

Приложение 5
к рабочей программе
МДК.02.02 Инструментальные сред-
ства разработки программного обеспечения

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Димитровградский инженерно-технологический институт –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
(ДИТИ НИЯУ МИФИ)

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

**по организации внеаудиторной самостоятельной работы студентов
по дисциплине**

МДК.02.02 Инструментальные средства разработки программного обеспечения

специальности 09.02.07 Информационные системы и программирование

Составитель: Р.К. Казынбаева, преподаватель техникума ДИТИ НИЯУ МИФИ

Димитровград

Содержание

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
2. ПРОГРАММА ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ	5
3. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ...	7
4. РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	15
5. ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ УЧЕБНЫХ ИЗДАНИЙ, ИНТЕРНЕТ-РЕСУРСОВ, ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ	25
ПРИЛОЖЕНИЕ 1	26

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время актуальным становятся требования к личным качествам современного студента – умению самостоятельно пополнять и обновлять знания, вести самостоятельный поиск необходимого материала, быть творческой личностью. Ориентация учебного процесса на саморазвивающуюся личность делает невозможным процесс обучения без учета индивидуально-личностных особенностей обучаемых, предоставления им права выбора путей и способов обучения. Появляется новая цель образовательного процесса – воспитание личности, ориентированной на будущее, способной решать типичные проблемы и задачи исходя из приобретенного учебного опыта и адекватной оценки конкретной ситуации.

Решение этих задач требует повышения роли самостоятельной работы студентов над учебным материалом, усиления ответственности преподавателя за развитие навыков самостоятельной работы, за стимулирование профессионального роста студентов, воспитание их творческой активности и инициативы.

Введение в практику учебных программ и модулей с повышенной долей самостоятельной работы активно способствует модернизации учебного процесса.

В соответствии с ФГОС СПО по специальности 09.02.07 Информационные системы и программирование в учебный процесс введена дисциплина «МДК.02.02 Инструментальные средства разработки программного обеспечения». Данная дисциплина состоит из одного раздела: Инструментарий Web-разработки, двух тем: Тема 1. Введение. Общие требования к созданию и оформлению Web-сайтов, Тема 2. Инструментальные средства разработки Web-сайтов в соответствии с Worldskills по компетенции «Веб разработка» и подтем, Тема 2.1 Визуальная среда Web-разработки WYSIWYG в соответствии с Worldskills по компетенции «Веб разработка». Adobe Dreamweaver, Тема 2.2 Визуальный HTML-редактор FrontPage для разработки Web-сайтов в соответствии с Worldskills по компетенции «Веб разработка».

Методические рекомендации для внеаудиторной самостоятельной работы (ВСР) по дисциплине «Инструментальные средства разработки программного обеспечения» предназначены для студентов четвертого курса специальности «Программирование в компьютерных системах (базовая подготовка)».

Внеаудиторная самостоятельная работа студентов является обязательной для каждого студента.

Основными целями внеаудиторной самостоятельной работы студентов являются:

- овладение знаниями и умениями по специальности;
- формирование готовности к самообразованию, самостоятельности и ответственности;
- развитие творческого подхода к решению проблем учебного и профессионального

уровня.

Выполнение студентами ВСР способствует формированию общих компетенций:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере.

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

2. ПРОГРАММА ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

МДК.02.02 Инструментальные средства разработки программного обеспечения

Наименование разделов и тем дисциплины	Объем, часов	Коды формируемых компетенций	Виды СРС	Формы контроля СРС	Сроки выполнения
1	2		3	4	5
Раздел 1. Инструментарий Web-разработки.					
Тема 1. Введение. Общие требования к созданию и оформлению Web-сайтов	1	ОК01- ОК09 ПК 2.1, ПК 2.5	Подготовка реферата по теме: «Языки программирования для Web-разработки». (0,5ч) Составление технического задания на разработку сайта с помощью Визуальной среды Web-разработки Adobe Dreamweaver. (0,5ч)	Выступление на занятии Индивидуальная проверка	к текущему занятию
Тема 2. Визуальная среда Web-разработки WYSIWYG в соответствии с Worldskills по компетенции «Веб разработка». Adobe Dreamweaver	2	ОК01- ОК09 ПК 2.1 - ПК 2.5	Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление практических работ. (1ч) Разработка своего сайта в соответствии с текущим стандартам W3C. Тестирование отладка готовых программ. (1ч)	Самостоятельная работа Индивидуальная проверка	
Тема 2.1 Визуальный HTML-редактор FrontPage для разработки Web-сайтов в соответствии с Worldskills по компетенции «Веб разработка»	1	ОК01- ОК09 ПК 2.1 - ПК 2.5	Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление практических работ. (1ч) Разработка сайта-визитки в соответствии с текущим стандартам W3C.	Самостоятельная работа Индивидуальная проверка	к текущему занятию

			Тестирование отладка готовых программ. (4ч)		
Всего	4				

3. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельные работы выполняются индивидуально на домашнем компьютере или в компьютерном классе в свободное от занятий время.

Студент обязан:

- перед выполнением самостоятельной работы, повторить теоретический материал, пройденный на аудиторных занятиях;
- выполнить работу согласно заданию;
- по каждой самостоятельной работе представить преподавателю отчет в виде результирующего файла на внешнем носителе;
- ответить на поставленные вопросы.

При выполнении самостоятельных работ студент должен сам принять решение об оптимальном использовании возможностей программного обеспечения. Если по ходу выполнения самостоятельной работы у студентов возникают вопросы и затруднения, он может консультироваться у преподавателя. Каждая работа оценивается по пятибалльной системе. Критерии оценки приведены в конце методических рекомендаций.

Раздел 1. Инструментарий Web-разработки

Тема 1. Введение. Общие требования к созданию и оформлению Web-сайтов

Формируемые компетенции: ОК01- ОК09, ПК 2.1, ПК 2.5

Кол-во часов: 1ч

Теоретический блок

1. Инструменты и методы разработки веб-сайтов

Веб-страницы и веб-сайты широко применяются во Всемирной паутине.

Для создания веб-сайтов используется язык разметки гипертекстовых документов HTML (HyperText Markup Language).

Методы конструирования веб-сайтов и инструменты, с помощью которых выполняется конструирование, очень разнообразны, их можно классифицировать по различным признакам. Важнейшими из них являются степень автоматизации конструирования сайта и методика процесса разработки.

По степени автоматизации конструирования веб-сайтов следует выделить два основных метода: визуальный и программируемый.

2. Визуальный метод

Визуальный метод позволяет конструировать веб-сайт с высокой степенью автоматизации. Этот метод возник из необходимости уменьшить трудоёмкость создания сайта и сократить сроки выполнения работ. При использовании этого способа создатель конструирует свою страницу на экране визуального редактора (онлайнового или программного), после чего программа по этой конструкции генерирует соответствующий код HTML для каждой страницы, который представляет собой набор конструкций языка HTML. Один из главных принципов визуальных редакторов формулируется как WYSIWYG - What You See Is What You Get, который переводится как «Что видишь при разработке, то и получаешь при просмотре странички». При создании сайтов с использованием визуального метода необходимо выбрать инструменты - редакторы визуального конструирования.

Во время создания веб-сайта эти редакторы обеспечивают пользователю возможность работать без непосредственного подключения к сети Интернет. В дальнейшем созданный веб-сайт можно разместить в сети Интернет.

Как уже было сказано, визуальные редакторы делятся на две большие группы, сильно отличающиеся друг от друга. Первый вид -- онлайновые редакторы. Онлайновые редакторы -- это

специальные механизмы для создания сайтов (как правило, очень простых), располагающихся непосредственно в Интернете, чаще всего на сервисах, предоставляющих хостинг (платный или бесплатный). Например, Narod.ru и Boom.ru.

Второй вид -- программы-клиенты (офлайновые), устанавливаемые на компьютер. Офлайновые редакторы позволяют создать достаточно сложные проекты. Среди наиболее известных редакторов можно отметить: Microsoft FrontPage, DreamWeaver (Macromedia), NamoWEbEditor, Adobe GoLive и др.

Но, даже разрабатывая сайт с помощью визуальных редакторов, разработчику необходимо знать основы языка HTML, потому что каждый редактор может допустить ошибку в коде документа. И разработчик должен уметь исправить возможные ошибки.

3. Программируемый метод

HTML (HyperText Markup Language) является стандартным языком, предназначенным для создания гипертекстовых документов в Интернет среде. Суть языка HTML - в разметке текста с помощью управляющих символов - тегов, которые располагаются в угловых скобках.

Html-документы могут просматриваться различными типами браузеров. Когда документ создан с использованием html, веб-браузер может интерпретировать html для выделения различных элементов документа и первичной их обработки. Использование html позволяет форматировать документы для их представления с использованием шрифтов, линий и других графических элементов на любой системе, их просматривающей.

Большинство документов имеют стандартные элементы, такие, как заголовок, параграфы или списки. Используя тэги html, можно обозначать данные элементы, обеспечивая веб-браузеры минимальной информацией для отображения данных элементов, сохраняя в целом общую структуру и информационную полноту документов. Все что необходимо, чтобы прочитать html-документ - это веб-браузер, который интерпретирует тэги html и воспроизводит на экране документ в виде, который ему придает автор.

В большинстве случаев автор документа строго определяет внешний вид документа. Html позволяет отметить, где в документе должен быть заголовок или абзац при помощи тэга, а затем предоставляет браузеру интерпретировать эти тэги. Например, один браузер может распознавать тэг начала абзаца и представлять документ в нужном виде, а другой не имеет такой возможности и представляет документ в одну строку. Пользователи некоторых браузеров также могут настраивать размер и вид шрифта, цвет и другие параметры, влияющие на отображение документа.

Html-тэги могут быть условно разделены на две категории:

- тэги, определяющие, как будет отображаться браузером тело документа в целом;
- тэги, описывающие общие свойства документа, такие как заголовок или автор документа.

Основное преимущество html заключается в том, что документ может быть просмотрен на веб-браузерах различных типов и на различных платформах. Языка html достаточно для создания полноценного сайта, но эти сайты статичны, без обратной связи с пользователями. Кроме того, обновления таких статичных сайтов трудоёмки. Создание сайтов на языке html относится к программируемым способам, также иногда называют ручным.

Также к программируемым способам относится написание сайтов на языках веб-программирования и Фреймворках: ASP.NET, Java, PHP, Perl, Ruby и т. д., но в основе любого кода сайта лежит язык гипертекстовой разметки Html. Его должны знать все разработчики.

Как правило, выбор языка и Фреймворка определяется тем, какими знаниями обладают программисты, реализующие проект. Многие задачи можно эффективно решить с помощью любого современного популярного языка программирования.

Необходимо понимать разницу между языком программирования и Фреймворком. Язык программирования -- это некоторый базовый синтаксис (возможно со стандартными библиотеками), с помощью которого можно создавать некоторые приложения. Фреймворк же предоставляет разработчику различные библиотеки, значительно упрощающие создание программ и сайтов. Некоторые языки и Фреймворки представляют собой неразрывное целое (например, ASP.NET и JSP). Другие языки могут использоваться без Фреймворка (PHP и Perl).

Рекомендуемая литература:

1. Снелл Абсолютно ясно о создании Web-страниц и Web-сайтов / Снелл, Нэд. - М.: Триумф, 2012. - 224 с.
2. Хассей, Трис WordPress. Создание сайтов для начинающих (+ CD-ROM) / Трис Хассей. - М.: Эксмо, 2012. - 432 с
3. Дакетт Джон HTML и CSS. Разработка и дизайн веб-сайтов (+ CD-ROM); Эксмо - Москва, 2013. - 480 с.
4. Китинг, Джоди Flash MX. Искусство создания web-сайтов; ТИД ДС - Москва, 2012. - 848 с.
5. Митчелл, Скотт 5 проектов Web-сайтов от фотоальбома до магазина; М.: ИТ Пресс - Москва, 2013. - 224 с.
6. Чебыкин Ростислав Разработка и оформление текстового содержания сайтов; БХВ-Петербург - Москва, 2014. - 528 с.
7. Петюшкин, Алексей HTML в Web-дизайне / Алексей Петюшкин. - М.: БХВ-Петербург, 2013. - 400 с.

Практический блок

– Подготовка реферата по теме: «Языки программирования для Web-разработки»

Примерный план № 1

1. Составьте план работы над рефератом.
2. Пользуясь интернет-источниками и литературой, выполните подбор материала (текстовый и графический) по выбранной теме. Напишите введение к реферату (объем 2-3 страницы).
3. Определите разделы и подразделы основной части реферата. Соотнесите найденный материал с содержанием.
4. Напишите заключение.
5. Оформите реферат и подготовьте его к печати. Распечатайте и сдайте на проверку.

Примечание: Требования к реферату смотрите в пункте 5.1. Образец оформления титульного листа представлен в приложении А.

Критерии оценивания

«Отлично» – работа выполнена с учетом всех требований, как к ее содержанию, так и оформлению.

«Хорошо» – работа выполнена и оформлена грамотно и правильно, но в ней встречаются некоторые неточности.

«Удовлетворительно» – к работе предъявлены 2 существенных замечания по содержанию и оформлению.

– Составить техническое задание на сайт

ТЗ – это документ, который описывает будущий проект детально и полностью. Чем детальней он будет, тем точнее будет реализована задумка и тем меньше конфликтов и спорных ситуаций в ходе выполнения проекта будет возникать, ведь абсолютно любую вещь можно сделать по-разному. На него можно ссылаться, если что-то не выполнено или выполнено не так или допущены другие ошибки. Перед началом работ заказчик обычно в тезисном виде описывает будущий проект или заполняет бриф, а исполнитель формализует все эти требования и пожелания, при необходимости, предлагает корректировки. При этом заказчику необходимо убедиться, что все его "хотелки" зафиксированы в тех задании.

