МИНИСТЕРСТВО КУЛЬТУРЫ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ КУЛЬТУРЫ»

УТВЕРЖДЕНО
Деканом факультета МАИС
О.А. Будариной
«06» октября 2015 г.

УТВЕРЖДЕНО
Зав. кафедрой дизайна
М.В. Решетовой
«06»октября 2015 г.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

Веб-дизайн

Направление подготовки: «Дизайн»

Профиль подготовки: Графический дизайн

Квалификация Бакалавр

Форма обучения Очная, Заочная

Согласовано:

С председателем методического совета по качеству по направлению

Москва
2015

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Данный практикум, составленный в соответствии с государственным образовательным стандартом и рабочей программой по дисциплине *«Web-дизайна»,* предназначен студентам, обучающимся по направлению подготовки бакалавров Реклама для выполнения практических работ.

В издании определены содержание, объем и порядок выполнения практических работ по данной дисциплине, а также требования к результатам работы студентов. В практикуме представлены краткие теоретические сведения по каждой теме, что предполагает обязательное посещение лекций и изучение соответствующей литературы.

Основная цель выполнения практических работ – освоить базовые навыки создания сайта инновационного проекта или технологии и изучить основные аспекты *web*-дизайна.

Цель настоящего практикума – методическое обеспечение организации самостоятельной работы студентов при проведении практических занятий.

Данный практикум *обеспечивает*:

 системное представление о задачах и содержании практической составляющей учебной дисциплины *«Web-дизайна»*;

 индивидуальную работу студентов;

 закрепление знаний, умений и навыков, формирование профессиональных компетенций студентов;

 организацию планомерной работы и стимулирование познавательного интереса студентов к учебной дисциплине;

 развитие творческого подхода к решению задач профессиональной деятельности;

 возможность для студентов проводить самоконтроль качества обучения.

После выполнения практических работ студент должен представить в письменном виде итоговый отчет о проделанной работе (редактор Word, формат А4, двухсторонняя печать), в который входят:

 описание проекта сайта;

 результаты обработки изображений;

 образцы оформления сайта;

 описание интерактивных и подгружаемых элементов сайта.

В книге представлен библиографический список, позволяющий самостоятельно изучить данный материал по другим источникам, а также список вопросов и тесты для подготовки к зачету.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ И ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

Перед выполнением каждой практической работы студент должен изучить теоретический материал.

Практические работы выполняются в компьютерных классах академии и должны быть продемонстрированы преподавателю.

По результатам выполнения практической работы оформляется краткий отчет (редактор Word, формат А4, двухсторонняя печать). Отчет составляется в одном экземпляре на 2 студентов и подлежит защите. Для защиты практической работы студент должен подготовить ответы на контрольные вопросы, которые находятся в конце каждой работы.

Титульный лист отчета необходимо оформить по стандартной форме (прил. 1).

Отчет должен содержать следующие компоненты:

– постановка задачи;
– содержание работы;
– полученные результаты;
– ответы на контрольные вопросы.

Предполагается, что студенты смогут изучать практическое создание сайтов начального уровня и начать редактировать и править их самостоятельно. Язык разметки текстовых потоков (HTML) считается простым, по сравнению с другими языками программирования, но изучение большого количества тегов (tags) требует времени.

Практическая работа № 1
РАЗРАБОТКА ПРОЕКТА САЙТА
«ИННОВАЦИОННЫЕ ТЕХНОЛОГИИ, ТОВАРЫ И УСЛУГИ»

Время выполнения – 4 часа.

Цель работы: изучение и освоение принципов разработки сайта.

Задачи работы

1. Изучить общие принципы разработки сайта.
2. Определить основную тему разрабатываемого сайта.
3. Разработать структуру и содержание сайта.
4. Написать сценарий сайта.
5. Разработать макет сайта.
6. Собрать и подготовить информацию и материалы для сайта.
7. Определить аппаратно-программные потребности сайта.

Перечень обеспечивающих средств, необходимых для выполнения практической работы:

– индивидуальное задание на выполнение работы для 2 человек;
– персональный компьютер;
– программное обеспечение (ПО) Microsoft Office Word.

ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Глобальная сеть состоит из ***сайтов***, доступных для общего пользования, закрытых приватных сайтов, корпоративных и локальных (доступных на уровне какой-либо локальной сети).

Сайт (веб-сайт, англ. ***website***, от ***web*** – паутина, и ***site*** – место) – это место во всемирной сети (интернете), которое имеет свой адрес, собственного хозяина и состоит из отдельных *web*-страниц, которые мы видим как одно целое.

Все странички каждого сайта объединяются одним корневым адресом (то, что мы набираем в адресной строке *браузера*), тематикой, системой и дизайном. Каждая страница *web*-сайта – это документ, структура которого описана при помощи языка разметки (X) HTML. Страницы сайтов могут быть как простыми наборами информации – тексты и картинки, так и сложными, с огромным количеством функций.

У каждого сайта свои *цели, задачи, функции, характеристики объемов и др*., которые реализуются благодаря возможностям и преимуществам интернет-технологий_____.

Сайты можно условно *классифицировать* по следующим признакам:

1. *Цели и задачи* его создания и функционирования – коммерческий, некоммерческий, информационный, рекламный, учебный сайт, поисковая система и др.).
2. *Объемность* – хоум-пейдж, т. е. домашняя страничка, сайты-

визитки, интернет-представительства, веб-порталы и т. п.

3. *Интерактивность* – наличие интерактивных модулей (гостевые книги, форумы, модули голосования, формы заказов, авторизация и т. д.).

4. *Метод их создания и функционирования* – динамические и статические (последние в настоящее время уже практически нигде не встречаются).

Создание сайта начинается с ***концепции и структуры***, т. е. с постановки задачи в соответствии с целями сайта. И уже на основе этой информации необходимо подбирать нужные материалы. Очень важно представить всю информацию удобным и понятным образом, чтобы то, что вы хотите передать, узнал бы и пользователь сайта.

Очень важна ***информационная структура сайта.*** Продуманная информационная структура гарантирует, что пользователи потратят меньше времени на поиск нужной информации и никогда не скажут, что они чего-то не нашли. При хорошей структуре они всегда обнаружат, что один документ связан ссылками с другими документами по той же теме. Они всегда смогут легко переключаться с поиска документов на их просмотр и обратно. Они лучше будут понимать, какую информацию сайт может им предложить.

Содержание сайта – это совокупность информационных ресурсов, удовлетворяющих следующим условиям:

 во-первых, и это самое главное условие, информация, которую вы хотите донести до посетителей должна быть полезной для них. Информация по возможности должна быть уникальной, т. е. такой, чтобы ее нигде не могли бы найти кроме как у вас;

 во-вторых, постарайтесь наиболее полно охватить ту тему, которой посвящен ваш сайт. Постарайтесь специализироваться, т. е. давать информацию на какую-то конкретную тему. Лучше, чтобы было много о малом, чем мало о многом. К тому же, сейчас очень много серьезных и раскрученных порталов, и с ними сложно конкурировать. Да и сами посудите, куда бы вы пошли, если бы хотели найти специализированную информацию?

Если ваш сайт посвящен бизнесу, публикуйте новости вашей компании или фирмы. Смотрите широко и предоставляйте информацию и новости не только по вашей компании, но и по всей отрасли, в которой она работает. Если вы продаете товары, то предоставляйте новости о родственных им товарах. Эта информация может быть полезной для ваших посетителей и они, возможно, захотят вернуться к вам на сайт (полезным для этого может стать ссылка «добавить в избранное»). Если же ваш сайт является информационным, то можно разместить ссылки на другие сайты той же тематики, что и ваш. Конечно, не все решатся на этот шаг – рекламировать собственных конкурентов, и если вы боитесь, что посетители уйдут от вас, то можно открывать ссылки на другие сайты в новых окнах.

Возможно, вы являетесь профессионалом своего дела, и тогда вы можете предоставить на своем сайте бесплатные консультации по интересующим посетителей вопросам. Для этого достаточно иметь гостевую книгу.

Теперь немного о том, как должна быть оформлена и подана информация на сайте. Постарайтесь не делать статьи очень длинными. В то же время, не дробите их на маленькие кусочки так, чтобы посетителю приходилось все время переходить с ссылки на ссылку. Постарайтесь умещать всю статью на одной страничке, чтобы можно было бы ее сохранить и прочитать как отдельный рассказ. Отделяйте между собой разные смысловые кусочки статьи пустыми строками или названиями (подразделы).

Используйте удобную навигацию по сайту. На каждой страничке должна быть хотя бы ссылка на главную страницу, а еще лучше будет, если вы

поставите ссылки на другие свои статьи по ходу текста (если она уместна в этом месте) или в конце страницы. Так человек, заинтересовавшийся вашей статьей, захочет прочитать и другие ваши или чужие работы на данную тему. И еще один важный совет: старайтесь обновлять сайт по мере возможностей. Посетители должны быть уверены, что, вновь зайдя на ваш сайт, они найдут для себя что-то новенькое. Иначе они просто не будут больше вас посещать.

Дизайн сайта – это совокупность графических элементов, шрифтов и цветов, реализованных на сайте.

Основная задача дизайна сайта – объединение всех информационных блоков на художественно проработанной основе и формирование у посетителя приятного впечатления. По сути, дизайн задает общий стиль вашего сайта, помогает посетителю с первого взгляда понять, что его здесь ждет. Грамотно разработанный дизайн является одним из важнейших факторов, определяющих посещаемость вашей web-страницы.

Как правило, дизайн сайта – это внешнее его оформление, которое призвано, как минимум, не отпугнуть посетителя Вашего интернет-ресурса, и, как максимум, завлечь, заинтересовать его. Как говорят опытные специалисты, *хороший дизайн сайта* – это незаметный, ненавязчивый дизайн, который не отвлекает посетителя от основного – от предоставленной целевой информации (вспомните хотя бы дизайн страниц известных поисковых систем *Google и Яндекс*). Думаю, Вы согласитесь с тем, что человек, заходя на какой-либо ресурс Интернет, прежде всего осуществляет поиск необходимой информации (исключением является лишь дизайнер, ищущий для себя новые решения и интересующийся именно дизайнерской тематикой). И в этой ситуации любая отвлекающая информация (слишком яркий, броский дизайн сайта, излишняя анимация, всплывающие рекламные окна) будут только мешать в достижении основной цели – получении необходимых данных, в поиске которых на Ваш сайт и заходил посетитель.

Задание

Для выполнения практической работы № 1 необходимо изучить общие принципы разработки сайта. Студент должен выбрать из табл. 1.1 один из вариантов названия будущего разрабатываемого сайта. Для определения основной темы разрабатываемого сайта необходимо сформулировать, что должно присутствовать на сайте, представляющем компанию (или проект). Разработать структуру, содержание, дизайн, макет сайта.

Написать техническое задание (сценарий), собрать, подготовить информацию и материалы для разрабатываемого сайта. Определить аппаратно-программные требования сайта.

Таблица 1.1

№ варианта	*Название проектов и технологий* (расположение проектов: www.SSGA.RU/ Проекты и технологии)
1	Сеть активных базовых станций ГЛОНАСС/GPS
2	Трехмерное лазерное сканирование
3	Карты бумажные и настенные
4	Цифровые карты и ГИС
5	Инженерно-геодезические работы
6	Геодинамика

7 Метрология геодезических приборов
8 Физические явления в оптике
9 Системы тепловидения
10 Справочно-информационные ГИС на CD
11 3D-модели природных объектов
12 Голограмма
13 Цифровая фотограмметрия
14 Планетарий
15 Повышение квалификации
16 Центр тестирования и профориентации
12

Порядок выполнения практической работы

1. Определение основной темы разрабатываемого сайта.

После того, как будет определена основная тема разрабатываемого сайта, необходимо сформулировать, что должно присутствовать на сайте, представляющем компанию или проект.

Сформулировать информационную архитектуру сайта. Любая информационная система – будь то книга или интранет-сеть предприятия – имеют свою информационную архитектуру, которая планируется с самого начала. Существует достаточно много способов организовывать информацию в систему. Хорошо продуманная архитектура сайта помогает пользователям тем, что использует некоторый набор из всех этих способов, т. е. набор, который больше всего подходит для сайта, исходя из целей сайта, информационных потребностей посетителей сайта, материалов сайта, бюджетных ограничений.

