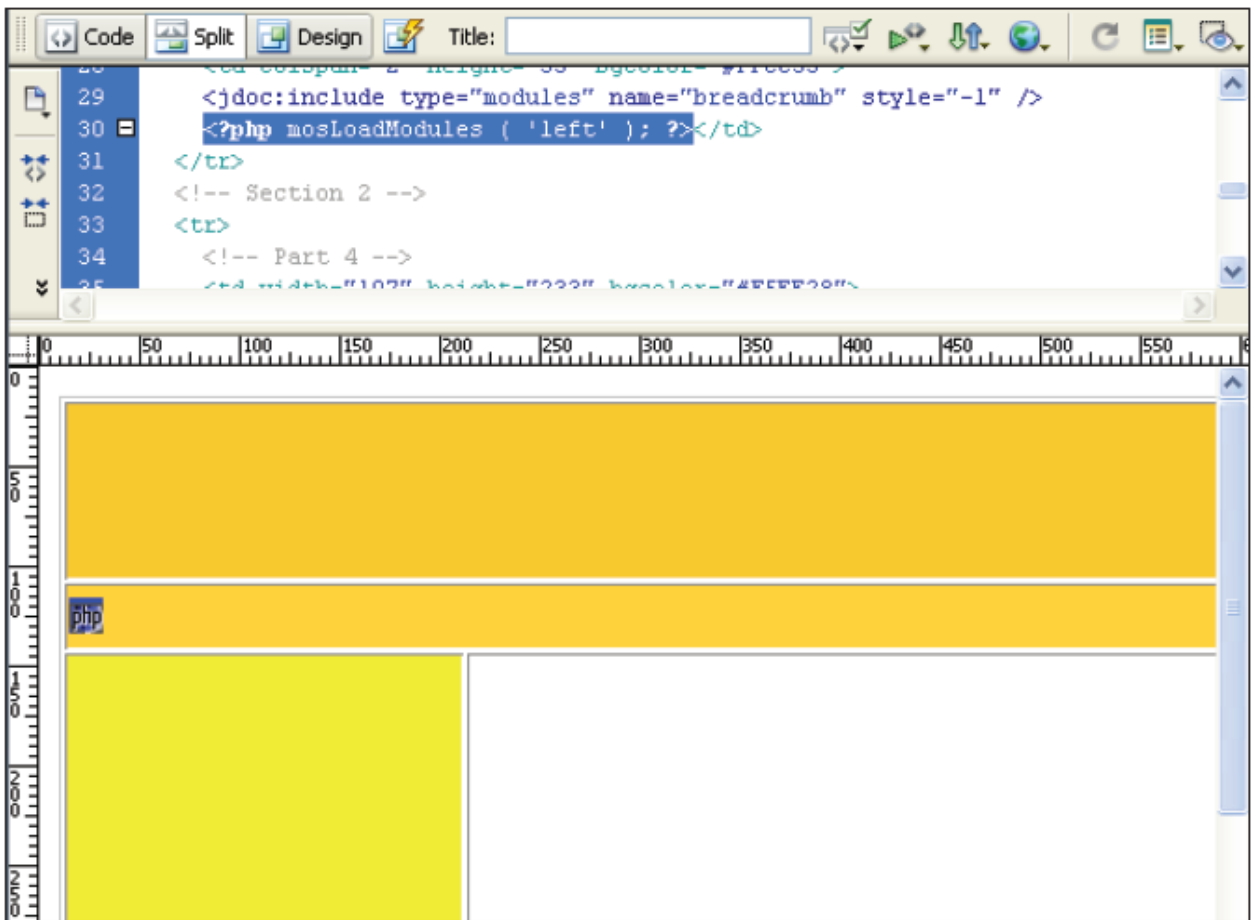


Рис. 22: Шаблон с модулями Joomla!