Если с веб-студией или фрилансером будет заключаться договор на разработку сайта, то тех задание обычно идет, как приложение к нему. И в спорных ситуациях, руководствуются тем, что там написано.

Из чего состоит ТЗ?

Предположим, в рамках проекта требуется составить техническое задание на разработку сайта студии копирайтинга «Перо». Какие пункты оно должно содержать?

Общие сведения (описание)

Здесь указываются:

Сведения о компании. Общая информация о студии, чем она занимается. Не лишним будет привести список оказываемых услуг. Сюда же можно добавить адрес будущего сайта, контактные данные.

Этапы и сроки реализации проекта. Очень важный момент, как правило, календарный план по всем этапам работ составляют в самом конце. Эта часть даёт понимание, что и когда будет делаться. Например, (с указанием дат):

- Подготовительный этап;
- Проработка концепции сайта;
- Проектирование;
- Создание дизайн-макета;
- Разработка дизайна страниц;
- Вёрстка;
- Программирование;
- Наполнение контентом;
- SEO-оптимизация;
- Тестирование;
- Запуск.

Каких-то этапов, например, SEO-продвижения может и не быть. Зависит от целей и задач заказчика и компетенций исполнителя.

Назначение и цели

Здесь формулируется, какие функции будет выполнять сайт и для кого он предназначен.

Назначение сайта. Каких целей созданием сайта необходимо достичь? Для чего он нужен, какие задачи решает?

- Реклама и привлечение новых клиентов;
- Поддержка заказчиков и партнёров;
- Демонстрация выполненных работ;
- Ознакомление со списком услуг;
- Создание и поддержание имиджа компании.

Возможно, некоторые пункты стоит расписать более подробно. Например, если перед сайтом стоит задача в том числе и информировать посетителей, то лучше пояснить, о чём именно.

Целевая аудитория. Кто будет пользоваться сайтом, для кого он создаётся?

- Веб-мастера, блогеры;
- Владельцы интернет-магазинов;
- Владельцы информационных порталов;
- Рекламные студии;
- Представители присутствующих в онлайн-пространстве фирм и компаний.

Требования

Большой и крайне важный раздел, в котором учитывается как можно больше моментов проектирования и разработки, потому как за неоговоренный в ТЗ функционал заказчику придётся доплачивать.

Тип. К какой категории принадлежит веб-ресурс?

- Посадочная страница;
- Сайт-визитка;
- Корпоративный сайт;
- Информационный портал;
- Интернет-магазин.

Требования к оформлению. Они могут быть следующего вида:

- Сайт должен быть минималистичным и при этом отражать род деятельности компании.
- Основные цвета: зелёный и белый, по брендбуку или на усмотрение дизайнера.
- В оформлении нельзя использовать анимацию, всплывающие окна, Flash-элементы, дизайнерские излишества.
- Нельзя использовать шрифты с засечками (можно применять стандартные: Verdana, Arial, Tahoma и т. д.). Кегль должен обеспечивать максимальное удобство чтения (12-16 пт.).

Что касается требований к дизайну, то тут можно применять разные подходы. Если заказчик сам точно знает, что хочет получить, то описывает подробно свои пожелания, приводит примеры сайтов, которые ему нравятся и дает прочую конкретику. Но иногда бывает так, что и он сам точно не знает, как это все должно выглядеть, в этом случае обычно исходят из задач, которые должен решать дизайн. Исполнитель прорабатывает концепции, предлагает решения, защищает свою идею и корректирует его по замечаниям заказчика. Второй вариант стоит дороже и требует большей квалификации от подрядчика.

Языковые требования. Носители какого языка смогут посещать ресурс? Какие языковые версии сайта должны быть?

- Русский;
- Английский;
- Эсперанто.

Требования к совместимости. С каких устройств и какими браузерами сайт точно будет открываться корректно? В последнее время наметилась тенденция к адаптивной вёрстке, когда страница правильно отображается на любом устройстве с любым соотношением сторон и разрешением экрана. Здесь можно перечислить браузеры, с которыми однозначно должен быть совместим ресурс. Обычно на всех современных браузерах сайты отображаются одинаково, бывают только проблемы со старыми версиями Internet explorer.

Требования к CMS. Возможности администрирования сайта определяют, какие именно блоки можно редактировать и настраивать через панель управления, не вмешиваясь в код и не редактируя напрямую базу данных, а используя удобный визуальный интерфейс. Например, можно сформулировать так:

- Возможность изменения контента на страницах сайта;
- Возможность управления страницами (добавление, переименование, удаление и т. д.);
- Возможность редактирования структуры сайта и пунктов меню;
- Функции автоматической обработки графики (создание превью, трансформация под заданный размер и т. п.);
- Возможность прописывать уникальные Мета-теги;
- ...

Как и в других подразделах, нужно описывать все требования и пожелания.

Часто заказчик уже имеет опыт работы с одним из популярных CMS, тогда целесообразно искать порядчиков под конкретный движок. Также при выборе CMS лучше не соглашаться на самописные решения, т.к. в дальнейшем это поставит в зависимость от исполнителя. Самописные движки, на мой взгляд, оправданы только в очень крупных проектах, где требуется специфичный функционал или оптимизация больших нагрузок.

Структура и навигация. Какие разделы, подразделы и отдельные страницы будет содержать проект?

- Главная страница
- Услуги
- Копирайтинг
- Рерайтинг
- SEO-копёррайтинг
- Корректурa
- Транскрибация
- Контент-менеджмент

- Контент-маркетинг
- Портфолио
- О нас
- Контакты

Сделайте и краткое описание каждой страницы, дайте определения. Например, что подразумевается под страницей «Контакты»? Она должна содержать адрес, телефон и электронную почту в текстовом виде? Или там должна присутствовать форма обратной связи? А может, нужно встроить код Яндекс Карт? Или же на странице контактов должно размещаться всё перечисленное, да ещё и ссылки на представительства в социальных сетях?

Желательно контент или хотя бы его наброски приготовить еще до начала работ с подрядчиком. Это будет способствовать более эффективной коммуникации.

Дополнительные требования. Всё, что не вошло в другие пункты раздела.

Описание разделов сайта

В этом пункте – детали. Обычно расписывается содержание всех уникальных страниц: какие элементы там будут, как пользователь будет взаимодействовать с ними.

Главная страница. Формулировка задачи может быть в следующем виде.


Основная часть главной страницы должна быть выполнена в виде Landing Page. На ней сверху вниз должны располагаться следующие элементы:

- Шапка – логотип, название фирмы;
- Меню навигации;
- Информация об акциях и скидках;
- Кнопка заказа;
- Рекламный текст;
- Блок с пятью лучшими работами и ссылкой на раздел портфолио;
- Отзывы клиентов;
- Штат студии;
- Блок партнёров компании;
- Информация о тарифах;
- Дублирующий блок скидок и кнопка заказа;
- Карта ссылок;
- Кнопки социальных сетей.

Схема главной страницы. Если есть понимание будущей структуры сайта, то будет нелишним вставить макет, отражающий структуру хотя бы главной страницы. Это может быть нарисованный от руки и отсканированный набросок, созданный в Photoshop макет – что угодно. Также можно использовать программы для прототипирования, который может освоить даже начинающий пользователь.

Внутренние страницы. Аналогичным образом можно отразить и другие страницы сайта. Если в целом они будут похожи друг на друга (например, изменяется только контент главной части страниц), то можно обобщить.

http://
🔍 Поиск



Обучающие материалы по созданию и продвижению сайтов и заработку в интернете

Главная	Содержание	Блог	Услуги	О сайте
---------	------------	------	--------	---------

Создание сайтов	Продвижение сайтов	Заработок в интернете	Полезное
С чего начать?	С чего начать?	С чего начать?	PCD шаблоны
Сайт на HTML, CSS	Внутренняя оптимизация	Все способы	HTML шаблоны
Сайт на Joomla	Контекстная реклама	Контекстная реклама	Алгоритмы
Сайт на Wordpress	Социальные сети	Партнерские программы	Аналитика сайта
Онлайн конструкторы	Аналитика	Бизнес в интернете	Другие сервисы

Популярное

Как создать свой сайт

- на UMI
- на Redhat
- на Wordpress
- на Joomla
- на NetHouse
- на Wix
- на A5

Как создать landing page

- на Wix
- на A5

Как создать интернет-магазин

- на InSales

Обзор конструкторов сайтов

Как заработать на сайте

Как продвигать сайт

Мои услуги

Баннер

Друзья сайта

Как создать свой сайт?

Добро пожаловать. Здесь я рассказываю о том, как создать свой сайт самому с нуля и заработать на нем деньги в интернете. Вся информация основана на моем личном опыте, ведь когда-то я сам искал ответы на эти вопросы и теперь точно знаю, что нужно знать новичку, чтобы создать сайт бесплатно, раскрутить его и начать зарабатывать на этом деньги. Теперь готов делиться всему накопленным знаниями тут.

1 Создание сайта

Добро пожаловать. Здесь я рассказываю о том, как создать свой сайт самому с нуля и заработать на нем деньги в интернете. Вся информация основана на моем личном опыте, ведь когда-то я сам искал ответы на эти вопросы и теперь точно знаю, что нужно знать новичку, чтобы создать сайт бесплатно, раскрутить его и начать зарабатывать на этом деньги. Теперь готов делиться всему накопленным знаниями тут.

2 Продвижение сайта

Добро пожаловать. Здесь я рассказываю о том, как создать свой сайт самому с нуля и заработать на нем деньги в интернете. Вся информация основана на моем личном опыте, ведь когда-то я сам искал ответы на эти вопросы и теперь точно знаю, что нужно знать новичку, чтобы создать сайт бесплатно, раскрутить его и начать зарабатывать на этом деньги. Теперь готов делиться всему накопленным знаниями тут.

3 Заработок в интернете

Добро пожаловать. Здесь я рассказываю о том, как создать свой сайт самому с нуля и заработать на нем деньги в интернете. Вся информация основана на моем личном опыте, ведь когда-то я сам искал ответы на эти вопросы и теперь точно знаю, что нужно знать новичку, чтобы создать сайт бесплатно, раскрутить его и начать зарабатывать на этом деньги. Теперь готов делиться всему накопленным знаниями тут.

4 Создание сайта

Добро пожаловать. Здесь я рассказываю о том, как создать свой сайт самому с нуля и заработать на нем деньги в интернете. Вся информация основана на моем личном опыте, ведь когда-то я сам искал ответы на эти вопросы и теперь точно знаю, что нужно знать новичку, чтобы создать сайт бесплатно, раскрутить его и начать зарабатывать на этом деньги. Теперь готов делиться всему накопленным знаниями тут.

5 Продвижение сайта

Добро пожаловать. Здесь я рассказываю о том, как создать свой сайт самому с нуля и заработать на нем деньги в интернете. Вся информация основана на моем личном опыте, ведь когда-то я сам искал ответы на эти вопросы и теперь точно знаю, что нужно знать новичку, чтобы создать сайт бесплатно, раскрутить его и начать зарабатывать на этом деньги. Теперь готов делиться всему накопленным знаниями тут.

Новое в блоге

X

X

X

Рубрика: SEO
Заголовок 1 строка

Рубрика: SEO
Заголовок 2 строки
Заголовок 2 строки

Рубрика: SEO
Заголовок 3 строки
Заголовок 3 строки
Заголовок 3 строки

[Вк](#) | [Фейсбук](#) | [твиттер](#)

Создание сайтов	Продвижение сайтов	Заработок в интернете	Полезное
С чего начать?	С чего начать?	С чего начать?	PCD шаблоны
Сайт на HTML, CSS	Внутренняя оптимизация	Все способы	HTML шаблоны
Сайт на Joomla	Контекстная реклама	Контекстная реклама	Алгоритмы
Сайт на Wordpress	Социальные сети	Партнерские программы	Аналитика сайта
Онлайн конструкторы	Аналитика	Оказание услуг	Другие сервисы

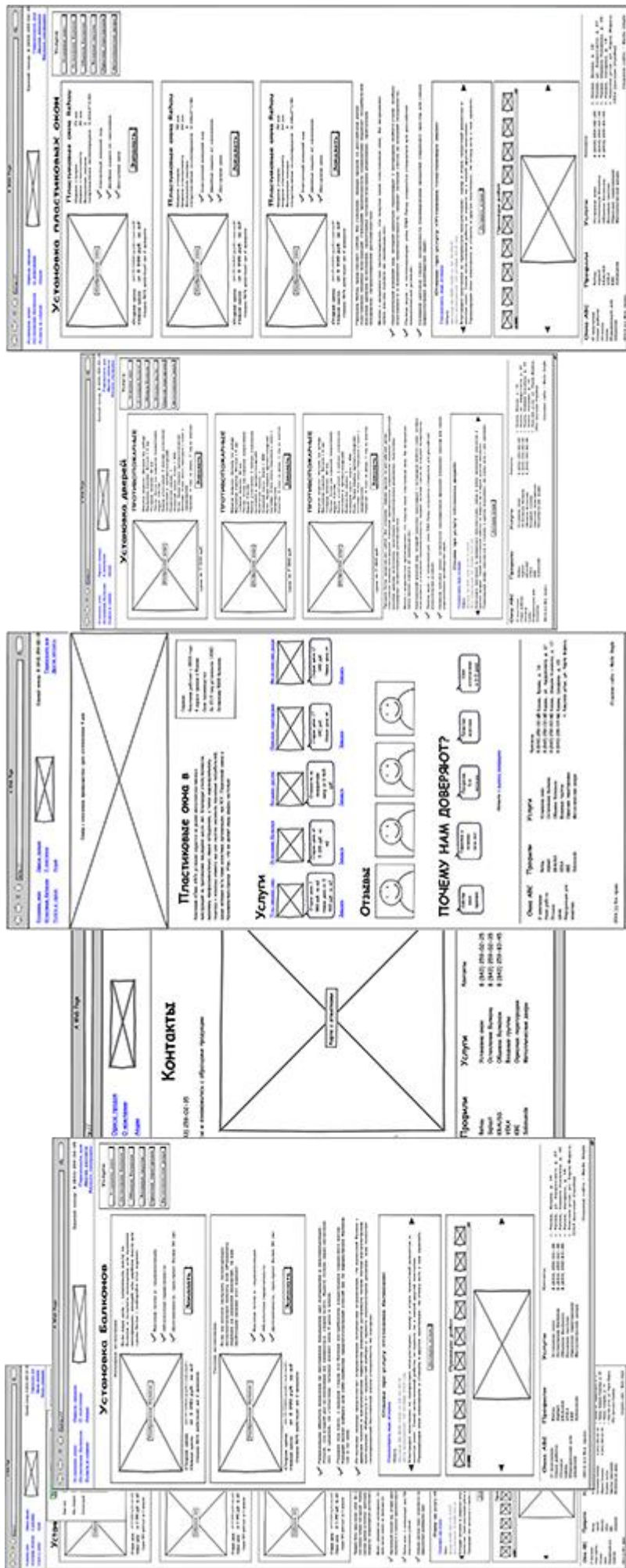
Мы в соц сети Вк, фейсбук

Создание и продвижение сайта - Media Simple

Качественные услуги по созданию и продвижению сайтов по всей России

Copyright © ИП Аметжан Булат Раисович 2010 - 2016
При копировании материалов сайта рабочая ссылка на веосторги обязательна.

