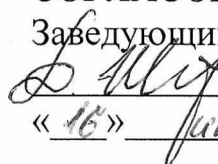


Учреждение образования
«Белорусский государственный университет культуры и искусств»

Факультет культурологии и социально-культурной деятельности
Кафедра информационных технологий в культуре

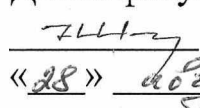
СОГЛАСОВАНО

Заведующий кафедрой

 Т.С. Жилинская
«16» июня 2023 г.

СОГЛАСОВАНО

Декан факультета

 Н.Е. Шелупенко
«28» августа 2023 г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

по учебной дисциплине учебного модуля

«Веб-дизайн и проектирование информационных ресурсов и систем»

ОСНОВЫ ФРОНТЕНД РАЗРАБОТКИ

*для специальности 1-21 04 01 Культурология (по направлениям),
направления специальности 1-21 04 01-02 Культурология (прикладная)
специализации 1-21 04 01-02 04 Информационные системы в культуре*

Составители:

О.М. Кунцевич, старший преподаватель кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств»

С.А. Гончарова, доцент кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств», кандидат технических наук, доцент

Рассмотрено и утверждено на заседании Совета факультета культурологии и социально-культурной деятельности

«28» августа 2023 г. протокол № 12

СОСТАВИТЕЛИ:

О.М. Кунцевич, старший преподаватель кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств»

С.А. Гончарова, доцент кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств» кандидат технических наук, доцент

РЕЦЕНЗЕНТЫ:

В.А. Касан, профессор кафедры информационных ресурсов и коммуникаций УО «Белорусский государственный университет культуры и искусств», кандидат педагогических наук, доцент

Кафедра дискретной математики и алгоритмики учреждения образования «Белорусский государственный университет»

Рассмотрен и рекомендован к утверждению:

Кафедрой информационных технологий в культуре
(протокол от 06.04.2023 № 9)

Советом факультета культурологии и социально-культурной деятельности

(протокол от «26» апреля 2023 г. № 10)

СОДЕРЖАНИЕ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	4
2. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ.....	5
2.1. Тематика лекционных занятий	5
2.2. Конспект лекций.....	5
3. ПРАКТИЧЕСКИЙ РАЗДЕЛ.....	26
3.1. Описание лабораторных работ	26
3.2. Описание практических занятий	44
4. РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ.....	46
4.1 Задания для управляемой самостоятельной работы.....	46
4.2 Перечень вопросов для проведения зачета.....	46
4.3 Перечень рекомендованных средств диагностики	48
4.4 Критерии оценки уровня знаний и умений студентов	48
5. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ.....	50
5.1 Учебная программа	50
5.2 Учебно-методическая карта учебной дисциплины для дневной формы получения высшего образования.....	57
5.4 Список основной литературы	59
5.5 Список дополнительной литературы	59

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В условиях современного общества неотъемлемым качеством квалифицированного специалиста сферы культуры и искусства становится высокий уровень владения интернет-технологиями для решения профессиональных задач по использованию, разработке, сопровождению и продвижению интернет-ресурсов в гипермедийном пространстве глобальной сети. Это предполагает знание основных технологий разработки сайтов, методов и инструментов их поискового и социального продвижения, умение применять их интегрировано для решения задач маркетинга и менеджмента.

Учебно-методический комплекс представляет собой совокупность учебно-методических материалов, необходимых и достаточных для организации учебного процесса по дисциплине «Основы фронтенд разработки». Для эффективного освоения учебного материала выполнена его систематизация в удобной и приемлемой форме в соответствии с учебной программой и образовательным стандартом подготовки специалистов по культурологическому профилю. Изучение учебной дисциплины «Основы фронтенд разработки» основывается на знаниях и умениях, полученных студентами в процессе освоения таких учебных дисциплин, как «Основы информационных технологий», «Информационная культура специалиста», «Системный анализ и моделирование информационных процессов». Знания, умения и навыки, полученные в рамках изучения данной учебной дисциплины необходимы для дальнейшего усвоения учебных дисциплин в первую очередь «Проектирование информационных ресурсов и систем», а также «Языки и системы программирования», «Технологии компьютерной графики», «Реклама в сфере культуры», «Технологии создания баз данных сферы культуры», «Мультимедийная техника и системное программное обеспечение», «Медиакультура специалиста».