## Сайт «в живую»

Чтобы получить возможность работать с только что созданным шаблоном в Joomla!, необходимо создать файл `templateDetails.xml`. Базовая версия без картинок выглядит так:

```

<install version="1.5" type="template">
<name>joomla150_book_dw</name>
<version>0.1</version>
<creationDate>28.07.2006</creationDate>
<author>Hagen Graf</author>
<copyright>GNU/GPL</copyright>
<authorEmail>hagen@cocoate.com</authorEmail>
<authorUrl>http://www.cocoate.com</authorUrl>
<version>0.1</version>
<description>... Description ... Dreamweaver Version </description>
<files>
<filename>index.php</filename>
<filename>templateDetails.xml</filename>
<filename>template_thumbnail.png</filename>
<filename>css/template.css</filename>
</files>
</install>

```

После создания XML файла вы сможете выбрать новый шаблон в Менеджере шаблонов (Template Manager) административной части Joomla! Установите шаблон в качестве шаблона по умолчанию и загрузите ваш сайт Joomla! в веб обозревателе.



Рис. 23: Менеджер шаблонов с вашим собственным шаблоном

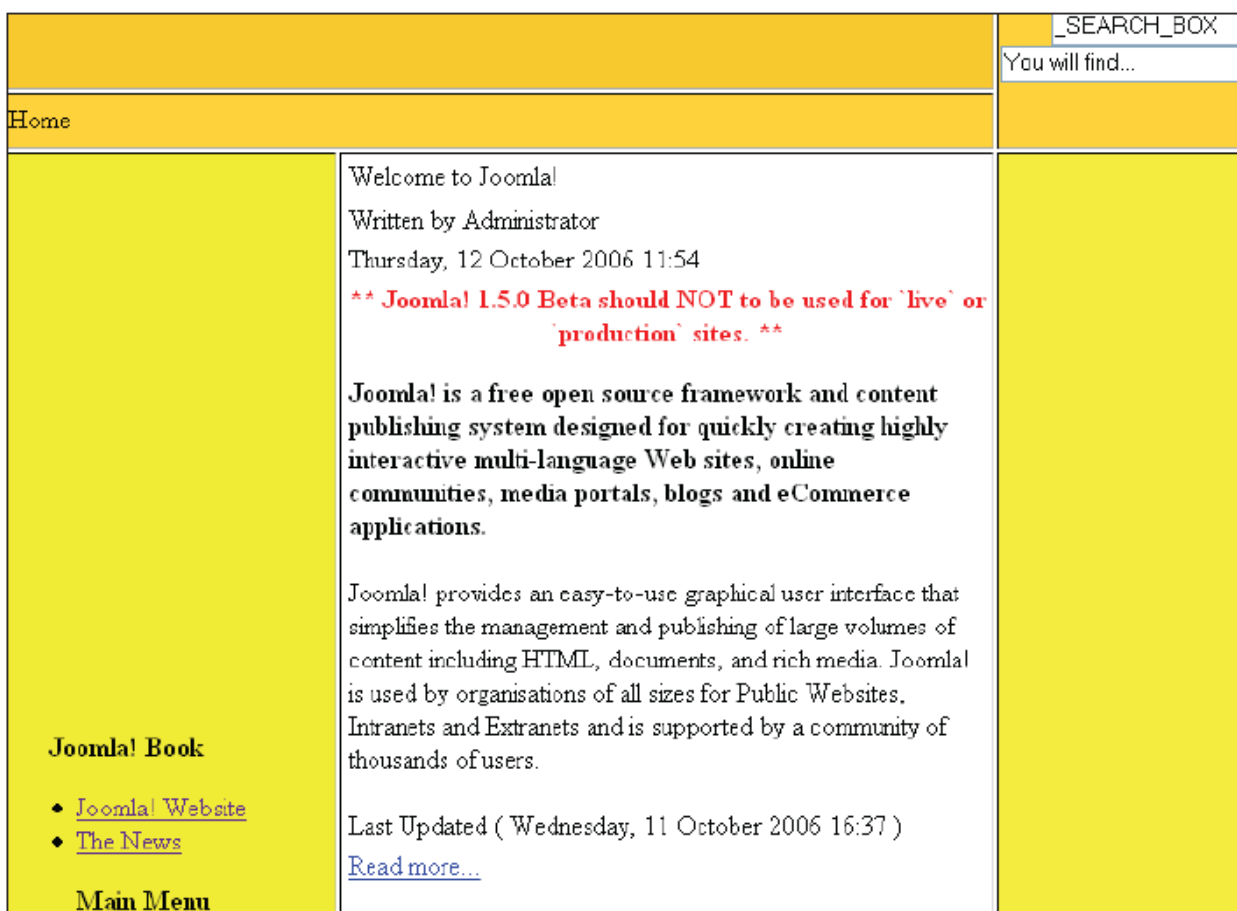


Рис. 24: Ваш сайт с вашим шаблоном

Далее, пользуясь широкими возможностями Dreamweaver вы можете продолжить работу над усовершенствованием дизайна и программной части шаблона. Dreamweaver позволяет просмотреть результаты редактирования шаблона сразу после его сохранения. Вы также можете настроить Dreamweaver на автоматическое обновление измененного шаблона, находящегося на FTP или WebDAV сервере! Для этого в Dreamweaver необходимо указать

путь к файлам Joomla! и ввести информацию, необходимую для подключения к удаленному серверу.