Сюда же можно поместить схемы (или схему) страниц.



Кстати, эти прототипы сделаны с помощью программы Balsamiq Mockups, который довольно прост в освоении.

Описание функциональной части

Здесь нужно описать о функциональности сайта в целом и перечислить страницы с особыми возможностями. Например, если большинство страниц будут представлять собой простой текст с повторяющимися в каждом разделе блоками, а на странице контактов необходимо разместить форму заказа, это необходимо отразить.

При нажатии кнопки "Заказать" открывается форма заказа с раскрывающимся списком выбора услуги, полями: Имя, Телефон, Комментарий, возможностью прикрепить файл и кнопкой Отправить, нажатие которой приведёт к созданию и отправке письма на e-mail администратора.

Необходимо описать логику работы ресурса в целом, проработать основные сценарии поведения пользователей на страницах, указать все нюансы и особые функции каждого блока и раздела сайта. Например, что должно находиться в «шапке» и как это будет работать, по какому критерию сортируются работы в портфолио, скачиваются они или открываются для просмотра, в каком формате хранятся и т. д.

Если какой-то функционал не описан, то обычно делается так, как это предусмотрено в стандартной модификации CMS.

Критерии оценивания

«Отлично» – работа выполнена с учетом всех требований, как к ее содержанию, так и оформлению.

«Хорошо» – работа выполнена и оформлена грамотно и правильно, но в ней встречаются некоторые неточности.

«Удовлетворительно» – к работе предъявлены 2 существенных замечания по содержанию и оформлению.

Раздел 1. Инструментарий Web-разработки

Тема 2. Инструментальные средства разработки Web-сайтов в соответствии с Worldskills по компетенции «Веб разработка»

Формируемые компетенции: ОК01- ОК09, ПК 3.1-ПК 3.5

Кол-во часов: 1 час

Теоретический блок Текстовый редактор Geany

Geany представляется как легкая IDE (Integrated Development Environment), то есть интегрированная среда разработки программного обеспечения.

Вступление

Geany разрабатывается Энрико Трёгером (Enrico Tröger) и Ником Трелевеном (Nick Treleaven), базируется на библиотеке Gtk, распространяется под лицензией GNU GPL v2 (по крайней мере до сих пор). Он присутствует практически во всех полнофункциональных дистрибутивах Linux, в портах FreeBSD и, вероятно, в других BSD-системах.

Geany способен выполнять практически все функции обычного текстового редактора, как-то: инверсию регистров, дублирование текущей строки или выделения, подсветку синтаксиса многих языков программирования и разметки, развитые средства поиска и замены (в том числе с использованием регулярных выражений и escape-последовательностей, учетом регистра и так далее), включать или выключать динамический перенос строк; короче, практически всё, что требуется при наборе и редактировании текста. И не обязательно текста исходного – нарративного тоже, о чем будет рассказано в конце этой заметки.

Поддержка проектов выводит эту программу в категорию редакторов развитых, делая его способным к обработке серии взаимосвязанных файлов. А встроенный эмулятор терминала полезен не только программистам, но незаменим также для линуксописателей. Автодополнение языковых конструкций (имеются ввиду языки программирования и разметки) – также функция, подчас не лишняя для простых юзеров, имеющих дело, например, с созданием HTML-документов.

Настоящая заметка посвящена общему описанию редактора Geany и методам его использования при работе с обычными текстами и HTML-документами. Не будучи программистами, авторы не затрагивают вопросы применения этой программы в качестве собственно IDE.

Текстовый редактор Notepad++

«**Notepad++**» – замечательная программа, которая позволяет писать код на огромном количестве языков, в том плане, что это будет весьма и весьма удобно. Для таких языков, как **HTML**, **CSS**, **JavaScript** и даже **PHP** – очень хороший выбор.

Sublime Text 2 – это платный текстовый редактор, написанный на C++, который:

- Работает в Linux, OS X и Windows
- Обладает приличной скоростью работы
- Приятным интерфейсом (включая всевозможные анимации)
- Гибко настраиваем (правда, не в GUI, а в json-конфигах)
- Имеет множество плагинов, число которых растёт как на дрожжах
- Поддерживает VIM-режим
- Использует fuzzy-поиск

Рекомендуемая литература:

1. Хоган, Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения / Б. Хоган. - М.: Питер, 2014. - 591 с.

Практический блок

– **Подготовка к практическим занятиям с использованием методических рекомендаций преподавателя, оформление практических работ.**

Цель: поиск и конспектирование информации по исходным данным.

Задание:

1. Оформить практические задания, согласно требованиям
2. Сделать выводы по занятию
3. Оформленную практическую сдать преподавателю.

Критерии оценивания

«Отлично» – работа выполнена с учетом всех требований, как к ее содержанию, так и оформлению.

«Хорошо» – работа выполнена и оформлена грамотно и правильно, но в ней встречаются некоторые неточности.

«Удовлетворительно» – к работе предъявлены 2 существенных замечания по содержанию и оформлению.

Раздел 1. Инструментарий Web-разработки

Тема 2.1 Визуальная среда Web-разработки WYSIWYG в соответствии с Worldskills по компетенции «Веб разработка». Adobe Dreamweaver Формируемые компетенции: ОК01- ОК09, ПК 2.1-ПК 2.5 Кол-во часов: 2 часов

Теоретический блок

Дримвейвер – это программа для создания веб-сайтов. Изучение этой программы может оказаться довольно сложной задачей, особенно если вы – новичок в веб-дизайне и не знаете языка html.

При первом запуске программы вы увидите окно приветствия, которое выглядит довольно устрашающе. Кажется, что там целая тысяча различных инструментов, которые невозможно изучить. Но в этой статье мы остановимся лишь на нескольких функциях программы, которые полезны как профессионалам веб-дизайна, так и новичкам. Цель этой статьи состоит в том, чтобы обобщить факты и прийти к выводу, является ли Дримвейвер просто редактором кода или это жизненно необходимый инструмент для создания сайтов.

Дримвейвер. Для чего нужна эта программа?

Дримвейвер является одним из флагманов в области программ компании Adobe. В какой-то степени он объединяет и другие программы от Adobe. Например, сначала дизайнеры создают дизайн будущего сайта в программе Photoshop или Fireworks, затем разрезают картинку, подготавливая ее к верстке, и далее формируют непосредственно html-код шаблона, где как в Дримвейвере. То же самое касается и флэш-роликов, которые после разработки в Adobe Flash попадают в Дримвейвер для внедрения в сайт.

Основная функциональность программы сосредоточена на создании HTML и CSS кода, а также работе с файлами сайта. Визуальный интерфейс программы позволяет работать с этими вещами быстро и интуитивно понятно. Вы можете наглядно видеть результат вашего кода, однако отображаемый в программе результат не всегда соответствует действительности. А если же внимательно и точно писать код, то сторонники написания кода в блокноте могут сказать, что по сути код можно качественно писать и в любой другой программе, зачем для этого использовать Дримвейвер? Нужно ли в таком случае все многообразие его инструментов? Нужен ли по такой огромной цене (около 800 долларов за версию CS5) просто редактор кода и файловый менеджер?

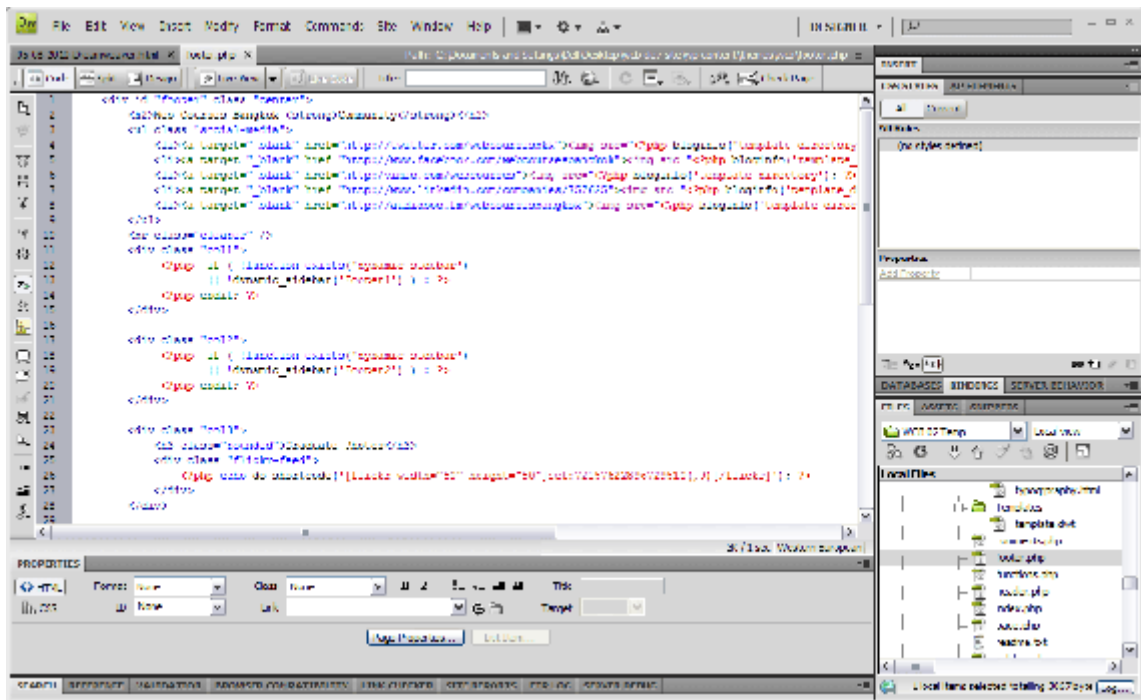
Давайте попробуем разобраться и выделим десять плюсов и десять минусов этой программы.

10 плюсов программы Adobe Dreamweaver.

1. Подсветка кода.

Эта функция поможет вам быстро просканировать ваш код, а также найти возможные ошибки. Каждый тип кода имеет различную подсветку, что позволит вам наглядно отличать HTML, CSS, PHP, Javascript и др.

Это очень удобно, когда ваша веб-страница представляет собой сочетание html и динамических языков, таких как например PHP или ASP.



Для начинающих веб-разработчиков подсветка кода помогает лучше понять каждый элемент кода, а также помогает выявить возможные ошибки. Например, ниже мы видим, что в написании атрибута пропущены кавычки, и в результате, благодаря подсветке (код подсвечен светло-голубым цветом), мы отслеживаем код чуть выше и сразу видим, где была допущена ошибка.

```

1 <div id="footer" class="center">
2 <h2>Web Courses Bangkok <strong>Community</strong></h2>
3 <ul class="social-media">
4 <li><a target="_blank" href="http://twitter.com/webcoursesbkk">
5 <li><a target="_blank" href="http://www.facebook.com/webcoursesbkk">
6 <li><a target="_blank" href="http://vimeo.com/webcourses"><img src="" alt="Vimeo icon" />
7 <li><a target="_blank" href="http://www.linkedin.com/companies/webcoursesbkk">
8 <li><a target="_blank" href="http://audioboo.fm/webcoursesbangkok">
9 </ul>

```

```

1 <div id="footer" class="center">
2 <h2>Web Courses Bangkok <strong>Community</strong></h2>
3 <ul class="social-media">
4 <li><a target="_blank" href="http://twitter.com/webcoursesbkk">
5 <li><a target="_blank" href="http://www.facebook.com/webcoursesbkk">
6 <li><a target="_blank" href="http://vimeo.com/webcourses"><img src="" alt="Vimeo icon" />
7 <li><a target="_blank" href="http://www.linkedin.com/companies/webcoursesbkk">
8 <li><a target="_blank" href="http://audioboo.fm/webcoursesbangkok">
9 </ul>

```

2. Автоподстановка кода.