Продуманная информационная архитектура гарантирует, что пользователи потратят меньше времени на поиск нужной информации, и почти никогда не скажут, что они чего-то не нашли. При хорошей архитектуре они всегда обнаружат, что один документ связан ссылками с другими документами по той же теме. Они всегда смогут легко переключаться с поиска документов на их просмотр и обратно. Они лучше будут понимать, какую информацию сайт может им предложить.

2. Разработка содержания, структуры и дизайна сайта.

После того, как вы определились с целями и задачами сайта, следует приступить к *разработке его содержания.* На этом этапе необходимо составить перечень информационных разделов сайта и по каждому разделу установить состав входящей в него информации и сформулировать содержание массивов информации («контент»). Содержание сайта – основа любого сайта, и от этого, во многом зависит его успех. Так что же должно размещаться на сайте, какая информация, и главное – как, чтобы посетители, раз зашедшие к вам на сайт, заходили на него снова и снова?

Далее необходимо *разработать структуру сайта* – один из важнейших, после *цели,* вопросов создания сайта. Дизайн, стилистика, компоновка, программные средства, технологии и прочие разрабатываются и/или выбираются исходя из структуры – можно даже сказать, что они являются структурными элементами сайта. Именно выбор правильной структуры обеспечивает «высокое юзабилити» сайта (удобство пользования) и гарантирует доступ ко всей информации – быстроту и легкость нахождения, адекватность материалов, находящихся в разделах, их тематике.

Структурность – многоплановое понятие, которое применяется как к *информации*, так и к визуальным *элементам.*

Нельзя назвать верным такое решение, когда все содержательное наполнение сайта, традиционно называемое «информация», или «контент»,

расположено, что называется, «внавалку». Как правило, вся информация рассортировывается, в первую очередь, по тематическим разделам.

Например, для *web*-представительства _____©2ҐС7небольшой фирмы информационная структура может выглядеть так:

 Index (Главная страница – это страница, которая загружается при наборе доменного имени сайта в строке браузера. В подавляющем большинстве сайтов главная страница содержит колонку News);
 About (О нас, О компании и т. д.);
 News (Новости – периодически и регулярно дополняемая информация);
 Products (Наши услуги, Наши продукты, Наши туры);
 Price (Цены, прайс-лист);
 Support (Техническая и/или сервисная поддержка, горячая линия и/или e-mail связь);
 Search (Поиск – модуль поиска по введенному слову или словосочетанию).

Расположение информации в самих разделах тоже должно придерживаться какого-либо принципа: хронологический, алфавитный, по ID, по тематической важности и т. п.

Структура сайта – это не только то, что мы видим, за этим стоит программный продукт с разделением содержания и представления

Если рассматривать *web*-страницу не с визуальной стороны, а с точки зрения структурного подхода, то можно выделить несколько групп составляющих ее элементов, таких, как:

 текст – основное содержание страницы, так называемый контент;
 акцентуированный текст – специальным образом форматированный текст, чаще всего с использованием CSS (каскадная таблица стилей), использующийся и для оформления контента, и как элемент интерфейса сайта – меню, разделы, заголовки;
 фон – цвет страницы или ячейки таблицы;
 ссылки – ссылки гипертекста;
 картинки – фотографии и графика контента, графические элементы оформления сайта.

Индексная
страница
сайта
Руководство пользователя
Сведения о разработчиках
сайта
Раздел сайта
Текст
Фото
Видео, ани мация Рисунки, гр афики
Интернет-а дреса
Звук
разделы сайта
информация
о сайте
Главное меню
Названия разделов сайта Прочая информация

После этого можно приступить непосредственно *к разработке дизайна* проектируемого сайта, обратив особое внимание на следующие вопросы.

1. Визуальное оформление.

Для приятного восприятия информации сайта необязательно его загружать яркой, броской графикой. Недопустима излишняя насыщенность сайта картинками, фотографиями и другими графическими элементами, которые могут помешать пользователю найти нужную ему информацию на сайте. Здесь также важно уделить внимание психологии восприятия, грамотности подбора шрифтов и цвета, т. е. главное выделять крупным, жирным цветным шрифтом, а второстепенная информация и элементы оформления должны читаться на втором плане.

2. Эксклюзивность и оригинальность сайта.

При работе над дизайном сайта возможно применение двух различных подходов: создание эксклюзивного дизайна или использование шаблона. В первом случае материальные и временные затраты будут больше, а использование шаблоны упростит работу. Достаточно лишь существующий шаблон привести в соответствие с темой и оформлением сайта.

3. Соответствие сайта техническим требованиям.

Страницы сайта должны быть привлекательными (даже без рисунков), быстро загружаться и быть совместимыми с наиболее популярными браузерами (Internet Explorer, Opera, Mozilla Firefox, Google Chrome). После разработки дизайна необходимо подготовить техническое задание (сценарий) для создания сайта. В нем указываются примеры удачных сайтов и указания по стилю и цветовому оформлению. В результате изготавливается макет, который можно будет корректировать по ходу выполнения проекта.

4. Не забывайте: Ваш сайт – лицо Вашей компании!

Как известно, встречают по одежке, а это значит, что о солидности Вашей организации будут судить по дизайну ее сайта. Грамотный дизайн интернет-ресурса отражает индивидуальный стиль компании, помогает ей выделиться среди конкурентов.

3. Подготовка технического задания (сценария) для создания сайта.

Сайт с информацией создается на *web*-страницах с помощью языка гипертекстовых разметок и программы, которые сопровождаются и управляются *техническим заданием* (*сценарием*), обеспечивающих активность гипертекстовых страниц (представляет методы обработки данных и компоновку страниц). При создании сайта техническое задание (сценарий) является основным документом, который конкретизирует общие положения и требования к сайту.

ПРИМЕР

ВОЗМОЖНОГО ТЕХНИЧЕСКОГО ЗАДАНИЯ НА СОЗДАНИЕ САЙТА

Основная цель: продвижение бренда компании в Интернете. Под продвижением подразумевается оперативное донесение информации до целевой аудитории, формирование положительного имиджа компании посредством стильного дизайна сайта и актуальности представленной информации.

Основная задача: разработка нового корпоративного (имиджевого) сайта компании.

Целевая аудитория: менеджеры среднего и высшего звена управления, консультанты, владельцы и руководители отечественных и зарубежных компаний.

Предполагаемая структура сайта

1. О Компании: история компании, задачи компании, миссия и корпоративные ценности, вакансии, новости (ленты RSS), опросы и т. п.

2. Услуги: услуга 1, услуга 2, другие услуги.

3. Проекты: перспективные, выполненные.

4. Клиенты и партнеры: клиенты Компании, партнеры Компании, отзывы, рекомендации.

5. Справочная информация: термины, статьи, вопросы и ответы (FAQ), законодательство.

6. Контакты.

7. Карта сайта.

8. Реклама.

Дополнительная информация: изменение содержания сайта должно осуществляться в рамках системы управления, что позволяет компании получить в использование Интернет-ресурс, которым смогут управлять

ответственные сотрудники, не имеющие соответствующих специальных знаний по созданию сайтов (после определенной подготовки и инструктажа специалистами).

4. Разработка макета сайта.

Выбор варианта компоновки сайта (или используемых шаблонов сайтов) определяется условиями технического задания на разрабатываемый *web*-сайт. Существуют сайты с фиксированной шириной или «резиновые» сайты, изменяющие ширину в зависимости от разрешения монитора пользователя

Верхняя часть страницы («шапка», header) используется, как правило, для размещения логотипа, эмблемы, названия фирмы и другой важной информации.

Нижняя часть страницы («подвал», footer) используется для контактной информации или данных, структурно требующих разделения с данными, размещенными в верхней части сайта

Панель навигации (главное меню) сайта должна размещаться в удобном для частого использования месте, так как количество ссылок может повлиять на компоновку сайта (вертикальное или горизонтальное размещение панели навигации)

5. Сбор и подготовка информации и материалов для сайта.

Подготовка материалов и дальнейшее уточнение структуры сайта взаимосвязаны. В соответствии с информационным наполнением идет процесс корректировки структуры. Проанализируйте существующие материалы и с точки зрения содержания, и с точки зрения занимаемых объемов. Например, после дополнительной обработки можно использовать имеющиеся буклеты, печатные работы, фотографии и рисунки, в том числе, эмблемы. Очень важный момент при работе с материалами – соблюдение авторских прав. Относитесь с уважением к чужому труду, соблюдайте требования действующего законодательства в области интеллектуальной собственности, указывайте авторов текстов, фотографий и т. д.

В целом при компоновке страниц необходим учет ограничений, связанных с объемом размещенных на ней материалов. Ради обеспечения достойного качества конечного продукта некоторыми материалами придется пожертвовать, а некоторые следует разместить с соответствующим предупреждением о времени загрузки и возможностью их проигнорировать. Например, хорошим тоном является возможность такого выбора перед запуском флэш-элементов. А фотоальбомы, несмотря на неминуемую громоздкость даже при хорошей структуризации по разделам, могут быть очень интересны людям.

6. Определение аппаратно-программных потребностей сайта.

В зависимости от содержимого сайта, его структуры, функционала формируются различные требования к аппаратному и программному

обеспечению, на котором данный сайт будет функционировать.
К программным требованиям серверной части могут относиться требования к операционной системе, *web*-серверу, установленным скриптовым языкам, СУБД и прочее, к аппаратным – объем доступного дискового пространства, загрузка процессора и памяти, время выполнения запросов и максимальное их количество.
Также могут указываться требования к клиентскому программному обеспечению, как правило, это определенные версии браузеров, дополнительные модули к ним.
После выполнения этих требований нужно определить основную тему, подобрать информацию различных видов (тексты, изображения, видео, звуки и т. п.), позволяющую осветить выбранную тематику, спроектировать содержимое отдельных страниц и их взаимосвязи, выполнить макет сайта, содержащий типовые элементы, их расположение, цветовую схему и т. д.
20
Вопросы для защиты работы
1. Состав и содержание действий, предшествующих непосредственным техническим работам по созданию сайта.
2. Технологическая последовательность процесса создания сайта.
3. Недопустимые или нежелательные технические решения при создании сайта.
4. На что следует обратить особое внимание при разработке сайта?
21
Практическая работа № 2
СОЗДАНИЕ ШАБЛОНА САЙТА
«ИННОВАЦИОННЫЕ ТЕХНОЛОГИИ, ТОВАРЫ И УСЛУГИ»
Время выполнения – 4 часа.
Цель работы: изучение и освоение принципов работы с HTML тегами, их атрибутами и умение создавать базовые элементы *web*-страницы.
Задачи работы
1. Научиться работать с HTML тегами, их атрибутами.
2. Научиться задавать базовую структуру HTML документа.
3. Научиться создавать базовые элементы *web*-страницы.
Перечень обеспечивающих средств, необходимых для выполнения 8™Щ_@5_k_Д_ђ_практической работы**:**
– индивидуальное задание на выполнение работы для 2 человек;
– персональный компьютер;
– программное обеспечение (ПО): любой текстовый редактор или специализированная среда разработки.
ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ
HTML (от англ. *HyperText Markup Language* – «язык разметки гипертекста») – стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц создается при помощи языка ***HTML (****или* ***XHTML)***. Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме. В настоящее время стабильная и самая распространенная версия языка – ***HTML 4.01***, а также активно развивается ***HTML 5.0***, который поддерживает новейшие мультимедийные приложения.
22
Тег – это служебная инструкция, размещенная между символами меньше (<) больше (>). Теги могут быть открывающимися и закрываю-

щимися. Например, – открывающий тег, – закрывающий тег. Открывающий тег указывает на начало какого-либо блока, а закрывающий – на завершение этого блока.
Атрибут – это уточнение для браузеров, как «поточнее» задать тег; атрибуты описываются внутри открывающего тега в виде коллекции имя="значение". Например, .
Для комплексных разработок и ведения больших проектов существуют системы программных средств, которые называются **Интегрированными средами разработки** (англ. Integrated development environment или integrated debugging environment – IDE). IDE используются повсеместно: от разработки мелких статичных сайтов до полноценных игр и программных комплексов. Для web-разработок IDE требует определенной настройки. Обычно среда web-разработки в себя включает три компонента:
1) *текстовый редактор* – область для ввода кода;
2) *компилятор и/или интерпретатор* – компонент, который обрабатывает код;
3) *отладчик* – компонент, отвечающий за корректность кода и проверку синтаксиса.
Интегрированные среды разработки призваны максимально увеличить производительность программиста благодаря тесно связанным компонентам с простыми пользовательскими интерфейсами. Это позволяет разработчику сделать меньше действий для переключения различных режимов, в отличие от дискретных программ разработки. Поскольку IDE является сложным программным комплексом, то лишь после долгого процесса обучения среда разработки сможет качественно ускорить процесс разработки ПО. Обычно среда разработки предназначена и ориентирована на определенный язык программирования, предоставляя набор функций, который наиболее близко соответствует парадигмам этого языка программирования. Однако в настоящее время наиболее популярные IDE, такие, как ***Eclipse, ActiveState Komodo, NetBeans, Microsoft Visual Studio, WinDev и Xcode,*** могут поддерживать несколько языков программирования, установив определенный модуль.