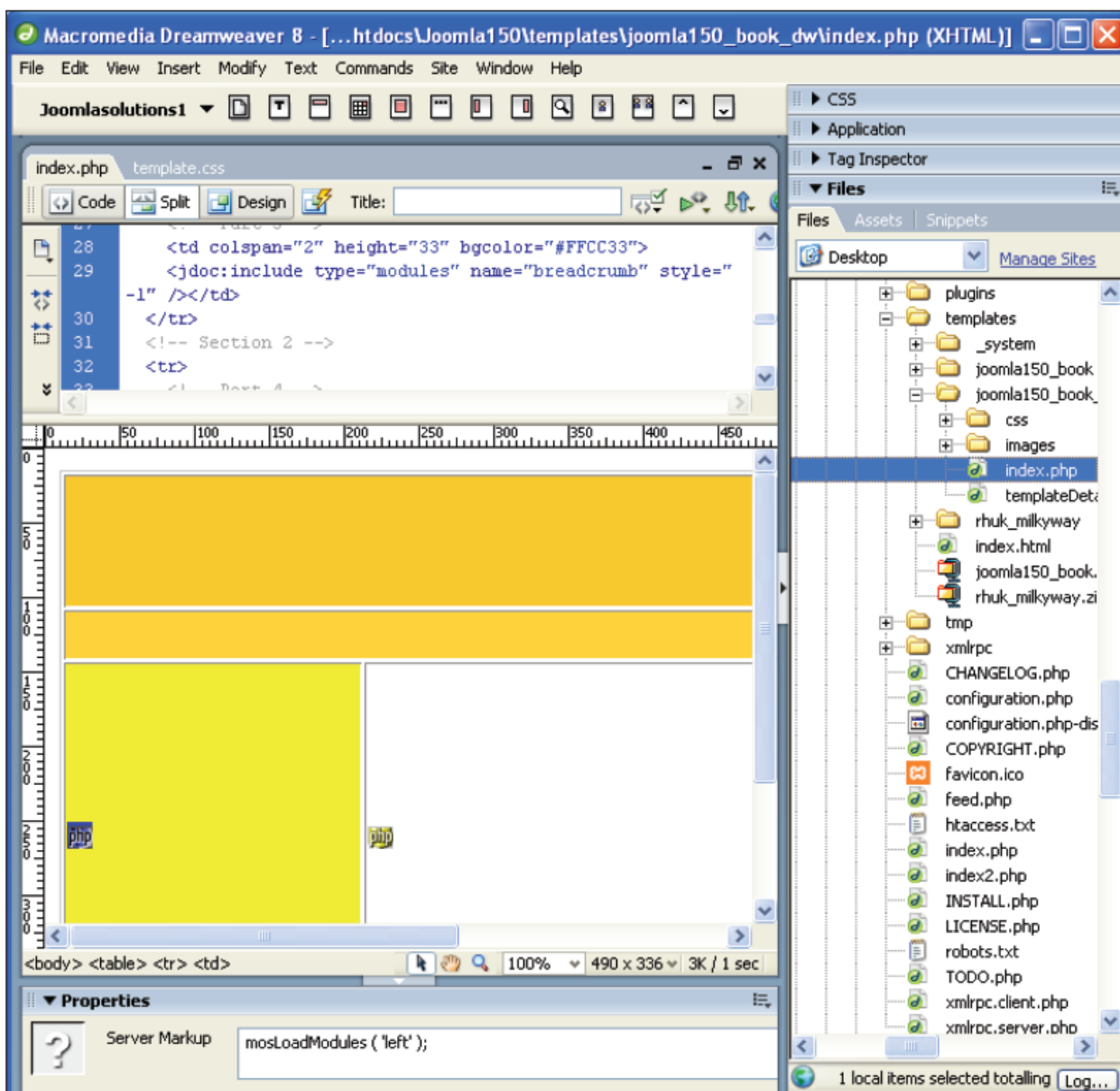


Рис. 25 Работа над сайтом в Dreamweaver

## Шаблоны и теги <div>

Joomla! до версии 1.5.0 (включительно) все еще может работать с таблицами при создании шаблонов. Однако, в последующих версиях Joomla! эта функциональность будет полностью убрана в целях реализации требований об устранении ограничений. Тег <div> заменит тег <table>

Тег <div> позволяет объединить в одном месте несколько элементов, например, текст, графику и т.д. Отличительным свойством такой группы элементов поначалу является только то, что она начинается с новой строки. Тег <div> не имеет больше ни каких свойств. Преимущество его использования открывается в комбинации с CSS выражениями. <div> был создан именно с этой целью: форматироваться с помощью CSS. <div> дает возможность создавать шаблоны, полностью контролируемые CSS файлами.