Учебно-методический комплекс включает следующие разделы: пояснительную записку, теоретический, практический, контроля знаний, вспомогательный. Теоретический раздел состоит из конспектов лекционного материала по темам программы. Практический раздел содержит задания и методику выполнения практических и лабораторных работ. В разделе контроля знаний находятся задания для самостоятельной контролируемой работы, вопросы к итоговому контролю. Вспомогательный раздел учебно-методического комплекса содержит учебную программу, учебно-методические карты для дневной и заочной формы обучения, информационно-методическую часть, поясняющую методические приемы и критерии оценки учебных достижений студентов, список источников.

2. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

2.1. Тематика лекционных занятий

Лекция 1. Технологии веб-дизайна.

Лекция 2. Язык HTML. Технология CSS.

Лекция 3 Этапы анализа и проектирования сайта.

2.2. Конспект лекций

Лекция 1. Технологии веб-дизайна

Цель. Ознакомление с основными технологиями веб-дизайна. Изучение терминологии предметной области.

Основные вопросы:

1. Понятие веб-пространства Интернет. История создания www.
2. Классификация интернет-ресурсов, критерии оценки.
3. Понятие веб-клиента и веб-сервера.
4. Технологические средства веб: URL-адресация, HTML-язык и HTTP-протокол в контексте единой модели клиент-сервер.
5. Клиентские и серверные технологии веб-дизайна.

Интернет – всемирная система объединённых компьютерных сетей, построенная на базе протокола TCP/IP.

Интернет образует глобальное информационное пространство, служит физической основой для Всемирной паутины (World Wide Web, WWW) и множества других систем (протоколов) передачи данных.

3 июня 2011 года была принята резолюция ООН, признающая доступ в Интернет базовым правом человека. Отключение конкретных регионов от Интернета с июня 2011 года считается нарушением прав человека.

День Интернета – 30 сентября.

Ключевые принципы Интернета.

Интернет состоит из многих тысяч корпоративных, научных, правительственных и домашних компьютерных сетей. Объединение сетей разной архитектуры и топологии стало возможно благодаря протоколу IP (англ. Internet Protocol) и принципу маршрутизации пакетов данных.

Протокол IP был специально создан агностическим в отношении физических каналов связи. То есть любая система (сеть) передачи цифровых данных, проводная или беспроводная, для которой существует стандарт инкапсуляции в неё IP-пакетов, может передавать и трафик Интернета. Агностицизм протокола IP, в частности, означает, что компьютер или маршрутизатор должен знать тип сетей, к которым он непосредственно присоединён, и уметь работать с этими сетями; но не обязан (и в

большинстве случаев не может) знать, какие сети находятся за маршрутизаторами.

На стыках сетей специальные маршрутизаторы (программные или аппаратные) занимаются автоматической сортировкой и перенаправлением пакетов данных, исходя из IP-адресов получателей этих пакетов. Протокол IP образует единое адресное пространство в масштабах всего мира, но в каждой отдельной сети может существовать и собственное адресное подпространство, которое выбирается исходя из класса сети. Такая организация IP-адресов позволяет маршрутизаторам однозначно определять дальнейшее направление для каждого пакета данных. В результате между отдельными сетями Интернета не возникает конфликтов, и данные беспрепятственно и точно передаются из сети в сеть по всей планете.

Серф Винт и Роберт Кан – создатели протокола передачи данных TCP/IP, который по сей день является стандартом для передачи данных в Интернете.

Разница между понятиями Интернет и Веб (www).

Интернет: сеть компьютеров, объединенных каналами связи и использующих протоколы TCP/IP для связи.

Веб: сеть сайтов, использующих гиперссылки для переходов от страницы к странице.

Веб-сайт (от англ. *website*: *web* — «паутина», «сеть» и *site* — «место», букв. «место в сети») или просто сайт — в компьютерной сети объединённая под одним адресом (доменным именем или IP-адресом) совокупность документов частного лица или организации.

Второе определение веб-сайта подчеркивает его коммуникативную функцию.

Сайт — это набор информационных блоков и инструментов для взаимодействия с целевой аудиторией, которая может быть представлена реальными и потенциальными клиентами и партнерами, а также представителями средств массовой информации.