Задание
Для выполнения практической работы № 2 необходимо изучить процесс создания шаблона сайта. Научиться работать с HTML тегами, их атрибутами. Научиться задавать базовую структуру HTML документа. Научиться создавать базовые элементы *web*-страницы.
Порядок выполнения практической работы
1. Создание заголовочной части web-страницы.
Создать шаблон, содержащий все элементы типовой страницы. Вид HTML документа определяют инструкции или теги, обрамляющие те или иные элементы страницы.
HTML документ начинается с заголовочной части, содержащей служебную информацию и общие инструкции браузеру по отображению контента.
Листинг 02.01
```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta http-equiv="Content-Type" content="text/html; charset=win-1251" />
05 <title>Шаблон страницы HTML5</title>
```

06 </head>
Тег <!DOCTYPE html> сообщает браузеру, что страница создана по стандарту HTML5.
Парный тег <html> обрамляет все остальные секции любого HTML документа (данный тег должен быть в единственном экземпляре и должен присутствовать на каждой странице).
Парный тег <head> отделяет заголовочную часть страницы со служебной информацией.
Одиночный тег <meta> Тег <meta> предоставляет метаданные о документе HTML браузеру. Метаданные не отображаются, а только используются для служебных целей либо движком браузера, либо поисковыми системами. Метаэлементы, как правило, используется для описания стра-

ницы (description), указания ключевых слов (keywords), указания 02 ю u1072 автора документа (author), указания типа контента (content) и его кодировки (charset).
Парный тег <title> – единственный тег из этой секции, который выводит видимую информацию, он определяет заголовок *web*-страницы, отображающийся в строке заголовка окна браузера.

2. Создание основной части web-страницы.

Создание тега HTML документа.
В HTML 5 для каждой части страницы имеется свой элемент.
Вот их список и краткое назначение:
1) section; назначение – определение секций. Его используют для описания определенного блока текста, например, хорошим применением этого элемента будет разбиение большой части текста на более малые, как происходит разбиение одной статьи на несколько абзацев;
2) header; назначение – определение верхней секции на странице;
3) footer; назначение – определение нижней секции на странице;
4) nav; назначение – определяет набор ссылок на другие страницы (часто используют для навигации по сайту);
5) article; назначение – выделить определенную часть текста.

Листинг 02.02

```
01 <body>
02 <section id="page">
03 <header> <!-- Шапка -->
04 <h1>Логотип</h1>
05 <h3>Слоган</h3>
06 <nav class="clear">
07 <ul>
08 <li><a href="#article1">Главная</a></li>
09 <li><a href="#article2">Статья 1</a></li>
10 <li><a href="#article3">Статья 2</a></li>
11 </ul>
12 </nav>
13 </header>
```

Парный тег <body> содержат все теги, определяющие структуру и содержание web-страницы.

Парный тег <section> используется для разделения страницы на семантические части.
Парный тег <header> определяет шапку страницы. В создаваемом

шаблоне она содержит заголовки первого и третьего уровней (<h1>, <h3>) для логотипа и слогана соответственно, и элементы навигации (<nav>) в виде маркированного списка (<ul> определяет список, <li> – элементы списка).
Атрибут href тега <a> содержит ссылку на документ, который связан с данным пунктом навигационного меню, часть после # отвечает за ID статьи, к которой мы хотим перейти.

3. Добавление элементов на страницу.

Разделим линией заголовочную и основную части и расположим текст статьи.

Листинг 02.03

```
01 <!-- Статья 1 начало -->
02 <div class="line"></div>
03 <article id="article1">
04 <h2>Статья 1</h2>
05 <div class="line"></div>
06 <div class="articleBody clear">
07 <figure>
08 </figure>
09 <p>Текст первой статьи</p>
10 </div>
11 </article>
12 <!-- Статья 1 конец -->
13 <footer> <!-- Подвал -->
14 <div class="line"></div>
15 <p>Copyright 2011</p>
16 </footer>
17 </section>
18 </body>
19 </html>
```

Парный тег – тег-контейнер, выделяет логический блок.
Парный тег <figure> служит для отображения рисунка для статьи.
Парный тег <p> обозначает отдельный абзац текста.

Опишем оставшуюся часть страницы – «подвал», а также закроем секцию page и тело документа.
В результате объединения вышеописанных частей получаем текст HTML страницы, просмотрев которую в браузере получим следующий результат

Вопросы для защиты работы

1. Что такое базовые элементы *web*-страницы?
2. Основные компоненты IDE.
3. Назначение тегов <html>, <body>, <header>, <section>, <p>, <figure>, .
4. Что входит в состав метаданных о документе HTML?
5. С помощью какого тега задается заголовок web-страницы?

Практическая работа № 3
ОБРАБОТКА ИЗОБРАЖЕНИЙ ДЛЯ ПРОЕКТА САЙТА «ИННОВАЦИОННЫЕ ТЕХНОЛОГИИ, ТОВАРЫ И УСЛУГИ»

Время выполнения – 2 часа.

Цель работы: изучить базовые принципы исправления и редактиро-

вания растровых изображений в ПО Adobe Photoshop.

Задачи работы

1. Изучить понятия мультимедийных форматов, применяемых в Интернете.
2. Научиться делать корректуру фотографий.
3. Научиться вносить изменения в растровые изображения.

Перечень обеспечивающих средств, необходимых для выполнения практической работы:

– индивидуальное задание на выполнение работы для 2 человек;
– персональный компьютер;
– программное обеспечение (ПО): Adobe Photoshop или имеющий аналогичные функции редактор растровой графики.

ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Графика на web-страницах

В настоящее время существует достаточно большое количество различных форматов графических файлов, которое можно разделить на две группы:

1) файлы, хранящие векторную графику;
2) файлы, хранящие растровую графику.

Растровая графика – это когда изображение хранится в виде маленьких ячеек – пикселей. Соответственно качество такой картинки ограничивается двумя факторами: собственно размером самой картинки в пикселях и разрешением изображения, то есть количеством пикселей на единицу длины (наиболее распространен показатель разрешения растра в пикселях на дюйм – dpi). Файлы, хранящие растровую графику, это jpg, gif, bmp, tiff, png, psd и прочие.

Векторная графика – это когда изображение хранится в виде массива чисел, описывающих построение изображения в виде векторов, заданных координатами ключевых точек-вершин. Форматы векторной графики – dxf, swf, cdr, max, ai, svg, частично pdf.

В практике *web*-графики в основном используются три формата растровой графики – gif, jpeg и png, и один формат векторной графики – svg.

Форматы растровой графики различаются различными алгоритмами сжатия изображения. Так как в web размер файла до сих пор играет весьма существенное значение в силу ряда причин, следовательно, малый вес графического файла существенно повышает скорость загрузки изображения.

Для разных «видов» картинок подходит тот или иной формат графики. Определение необходимого формата – основное умение *web*-дизайнера. Для фотографий, портретов, картинок большого размера, насыщенных сложными деталями, лучше всего подходит формат ***JPEG.***

Алгоритм сжатия этого формата ***JPEG*** работает таким образом, что при уменьшении «веса» картинки, а, следовательно, качества, изображение как бы «размывается», становятся плохо различимы четкие переходы между цветами и появляются паразитные цветные пиксели как побочный эффект действия алгоритма. Степень компрессии файла определяется каждый раз дизайнером исходя из его нужд, но оптимальным соотношением размер/качество изображения считается процент сжатия, равный 65.

Формат ***GIF*** наилучшим образом подходит для изображений небольшого размера, там, где необходима прозрачность (альфа-канал), и для анимированной растровой графики.

Алгоритм сжатия этого формата основывается на том, что изображению задается фиксированная цветовая палитра (от 2 до 256 цветов), а все

близкие оттенки выкидываются либо заменяются соседними цветами. Также алгоритм просчитывает изображение линиями – слева направо и хранит информацию не о каждом пикселе в отдельности, а считает,

сколько пикселей одинакового цвета стоят в ряд, и сохраняет информацию только о цвете и количестве пикселей. Это существенно снижает вес файла. Нетрудно заметить, что вертикальный градиент (сверху вниз) будет весить в таком случае намного меньше градиента горизонтального (слева направо). Это следует учитывать при создании изображений, особенно при создании анимированных баннеров, где вес складывается еще и из количества кадров и идет буквально война за каждый байт. Также возможно загружать *GIF*-изображения чересстрочным способом. Если картинка имеет очень большой размер и долго загружается, то будут сначала видны нечеткие контуры изображения, а по мере загрузки оно будет «проявляться». Для этого при сохранении *GIF*-файла нужно включить метод Interlaced (чересстрочный).

Формат ***PNG*** имеет большие возможности, он сочетает в себе свойства *GIF*- и *JPEG*-сжатия: отбирать только нужные цвета, использовать палитру шкалы полутонов, обеспечивать прозрачность, сложная последовательная загрузка. Хотя формат *PNG* поддерживается всеми современными браузерами, но все же некоторые браузеры могут некорректно его отображать.

При именовании файлов следует придерживаться простых правил. Во-первых, следует избегать как бессмысленных, так и «говорящих» названий. Название файла должно сразу определять его место в структуре *web*-страницы. То есть, если это картинка к статье, то она должна располагаться в папке articles, и ее названием служит id статьи. Если это позиция в каталоге, то в соответствующей папке (items, groups) картинка должна иметь названием ID группы, подгруппы или товара.

В то же время, если, к примеру, на сайте очень редко пишутся статьи, можно не вводить дополнительные папки, но тогда файл должен по названию однозначно ассоциироваться с данным разделом. Это можно сделать, к примеру, добавлением слова news или art перед id картинки (например, news-34.jpeg). Если к одному id относится несколько картинок разного размера, необходимо добавлять после id картинки расширение, обозначающее размер (для больших картинок – b, для маленьких – s) (например, 38-s.gif, art-08-b.jpeg). Для нескольких картинок одного размера, можно ввести порядковые номера (например, art08-b_01.gif).

SVG (от англ. Scalable Vector Graphics – масштабируемая векторная графика) – язык разметки масштабируемой векторной графики, созданный Консорциумом Всемирной паутины (W3C) и входящий в подмножество расширяемого языка разметки XML, предназначен для описания двумерной векторной и смешанной векторно-растровой графики в формате XML. Поддерживает как неподвижную, так анимированную и интерактивную графику, или, в иных терминах, декларативную и скриптовую. Не поддерживает описание трехмерных объектов (не путать с имитацией трехмерности путем светотени).

Достоинства формата.