Один из базовых шаблонов, `rhuk_milkyway`, использует именно такую технологию. Он не содержит таблиц (тег `<table>`), вместо этого структура шаблона определяется тегами `<div>`. Просмотрите исходный код этого шаблона, чтобы понять, как работает такая технология. Dreamweaver также поддерживает работу с тегами `<div>`.

## Размещение модуля «в деталях»

Давайте ближе рассмотрим панель модуля, расположенную слева:

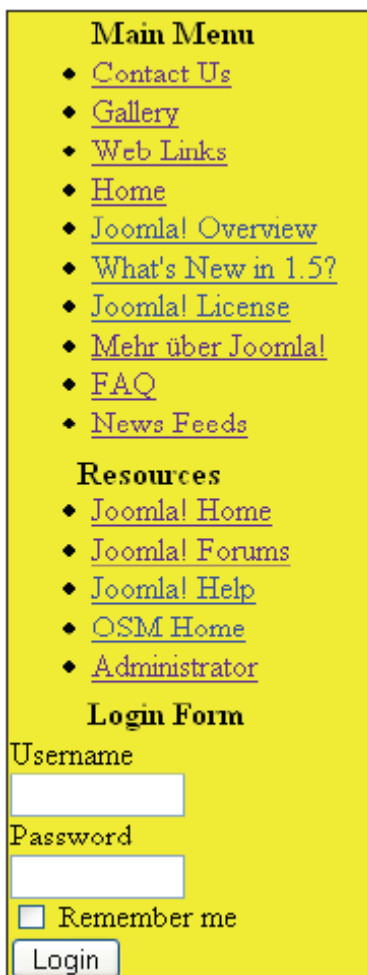


Рис. 26: Модуль, расположенный слева

В HTML коде файла `index.php` имеется следующий фрагмент:

```
<div id="leftcolumn">
  <jdoc:exists type="modules" condition="left" >
  <jdoc:include type="modules" name="left" style="-3" />
  </jdoc:exists>
</div>
```

В форматировании тега `<div>` используется идентификатор CSS под названием `leftcolumn`. Контент между тегами `<jdocs:exists>` и `</jdocs:exists>` отображается только если выполняются условия параметров. В данном случае отображение будет зависеть от наличия модуля или модулей в положении «слева» (`left`). Если таких модулей нет, общая разметка шаблона динамически изменится, чтобы заполнить пустое место.