То, какая информация и как будет представлена на сайте, техническое оформление сайта находится в сильной зависимости от того, кто является целевой аудиторией и что сайт должен до нее донести и какие возможности предоставить.

Профессионально разработанный веб-сайт может служить как высокоэффективным инструментом ведения бизнеса, так и информационным или имиджевым ресурсом, рассказывающим о деятельности какой-либо общественной организации.

Задачи, которые решают сайты, многообразны – от организации внутренних или внешних коммуникаций компании (корпоративные ресурсы) и интернет-торговли (интернет-магазины) до самовыражения (сетевые

дневники) и объединения людей по интересам (комьюнити образующие ресурсы).

Сайт – это структурированная информационная единица всемирной паутины.

Классификация сайтов.

Виды сайтов по технологиям, влияющим на дизайн и функциональность:

- статические сайты и веб-страницы;
- динамические сайты и веб-страницы;
- флэш-сайты.

По принадлежности сайты подразделяются на:

- личные (персональные) сайты;
- сайты коммерческих организаций;
- сайты некоммерческих организаций.

К официальному сайту предъявляются более жесткие требования в части информационного содержимого, графического дизайна, навигации, хостинга.

Официальный сайт обычно имеет следующие разделы:

- новостная информация;
- нормативные документы, положения;
- направления деятельности;
- структура учреждения;
- кадровый состав;
- контактная информация (список ответственных лиц, их должности, координаты и часы приема).

Законодательство Республики Беларусь определяет свои требования к отдельным категориям сайтов.

Виды сайтов *по величине*, по уровню решаемых ими задач:

- простые сайты, содержащие немного информации и состоящие из нескольких страничек («сайты-визитки», домашние странички и т.п.);
- тематические, узконаправленные сайты;
- многофункциональные сайты (порталы).

Виды сайтов *по назначению*:

- сайты, предоставляющие контент;
- сайты для онлайн-контактов и общения;
- сайты электронной коммерции;
- сайты, предоставляющие онлайн-сервисы.

По характеру предоставляемого контента можно выделить:

- информационно-тематические сайты;
- новостные сайты;
- развлекательные сайты;
- сайты-библиотеки;

- сайты-базы определённого рода документов;
- разнообразные сайты-справочники, онлайн-энциклопедии и словари,
- сайты-каталоги, обобщающие информацию о других сайтах и т.д.

По доступности сайты бывают:

- открытые – открыты для всех посетителей;
- полуоткрытые – часть информации открыта для всех, а часть скрыта, поэтому для просмотра скрытой информации на таком сайте необходима регистрация;

- закрытые.

<http://archive.org/web/> – веб-архив сайтов с момента появления Веба.

Способы активного отображения информации во Всемирной паутине.

Информация в вебе может отображаться как пассивно (то есть пользователь может только считывать её), так и активно — тогда пользователь может добавлять информацию и редактировать её. К способам активного отображения информации относятся:

- гостевые книги,
- форумы,
- чаты,
- блоги,
- вики-проекты,
- социальные сети,
- системы управления контентом.

Перспективы развития Всемирной паутины.

В настоящее время наметились две тенденции в развитии Всемирной паутины: семантическая паутина и социальная паутина. Семантическая паутина предполагает улучшение связности и релевантности информации во Всемирной паутине через введение новых форматов метаданных. Социальная паутина полагается на работу по упорядочиванию имеющейся в Паутине информации, выполняемую самими пользователями Паутины. В рамках второго направления разработки, являющиеся частью семантической паутины, активно используются в качестве инструментов (RSS и другие форматы веб-каналов, микроформаты XHTML).

12 критериев качества ресурсов: прозрачность, эффективность, поддержка, доступность, ориентация на пользователя, реактивность, многоязычность, совместимость, управляемость, сохранность, производительность, посещаемость.

Персоналии веб-дизайна.

Сэр Тимоти Джон Бернерс-Ли (англ. Sir Timothy John «Tim» Berners-Lee; род. 8 июня 1955) — британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и

действующий глава Консорциума Всемирной паутины. Автор концепции семантической паутины. Автор множества других разработок в области информационных технологий.

Роберт Кайо (англ. Robert Cailliau) (родился 26 января 1947) совместно с сэром Тимом Бернерсом-Ли изобрёл технологию Всемирной паутины (World Wide Web). Место рождения: Тонгерен, Бельгия. В 1989 году независимо от Тима Бернерса-Ли предложил систему гипертекст для доступа к документации CERN. В 1990 году это привело к совместному предложению этой технологии, а затем и к созданию Всемирной паутины (World Wide Web).