1. Текстовый формат – файлы SVG можно читать и редактировать (при наличии некоторых навыков) при помощи обычных текстовых редакторов. При просмотре документов, содержащих SVG графику, имеется

доступ к просмотру кода просматриваемого файла и возможность сохранения всего документа. Кроме того, SVG файлы обычно получаются меньше по размеру, чем сравнимые по качеству изображения в форматах JPEG или GIF, а также хорошо поддаются сжатию.
2. Масштабируемость – SVG является векторным форматом. Существует возможность увеличить любую часть изображения SVG без потери качества. Дополнительно к элементам SVG документа возможно применять фильтры – специальные модификаторы для создания эффектов, подобных применяемым при обработке растровых изображений (размытие, выдавливание, сложные системы трансформации и др.) В тексте SVG-кода фильтры описываются тегами, визуализацию которых обеспечивает средство просмотра, что не влияет на размер исходного файла, при этом обеспечивается необходимая иллюстративная выразительность.
3. Широко доступно использование растровой графики в SVG документах. Имеется возможность вставлять элементы с изображениями в форматах PNG, GIF или JPG.
4. Анимация реализована в SVG с помощью языка SMIL (Synchronized Multimedia Integration Language), разработанного также консорциумом W3C. Поддерживаются скриптовые языки на основе спецификации ECMAScript. SVG-элементами можно управлять с помощью JavaScript. Применение скриптов и анимации в SVG позволяет создавать динамич-

ную и интерактивную графику. В SVG обеспечивается событийная модель, отслеживаются события (загрузка страницы, изменение ее параметров, события мыши, клавиатуры и др.). Анимация может запускаться по определенному событию (например, «onmouseover» или «onclick»), что придает графике интерактивность. У каждого элемента есть свои собственные события, к которым можно привязывать отдельные скрипты.
5. SVG – открытый стандарт. В отличие от некоторых других форматов, SVG не является чьей-либо собственностью.
6. SVG документы легко интегрируются с HTML и XHTML документами.
7. Совместимость с CSS (англ. Cascading Style Sheets). Отображением (форматированием и декорированием) SVG элементов можно управлять с помощью таблицы стилей CSS 2.0 и ее расширений, либо напрямую с помощью атрибутов SVG элементов.
Недостатки формата.
1. SVG наследует все недостатки XML, такие, как большой размер файла (впрочем, последний компенсируется существованием сжатого формата SVGZ).
2. Сложность использования в крупных картографических приложениях из-за того, что для правильного отображения маленькой части изображения документ необходимо прочитать целиком.
3. Чем больше в изображении мелких деталей, тем быстрее растет размер SVG-данных. Предельный случай – когда изображение представляет собой белый шум. В этом случае SVG не только не дает никаких преимуществ, но и даже обладает чрезмерно избыточным по отношению к растровому формату размером. На практике SVG становится невыгоден уже задолго до того, как изображение дойдет до стадии белого шума.
Звук на web-страницах
Есть четыре основных варианта на выбор: WAV, MP3, OGG и AAC.
Но не все браузеры обеспечивают полную их поддержку.

WAV (его наиболее распространенный формат PCM) является несжатым аудио. В результате, файлы, как правило, очень велики, и из-за этого очень сильно возрастает время загрузки *web*-страницы.
32
Хоть *MP3* и является в данный момент самым распространенным, этот формат не открытый. За возможность с ним работать требуется платить определенную сумму обладателям патента.
Ogg Vorbis – это относительно новый универсальный формат аудиокомпрессии, официально вышедший летом 2002 г. Он принадлежит к тому же типу форматов, что и MP3, AAC, VQF и WMA, т. е. к форматам компрессии с потерями. Психоакустическая модель, используемая в Ogg Vorbis, по принципам действия близка к MP3 и иже с ними, но и только – математическая обработка и практическая реализация этой модели в корне отличаются, что позволяет авторам объявить свой формат совершенно независимым от всех предшественников.
Главное неоспоримое преимущество формата Ogg Vorbis – это его полная открытость и свободность. Более того, в нем использована новейшая и наиболее качественная психоакустическая модель, из-за чего соотношение битрейт/качество значительно ниже, чем у других форматов. Как результат – качество звука лучше, но размер файла меньше.
Advanced Audio Coding (AAC) – собственнический (патентованный) формат аудиофайла с меньшей потерей качества при кодировании, чем MP3 при одинаковых размерах.
Также AAC – это широкополосный алгоритм кодирования аудио, который использует два основных принципа кодирования для сильного уменьшения количества данных, требуемых для передачи высококачественного цифрового аудио. Данный формат является одним из наиболее качественных, использующих сжатие с потерями, поддерживаемый большинством современного оборудования, в том числе портативного.
Видео на web-страницах
Видеофайл в любом из видеоформатов, следует воспринимать как zip-архив содержащий видеопоток и аудиопоток. Вот три видеоформата, наиболее распространенных в сети:
1) mp4 = H.264 + AAC;
2) .ogg/.ogv = Theora + Vorbis;
3) .webm = VP8 + Vorbis.
33
На звание кодека для HTML5 video в данный момент претендуют два кодека – Ogg Theora и H.264.
В основе *Ogg Theora* лежит кодек VP3, разработанный On2 Technologies. В 2002 г. On2 Technologies передали код VP3 под свободной BSD-подобной лицензией в руки Xiph.org Foundation, а также отказались от патентов на кодек (технически не отказались, а просто передали право их использовать всем, но это по сути то же самое). С тех пор Xiph.org продолжает развитие этого кодека.
Использовать Ogg Theora можно везде, всегда, без лицензионных или патентных отчислений.
H.264 – это лицензируемый стандарт сжатия видео. Его использование требует платы в странах, где действуют патенты на него (в первую очередь, это США). Однако, на сегодняшний день это один из самых лучших способов сжимать видео. Именно H.264 является стандартом дефакто сжатия HD-видео, к примеру, H.264 заметно эффективнее Ogg

Theora по соотношению качество/битрейт.
Если кратко, H.264 – лучше, но даже его open-source реализации не могут быть использованы свободно в странах, где действуют патенты на него.
Другим вариантом воспроизведения видео является декодирование видео в браузере с использованием модульного подхода без привязки к определенному кодеку. Мало того, в каждой операционной системе уже и так есть модульная инфраструктура кодеков. В Windows это DirectShow, в Mac OS X это QuickTime, в Linux это gstreamer. A gstreamer еще и кроссплатформенный, и, между прочим, уже используется в кроссплатформенных программах, к примеру, Songbird для воспроизведения музыки использует именно gstreamer на всех платформах.
Использование gstreamer решит все проблемы с кодеками в браузерах один раз и навсегда. В частности, не будет никаких проблем с патентами, так как браузер будет распространяться без защищенных патентами кодеков, но на системе пользователя он сможет найти установленный плагин для этого кодека и использовать его.
Кроме того, в gstreamer предусмотрена возможность использовать кодеки, установленные в родном для данной системы фреймворке (для Windows – DirectShow, для Mac OS – QuickTime).
34

Задание

Для выполнения практической работы № 3 необходимо изучить редактирование растровых изображений в ПО Adobe Photoshop, научиться делать корректуру фотографий, научиться вносить изменения в растровые изображения.

Порядок выполнения практической работы

Рассмотрим несколько примеров по корректуре фотографий, содержащих какие-либо недостатки.

1. Улучшение «недодержанных» фотографий.

Данная проблема возникает, как правило, в недостаточно освещенной обстановке, когда через объектив фотокамеры проникает недостаточно света при стандартных установках выдержки и диафрагмы.
Ниже приведен один из вариантов решения данной проблемы.
Откроем это изображение в программе Adobe Photoshop и перейдем к палитре слоев. Нажмем правой кнопкой по слою “Фон“ и выберем команду *Создать дубликат слоя*, в открывшемся диалоге нажмем кнопку ОК, оставив название нового слоя таким, как предлагаем программа.
Выделим новый слой и в списке методов наложения изменим *Обычный (Normal) на Экран (Screen)*
При необходимости предыдущие операции можно повторить, снова создав копию исходного слоя и выбрав режим наложения Экран (эти операции можно осуществить, нажав сочетание клавиш Ctrl + J. Вполне возможно, что на определенном этапе снимок будет выглядеть лишь немного недодержанным, поэтому создание очередной копии слоя приведет к передержке, т. е. нам будет нужна не целая часть слоя, а только половина или меньше.
Для этого можно уменьшить непрозрачность последнего слоя: плавно регулируем ползунок *Непрозрачность* (Opacity), до нужного результата

2. Исправление искажений перспективы.

В качестве примера возьмем следующую фотографию
Исходное изображение
Откроем изображение в программе Adobe Photoshop, создадим дубли-

кат слоя.

Включим линейки (Просмотр – Линейки) –

Теперь вытянем из включенных линеек направляющие, которые помогут исправить искажение. Установим три направляющие: две – слева и справа в основании ярко выраженных контуров и одну – в верхней точке крыши, чтобы видеть, что у изображения не слишком исказились пропорции по горизонтали и вертикали

Включим инструмент Искажение (Редактирование – Трансформирование – Искажение). Перетаскивая верхние левый и правый маркеры,

добьемся вертикальности контуров видим, что в результате деформаций верхняя граница крыши сместилась вниз. Вернем ее на прежний уровень, переместив центральный верхний маркер.

3. Корректировка цвета фотографии.

Для примера возьмем изображение, на котором слишком много красных цветов.

Откроем диалог *Изображение – Коррекция – Кривые*

Переключим цветовой канал на *Красный*

Далее укажем на фотографии с черным, серым и белым цветами с помощью трех пипеток.

При необходимости повторим те же операции для других цветовых каналов. Когда получим приемлемый результат, нажимаем кнопку ОК

Результат изменения цветовых кривых дано сравнение исходного и получившегося изображения.

4. Изменение цвета объекта на фотографии.

Изменим цвет купола зонта с фотографии.

Создадим новый слой и нарисуем в нем прямоугольник произвольного цвета

Изменим тип смешивания на *Цветность*

Купол зонта изменил свой цвет, но вместе с ним перекрасились ручка и спицы. Вернем им исходный цвет. Для этого создадим еще один слой,

затем включим инструмент *Архивная кисть***.** Подберем подходящий радиус и проведем кистью по тем элементам фотографии, которым нужно вернуть первоначальный цвет.

Вопросы для защиты работы

1. Какие форматы *web*-графики в основном используются на практике?
2. Что означает показатель dpi?
3. Назначение и основные особенности кодеков Ogg Theora и H.264.
4. Порядок действий по исправлению искажений перспективы.
5. Порядок изменения цвета объекта на фотографии.
6. Назначение и особенности форматов WAV, MP3.
7. Основные характеристики и назначение формата mp4.
8. Для каких изображений наилучшим образом подходит формат GIF?

Практическая работа № 4

ОФОРМЛЕНИЕ САЙТА «ИННОВАЦИОННЫЕ ТЕХНОЛОГИИ, ТОВАРЫ И УСЛУГИ» С ПОМОЩЬЮ КАСКАДНЫХ СТИЛЕЙ (CSS)

Время выполнения – 4 часа.

Цель работы: изучить принципы создания и использования каскадных стилей (CSS) для оформления элементов *web*-страницы.

Задачи работы

1. Изучить понятие каскадных стилей.

2. Научиться создавать стили и атрибуты и привязывать их к элементам HTML страницы.

3. Научиться задавать оформление элементов HTML страницы с помощью каскадных стилей.

4. Научиться создавать интерактивные элементы с помощью каскадных стилей.

Перечень обеспечивающих средств, необходимых для выполнения практической работы:

– индивидуальное задание на выполнение работы для 2 человек;

– персональный компьютер;

– программное обеспечение (ПО): любой текстовый редактор или специализированная среда разработки.

ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Для создания представления *web*-страниц предназначена технология ***каскадных таблиц стилей*** (Cascading Style Sheets, CSS), или просто *таблиц стилей*. Таблица стилей содержит набор правил *(стилей)*, описывающих оформление самой *web*-страницы и отдельных ее фрагментов. Эти

правила определяют цвет текста и выравнивание абзаца, отступы между графическим изображением и обтекающим его текстом, наличие и параметры рамки у таблицы, цвет фона *web*-страницы и многое другое. Стоит отметить, что в настоящее время, при достаточно большом количестве *web*-браузеров, каждый из них интерпретирует CSS-стили со своими особенностями, что следует учитывать при верстке HTML-страниц.

Каждый стиль должен быть привязан к соответствующему элементу *web*-страницы (или самой *web*-странице). После привязки описываемые выбранным стилем параметры начинают применяться к данному элементу. Привязка может быть явная, когда мы сами указываем, какой стиль к какому элементу *web*-страницы привязан, или неявная, когда стиль автоматически привязывается ко всем элементам *web*-страницы, созданным с помощью определенного тега.

Таблица стилей может храниться прямо в HTML-коде *web*-страницы или в отдельном файле. Последний подход более соответствует концепции Web 2.0; она требует, чтобы содержимое и представление *web*-страницы были разделены. Кроме того, отдельные стили можно поместить прямо в тег HTML, создающий элемент *web*-страницы; такой подход используется довольно редко и, в основном, при экспериментах со стилями.