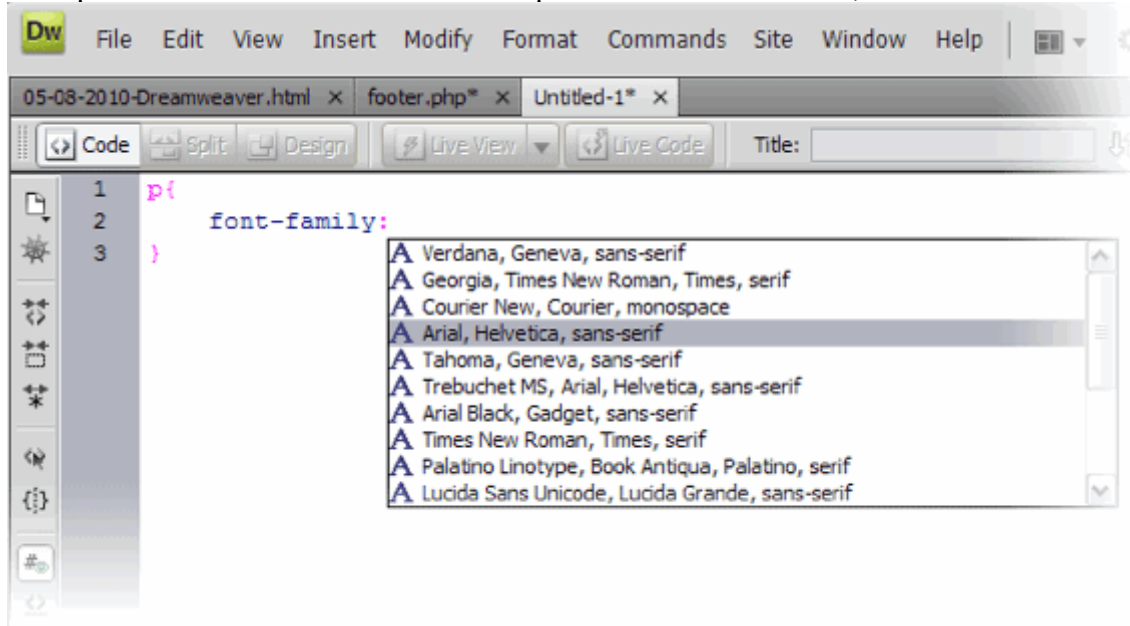
Еще одна полезная функция, которая позволяет экономить время – это автоматическое дописывание кода. Когда вы начинаете писать какие-нибудь html-теги, перед вами открывается список возможных вариантов на основе введенных первых букв. Вам остается лишь выбрать подходящий вариант. Причем это удобно не только для написания самих тегов, но и для их атрибутов.

Возьмем, например, добавление изображения. Как правило, вы должны знать такие вещи, как расположение изображения и его размеры. С помощью Дримвейвера вы можете просто написать `<img`, а далее, как только вы нажмете пробел и напишете букву **S**, то вам мгновенно выведут атрибут `src=""`, далее просто дважды нажмите **Enter** и перед вами откроется окно проводника, чтобы вы могли выбрать подходящее изображение, а далее Дримвейвер автоматически пропишет его путь, а также подставит атрибуты ширины и высоты этой картинки.

Еще один пример этой функции: допустим, вы пишете CSS-код, и вам нужно прописать свойство семейства шрифтов `"font-family:Arial, Helvetica, sans-serif;"`, т.е. нам в данном случае нужно ввести 41 символ. Или же:

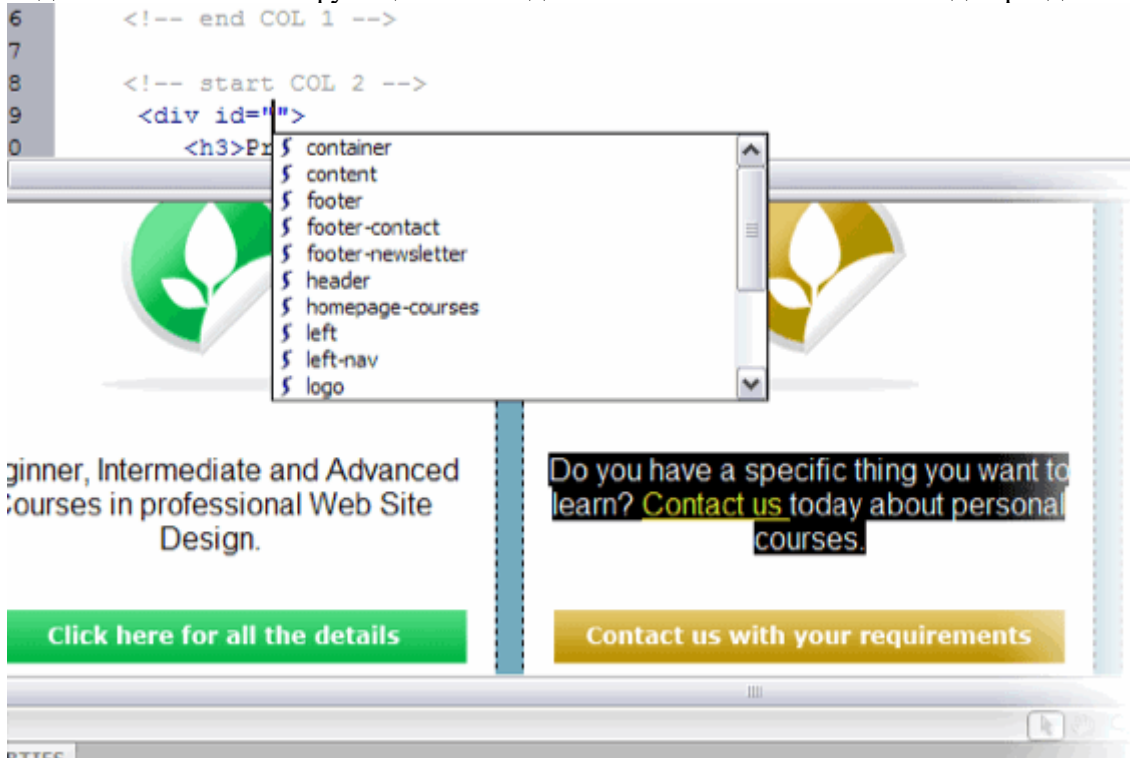
1. пишем **fon** и один раз нажимаем стрелку Вниз на клавиатуре;
2. нажимаем **Enter**, чтобы выбрать опцию "**font-family**";
3. выбираем из списка "**Arial, Helvetica, sans-serif**";
4. добавляем точку с запятой, и все готово!

Набор 41 символа или выполнение 4 простых шагов... Я знаю, что из этого вы выберете!)



Если вы только изучаете HTML, то такая помощь оказывается действительно необходимой, т.к. предоставляет вам полный контроль над кодом, но в то же время дает возможность не ошибиться в написании каких-то деталей.

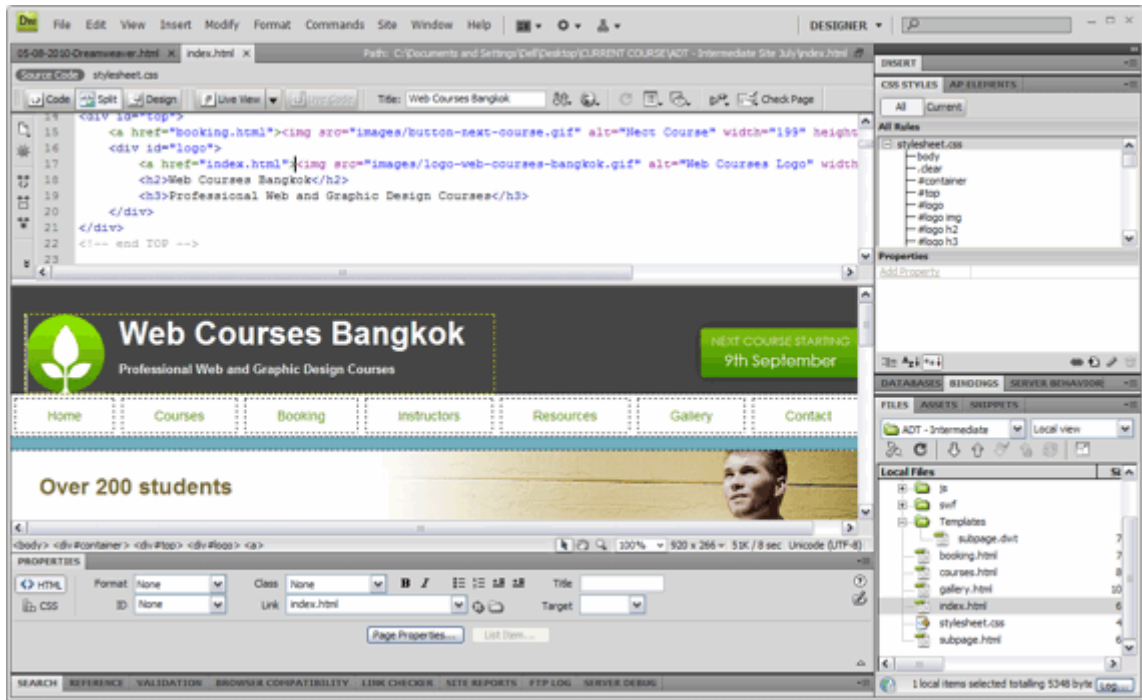
А для более опытных функция автоподстановки позволяет писать код гораздо быстрее.



3. Режим кода и режим дизайна

Всегда здорово видеть, что получается в результате написания того или иного кода, поэтому функция визуального отображения очень полезна. В Дримвейвере есть возможность быстрого переключения режима кода и режима дизайна.

Например, когда мы размещаем блоки div, создавая каркас сайта, нам удобнее работать в режиме кода, а когда нужно добавить на страницы какой-то контент, то в этом случае больше подойдет режим дизайна.

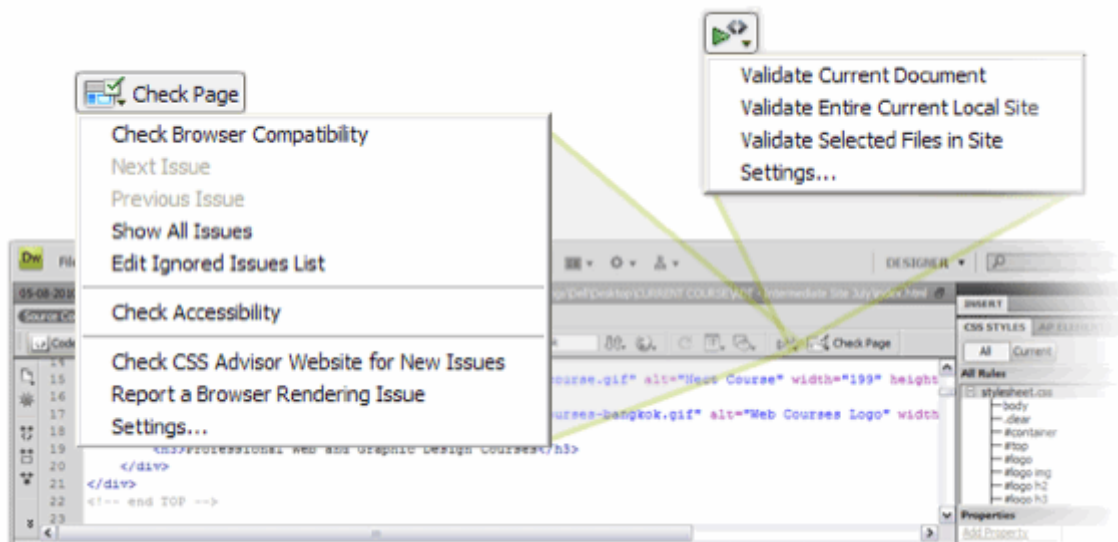


Противники работы в Дримвейвере здесь могут возразить, что реальную работу кода можно увидеть только при просмотре в браузере. Однако все-таки удобно при каких-то основных изменениях кода увидеть сразу этот результат. Зачем запускать браузер, если можно просто кликнуть на вкладку визуального отображения и увидеть изменения?

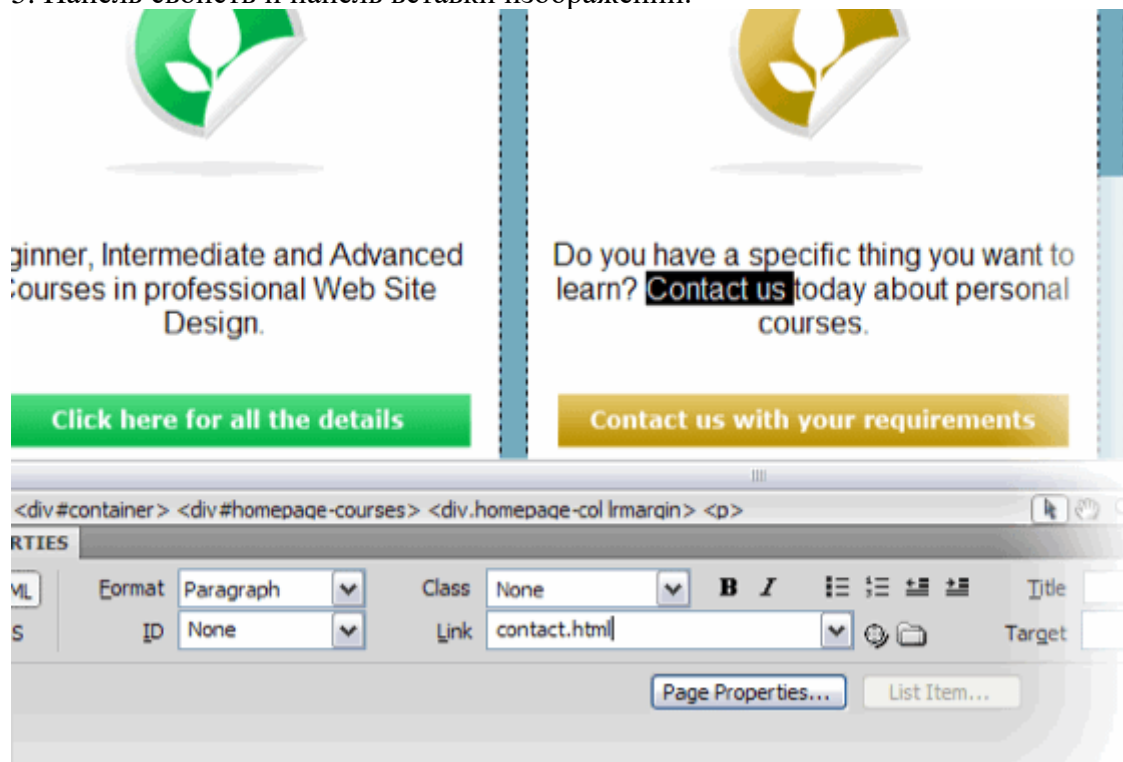
4. Проверка кода.