17 мая 1991 более 20 лет назад сотрудник Европейской лаборатории по ядерным исследованиям CERN в Женеве, консультант по программному обеспечению Тим Бернес-Ли приклеил на один из компьютеров в своей лаборатории наклейку с надписью «This machine is a server, DO NOT POWER IT DOWN!!». Этот первый в мире веб-сервер реализовал „большую технологическую тройку“: URL-адресацию, HTML-разметку и HTTP-протокол в контексте единой модели клиент-сервер.

Первый в мире веб-сайт появился в Интернете 6 августа 1991 года по адресу <http://info.cern.ch/>.

На этом сайте описывалось, что такое Всемирная паутина, как установить веб-сервер, как заполнить браузер и т. п. Этот сайт также являлся первым в мире интернет-каталогом, потому что позже Тим Бернерс-Ли разместил и поддерживал там список ссылок на другие сайты.

W3C. 1994 г. – основную работу по развитию Всемирной паутины взял на себя Консорциум Всемирной паутины (англ. World Wide Web Consortium, W3C), основанный и до сих пор возглавляемый Тимом Бернерсом-Ли. Консорциум разрабатывает и внедряет единые принципы и технологические стандарты для Интернет и Всемирной паутины (называемые «Рекомендациями», англ. W3C Recommendations).

Миссия W3C: «Полностью раскрыть потенциал Всемирной паутины, путём создания протоколов и принципов, гарантирующих долгосрочное развитие Сети». Две другие важнейшие задачи Консорциума – обеспечить полную «интернационализацию Сети» и сделать Сеть доступной для людей с ограниченными возможностями.

Все Рекомендации Консорциума Всемирной паутины открыты, то есть не защищены патентами и могут внедряться любым человеком без всяких финансовых отчислений консорциуму.

Официальный сайт Консорциума Всемирной паутины (W3C) www.w3.org

О нескольких важных принципах:

Возможность редактировать информацию Паутины не менее важна, чем возможность просто путешествовать по ней. В этом смысле Бернерс-Ли очень рассчитывает на концепцию WYSIWYG, хотя Wiki — это тоже шаг в нужном направлении.

Компьютеры могут быть использованы для «фоновых процессов», помогающих людям работать сообща.

Каждый аспект Интернета должен работать как паутина, а не как иерархия. В этом смысле очень неприятным исключением является система имён доменов (англ. Domain Name System, DNS), управляемая организацией ICANN.

Учёные-компьютерщики несут не только техническую ответственность, но и моральную.

Развитие веб-дизайна шло по двум основным направлениям:

– с развитием программных и инструментальных средств:

1. HTML-дизайн; использование в дизайне таблиц для разметки;
2. CSS (каскадные таблицы стилей);
3. Flash (динамичекий дизайн);
4. CMS (система управления контентом);
5. HTML5, CSS3, Ajax (Asynchronous Javascript and XML).

– с развитием технических средств:

1. совершенствование средств вывода информации (мониторы, их разрешение и цветопередача);
2. увеличение пропускной способности каналов связи.

Браузеры. Текстовые документы, содержащие код на языке HTML (такие документы традиционно имеют расширение .html или .htm), обрабатываются специальными приложениями, которые отображают документ в его форматированном виде. Такие приложения, называемые браузерами, обычно предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер.

Веб-сервер – это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, обычно вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными.

Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает.

Клиент, которым обычно является веб-браузер, передаёт веб-серверу запросы на получение ресурсов, обозначенных URL-адресами. Ресурсы – это HTML-страницы, изображения, файлы, медиа-поток или другие данные, которые необходимы клиенту. В ответ веб-сервер передаёт клиенту запрошенные данные. Этот обмен происходит по протоколу HTTP.

Наиболее распространённые веб-серверы:

Apache – свободный веб-сервер, наиболее часто используемый в Unix-подобных операционных системах;

NGINX – веб-сервер для высоко нагруженных ресурсов, используемый в Unix-подобных операционных системах;

IIS от компании Microsoft, распространяемый с ОС семейства Windows.

Технологии построения веб-приложений

Клиентские средства разработки веб-приложений: HTML, CSS, JavaScript.