В нашем случае имеется три модуля.

CSS идентификатор левой колонке выглядит в CSS файле следующим образом:

```
#leftcolumn {  
padding: 0;  
margin: 0;  
width: 20%;  
float:left;  
}
```

Если вы посмотрите на исходный код страницы, отображаемой в веб браузере (нажать правой кнопкой на странице и выбрать Просмотреть исходный код страницы (View Page Source)), то увидите, что jdoc запрос Joomla! сгенерировал на самом деле значительно больший объем кода.

## Листинг 7: Выдержка из исходного кода HTML страницы

... дополнительные HTML команды

```
<div id="leftcolumn">  
  
  <div class="module_menu">  
    <div>  
    <div>  
    <div>  
    <h3>Main Menu</h3><ul class="mainmenu">  
    <li class="level1 item1 active current">...</li>  
    <li class="level1 item2">...</li>  
    </div>  
    </div>  
    </div>  
    </div>  
    <div class="module_menu">  
    <div>  
    <div>  
    <div>  
    <table ...>  
    <tr ><td>...</td></tr>  
    <tr ><td>...</td></tr>  
    </table>  
    </div>  
    </div>  
    </div>  
    </div>
```

... дополнительные HTML команды

В данном фрагменте кода особо отметим CSS класс `module_menu`. Описание данного класса в CSS файле выглядит следующим образом:

```
div.module_menu {  
background: url(..images/mw_box_blue_br.png)  
100% 100% no-repeat;  
... дополнительные команды ...  
}  
div.module_menu div {  
background: url(..images/mw_box_blue_bl.png)  
0 100% no-repeat;  
}  
div.module_menu div div {  
background: url(..images/mw_box_blue_tr.png)  
100% 0 no-repeat;  
}
```

```

div.module_menu div div div {
background: url(../images/mw_box_blue_tl.png)
0 0 no-repeat;
padding: 10px;
padding-top: 30px;
padding-bottom: 15px;
width: auto;
}
div.module_menu div div div div {
background: none;
padding: 0;
}

```

Четыре блока CSS активируют определенные фоновые рисунки в каждом из тегов <div>. В нашем случае результатом использования этого кода является меню с закругленными углами. Также в этом CSS файле имеются дополнительные строки, формирующие заголовки третьего порядка (<h3></h3>), и несортированный список элементов меню.

Пока все идет нормально. Тип форматирования выбран, а отображение тегов <div> можно в любой момент изменить с помощью атрибута style (список всех параметров можно найти в дополнении). В первом примере мы использовали код -1, в этом примере мы используем код -3. Использование именно этого кода необходимо при работе с тегами <div>. Однако, если вы продолжите читать листинг 7 далее, то снова наткнетесь на таблицы. Модуль входа в систему (Login Module) все еще работает с помощью таблиц. На данном этапе развития Joomla! редактирование модуля входа в систему без использования таблиц может быть проблемой. Однако и здесь есть решение.

## **Изменение отображения HTML без изменения базовых файлов**

Для того чтобы решить проблему разметки информации при выводе в HTML страницу в виде таблиц, как правило, нужно вносить изменения в базовые файлы. Этот метод позволит убрать таблицы из модуля входа в систему, однако, при следующем обновлении Joomla! он может быть снова заменен файлом с таблицами.

Начиная с версии Joomla! 1.5.0 для решения этой проблемы используются так называемые виды (views). Каждый компонент имеет дополнительную директорию views, в которой могут храниться различные варианты его отображения. Например, у компонента контента com\_content имеются следующие варианты отображения: archive, article, category, frontpage, section. Директории с такими названиями размещены в папке views и содержат файлы для поддержки того или иного способа отображения компонента. Каждая из этих директорий также содержит папку tmpl, которая в свою очередь содержит PHP файл с именем данного вида, в нашем примере – это article.php.