В Дримвейвере есть отличный инструмент проверки кода, который не позволит ни одному веб-дизайнеру писать код, не соответствующий стандартам.

Ошибки и проблемы совместимости могут быть быстро найдены именно во время написания кода, а не после того, как файлы с кодом уже загружены на сервер и проверены онлайн-валидатором.

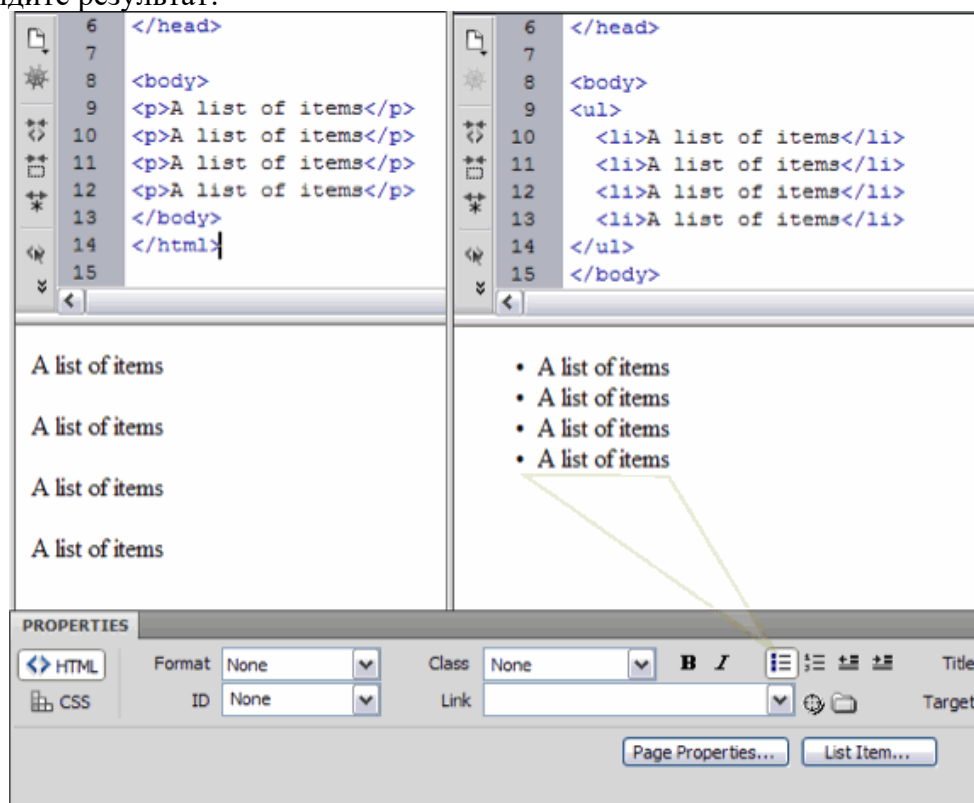


5. Панель свойств и панель вставки изображений.



Панель свойств – это лучший инструмент для быстрого создания ссылок, выделения шрифта полужирным или курсивом, создания нумерованных и ненумерованных списков и т.п. Суть в том, что вы концентрируетесь на содержании при разработке веб-сайта, а не на его коде. Например, просто выделите какой-нибудь текст и напишите ссылку, по которой посетитель вашего сайта должен будет перейти по клику.

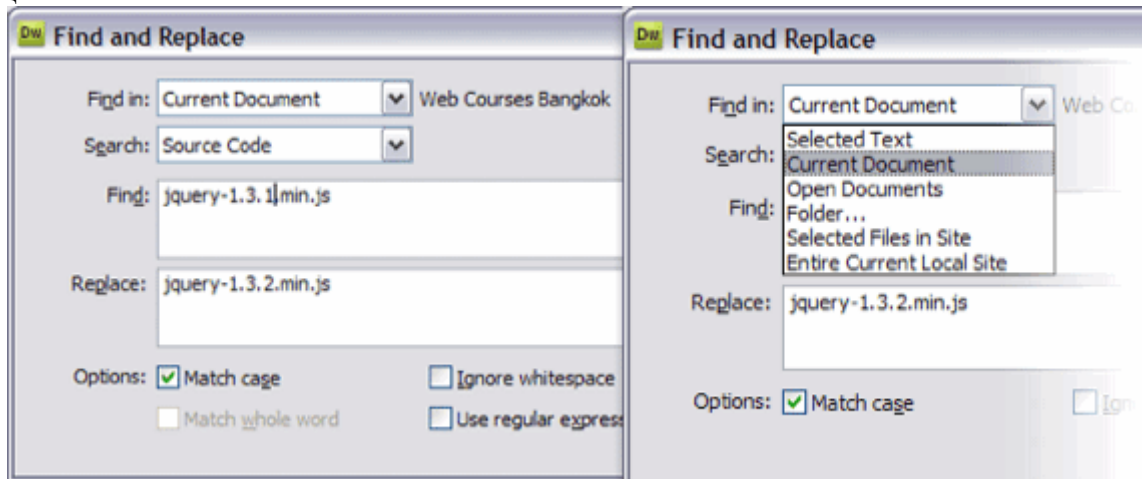
Или еще пример: если вам кажется, что эти три абзаца будут лучше смотреться в виде списка, выделите их и нажмите соответствующую иконку на панели свойств и... через мгновение вы увидите результат!



6. Глобальная функция "Найти/Заменить".

Это может спасти часы драгоценного времени. Вы можете находить и заменять код не только на одной странице, но и полностью на сайте. Но на этом чудеса не заканчиваются: вы мо-

жете осуществлять поиск только по тексту на странице, только по исходному коду и даже только по специальным тегам.



Например, компания, которой вы делаете сайт, поменяла в своем названии литеры LTD на PLC. Без Дримвейвера вам бы пришлось перебирать сотни или тысячи страниц вашего сайта, а с Дримвейвером вы сможете все заменить за пару секунд.

Или еще вариант: например, вам нужно обновить версию jQuery на статичном сайте. Нажимаем Ctrl+F, чтобы вызвать окно поиска/замены, в поле "Найти" вводим "jquery.1.XXXXXXX" а в поле "Заменить" - "jquery.XXXXXX", выбираем опции "Искать на сайте" и "Искать в исходном коде" и кликаем кнопку "Заменить все". Через две секунды все страницы вашего сайта обновлены и готовы к загрузке на сервер вместе с последней версией JQuery.

Но прежде чем запускать функцию "Найти/Заменить", пожалуйста, будьте очень аккуратны и внимательны, дважды все проверив, потому что отменить результат применения этой функции нельзя. И наконец, если вдруг вы захотите использовать это для замены названий файлов, прочитайте следующий плюс программы Дримвейвер.

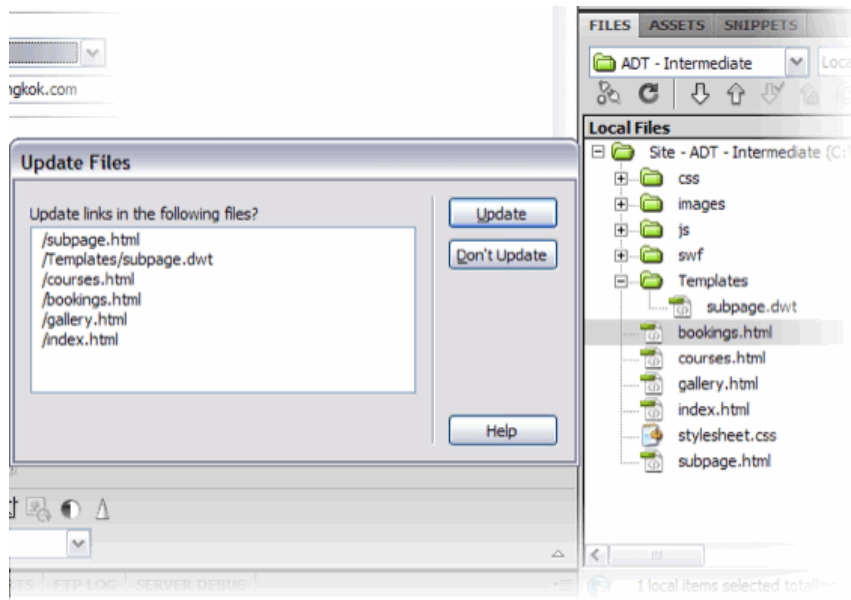
7. Файловый менеджер.

Система работы с файлами – это, наверное, одна из главных особенностей, которая превращает Дримвейвер в нечто большее, что просто редактор кода.

Прежде всего определите расположение вашего сайта (**Site > New Site > Имя сайта > Затем выберите папку в которой находится ваш локальный сайт**) и дальше позвольте Дримвейверу позаботиться о всех файлах вашего сайта.

Если вы измените имя какого-либо файла, то Дримвейвер предложит вам обновить ссылки. Например, если вы изменили имя файла "about-web-courses-bangkok.html" на просто "about.html" Дримвейвер обновит все ссылки, которые должны быть связаны с этим файлом.

Это также относится и к файлам контента, например при изменении имени картинки с "dc2234.jpg" на "product-name-2345.jpg" все ссылки, ведущие к этому изображению будут немедленно обновлены.

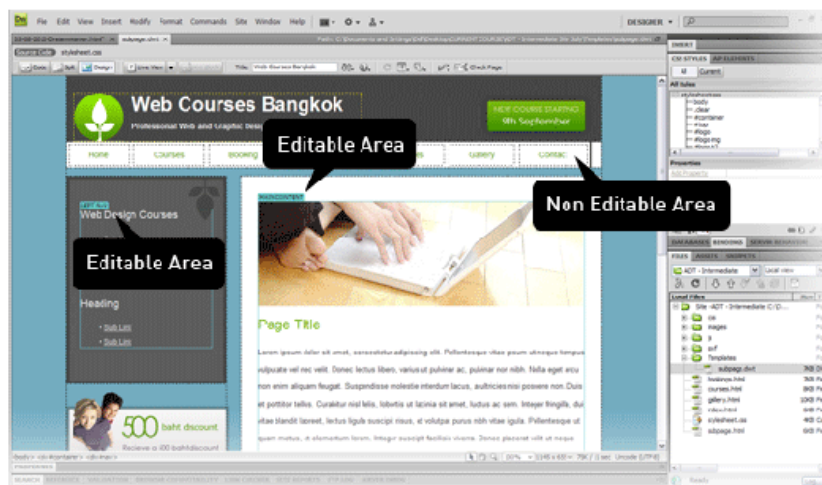
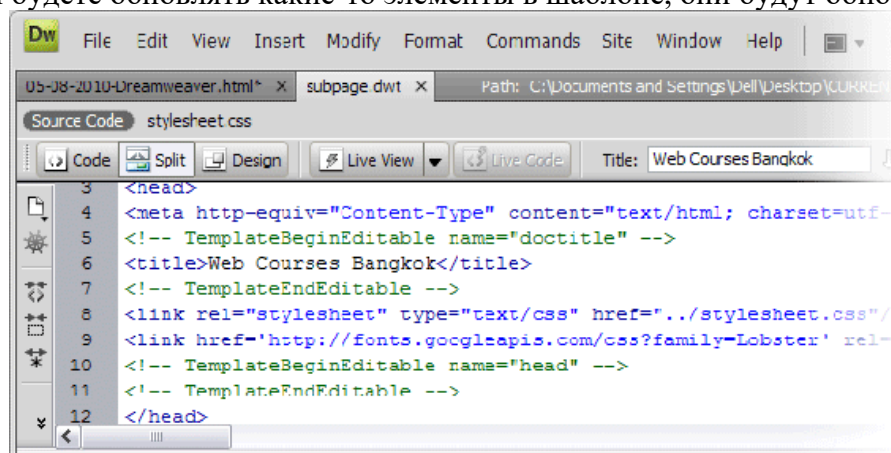


Также файловый менеджер может соединиться по ftp с вашим удаленным сервером и при сохранении изменений в каком-либо файле этот файл будет автоматически загружаться на сервер. Еще вы можете включить отображение всех удаленных файлов и работать непосредственно с файлами на удаленном сервере.