Обычный формат определения стиля CSS:

Листинг 04.01

```
01 <селектор> {
02 <атрибут стиля 1>: <значение 1>;
03 <атрибут стиля 2>: <значение 2>;
04 . . .
05 <атрибут стиля n-1>: <значение n-1>;
06 <атрибут стиля n>: <значение n>
07 }
```

Селектор используется для привязки стиля к элементу *web*-страницы, на который он должен распространять свое действие. Фактически селектор однозначно идентифицирует данный стиль. За селектором через пробел указывают список атрибутов стиля и их значений, заключенный в фигурные скобки.

Атрибут стиля представляет один из параметров элемента *web*-страницы: цвет шрифта, выравнивание текста, величину отступа, толщину рамки и др.

З*начение* атрибута стиля указывают после него через символ: (двоеточие). В некоторых случаях значение атрибута стиля заключают в кавычки. Пары <*атрибут стиля*>:<*значение*> отделяют друг от друга символом: (точка с запятой).

Задание

Для выполнения практической работы № 4 необходимо изучить процесс использования каскадных стилей, научиться создавать стили и атрибуты и привязывать их к элементам HTML страницы, научиться задавать оформление элементов HTML страницы с помощью каскадных стилей, научиться создавать интерактивные элементы с помощью каскадных стилей.

Порядок выполнения практической работы

1. Установка визуального оформления базовых элементов.

Следуя этим инструкциям, последовательно установим визуальное отображение элементов страницы.

Один и тот же набор атрибутов можно одновременно устанавливать для нескольких элементов страницы, последовательно указав их селекторы через запятую.

Листинг 04.02

```
01 header,footer,
02 article,section,
03 hgroup,nav,
04 figure{
05 display: block;
06 }
```

Свойство display:block означает, что элемент будет отображаться как элемент блокового уровня. Блоковый элемент имеет отступы до и после

себя. Кроме этого в одной строке вместе с этим элементом не могут находиться другие элементы (если только дополнительно для них не применять свойство float).

Листинг 04.03

```
01 body{
02 font-size: 0.825em;
03 color: #fcfcfc;
04 background-color: #355664;
05 font-family: Arial, Helvetica, sans-serif;
06 }
```

Здесь устанавливаются значения по умолчанию шрифта (font-family, несколько шрифтов перечисляются на тот случай, если предпочтительный не будет найден у пользователя), его размера (font-size), цвета (color), цвета фона страницы (background-color).

В следующем блоке устанавливаем внешний вид ссылок (textdecoration – оформление текста, outline – наличие рамки, border – наличие окантовки):

Листинг 04.04

```
01 a, a:visited {
02 color: #0196e3;
03 text-decoration: none;
04 outline: none;
05 }
06 a:hover{
```

```
07 text-decoration: underline;
08 }
09 a img{
10 border: none;
11 }
```

2. Установка параметров шрифта.

Далее оформим текст логотипа, слогана, заголовков статей:

50

Листинг 04.05

```
01 h1,h2,h3{
02 font-family: "Myriad Pro","Helvetica Neue",Helvetica,Arial,Sans-
Serif;
03 text-shadow: 0 1px 1px black;
04 }
05 h1{
06 /* текст логотипа */
07 font-size: 3.5em;
08 padding: 0.5em 0 0;
09 text-transform: uppercase;
10 }
11 h3{
12 /* текст слогана */
13 font-family: forte,"Myriad Pro","Helvetica Neue",Helvetica,Arial,Sans-
Serif;
14 font-size: 2em;
15 font-weight: normal;
16 margin: 0 0 1em;
17 }
18 h2{
19 font-size: 2.2em;
20 font-weight: normal;
21 letter-spacing: 0.01em;
22 text-transform: uppercase;
23 }
```

Атрибут стиля text-shadow задает параметры тени:

text-shadow: none | <цвет> <горизонтальное смещение>
<вертикальное смещение> [<радиус размытия>]

Атрибут стиля padding позволяет сразу указать величины внутренних отступов со всех сторон элемента web-страницы:

padding: <отступ 1> [<отступ 2> [<отступ 3> [<отступ 4>]]]

Атрибут стиля text-transform позволяет изменить регистр символов текста:

text-transform: capitalize|uppercase|lowercase|none|inherit

Можно преобразовать текст к верхнему (значение uppercase этого атрибута) или нижнему (lowercase) регистру, преобразовать к верхнему регистру первую букву каждого слова (capitalize) или оставить в изначальном виде (none).

51

Атрибут стиля font-weight устанавливает «жирность» шрифта:

font-weight: normal|bold|bolder|lighter|100|200|300|400|500|600|
700|800|900|inherit

Атрибут margin задает величины отступа одновременно со всех сто-

рон элемента web-страницы:
margin: <отступ 1> [<отступ 2> [<отступ 3> [<отступ 4>]]]
Атрибут стиля letter-spacing позволяет задать дополнительное расстояние между символами текста:
letter-spacing: normal|<расстояние>
Оформление абзацев текста:
Листинг 04.06
```
01 p{
02 line-height: 1.5em;
03 padding-bottom: 1em;
04 }
```
3. Добавление элементов оформления секций и статей.
Определим вид линии, разделяющей секции страницы:
Листинг 04.07
```
01 .line{
02 height: 1px;
03 background-color: #24404c;
04 border-bottom: 1px solid #416371;
05 margin: 1em 0;
06 overflow: hidden;
07 }
```
Атрибут стиля overflow задает поведение контейнера при переполнении:
overflow: visible|hidden|scroll|auto|inherit
Также установим представление линии, отделяющей заголовок статьи от текста

Листинг 04.08
```
01 article .line{
02 background-color: #15242a;
03 border-bottom-color: #204656;
04 margin: 1.3em 0;
05 }
```
и линию, отделяющую нижнюю часть страницы:
Листинг 04.09
```
01 footer .line{
02 margin: 2em 0;
03 }
```
4. Установка визуального стиля для навигационной панели.
Теперь перейдем к установке стилей отображения навигационной панели и ее элементов:
Листинг 04.10
```
01 nav{
02 background: #f8f8f8;
03 padding: 0 5px;
04 position: absolute;
05 right: 0;
06 top: 4em;
07 border: 1px solid #FCFCFC;
08 -moz-box-shadow: 0 1px 1px #333333;
09 -webkit-box-shadow: 0 1px 1px #333333;
10 box-shadow: 0 1px 1px #333333;
```

```
11 }
12 nav ul li{
13 display: inline;
14 }
15 nav ul li a, nav ul li a:visited{
16 color:#565656;
17 display: block;
18 float: left;
19 font-size: 1.25em;
20 font-weight: bold;
21 margin: 5px 2px;
53
22 padding: 7px 10px 4px;
23 text-shadow: 0 1px 1px white;
24 text-transform: uppercase;
25 }
26 nav ul li a:hover{
27 text-decoration: none;
28 background-color: #f0f0f0;
29 }
30 nav, article, nav ul li a,figure{
31 /* Применение закругленных углов у элементов */
32 -moz-border-radius: 10px;
33 -webkit-border-radius: 10px;
34 border-radius: 10px;
35 }
```

Конструкция nav ul li a является комбинированным стилем и означает, что ссылка будет выглядеть соответствующе, только если окажется последовательно вложенной в элементы nav ul li.

В конструкции nav ul li a:hover элемент hover является новым компонентом CSS3. Он является псевдоклассом и позволяет создавать альтернативные стили для элементов. Эти стили будут присваиваться элементам при наведении на них курсором мыши. Указывать псевдоклассы следует после имени элемента или класса после знака ":". Данный псевдокласс применим не только к ссылкам, по большому счету, его можно применять для всех тегов.

Атрибут стиля box-shadow создает тень вокруг изображения или другого элемента. Box-shadow – это стандартное написание, варианты -mozbox-shadow и –webkit-box-shadow используются для браузеров FireFox и Chrome, Safari соответственно.

Display:inline означает, что элемент отображается как встроенный, содержимое блочных элементов начинается с того места, где окончился предыдущий элемент.

5. Установка визуального стиля для статей.

Установим оформление основных элементов содержания страницы – статей. Зададим окантовку, фон, размещение рисунков и текста:

54

Листинг 04.11

```
01 #page{
02 width: 960px;
03 margin: 0 auto;
04 position: relative;
```

```
05 }
06 article{
07 background-color: #213E4A;
08 margin: 3em 0;
09 padding: 20px;
10 text-shadow: 0 2px 0 black;
11 }
12 figure{
13 border: 3px solid #142830;
14 float: right;
15 height: 300px;
16 margin-left: 15px;
17 overflow: hidden;
18 width: 500px;
19 }
20 figure:hover{
21 -moz-box-shadow: 0 0 2px #4D7788;
22 -webkit-box-shadow: 0 0 2px #4D7788;
23 box-shadow: 0 0 2px #4D7788;
24 }
25 figure img{
26 margin-left: -60px;
27 }
```

Атрибут стиля overflow задает поведение контейнера при переполнении.

Доступны четыре значения:

- visible – высота контейнера увеличится, чтобы полностью вместить все содержимое (обычное поведение);
- hidden – не помещающееся в контейнер содержимое будет обрезано. Контейнер сохранит свои размеры;
- scroll – в контейнере появятся полосы прокрутки, с помощью которых можно просмотреть не помещающуюся часть содержимого. Эти по-

лосы прокрутки будут присутствовать в контейнере всегда, даже если в них нет нужды;
- auto – полосы прокрутки появятся в контейнере, только если в них возникнет необходимость.

6. ***Установка визуального стиля для нижней части страницы.***

И теперь осталось задать внешний вид нижней части страницы:

Листинг 04.12

```
01 footer{
02 margin-bottom: 30px;
03 text-align: center;
04 font-size: 0.825em;
05 }
06 footer p{
07 margin-bottom: -2.5em;
08 position: relative;
09 }
10 footer a,footer a:visited{
11 color: #cccccc;
12 background-color: #213e4a;
```

```
13 display: block;
14 padding: 2px 4px;
15 z-index: 100;
16 position: relative;
17 }
18 footer a:hover{
19 text-decoration: none;
20 background-color: #142830;
21 }
22 footer a.by{
23 float: left;
24
25 }
26 footer a.up{
27 float: right;
28 }
56
```

После того как в файл внесены все необходимые настройки, его нужно сохранить с расширением css и привязать к созданному ранее HTML файлу, дописав после тега <title> строку:

Листинг 04.13

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

В итоге получится примерно такой результат к ее элементам каскадных стилей

7. Создание меню средствами CSS.

CSS3 позволяет не только устанавливать оформление для элементов *web*-страницы, но и создавать анимацию и некоторые интерактивные элементы.

Для примера создадим выпадающее меню.

Сначала в HTML файле добавим дополнительные пункты списка в меню панели навигации:

Листинг 04.14

```
…
<ul>
<li><a href="#article1">Главная</a></li>
<li><a href="#article2">Раздел 1</a>
<ul>
<li><a href="#">Подраздел 1</a></li>
<li><a href="#">Подраздел 2</a></li>
<li><a href="#">Подраздел 3</a></li>
</ul>
</li>
<li><a href="#article3">Раздел 2</a>
<ul>
<li><a href="#">Подраздел 4</a></li>
<li><a href="#">Подраздел 5</a></li>
</ul>
</li>
</ul>
…
```

8. Добавление интерактивности к элементам меню.

Далее перейдем в файл CSS и добавим инструкции для новых элементов:

Листинг 04.15

```
01 nav ul li{
02 display:block;
03 float: left;
04 position: relative;
05 }
06 nav ul ul{
58
07 position:absolute;
08 top:40px;
09 }
10 nav li ul{
11 display: none;
12 }
13 nav li:hover > ul {
14 display:block;
15 }
16 nav ul ul li a, nav ul ul li a:visited{
17 font-size:0.95em;
18 padding:1px 3px 3px;
19 color:#ffffff;
20 }
21 nav ul ul li a:hover{
22 background-color:#213E4A;
23 }
```

Когда курсор мыши не наведен ни на один из элементов меню, вложенный список не отображается (строки 10-12).

Когда же курсор попадает на ссылку, скрытые элементы становятся видимыми (строки 13-15).