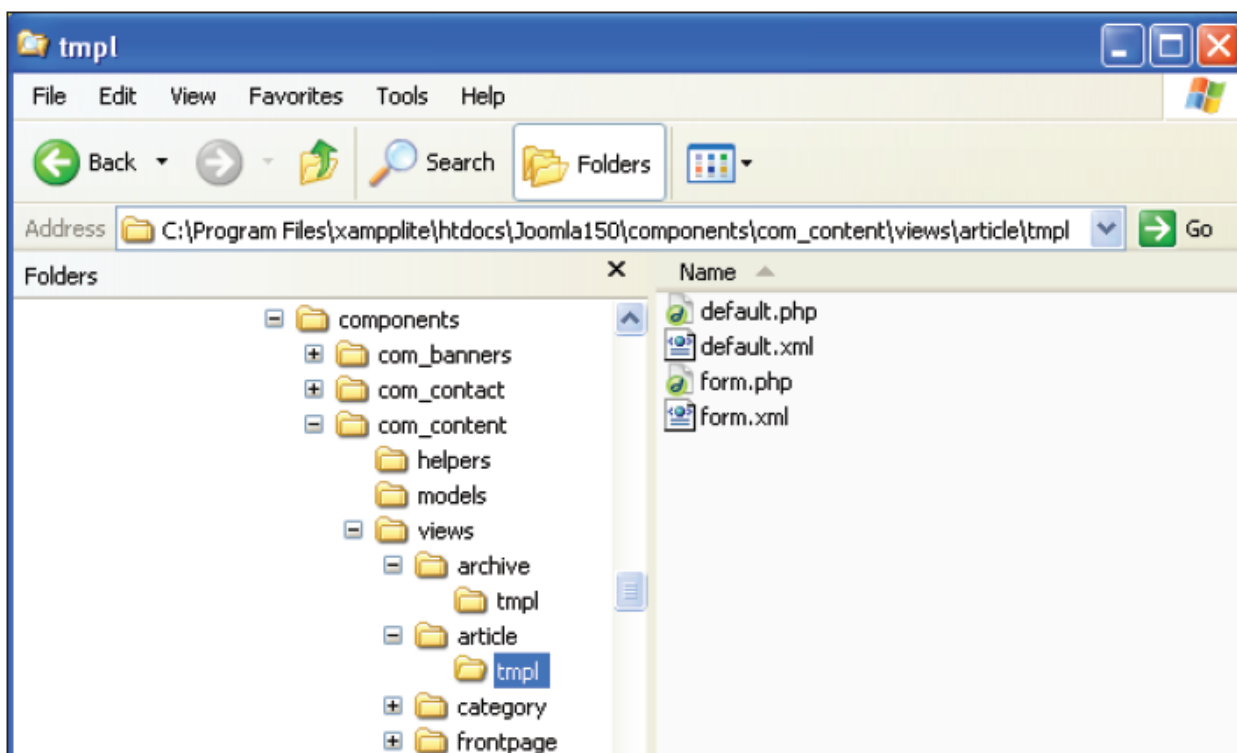


Рис. 27: Структура директорий папки виды (views)

Для того чтобы использовать данный метод, в директории шаблона необходимо создать еще одну папку. В данном примере такую папку нужно назвать `com_content`, поскольку мы хотим изменить разметку компонента `content`: `[ПутьКJoomla]/templates/joomla150_book_dw/com_content/`.

Теперь можно копировать файл `[ПутьКJoomla]/components/com_content/view/article/tmpl/default.php` в только что созданную директорию и изменить его по своему вкусу. В первую очередь Joomla! использует виды (views) из папки шаблона, если они отсутствуют, то используются стандартные виды компонентов.

## Web доступность и Joomla!

Важность доступности и удобства (web accessibility) веб страниц сложно переоценить. Во многих странах эта тема регулируется специальными правилами. Полную доступность и дружелюбность к пользователю, однако, достаточно сложно достичь. Но к этому нужно стремиться. При разработке сайта необходимо учесть требования и возможности как можно большего числа пользователей.

Широкая доступность (accessibility) сайта – это то же самое, что и рампа около ступенек в супермаркете для инвалидов на креслах-каталках, или звонок, дублирующий зеленый цвет светофора, для слепых.