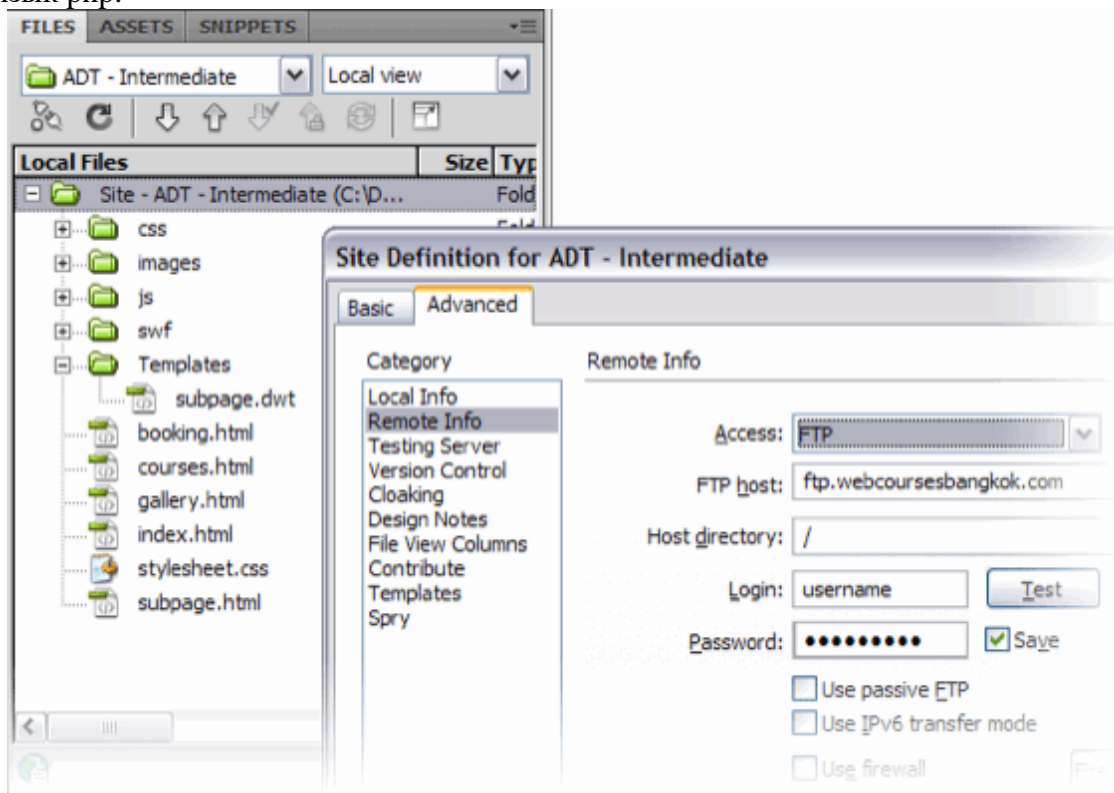
8. Шаблоны.

Возможно, вы захотите создавать все больше и больше сайтов. И следующий шаг к созданию CMS – это использование шаблонов. Вы создаете одну страницу, дизайн которой будет использоваться для всего сайта, и превращаете ее шаблон, выбирая те области, которые будут редактируемыми и те области которые будут одними и теми же на всем сайте.

Таким образом, через функцию **"Save as"** можно создать из шаблона новую страницу. Когда вы будете обновлять какие-то элементы в шаблоне, они будут обновляться на всех страницах.



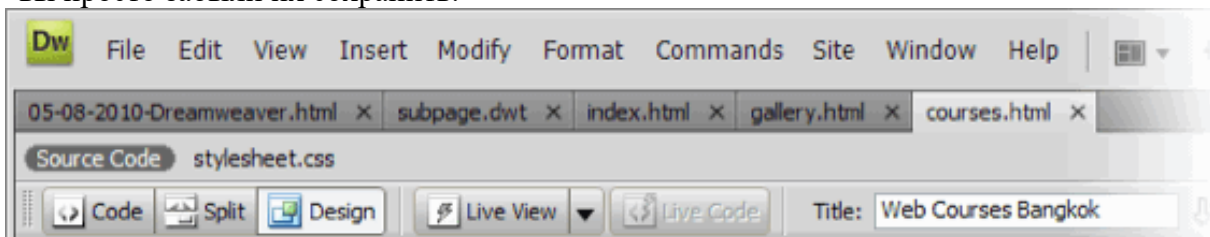
Следовательно, у вас получится подобие динамического сайта, когда обновляя одну страницу, вы обновляете все остальные. И в то же время в данном случае вам нет необходимости знать язык php.



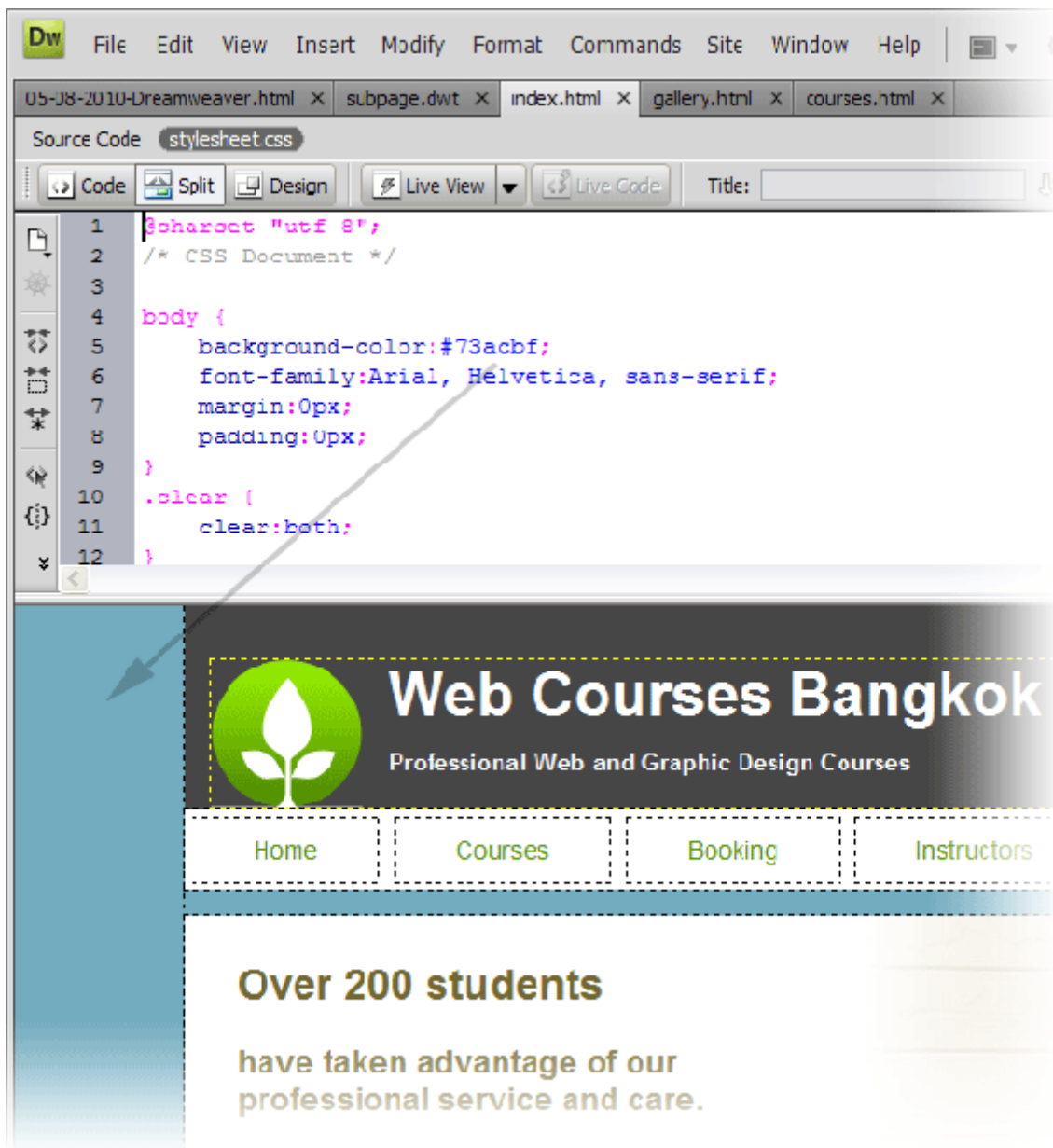
9. Вкладки и связанные файлы.

Вкладки были представлены начиная с версии CS3 Дримвейвера и значительно упростили работу с текущими файлами. Это очень удобно, когда вам приходится ставить ссылки с одной страницы на другую, а также перемещать контент из одного места в другое.

Еще одной действительно полезной вещью является появление значка * на вкладке тех файлов, изменения в которых вы уже сделали, но еще их не сохранили. Таким образом, вы можете сразу понять, почему те изменения, которые вы уже сделали в редакторе не отображаются в браузере – вы просто забыли их сохранить.



Начиная с версии CS4 появилась возможность видеть все подключенные к странице файлы, такие как файлы таблиц стилей или яваскрипт-файлы. Они отображаются чуть ниже вкладок файлов. Это позволяет быстро переключиться например на css-файл, подключенный к редактируемой странице. Также в режиме split (одновременное отображение режима кода и режима визуального отображения) очень удобно видеть изменения на странице при редактировании css-стилей.



10. Новые интересные возможности, появившиеся в версии CS5.

— Больше нет головной боли из-за различного отображения сайта в разных браузерах.



Когда вы занимаетесь разработкой сайта, то вы должны проверять его отображение во всех браузерах. Возможно, вам не удастся добиться абсолютной идентичности отображения, но в любом случае вы должны проверить, что все блоки и элементы стоят на своих местах.

BrowserLab когда-то был отдельным сервисом, разработанным Adobe, который позволял проверить сайт во всех браузерах, а теперь этот сервис внедрен в CS5, чтобы вы смогли проверить, все ли в порядке с дизайном до запуска сайта.

"Предпросмотр динамических страниц и локального контента с различными видами отображения, возможностью диагностики и инструментами для сравнения" – Adobe о сервисе BrowserLab.

— Поддержка "большой тройки" CMS.

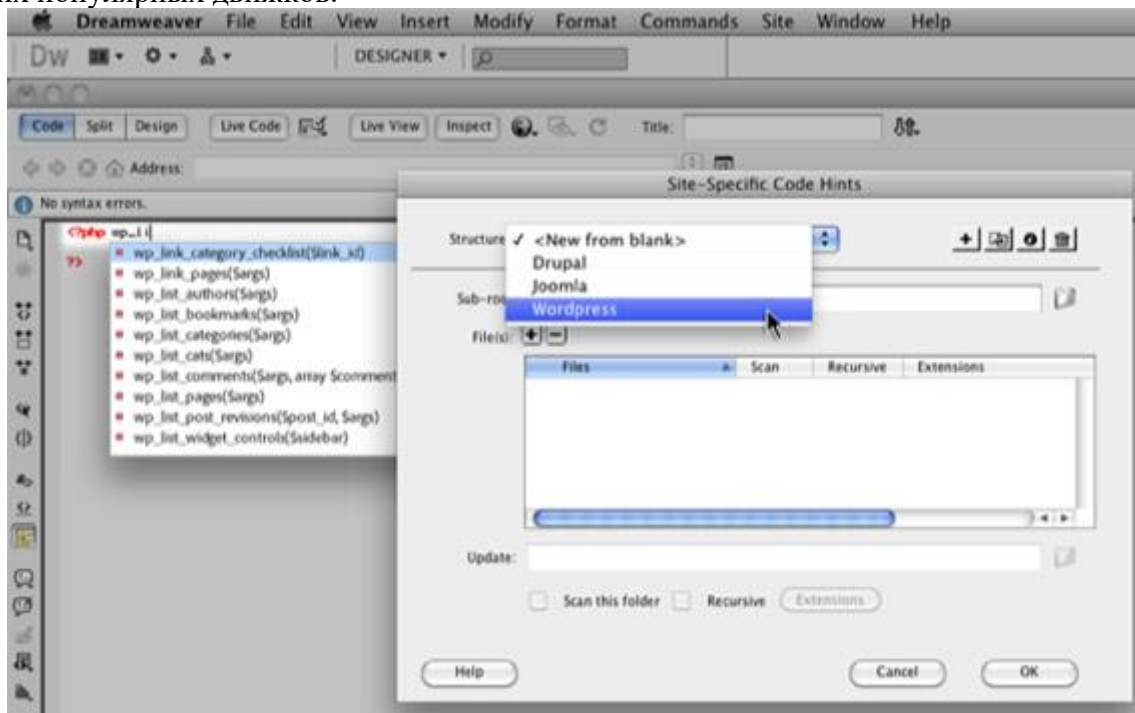


Единственный способ эффективно управлять большими сайтами – это, конечно, использование CMS. Просто-напросто непрактично делать каждую страницу в Дримвейвере, затем загружать ее на сервер с остальным контентом (изображения и т.п.). Как раз в этом случае на помощь приходят CMS.

До появления версии Дримвейвера CS5 мы должны были создать весь код отдельно для нашего шаблона, загрузить его и затем проверить онлайн непосредственно на движке, как сделанные изменения работают.

Теперь же Дримвейвер может продемонстрировать результат в режиме Live View, вместе с данными, загруженными из базы данных движка.

Кроме того, как мы уже упоминали, Дримвейвер имеет возможность автоподстановки кода для этих популярных движков.