Символ “>” означает, что описанные атрибуты будут применены только к указанному типу, без влияния на остальные вложенные элементы

Вопросы для защиты работы

1. Роль селектора в технологии CSS.
2. Какие атрибуты используются для задания размера, цвета, «жирности», вида шрифта и расстояния между символами в тексте?
3. Перечислить применяемые значения атрибута стиля overflow.
4. Порядок действий при создании выпадающего меню.

60

Практическая работа № 5

ПРИМЕНЕНИЕ ЯЗЫКА СЦЕНАРИЕВ JAVASCRIPT В ПРОЕКТЕ САЙТА «ИННОВАЦИОННЫЕ ТЕХНОЛОГИИ, ТОВАРЫ И УСЛУГИ»

Время выполнения – 4 часа.

Цель работы: изучить основы использования языка сценариев JavaScript для создания интерактивных элементов на *web*-страницах.

Задачи работы

1. Изучить основы JavaScript.
2. Научиться создавать интерактивные элементы с использованием JavaScript.
3. Научиться пользоваться сторонними фреймворками.

Перечень обеспечивающих средств, необходимых для выполнения практической работы:

– индивидуальное задание на выполнение работы для 2 человек;
– персональный компьютер;
– программное обеспечение (ПО): любой текстовый редактор или специализированная среда разработки.

ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Объектная модель документа делает все элементы страницы программируемыми объектами. С ее помощью через языки сценариев можно получить доступ и управлять всем, что есть в документе. Каждый элемент HTML доступен как индивидуальный объект, а это означает, что можно изменять значение любого параметра любого тега HTML-страницы, и, как следствие, документ действительно становится динамическим. Любое действие пользователя (щелчок кнопкой мыши, перемещение мыши в ок-

не браузера или нажатие клавиши клавиатуры) объектной моделью документа трактуется как событие, которое может быть перехвачено и обработано процедурой сценария.

Поведение, т. е. набор правил, определяющих, как *web*-страница будет реагировать на действия посетителя, создается с помощью так называемых *web-сценариев* – программ, которые записывают прямо в HTML-коде *web*-страниц или, что предпочтительнее, в отдельных файлах. Эти программы пишут на языке *JavaScript. Web*-обозреватель считывает JavaScript-код и последовательно выполняет записанные в нем выражения, проводя вычисления и выполняя на основе полученного результата заданные манипуляции над *web*-страницей.

Общим заблуждением является то, что JavaScript аналогичен или тесно связан с Java, но это не так. Оба языка имеют C-подобный синтаксис, являются объектно-ориентированными и, как правило, широко используются в клиентских веб-приложениях, на этом их сходство заканчивается.

1. Java реализует ООП подход, основанный на классах, JavaScript на прототипах;
2. Java имеет статическую типизацию, JavaScript динамическую типизацию;
3. Java загружается из скомпилированного байт-кода; JavaScript интерпретируется напрямую из файла (но часто с незаметной JIT-компиляцией).

Динамическое содержимое на *web*-странице можно разделить на автономное, независимое от пользователя (анимированный логотип, падающие снежинки) и интерактивное, напрямую зависящее от его действий (игры, галереи изображений, перемещение элементов).

Задание

Для выполнения практической работы № 5 необходимо изучить принципы использования сценариев JavaScript, научиться создавать интерактивные элементы с использованием JavaScript, научиться пользоваться сторонними фреймворками.

Порядок выполнения практической работы

Для примера создадим на странице объекты обоих типов.

Автономным элементом будет прямоугольник, на котором рисуется волна, а интерактивным – фотогалерея.

1. Объект canvas.

Для создания визуального эффекта волны воспользуемся новым тегом HTML – canvas. Он предназначен для создания растрового изображения при помощи JavaScript. На сегодняшний день canvas чаще используется

для построения графиков, простой анимации и игр в веб-браузерах. Создадим холст в объекте, предназначенном для вставки изображения для первой статьи.

Листинг 05.01

```
01 <figure>
02 <canvas id="canvas"></canvas>
03 </figure>
```

Код на языке JavaScript может находиться как внутри текста HTML страницы, так и во внешнем файле. Второй способ удобен более «прозрачной» файловой структурой проекта, уменьшением размера и упрощением содержимого HTML документов, возможностью использовать одни и те же скрипты на разных страницах, при этом изменения в коде файла сразу отразятся на всех связанных с ним страницах.

2. Добавление сценариев JavaScript на web-страницу.

Код создаваемого скрипта поместим в файл waves.js и присоединим его к странице, добавив следующую строчку в конце секции body:

Листинг 05.02

```
01 </section> <!-- Закрываем секцию #page -->
02 <script language="JavaScript" src="src/waves.js"></script>
03 </body>
```


3. Установка оформления.

В файле CSS определим фоновый цвет, размеры и положение объекта canvas.

Листинг 05.03

```
01 #canvas {
02 background-color: #FAFAFA;
03 position: relative;
04 margin-top: 25px;
05 margin-left: 50px;
06 width: 400px; height: 250px;
07 }
```

Далее представлен код, содержащийся в файле waves.js:

Листинг 05.04

```
01 var canvas = document.getElementById('canvas');
02 var ctx = canvas.getContext('2d');
03 canvas.width = window.innerWidth;
04 canvas.height = window.innerHeight;
05 var waves = ["rgba(157, 187, 210, 0.3)"]
06 var i = 0;
07
08 function draw() {
09 canvas.width = canvas.width;
10
11 var offset = i + 1 * Math.PI * 12;
12 ctx.fillStyle = "#138CCB";
13
14 var randomLeft = (Math.sin(offset/100)+ 1) / 2 * 150;
15 var randomRight = (Math.sin((offset/100) + 10) + 1) / 2 * 150;
16 var randomLeftConstraint = (Math.sin((offset/60)+ 2)+ 1) / 2 * 150;
17 var randomRightConstraint = (Math.sin((offset/60)+ 1)+ 1) / 2 * 150;
18
```

```
19 ctx.beginPath();
20 ctx.moveTo(0, randomLeft + 150);
21
22 // ctx.lineTo(canvas.width, randomRight + 100);
23 ctx.bezierCurveTo(canvas.width / 3, randomLeftConstraint, canvas.width /
3 * 2, randomRightConstraint, canvas.width, randomRight + 150);
24 ctx.lineTo(canvas.width , canvas.height);
25 ctx.lineTo(0, canvas.height);
26 ctx.lineTo(0, randomLeft + 150);
64
27
28 ctx.closePath();
29 ctx.fill();
30
31 i = i + 3;
32 }
33
34 setInterval("draw()", 50);
```

4. Подключение внешних библиотек.

Перейдем к созданию интерактивного элемента – фотогалереи. В секцию head добавим две строки:

Листинг 05.05

```
<script type="text/javascript" src="src/jquery.js"></script>
<script type="text/javascript" src="src/cycle_photo.js"></script>
```

В первой строке мы подключаем внешнюю библиотеку JQuery (http://jquery.com/), она упрощает создание сайтов и интерактивных пользовательских интерфейсов.

JQuery Framework позволяет:

1) обращаться к любому элементу DOM (объектной модели документа), и не только обращаться, но и манипулировать ими;

65

2) работать с событиями;

3) легко осуществлять различные визуальные эффекты;

4) работать с AJAX (технология, позволяющая общаться с сервером без перезагрузки страницы);

5) имеет огромное количество JavaScript плагинов, предназначенных для создания элементов пользовательских интерфейсов.

5. Добавление фотогалереи на web-страницу.

Во второй строке подключен плагин, взятый с сайта http://jquery.malsup.com/cycle/, реализующий функции фотогалереи. Для его работы необходимо добавить следующий код, содержащий объект div с классом slideshow и описанием изображений, которые будут использоваться.

Листинг 05.06

```
01 <figure>
02 <div class="slideshow">
03 <img src="img/1.jpg" width="200" height="200" />
04 <img src="img/2.jpg" width="200" height="200" />
05 <img src="img/3.jpg" width="200" height="200" />
06 <img src="img/4.jpg" width="200" height="200" />
07 <img src="img/5.jpg" width="200" height="200" />
08 </div>
```

09 </figure>
Сам текст скрипта, указывающего параметры работы (fx – эффект, speed – скорость, next – элемент страницы, содержащий список изображений, timeout – пауза между сменами изображений, если установлен 0, то смена происходит по щелчку мыши), помещаем в секцию head.
Листинг 05.07
```
01 <script type="text/javascript">
02 $(document).ready(function() {
03 $('.slideshow').cycle({
04 fx: 'fade', speed: 2000, next: '.slideshow',
timeout: 0 });
05 });
06 </script>
```
66
Файл CSS дополним настройками объекта slideshow.
Листинг 05.08
```
01 .slideshow {
02 height: 225px;
03 width: 225px;
04 position: relative;
05 margin-left: 205px;
06 margin-top: 38px;
07 }
08 .slideshow img { padding: 10px; border: 1px solid #ccc; background-
color: #eee; }
```
Вопросы для защиты работы
1. Определение и назначение *web-сценариев.*
2. Перечислить принципы использования сценариев JavaScript
3. Привести примеры динамического содержимого на *web*-страницах.
4. Порядок создания автономного элемента типа падающих снежинок.
5. Сформулировать назначение и перечислить настройки объекта slideshow.

Практическая работа № 6
ПОДГРУЖАЕМЫЕ ЭЛЕМЕНТЫ СТРАНИЦ САЙТА «ИННОВАЦИОННЫЕ ТЕХНОЛОГИИ, ТОВАРЫ И УСЛУГИ»

Время выполнения – 4 часа.
Цель работы: изучить принципы составления единой *web*-страницы из отдельных подгружаемых элементов.
Задачи работы
1. Научиться разделять HTML страницу на отдельные элементы.
2. Научиться формировать HTML страницу из отдельных модулей.

Перечень обеспечивающих средств, необходимых для выполнения практической работы:
– индивидуальное задание на выполнение работы для 2 человек;
– персональный компьютер;
– программное обеспечение (ПО): любой текстовый редактор или специализированная среда разработки.

ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Страницы сайтов обычно строятся по модульному принципу – например, вверху каждой страницы находится название сайта с логотипом, внизу – знак охраны авторских прав, слева – навигационная панель, посредине – собственно содержание *web*-страницы (статья, рецензия и т. п.).

При этом на всех страницах все части, кроме содержания, одинаковы (т. е. одинаковая шапка, панель навигации и т. п.). Делать такие страницы отдельно – это плохой тон. Представьте себе, что вы добавили на сайт еще один раздел, и вам понадобилось изменить панель навигации на каждой странице сайта. При очень большом количестве страниц в составе сайта

это становится достаточно серьезной проблемой. Поэтому мы сейчас и посмотрим, как можно строить страницу по модульному принципу.

Задание

Для выполнения практической работы № 6 необходимо изучить принципы модульного построения *web*-страниц, научиться разделять HTML страницу на отдельные элементы, научиться формировать HTML страницу из отдельных модулей.

Порядок выполнения практической работы

1. Создание отдельных подгружаемых элементов web-страницы.

Заполним текстовые блоки, содержащие статьи текстом из внешнего файла. Для этого создадим дополнительную папку **html,** в которую положим два html файла с простой структурой, содержащих тексты первой и второй статей.

В основном html файле обрамим теги параграфов с заготовками текста тегами **div**, в которые будут загружаться созданные страницы:

Листинг 06.01

```
01 <div id="txtart1">
02 <p>Текст первой статьи</p>
03 </div>
```

Листинг 06.02

```
01 <div id="txtart2">
02 <p>Текст второй статьи</p>
03 </div>
```

2. Подключение загружаемых элементов.

Теперь, чтобы поместить текст в элементы страницы, создадим в теле страницы внедренный скрипт со следующим текстом:

Листинг 06.03

```
01 <script language="JavaScript">
02 $('#txtart1').load('html/page1.html');
03 $('#txtart2').load('html/page2.html');
04 </script>
```


Данный вариант будет «работать», только если подключен фреймворк jQuery, как в предыдущей работе.

В итоге получим следующий результат.

Вопросы для защиты работы

1. В чем заключается сущность и преимущества модульного принципа построения *web*-страницы?
2. Порядок действий при реализации модульного подхода к созданию *web*-страницы.

Практическая работа № 7

СОЗДАНИЕ ИНТЕРАКТИВНЫХ ЭЛЕМЕНТОВ САЙТА «ИННОВАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОЕКТЫ, ТОВАРЫ И УСЛУГИ» С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ FLASH

Время выполнения – 4 часа.