Доступность – это огромное преимущество для любой компании, предлагающей публичный сервис. Каждый посетитель сайта – это потенциальный клиент. Поэтому, тема доступности на самом деле выходит далеко за рамки обслуживания инвалидов и пожилых людей.



## Критерии доступности веб сайтов

Чтобы понять, какие изменения необходимо внести, чтобы сделать сайт более доступным, обратите внимание на следующие критерии:

**Ясность:** Содержание любого сайта должно быть ясным и легким для понимания. Информация должна быть там, где пользователь ожидает ее увидеть.

**Совместимость с веб обозревателями:** Существует масса различных веб обозревателей. От текстового Lynx на консоли Linux и обозревателей для мобильных телефонов и КПК до веб обозревателей под OSX, Unix, Windows или OS2. Все они предполагают работу с определенным разрешением, отображают или не отображают графику, могут или не могут выполнять JavaScript, могут или не могут проигрывать Flash ролики и т.д. Не забывайте также про программы для распознавания символов и, например, клавиатуры с азбукой Брайля для слепых. Такие программы тоже являются обозревателями!

**Корректный исходный код и логически структурированная архитектура страницы:** Навигация, разметка и контент – это важные аспекты построения веб страницы. Они должны быть логично построены и семантически правильны. Стандартом в этом является XHTML. Ваш исходный код не должен содержать ошибок!

**Контраст:** Дизайн сайта должен быть в достаточной мере контрастным, чтобы удовлетворять требованиям людей с дефектами зрения.

**Графика и изображения:** Большинство устройств вывода для инвалидов не могут отображать графику. Поэтому альтернативный способ подачи информации без использования графических элементов должен быть предусмотрен.

**Размеры шрифтов:** Шрифты должны быть читаемы и на старых или альтернативных системах.

Основные требования к доступности изложены по адресу <http://www.section508.gov>.

## Реальность

Вначале реальность несколько пугает. Сложно найти сайт, полностью отвечающий всем требованиям доступности. Однако, как уже говорилось, нет пределов совершенству. Повсеместное использование CMS систем и формулировка четких критериев доступности также способствует распространению «правильно» оформленных сайтов.

Компании, торгующие через Интернет, тоже начинают понимать, что сайты, доступные более широкому кругу пользователей, способны принести большой доход.

В реальной жизни приходится иметь дело с различными группами пользователей, различными типами пользовательских терминалов и требованиями. Поэтому, ваш веб сайт всегда будет компромиссным. Не ставьте перед собой сразу заоблачные требования. Начинать с простого.

## Соответствует ли Joomla! требованиям доступности

Если кратко: нет. Однако, в Joomla! версии 1.5.0 уже заложен фундамент, который в следующих версиях разовьется в полноценную поддержку всех аспектов, необходимых для создания доступного сайта.

## Можно ли привести Joomla! в соответствие с требованиями доступности?

Если кратко: да!

### Технология

Joomla! до версии 1.5.0 все еще использует разметку в виде XHTML таблиц. Однако, при построении сайта со «свободной» конфигурацией элементов использование тегов таблиц недопустимо. Таким образом, программный код Joomla!, отвечающий за вывод информации необходимо изменить. Сделать это можно с помощью системы видов, описанной в подразделе *Изменение отображения HTML без изменения базовых файлов*.

Если вы сможете создать интуитивно понятную структуру с корректным XHTML кодом, масштабируемыми шрифтами, правильными комбинациями цветов и достаточной контрастностью, уровень доступности вашего сайта будет выглядеть вполне достаточным.

Проверить функциональность сайта можно с помощью массы различных инструментов. В поисках подходящего инструмента рекомендуем посетить сайт Яна Эрика Хеллбуша (Jan Eric Hellbusch).

### Люди

Теперь перейдем к людям, которые наполняют сайт содержимым – редакторам. В первую очередь необходимо пробудить в них чувство ответственности.