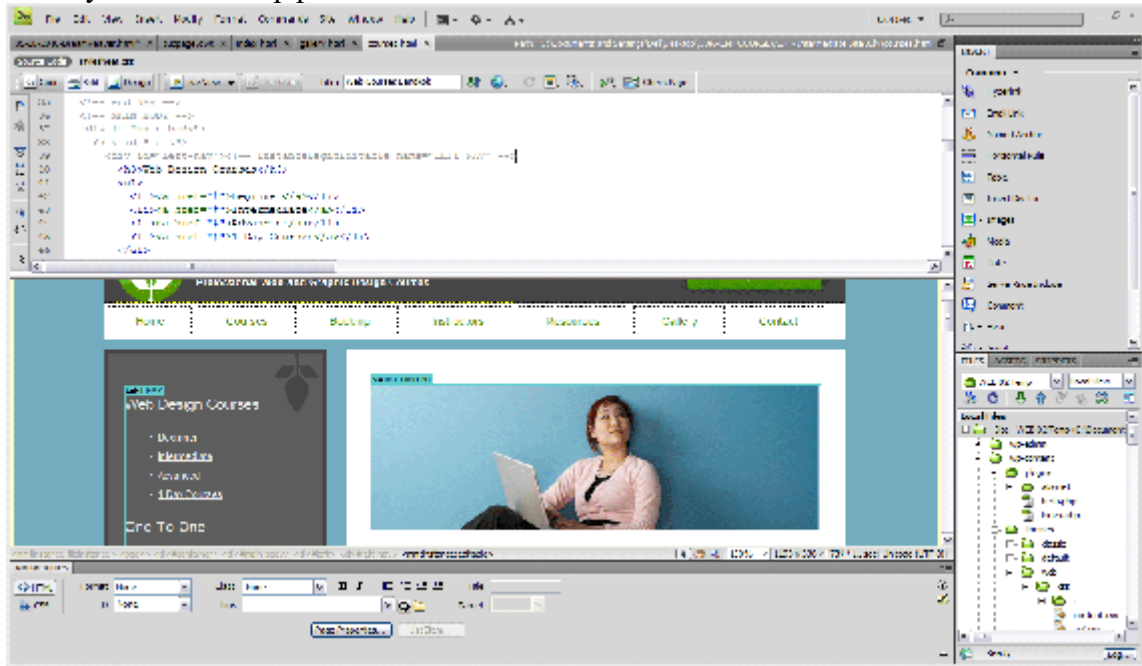


Вот еще некоторые новые возможности CS5:

- Возможность экспериментирования с HTML5;
- Возможность подсветки собственных php-классов и функций, что позволяет писать код более аккуратно;
- Интеграция со службой Business Catalyst, которая является сервисом для размещения чего угодно: от великолепных веб-сайтов и до мощных онлайн-магазинов.

10 минусов программы Adobe Dreamweaver.

1. Запутанный интерфейс.

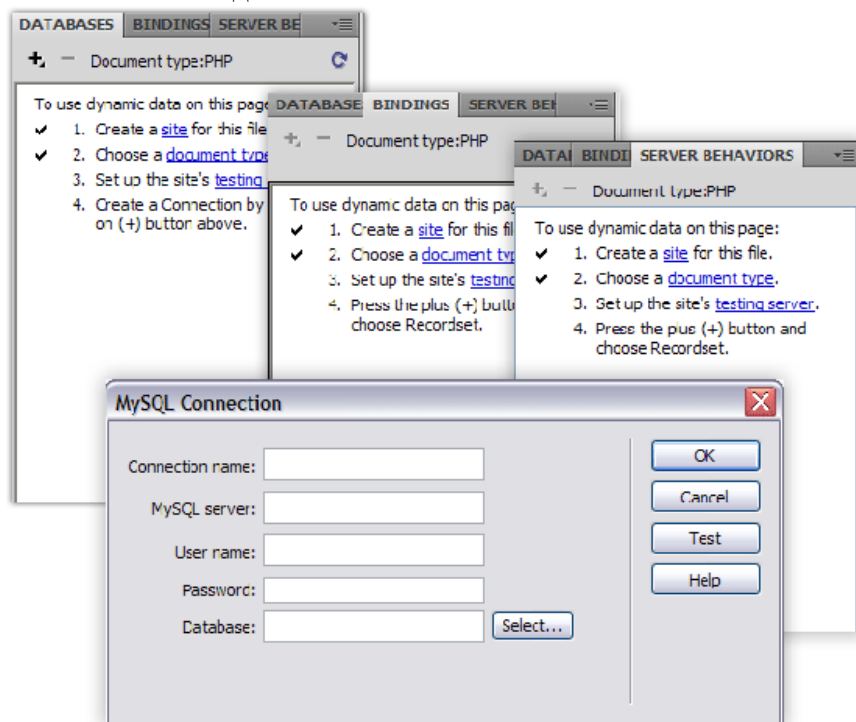


В самом верху программы мы видим 15 опций: это и пункты меню и иконки быстрого изменения интерфейса программы. Чуть ниже находятся еще от 5 до 50 опций (в зависимости от количества открытых файлов), а далее мы видим еще 15 элементов. Таким образом, перед нами одновременно находятся более 35 элементов, с помощью которых мы можем выполнять те или иные функции программы. Не каждый может удержать в голове значение всех этих элементов.

2. Непродуманная система обучения.

Если вы предпочтете изучать работу в программе Дримвейвер самостоятельно, то это будет очень непросто. Несмотря на то, что большинство функций программы сгруппированы логично и последовательно, все же некоторые элементы могут поставить в ступор даже профессионала.

Инструменты программы предназначены для реализации разнообразных задач при построении сайта. Например, вы захотите добавить данные соединения с базой данных вашего сайта, который вы разрабатываете на локальном сервере. Но если вы новичок, то это будет для вас практически невыполнимая задача.



Нужно признаться, что интерфейс Дримвейвера может быть удобен лишь немногим профессионалам, создающим мощные сайты и постоянно использующим все функции программы. А для рядовых дизайнеров все эти иконки интерфейса просто никогда не понадобятся. Например, в стандартном интерфейсе есть панели для работы с базами данных и с сервером, но практически никто их не использует (тем более, что со скриптами, выполняющимися через сервер, Дримвейвер работает ужасно).

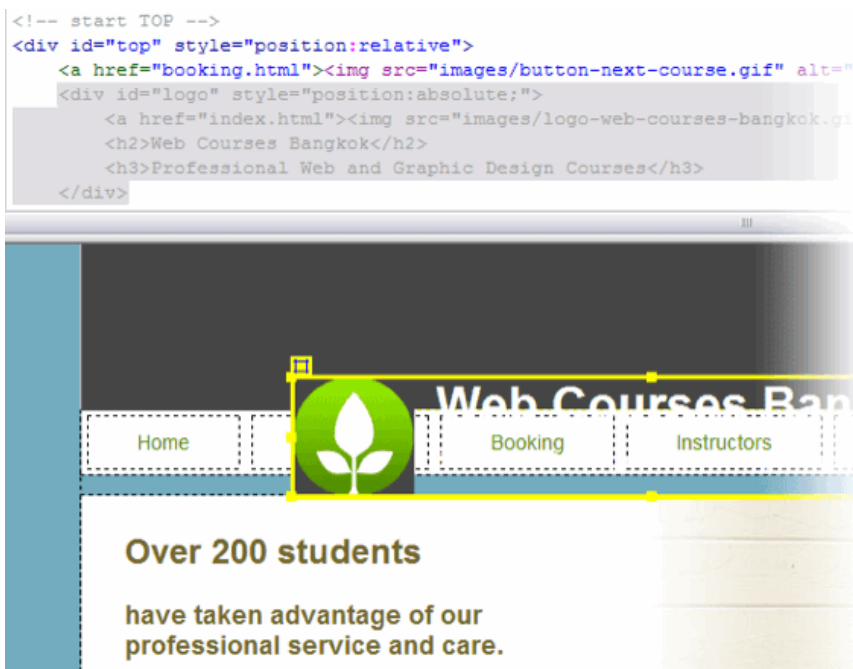
Как и при работе с любым многофункциональным и мощным инструментом, желательно, чтобы кто-то показал или научил вас, как с ним работать. Как вариант, можно купить книги по этой теме. Но проблема в том, что автор книги обзорно освещает все инструменты программы, не подозревая о том, какие конкретно из инструментов нужны именно вам.

Пошаговые уроки уже более целенаправлены, например **"Как создать сайт-портфолио"** или **"Куда ввести название сайта"**, но все же лучший вариант – это обучение у веб-дизайнера, который использует Дримвейвер в своей повседневной работе. Такие люди точно знают, какие инструменты программы могут пригодиться, и как использовать их наиболее эффективно. Таким образом вы будете концентрироваться именно на конкретной задаче, а не на абстрактных функциях того или иного инструмента.

3. Погрешности визуального редактора.

Как мы уже говорили выше, нельзя доверять отображению сайта в визуальном редакторе Дримвейвера, т.к. это все-таки не браузер. От версии к версии разработчики программы стараются сделать отображение кода в режиме дизайнера более точным, но все же пока оно далеко от реальности.

В частности проблемы наблюдаются с отображением элементов с абсолютным позиционированием, где это отображение может отличаться от заданного на 20 пикселей и даже больше. А говорить о качественном отображении результатов работы php-кода и javascript-кода вообще не приходится.



Поэтому вам всегда нужно проверять отображение и работу сайта именно в браузере, а лучше в нескольких.

4. Громоздкий динамический код.

Дримвейвер имеет возможность создания динамических сайтов, если вы работаете с локальным или удаленным сервером. Единственная проблема в том, что код, который создает дримвейвер (точно так же, как и встроенные сниппетты, о которых мы поговорим в следующем пункте), перегружен и сложен. Гораздо лучше писать код вручную или использовать готовые фреймворки такие как CakePHP, CodeIgnitor или Zend.

5. Плохо написанные сниппеты.

Заготовки участков кода в Дримвейвере были не самыми лучшими, еще когда Дримвейвер принадлежал компании Macromedia, но даже и сейчас в версии CS5 они не стали лучше.

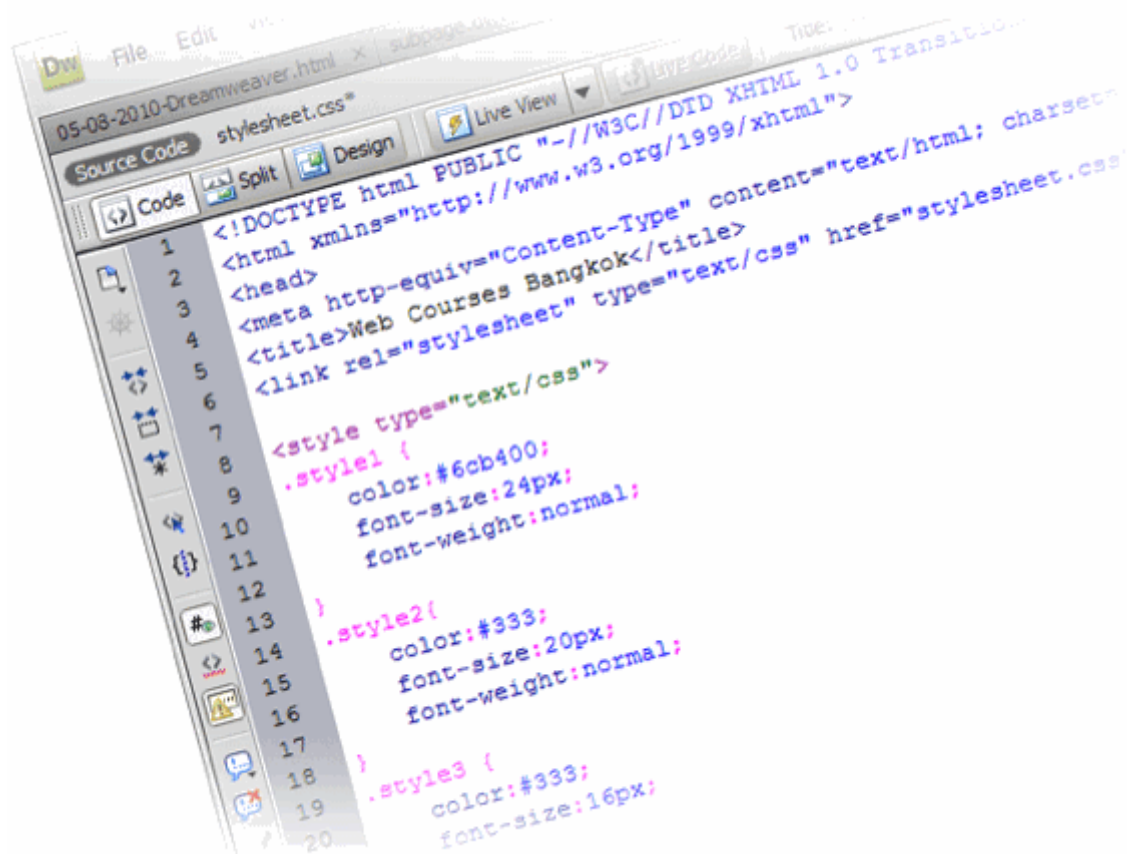
Причина в том, что эти заготовки кода неоправданно большие. Например, чтобы вставить проверку всего одного поля формы, Дримвейвер предложил нам вставить целых 19 строк javascript-кода.

```
7 <!-- InstanceEndEditable -->
8 <link rel="stylesheet" type="text/css" href="stylesheet.css"/>
9 <link href="http://fonts.googleapis.com/css?family=Roboto:400,700&subset=latin,latin-ext" rel="stylesheet" type="text/css">
10 <!-- InstanceBeginEditable name="Form" -->
11 <script type="text/javascript">
12 <!--
13 function MM_validateForm() { //v4.0
14   if (document.getElementById){
15     var i,p,q,nm,test,num,min,max,errors='',args=MM_validateForm.args;
16     for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=document.getElementById(test);
17       if (val) { nm=val.name; if ((val=val.value)!="") {
18         if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
19           if (p<1 || p==(val.length-1)) errors+='- '+nm+' must contain an email address.\n';
20         } else if (test!='R') { num = parseFloat(val);
21           if (isNaN(val)) errors+='- '+nm+' must contain a number.\n';
22           if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
23             min=test.substring(8,p); max=test.substring(p+1);
24             if (num<min || max<num) errors+='- '+nm+' must contain a number in the range '+min+' to '+max.\n';
25           } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' is required.\n';
26         } if (errors) alert('The following error(s) occurred:\n'+errors);
27         document.MM_returnValue = (errors == '');
28       } }
29   //-->
30 </script>
31 <!-- InstanceEndEditable -->
32 </head>
33 <body>
```

The code added just to validate one field!

6. Неопределенность в написании CSS-стилей (до версии CS4).