Цель работы: освоение методики создания интерактивных элементов сайта с использованием технологии Flash.

Задачи работы

1. Изучить базовые основы технологии Flash и ПО Adobe Flash.
2. Научиться применять анимацию движения.
3. Научиться применять анимацию формы.
4. Научиться вставлять Flash объекты на HTML страницу.

Перечень обеспечивающих средств, необходимых для выполнения практической работы:

– индивидуальное задание на выполнение работы для 2 человек;
– персональный компьютер;
– программное обеспечение (ПО) Adobe Flash.

ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Adobe Flash (ранее Macromedia Flash), или просто Flash, мультимедийная платформа компании Adobe для создания *web*-приложений или мультимедийных презентаций. Широко используется для создания рекламных баннеров, анимации, игр, а также воспроизведения на *web*-страницах видео- и аудиозаписей.

Платформа включает в себя ряд средств разработки, прежде всего Adobe Flash Professional и Adobe Flash Builder (ранее Adobe Flex Builder),

а также программу для воспроизведения flash-контента – Adobe Flash Player, хотя flash-контент умеют воспроизводить и многие плееры сторонних производителей. Например, SWF-файлы можно просматривать с помощью свободных плееров Gnash или swfdec, а FLV-файлы воспроизводятся через мультимедийные проигрыватели Quicktime, Windows Media Player и различные проигрыватели в UNIX-подобных системах при наличии соответствующих плагинов.

Adobe Flash позволяет работать с векторной, растровой и ограниченно – с трехмерной графикой, а также поддерживает двунаправленную потоковую трансляцию аудио и видео. Для КПК и других мобильных устройств выпущена специальная «облегченная» версия платформы Flash Lite, чья функциональность ограничена в расчете на возможности мобильных устройств и их операционных систем.

Стандартным расширением для скомпилированных flash-файлов (анимации, игр и интерактивных приложений) является .SWF (Shockwave Flash или Small Web Format). Видеоролики в формате Flash представляют собой файлы с расширением FLV (при этом Flash в данном случае используется только как контейнер для видеозаписи). Расширение FLA соответствует формату рабочих файлов в среде разработки.

Flash Player представляет собой виртуальную машину, на которой выполняется загруженный из Интернета код flash-программы.

В основе анимации во Flash лежит векторный морфинг, то есть плавное «перетекание» одного ключевого кадра в другой. Это позволяет делать сложные мультипликационные сцены, задавая лишь несколько ключевых кадров. Производительность Flash Player при воспроизведении анимации в несколько раз превышает производительность виртуальной машины Javascript в браузерах, поддерживающих предварительный стандарт HTML5, хотя во много раз уступает приложениям, работающим вообще без использования виртуальных машин.

Flash использует язык программирования ActionScript, основанный на ECMAScript.

Задание

Для выполнения практической работы № 7 необходимо изучить принципы создания элементов *web*-страниц с применением технологии Flash, научиться применять анимацию движения, научиться применять анимацию формы, научиться вставлять Flash объекты на HTML страницу. Используем технологию Flash для анимирования и наложения эффектов на буквы логотипа.

Порядок выполнения практической работы

1. Создание текста логотипа.

В Adobe Flash создадим пустой документ ActionScript 3.0

Выберем инструмент Text Tool и создадим в произвольном месте рабочей области объект с текстом *Логотип* и параметрами, примерно повторяющими текстовый логотип на странице

Зададим цвет фона рабочей области таким же, как и на странице. Для этого щелкнем правой кнопкой по рабочей области и выберем пункт Document Properties. Там установим размер рабочей области по содержимому Match – Contents и установим нужный цвет в Background color.

Теперь осталось поменять цвет шрифта на белый

Теперь создадим для этого текстового блока несколько эффектов.

2. Анимирование букв логотипа.

Для начала сделаем вращающиеся буквы при появлении логотипа.

Чтобы при изменении положения буквы не накладывались друг на друга, немного увеличим размер документа (Document Properties) и расстояние между буквами (Letter Spacing) –Выбрав объект, разделим надпись на отдельные составляющие и

применим команду Modify – Break Apart. И теперь, не снимая выделения, распределим получившиеся отдельные буквы по слоям Modify – Timeline-Distribute to Layers. В панели слоев мы должны увидеть следующий результат

Теперь проанимируем буквы. В начале ролика сделаем небольшую паузу, пропустив несколько кадров, о продолжительности этой паузы можно судить, зная что по умолчанию у ролика частота кадров составляет 24 кадра в секунду.

Добавим ключевой кадр для слоя, содержащего букву Л, щелкнув правой кнопкой мыши по 40-му кадру в этом слое и выбрав пункт меню Insert Keyframe

Это будет исходное состояние текстового объекта Л.

Выделим объект и сделаем из него символ с помощью команды меню Modify – Convert to Symbol. Все настройки в этом диалоге оставляем по умолчанию, кроме точки регистрации, ее переносим на центр.

Добавим следующий ключевой кадр с номером 60, не сбрасывая выделение с объекта, повернем его на 90 градусов по часовой стрелке командой Modify – Transform – Rotate 90° CW

Проделаем те же операции на 80-, 100- и 120-м кадрах.

Теперь поочередно щелкаем правой кнопкой мыши на созданных ключевых кадрах и выбираем пункт Create Classic Tween. На промежуточных кадрах при этом должна появляться стрелка на голубом фоне

Полученный результат можно проверить через меню Control – Test Movie – Test.

Возможно, что центр буквы не будет совпадать с центром вращения, для корркгировки нужно открыть символ для редактирования двойным

щелчком мыши и сместить букву относительно центральной точки.
Далее поочередно анимируем все остальные буквы с небольшим смещением по времени относительно предыдущей. Для того чтобы все буквы отображались, после анимации нужно продлить их статичное состояние до конца ролика, вставив в конце каждого слоя кадр с помощью команды контекстного меню Insert Frame
Проверяем полученный ролик, при наличии ошибок производим исправление.

3. *Добавление эффекта блика к логотипу.*

Теперь добавим эффект блика на последней букве после воспроизведения анимации.
Для этого на верхнем правом углу буквы П с помощью инструмента Oval Tool нарисуем круг с диаметром примерно 25 пикселей.

Вызовем панель Color (Window – Color), укажем, что обводка отсутствует. В списке типа заливки укажем *Radial Gradient.* Сдвинем левый регулятор градиента немного вправо и укажем цвет – #FFFFFF, значение прозрачности – 100 %. Правый регулятор переместим влево, цвет – #FECE41, прозрачность – 0 %.

Используя инструмент Line Tool , нарисуем следующую фигуру из пересекающихся линий поверх круга.
Создание базовых линий эффекта блика
Выделим все линии двойным щелчком по участку одной из линий и зададим в настройках цвета следующие параметры:

- заливка отсутствует;
- тип обводки – Radial Gradient;
- первый цвет градиента - #FECE41, прозрачность – 80 %;
- второй цвет градиента – #FFFFFF, прозрачность – 40 %.

Целиком выделим содержимое слоя, щелкнув по кадру, содержащему нарисованные объекты, и превратим их в символ командой Modify – Convert to Symbol.
В слое объектов блика создадим ключевые кадры на 10-м и 40-м кадрах относительно первого, содержащего исходные объекты.
Для примера: 370-й кадр будет начальным, в 380-м и 410-м кадрах созданы ключевые кадры.
Создание ключевых кадров для эффекта блика

Перейдем в последний кадр эффекта и выделим созданный символ, развернем панель Color Effect, выберем в списке Alpha и установим значение в 0 %.

Перейдем на промежуточный десятый кадр, в этой же панели выберем из списка Tint, установим значение эффекта в 100 %, цвет оставим по умолчанию (белый – 255,255,255).
Настройка цветового оттенка объекта
На этом же кадре в контекстном меню таймлайна выберем команду Create Classic Tween. В панели настроек установим значение параметра Rotate – CW x 1
Настройка анимации поворота
Перейдем в первый кадр анимации блика, выделим объект и включим инструмент Free Transform . Далее удерживая клавишу Shift, уменьшим размер символа до минимума. И теперь осталось в контекстном меню этого кадра указать команду Create Classic Tween.
Проверяем полученный эффект

4. Вставка полученного ролика на web-страницу.

Для того чтобы вставить полученный swf ролик в страницу, нужно поместить его в каталог img и заменить тег, содержащий текст ЛОГОТИП

Листинг 07.01

```
01 <h1>Логотип</h1>
```

на фрагмент

Листинг 07.02

```
01 <div id="flashContent">
02 <object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
width="150" height="50" id="logo" align="middle">
03 <param name="movie" value="logo.swf" />
04 <param name="quality" value="high" />
05 <param name="play" value="true" />
06 <!--[if !IE]>-->
07 <object type="application/x-shockwave-flash" data="img/logo.swf"
width="209" height="50">
08 <param name="movie" value="logo.swf" />
09 <param name="quality" value="high" />
10 <param name="play" value="true" />
11 </object>
12 </object>
13 </div>
```

Вопросы для защиты работы

1. Определение и основные возможности Adobe Flash.
2. В чем сущность анимации формы и анимации движения (привести примеры)?
3. Порядок действий при создании анимационного ролика.

ТЕСТОВЫЕ ЗАДАНИЯ
для подготовки к зачету по дисциплине
«Основы web-дизайна»

1. Заголовок *web*-страницы заключается в тег:
а) < HEAD > < /HEAD >;
б) < BODY > < /BODY >;
в) < HTML > < /HTML >;
г) < TITLE > < /TITLE >.
2. Строка таблицы обозначается тегом:
а) < p > < /p >;
б) < td > < td >;
в) < strong > < /strong >;
г) < tr > < /tr >.
3. Основное содержание *web*-страницы помещается в тег:
а) < p > < /p >;
б) < table > < / table >;
в) < title > < / title >;
г) < body > < /body >.
4. Выделенный элемент *web*-страницы, с которым связана информация об адресах переходов как внутри данной *web*-страницы, так и к другим *web*-страницам, называется:
а) тегом;
б) значком;
в) *web*-узлом;
г) гиперссылкой.

5. CSS – это:

а) технология описания внешнего вида документа;
б) метод установки PHP;
в) глобальный массив, хранящий переменные сессий;
г) директива в файле настройки php.ini.
6. Тег «…….» делает заключенный в него текст жирным:
а) < b > < /b >;
б) < u > < /u >;
в) < p > < /p >;
г) < h > < /h >.
7. Создать таблицу внутри уже существующей таблицы:
а) да, но не более чем в 3 строки;
б) да;
в) да, но только без рамки;
г) нет.
8. Использование цвета для оформления текста…
а) только стандартные 16 цветов;
б) 48 цветов палитры Редактора;
в) любые;
г) только черный.
9. Что произойдет с положением абзаца на странице при нажатии клавиши Align Right?
а) текст абзаца окажется посередине страницы;
б) текст абзаца прижмется к правому краю страницы;
в) текст абзаца прижмется к левому краю страницы;
г) текст абзаца прижмется к нижнему краю страницы.