Вот пример корректного и семантически правильного оформления текста:

```
<h1>заголовок первого уровня</h1>
<p>первый параграф</p>
<p>второй параграф</p>
```

Несмотря на всю простоту данного кода, зачастую, очень сложно вбить его в голову людям, которые пишут тексты. Можно заставить заголовок выглядеть заголовком путем изменения скриптовых атрибутов. Следующая команда отображает текст, выглядящий как заголовок, но при этом не являющийся им.

```
<span class="heading">заголовок первого уровня</span>
```

Корректная же версия команды будет выглядеть так:

```
<h1 class="heading">заголовок первого уровня</h1>
```

Как научить этому редактора?

Существуют две программы, являющиеся стандартом для обработки текста: **OpenOffice.org** и **Microsoft Word**. Редакторы должны иметь аналогичный инструмент с такими же возможностями для создания контента веб страниц. WYSIWYG редактор, встроенный в Joomla! реализует лишь малую часть этих возможностей. Вот еще один пример корректного размещения графики с использованием атрибутов title и longdesc:

```


```

Атрибут longdesc служит для размещения расширенного описания изображения

При наполнении сайта необходимо учесть еще массу таких «трюков». WYSIWYG редактор Joomla! не совсем подходит для таких целей.

Также очень важным является и корректное построение фраз. Язык текста должен быть легко понятен целевой аудитории. Журналистский талант может пригодиться очень к стати!

## Доступный сайт с Joomla!

Вот пример сайта Joomla!, отвечающего критериям веб доступности ([www.3tc4u.co.uk](http://www.3tc4u.co.uk)):



Рис. 28: Доступный веб сайт

На этом же ресурсе можно найти примеры и других подобных сайтов.

## Итог

Надеемся, что эта статья поможет вам в создании собственных шаблонов. Также мы постарались прояснить для вас понятие Web Accessibility (веб доступности) сайтов для различных групп пользователей. Дерзайте и у вас все получится!

## Дополнение

### Структура шаблона (файла index.php)

Элементы `<jdoc>` позволяют обращаться к динамическим секциям шаблона без использования PHP команд

#### Код шаблона

```
<jdoc:tmpl name="loadcss" varscope="document"
type="condition" conditionvar="LANG_DIR">
<jdoc:sub condition="rtl">
<link href="templates/{TEMPLATE}/css/template_rtl.css"
rel="stylesheet" type="text/css" />
</jdoc:sub>
</jdoc:tmpl>
```

#### Эффект

Установка CSS поддержки для языков с письмом справа на лево.

#### BODY

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

Отображение системных сообщений Joomla!

```
<jdoc:include type="modules"
name="[position]" style="[Style]" />
```

Загрузить модуль в позиции [position]. Стиль [style]: 0 – модуль отображается в табличной колонке (`<td>...</td>`)

1 – горизонтальное меню

-1 – отдельная строка

-2 – позиционирование с помощью XHTML тегов (`<div>...</div>`)

-3 – опция для форматирования «закругленных» углов.

```
<jdoc:exists type="modules" condition="left" >
<jdoc:include type="modules" name="left" style="-3" />
</jdoc:exists>
```

Условная инструкция: если имеется модуль для отображения слева, загрузить его в шаблон

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

Отображение компонента. Имя компонента извлекается из URL.

### Как заменить изображение (логотип) в шаблоне?

Перевод и публикация: Copyright © 2007 WWW.JOOMLAUA.COM

Чтобы заменить логотип в уже существующем шаблоне, необходимо просмотреть его исходный код. Графика и изображения могут быть определены как в HTML так и в CSS структурах. Размер графических изображений, обычно, оптимизируется под конкретный шаблон.

Заменить изображение в шаблоне можно одним из следующих методов:

#### **Метод 1**

1. Создайте графический объект с таким же размером и разрешением, как и у заменяемого.
2. Загрузите новый объект в **Медиа Менеджер (Media Manager) Joomla!**
3. Нажмите на графический файл, чтобы получить ссылку на него.
4. Замените соответствующий атрибут тега `img` (`<img src=`) в исходном коде шаблона.

#### **Метод 2**

1. Создайте новый графический объект и присвойте ему имя, аналогичное имени заменяемого.
2. Просто замените старый графический объект новым.