С этой проблемой приходится сталкиваться снова и снова. Используя панель свойств, можно оформлять участки кода при помощи CSS-стилей, однако новички слишком увлекаются этим, и в итоге получаются сотни строк кода с неопределенными стилями типа style1, style2 и т.д. Причем все стили автоматически прописываются Дримвейвером между тегами веб-страницы.



Позволяя Дримвейверу прописывать стили за вас, вы теряете контроль над кодом и, что еще хуже, вполне могут появиться конфликты стилей. Самое лучшее решение – это держаться по-

далее от стилей, которые предлагает Дримвейвер и полностью взять эту функцию на себя, прописывая те свойства, которые нужны именно вам, в которых вы не запутаетесь. И, что важно, стили должны быть в отдельном файле стилей, а не в том же файле, что и html-код.

К счастью Adobe убрали эту функцию в последних версиях программы.

7. Неполный контроль над кодом.

Использование визуального редактора – это самый простой путь потерять контроль над кодом. Когда люди используют только режим дизайна, то их код перегружается тегами, которые автоматически прописываются где только можно. Особенно много встречается пустых строк, оформленных таким образом:

```
<p><strong><strong>Текст</strong></strong></p>
<p><strong>.</strong></p>
<p> </p>
```

Еще одним распространенным примером является вставка изображений. Если вы редактируете какой-то текст, и вам нужно вставить изображение, то вставляйте его, не нажимая перед

этим клавишу **Enter**. В противном случае тег `` будет заключен в теги абзаца, что является абсолютно лишним.

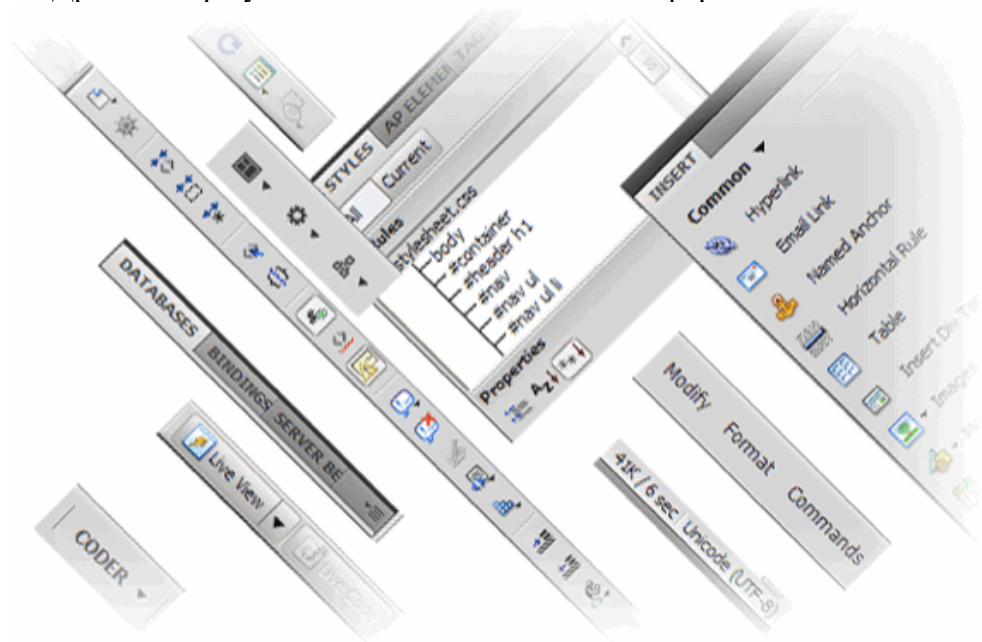
```
<body>
<div id="container">

<div id="header">
  <p></p>
  <h1>My First Web Site</h1>
</div>
```

Поэтому не забывайте всегда контролировать код, прописанный визуальным редактором, и, в случае необходимости, корректируйте его.

8. Действительно полезными являются только 10% функционала.

Вы только взгляните на интерфейс программы, как уже упоминалось, вы увидите больше 35 различных кнопочек, меню, иконок, большая часть из которых никогда вам не пригодится. Такое чувство, что Дримвейвер буквально напичкан лишней информацией.



Все веб-дизайнеры лишь хотят иметь удобный инструмент для написания кода, а кнопочки для каждого случая жизни им не нужны. Понятно, что некоторые панели можно просто отключить, но это лишние телодвижения.

Почему бы не начать, наоборот, с простого интерфейса, дав возможность профессионалам самостоятельно включать необходимые им панели и пиктограммы?

9. Дримвейвер развивает в веб-дизайнере лень.

Если вы не будете осторожным, то Дримвейвер сделает вас ленивым. Хотя бы взглянуть на многие школы, в которых обучают только работе с режимом "Дизайн" при разработке так называемых "профессиональных" сайтов. Это заставляет ужаснуться. Являясь инструментом для разработки кода, в данном случае Дримвейвер не выполняет эту функцию.

Нужно акцентировать внимание именно на кодировании, чтобы все остальные инструменты способствовали и помогали этому. Не полагайтесь на автоматические функции дримвейвера.

Разработка html-кода – это основа основ при разработке сайта, поэтому не стоит для его формирования необоснованно использовать иконки или пункты меню.

10. Очень дорого.

Конечно понятно, сколько работы вложено в разработку Дримвейвера, но в любом случае его цена в \$654,72 – это очень дорого.

Кроме того, при разработке сайта, вам, скорее всего, понадобится Фотошоп – а это еще 1456,14 долларов. Итого получается, что вам понадобится 2110,86 долларов, чтобы приступить к работе.

К счастью, существуют бесплатные альтернативы. Для фотошопа – это программа Gimp, а для дримвейвера такой альтернативой является Aptana. Вывод. Так что же такое Дримвейвер? Неужели это просто редактор кода с завышенной ценой? Или все-таки что-то большее?

Безусловно, Дримвейвер – очень мощный инструмент, но в то же время эта мощь является его "Ахилесовой пятой". Несмотря на то, что он насыщен функционалом, большая часть этих функций никогда не будет востребована. Т.е. вы переплачиваете огромную цену за те функции, которыми не будете пользоваться.

Но в последнее время специалисты Adobe стараются добавлять именно те функции, которые будут полезны. Например, сниппеты для работы с CMS.

Для начинающих Дримвейвер удивительно удобен, но лишь в случае, если вы уже неплохо знаете html и css-код, а не слепо доверяете инструментам программы, которые прописывают часть кода автоматически.+

Конечно, эту программу можно рекомендовать и начинающим пользователям, и профессиональным веб-разработчикам. Но все же хотелось бы иметь и более упрощенную версию этого редактора, как, например, Elements является упрощенной версией Фотошопа.

Конечно, это не просто редактор кода с завышенной ценой. Это, действительно, мощный инструмент, с которым должен познакомиться каждый веб-дизайнер.

Рекомендуемая литература:

2. Дронов, Владимир Самоучитель Adobe Dreamweaver CS5.5 / Владимир Дронов. - М.: БХВ-Петербург, 2012. - 384 с.
3. Adobe Dreamweaver CS6. Официальный учебный курс (+ CD-ROM). - М.: Эксмо, 2013. - 496 с.
4. Цифровой дизайн. Основы веб-проектирования с помощью инструментов Adobe. - М.: БХВ-Петербург, 2011. - 768 с.
5. Adobe Dreamweaver CS. Официальный учебный курс (+ CD-ROM): — Санкт-Петербург, Эксмо, 2014 г.- 496 с.
6. Черников, С. В. Dreamweaver CS3. Строим Web-сайты (+ CD-ROM) / С.В. Черников, Ф.А. Резников. - М.: 100 книг, Триумф, 2014. - 256 с.

Практический блок

– **Разработка своего сайта в Adobe Dreamweaver в соответствии с текущим стандартом W3C (2ч)**

Задание:

Разработка веб-сайта по любой выбранной теме, со следующими требованиями:

1. Аккуратная верстка. Блочная или табличная, предпочтительнее блочная.

2. Как минимум два уровня доступа пользователей: «Администратор» и «Обычные пользователи».

3. Как минимум пять основных страниц, реализующих описанные ниже возможности.

W3C-стандарты - это общий набор «правил» для браузеров с указанием, как использовать и показывать код страницы. Главная организация - это **World Wide Web Consortium (W3C)**, чей директор Тим Бернерс-Ли активно участвовал в развитии WWW с 1989 года.

Спросите бывалых веб-дизайнеров, что такое веб-стандарты, и вы получите пять разных ответов. Но все они основаны на следующих утверждениях:

- **Чистый код, соответствующий стандартам W3C, html, или xhtml**

Удостоверьтесь, что у вас чистый код, соответствующий стандартам W3C. Это легко проверить с помощью сайта <http://validator.w3.org/>. Убедитесь, что вы определили правильный DOCTYPE. Более подробно об этом можно прочитать в этой статье.

- **Семантически верный код**

Удостоверьтесь, что (x)html описывает только содержимое, но не внешний вид содержимого. Это означает использование тегов h1/h2 и др. и использование таблиц для представления табличных данных, но не для представления всей страницы.

- **Cascading Style Sheets (CSS) - каскадные таблицы стилей**

Используются для форматирования внешнего вида различных элементов веб-страницы: цвета, фона, шрифта и т.д. Они параллельны (x)html-коду и находятся отдельно. Лучший пример - CSS Zen Garden, где используется один xhtml файл и сотни css-стилей. Результат: одно содержимое, но различные стили оформления.

Сегодня, проектирование сайтов на Joomla!, соответствующих веб-стандартам - это большие трудности. Дело в том, что вся линейка Joomla! 1.0.x использует достаточно большое количество таблиц для представления содержимого. Но, даже в таком случае можно достигнуть результата, когда шаблон будет соответствовать W3C-стандартам.

Критерии оценивания

«Отлично» – работа выполнена с учетом всех требований, как к ее содержанию, так и оформлению.

«Хорошо» – работа выполнена и оформлена грамотно и правильно, но в ней встречаются некоторые неточности.

«Удовлетворительно» – к работе предъявлены 2 существенных замечания по содержанию и оформлению.

– **Подготовка к практическим занятиям с использованием методических рекомендаций преподавателя, оформление практических работ.**

Цель: поиск и конспектирование информации по исходным данным.

Задание:

4. Оформить практические задания, согласно требованиям
5. Сделать выводы по занятию
6. Оформленную практическую сдать преподавателю.

Критерии оценивания

«Отлично» – работа выполнена с учетом всех требований, как к ее содержанию, так и оформлению.

«Хорошо» – работа выполнена и оформлена грамотно и правильно, но в ней встречаются некоторые неточности.

«Удовлетворительно» – к работе предъявлены 2 существенных замечания по содержанию и оформлению.

5. ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ УЧЕБНЫХ ИЗДАНИЙ, ИНТЕРНЕТ-РЕСУРСОВ, ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ.

Основная

1. Снелл Абсолютно ясно о создании Web-страниц и Web-сайтов / Снелл, Нэд. - М.: Триумф, 2012. - 224 с.
2. Хассей, Трис WordPress. Создание сайтов для начинающих (+ CD-ROM) / Трис Хассей. - М.: Эксмо, 2012. - 432 с
3. Дакетт Джон HTML и CSS. Разработка и дизайн веб-сайтов (+ CD-ROM); Эксмо - Москва, 2013. - 480 с.
4. Китинг, Джоди Flash MX. Искусство создания web-сайтов; ТИД ДС - Москва, 2012. - 848 с.
5. Митчелл, Скотт 5 проектов Web-сайтов от фотоальбома до магазина; М.: ИТ Пресс - Москва, 2013. - 224 с.
6. Чебыкин Ростислав Разработка и оформление текстового содержания сайтов; БХВ-Петербург - Москва, 2014. - 528 с.
7. Петюшкин, Алексей HTML в Web-дизайне / Алексей Петюшкин. - М.: БХВ-Петербург, 2013. - 400 с.
8. Хоган, Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения / Б. Хоган. - М.: Питер, 2014. - 591 с.
9. Дронов, Владимир Самоучитель Adobe Dreamweaver CS5.5 / Владимир Дронов. - М.: БХВ-Петербург, 2012. - 384 с.
10. Adobe Dreamweaver CS6. Официальный учебный курс (+ CD-ROM). - М.: Эксмо, 2013. - 496 с.
11. Цифровой дизайн. Основы веб-проектирования с помощью инструментов Adobe. - М.: БХВ-Петербург, 2011. - 768 с.
12. Adobe Dreamweaver CC. Официальный учебный курс (+ CD-ROM): – Санкт-Петербург, Эксмо, 2014 г.- 496 с.
13. Черников, С. В. Dreamweaver CS3. Строим Web-сайты (+ CD-ROM) / С.В. Черников, Ф.А. Резников. - М.: 100 книг, Триумф, 2014. - 256 с.
14. Мак-Дональд, Мэтью Создание Web-сайта. Недостающее руководство / Мэтью Мак-Дональд. - М.: БХВ-Петербург, 2013. - 624 с.

Электронный ресурс

1. Долженко А.И. Технологии командной разработки программного обеспечения информационных систем [Электронный ресурс]. /А.И. Долженко. – М.: ИНТУИТ, 2013. - 519 с. <http://www.knigafund.ru/books/173587>

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Дмитровградский инженерно-технологический институт –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
(ДИТИ НИЯУ МИФИ)

Реферат

на тему: «**Название темы**»

по дисциплине

«**МДК.02.02 Инструментальные средства
разработки программного обеспечения**»

Выполнил(а):

студент(ка) 3-го курса, группа _____

_____ (ФИО)

Проверил:

преподаватель техникума ДИТИ НИЯУ МИФИ

Казынбаева Р.К.

Оценка _____

Дмитровград 20__