10. Использоваться на странице могут дополнительные элементы оформления:
а) звуки;
б) видео;
в) бегущие строки;
г) все вышеперечисленные.
11. Гиперссылку на E-mail можно создать:
а) да;
б) да, если адрес находится в пределах данного домена;
в) да, если на странице указано имя владельца адреса e-mail;
г) нет.
12. Изображения, вставляемые на страницу:
а) переводятся в двоичную форму и помещаются в HTML-код;
б) записываются в архив и прилагаются к HTML-файлу;
в) изображения не сохраняются, а при просмотре используются из библиотеки пользователя;
г) сохраняются как отдельные файлы, а в HTML-код вставляется только ссылка на них.
13. HTML (Hyper Text Markup Language) является:
а) сервером Интернет;
б) языком разметки гипертекста;
в) языком программирования;
г) средством просмотра *web*-страниц.
14. Графика, представляемая в памяти компьютера в виде совокупности точек, называется:
а) растровой;

б) векторной;
в) трехмерной;
г) фрактальной.
85
15. Элементарным объектом растровой графики является следующий:
а) рисуется одним инструментом;
б) пиксель;
в) символ;
г) примитив.
16 Инструмент, позволяющий залить изображение двумя, плавно перетекающими друг в друга цветами, называется:
а) банка краски;
б) заливка;
в) градиент;
г) узор.
86

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Кирсанов, Д. Веб-дизайн [Текст] / Д. Кирсанов. – СПб.: Символ-Плюс, 2004. – 376 с.
2. Куприянов, Н. И. Рисуем на компьютере: Word, Photoshop, CorelDRAW, Flash [Текст] / Н. И. Куприянов. – СПб.: Питер, 2005. – 128 с.
3. Жадаев, Б. Macromedia Flash 8 [Текст] / Б. Жадаев. – М.: 100 книг: Триумф, 2007. – 368 с.
4. Леонтьев, Б. Web-дизайн: хитрости и тонкости [Текст] / Б. Леонтьев. – М.: Изд-во «Познавательная книга плюс», «МиК», 2001. – 224 с.
5. Гультяев, А. К. WEB-дизайн от MACROMEDIA [Текст]: практическое пособие / А. К. Гультяев. – СПб.: КОРОНА принт, 2001. – 480 с.
6. Гультяев, А. К. Dreamweaver 4-инструмент создания интерактивных Web-страниц [Текст]: практическое пособие / А. К. Гультяев. – СПб.: «КОРОНА принт», 2001. – 224 с.
7. ADOBE Web-дизайн и публикация. Энциклопедия пользователя [Текст]: настольная книга дизайнера Web-продукции / Д. Браун, В. Фримен, Б.Б. Хол и др. – Киев: ДиаСофт, 1998. – 656 с.
8. Рейнбоу, В. Компьютерная графика [Текст] / В. Рейнбоу. – М.: Питер, 2003. – 766 с.
9. Курушин, В. Д. Графический дизайн и реклама [Текст] / В. Д. Курушин. – М.: ДМК Пресс, 2001. – 272 с.
10. Панкратова, Т. Photoshop 6 [Текст]: учебный курс (+CD) / Т. Панкратова. – СПб.: ПИТЕР, 2002. – 480 с.
11. Лисицкий, Д. В. Анализ и методы использования современных web-технологий для создания интерактивных мультимедийных учебных пособий [Текст] / Д. В. Лисицкий, Е. В. Комиссарова, А. А. Колесников, В. В. Мандругин // Интеграция образовательного пространства с реальным сектором экономики. Ч. 4: сб. материалов Международной научно-методической конференции, 27 февраля – 2 марта 2012 г., Новосибирск. – Новосибирск: СГГА, 2012. – С. 107–110.
87
12. Колесников, А. А. Применение web-ГИС и мультимедийных технологий для картографического моделирования [Текст] / А. А. Колесников, Е. В. Комиссарова, В. А. Ракунов // Интерэкспо ГЕО-Сибирь-2013. IX Междунар. науч. конгр. : Междунар. науч. конф. «Геодезия, геоинформатика, картография, маркшейдерия» : сб. материалов в 3 т. (Новосибирск,

15□26 апреля 2013 г.). □ Новосибирск: СГГА, 2013. Т. 2. □ С. 96–101.
13. Лисицкий, Д. В. Методика проведения лекционных занятий на базе мультимедийного проектора и диалоговой доски / Д. В. Лисицкий // Вестник СГГА. □ 2011. – Вып. 1 (14). – С. 147–152.

Кроме того, можно воспользоваться следующими интернет-ресурсами:

- Официальный сайт научно-технической библиотеки СГГА. – Режим доступа: http://lib.ssga.ru/.
- Электронно-библиотечная система научно-издательского центра «ИНФРА-М». – Режим доступа: http://znanium.com/.
- Электронно-библиотечная система издательства «Лань». – Режим доступа: http://e.lanbook.com/.
- Научная электронная библиотека. – Режим доступа: http://elibrary.ru/.

Вопросы к зачету по окончании семестра:

1. Разработка концепции сайта, исходя из поставленных задач, целевой аудитории, фирменного стиля организации.

2. Создание основного конструктива сайта посредством языка верстки HTML.

3. Тег, правила использования, основные теги.

4. Правила размещения и настройка графических изображений. Используемы форматы в браузере.

5.Бегущая строка.

6. Таблицы, правила размещения.

7. Каскадные таблицы стилей (CSS)

8. Правила подключения таблиц на страницу.

9. Web – usability, основные правила.

10. Правила подготовки графических изображений.

11. Динамические и статические форматы.

12. Основные инструменты web – мастера в программе AdobePhotoshop.

13. Создание баннера в программе ImageReady.

14. Виды рекламы размещаемой на сайте.

15. Программа AdobeFlash, создание анимационного баннера, основные правила.

16. Программа AdobeDreamweaver. Основные инструменты.

17. Создание блока DIV.

18. Фрагменты кода.

19. Разработка собственной структуры сайта.

20. Меню Properties.

21. Создание навигационного меню на сайте, посредством палитры Behaviors. Правила создания эффекта Rollover.

22. Оценка сайта.

23. Хостинг, виды хостингов.

24. Ресурсы, связанные с вопросами web-дизайна и web-usability.

25. FTP протокол.

МИНИСТЕРСТВО КУЛЬТУРЫ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ КУЛЬТУРЫ»

УТВЕРЖДЕНО
Деканом факультета МАИС
О.А. Будариной
«06» октября 2015 г.

УТВЕРЖДЕНО
Зав. кафедрой дизайна
М.В. Решетовой
«06»октября 2015 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Веб-Дизайн

Направление подготовки: «Дизайн»

Профиль подготовки: Графический Дизайн

Квалификация Бакалавр

Форма обучения Очная

Согласовано:

С председателем методического совета по качеству по направлению

Москва

2015

Формируемые компетенции в результате освоения дисциплины (модуля): ОК-1, ОК-2, ОК-14, ПК-1, ПК-2, ПК-3;

- умеет логически верно, аргументировано и ясно строить устную и письменную речь (ОК-2);
- стремится к саморазвитию, повышению своей квалификации и мастерства (ОК-6);
- Осознает сущность и значение информации в развитии современного общества; владеет основными методами, способами и средствами получения, хранения, переработки информации (ОК-14);
- **самостоятельно приобретать новые знания, используя современные образовательные и информационные технологии** (ПК-1);
- может самостоятельно изображать объекты предметного мира, пространство и человеческую фигуру на основе знания их строения и конструкции (ПК-2);
- способен к определению целей, отбору содержания, организации проектной работы; синтезированию набора возможных решений задачи или подходов к выполнению проекта; готов к разработке проектных идей, основанных на творческом подходе к поставленным задачам; созданию комплексных функциональных и композиционных решений (ПК-3);

Паспорт

фонда оценочных средств

по дисциплине «**Веб-дизайн»**

			Оценочные средства		
№	**Контролируемые разделы, темы, модули[1]**	**Формируемые компетенции**	**Количество тестовых заданий**	**Другие оценочные средства**	
				Вид	Количество
1	Создание структуры сайта	ОК-14	-	Практическая работа	1

2	Виды сайтов: простые, корпоративные и креативные сайты, internet-магазины и тестовые программы.	ПК-2	-	Практическая работа	1
3	Формирование основных принципов работы с кодом страниц.	ОК-14	-	Индивидуальное творческое задание	1
4	Варианты встраивания CSS в код HTML.	ОК-14	-	Практическая работа	1
Всего:					4

[1]Наименования разделов, тем, модулей соответствует рабочей программе дисциплины.

«Московский государственный институт культуры»
Кафедра _____Дизайн_____
(наименование кафедры)

Вопросы

по дисциплине **Веб-дизайн**

---

1. Понятие сайт.

2. Виды сайтов.

3. Тег. Открывающиеся, закрывающиеся теги.

4. Правила оформления страницы.

5. Браузеры, виды браузеров.

6. Форматы файлов размещаемых внутри html страницы.

7. Программы, используемые для создания статичной и динамичной графики.

8. Фрейм. Фреймовая структура как альтернативное конструирование интернет-страниц.

9. CSS. Возможности использования каскадных таблиц стилей.

10. Варианты встраивания CSSв HTML.

11. Основные виды макетов в web.

12. Эргономика сайта.

13. Программа AdobePhotoshopCS5. Правила создания растровых изображений.

14. Программа AdobeFlashCS5. Создание баннера.

15. Программа AdobeDreamweaverCS5. Разработка собственной структуры сайта.

16. Существующие ресурсы по web – дизайну.

17. Основные бесплатные хостинги.

Критерии оценки (Приложение 1):

1. Оценка "отлично" выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
2. Оценка "хорошо" выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
3. Оценка "удовлетворительно" выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
4. Оценка "неудовлетворительно" выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка "неудовлетворительно" ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

- оценка «зачтено» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения;
- оценка «не зачтено» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

Составитель _____Тавлеева О.А._____ И.О. Фамилия
(подпись)

«____»_____20__ г.

Приложение 1

Оценочный лист Реферата

ФИО _____

Группа_____ Преподаватель_____

ДАТА _____

1 вариант

Критерии	ДА	НЕТ	Комментарий
ФОРМА			
деление текста на введение, основную часть и заключение			
деление текста на введение, основную часть и заключение логичный и понятный переход от одной части к другой, а также внутри частей с использованием соответствующих языковых средств связи			
СОДЕРЖАНИЕ			
соответствие теме			
наличие тезиса в вводной части и ее обращенность к читателю развитие тезиса в основной части (раскрытие основных положений через систему аргументов, подкрепленных фактами, примерами и т.п.)			
наличие выводов, соответствующих тезису и содержанию основной части			

«Московский государственный институт культуры»
Кафедра __Дизайн___
(наименование кафедры)

Темы индивидуальных творческих заданий

по дисциплине _Информационные технологии и компьютерная графика_
(наименование дисциплины)

Индивидуальные творческие задания (проекты):

Целью практических работ по дисциплине **«ВЕБ-дизайн»** - является формирование у студентов способности к типографическому оформлению продукции в зависимости от ее вида, готовности к обоснованному выбору шрифтового и композиционного оформления текстовых документов при осуществлении дизайнерской деятельности.

1. *Схематичное создание структуры сайта..*

Цель работы: ознакомиться с понятием "Структура сайта". Применить полученные знания.

Форма контроля – просмотр работ.

2. *Создание макета в программе Adobe Photoshop сайта -визитки.*

Цель работы: закрепить базовые знания программы Adobe Photoshop CS5 на примере создания сайта с несложной структурой.

Форма контроля – просмотр работ.

3.Подключение каскадной таблицы стилей CSS.

Цель работы: ознакомиться с каскадной таблицей стилей.

Форма контроля – просмотр работ.

Критерии оценки:

5. Оценка "отлично" выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
6. Оценка "хорошо" выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на

вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.

7. Оценка "удовлетворительно" выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
8. Оценка "неудовлетворительно" выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка "неудовлетворительно" ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

- оценка «зачтено» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения;
- оценка «не зачтено» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

КРИТЕРИИ ОЦЕНКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Уровень подготовки	Реализуемые компетенции
Базовый	В результате изучения дисциплины студент должен: ***Знать:*** основные теоретические понятия курса; ***Уметь:*** применять значительную часть полученных знаний на практике; выполнять основные задачи профессиональной деятельности, связанные со спецификой изучаемой дисциплины; ***Владеть***: базовыми навыками использования имеющихся знаний в собственной профессиональной деятельности.
Повышенный	В результате изучения дисциплины студент должен: ***Знать*** и понимать на более высоком уровне теоретические понятия курса, их связь с проектной культурой дизайна; ***Уметь*:** ориентироваться в современных сферах дизайна и разрабатывать проектную документацию; пользоваться основными методами проектирования; эффективно применять полученные теоретические знания в проектной деятельности; ***Владеть***: устойчивыми навыками использования имеющихся профессиональных знаний в собственной дизайнерской практике.
Продвинутый	В результате изучения дисциплины студент должен: ***Знать:*** на углубленном, расширенном уровне теоретические понятия курса, их связь с проектной культурой дизайна; ***Уметь*:** свободно ориентироваться и применять на практике избранные решения задачи или подходы к выполнению дизайн-проекта; пользоваться всем спектром методов проектной

	деятельности; с высокой эффективностью применять полученные теоретические знания в профессиональной деятельности; ***Владеть***: в совершенстве устойчиво сформированными навыками использования имеющихся профессиональных знаний в собственной дизайнерской практике.

Составитель _____Тавлеева О.А.
(подпись)

«____»_____2015 г.