Серверные технологии разработки веб-приложений: PERL, PHP, PYTHON, ASP и др.

Актуальные технологии при разработке веб-приложений: XHTML, микроформаты, фреймворки, AJAX, JSON, RSS, Ruby/Ruby on Rails, ASP.NET, Silverlight.

Технология AJAX (асинхронный JavaScript + XML) была известно давно. Однако, благодаря появлению термина AJAX, который ввел Джис Джеймс Гаррет (Jesse James Garrett) в феврале 2005г., она стала необычайно модной. Суть технологии AJAX заключается в изменении содержимого загруженной веб-страницы без ее полной перезагрузки, благодаря чему достигается высокая динамичность сайтов. Технология основывается на разделении данных и подзагрузки тех или иных компонентов по мере необходимости.

AJAX технологией в строгом смысле слова не является. Это просто аббревиатура, обозначающая подход к созданию веб-приложений с помощью следующих технологий: стандартизированное представление средствами XHTML и CSS; динамическое отображение и взаимодействие с пользователем с помощью DOM; обмен и обработка данных в виде XML и XSLT; JavaScript; асинхронные запросы с помощью объекта XMLHttpRequest.

Преимущества:

Экономия трафика. Вместо загрузки всей страницы достаточно загрузить только изменившуюся часть, как правило довольно небольшую.

Уменьшение нагрузки на сервер. К примеру, на странице работы с почтой, когда вы отмечаете прочитанные письма, серверу достаточно внести изменения в базу данных и отправить клиентскому скрипту сообщение об успешном выполнении операции без необходимости повторно создавать страницу и передавать её клиенту.

Ускорение реакции интерфейса. Поскольку нужно загрузить только изменившуюся часть, пользователь видит результат своих действий быстрее.

Недостатки:

Отсутствие интеграции со стандартными инструментами браузера.

Динамически создаваемые страницы не регистрируются браузером в истории посещения страниц, поэтому не работает кнопка «Назад», предоставляющая пользователям возможность вернуться к просмотренным ранее страницам, но существуют скрипты, которые могут решить эту проблему.

Другой недостаток изменения содержимого страницы при постоянном URL заключается в невозможности сохранения закладки на желаемый материал. Частично решить эти проблемы можно с помощью динамического изменения идентификатора фрагмента (части URL после #), что позволяют многие браузеры.

Динамически загружаемое содержимое недоступно поисковикам (если не проверять запрос, обычный он или XMLHttpRequest).

Усложнение проекта. Перераспределяется логика обработки данных — происходит выделение и частичный перенос на сторону клиента процессов первичного форматирования данных. Это усложняет контроль целостности форматов и типов. Конечный эффект технологии может быть нивелирован необоснованным ростом затрат на кодирование и управление проектом, а также риском снижения доступности сервиса для конечных пользователей.

Лекция 2. Язык HTML. Технология CSS

Цель: Изучение структуры документа и синтаксиса языка HTML, таблиц каскадных стилей CSS, знакомство с методами реализации динамических страниц.

Основные вопросы

1. Язык HTML: история создания, версии.
2. Элементы веб-страницы.
3. Синтаксис языка HTML.
4. Таблицы каскадных стилей CSS.
4. Методы реализации динамических страниц.

Основные определения.

Веб-страница – гипермедийный HTML-документ.

HTML (Hyper Text Markup Language) – язык гипертекстовой разметки – язык создания веб-страниц.

Браузер – программа, предназначенная для просмотра веб-страниц.

Веб-сайт – группа веб-страниц, связанных единой темой, схемой оформления и гипертекстовыми ссылками.

Язык HTML – язык, предназначенный для создания форматированного текста, который насыщен изображениями, звуком, анимацией и ссылками на другие объекты, например, гипертекстовые документы, графические файлы и т.д., разбросанные по всему пространству веб-серверов Интернет/Интранет.

Версии HTML:

HTML – данный стандарт не использовался, но стал основой для других версий языка;

HTML+ – разработан в 1993 г.;

HTML 2.0 – данная спецификация появилась в 1994 г.;

HTML 3.0 – проект данной спецификации был опубликован в 1995 г., но в качестве стандарта реализован не был;

HTML 3.2 – первая публикация в 1996 г.;

HTML 4.0.

HTML 5 – с 2014 г.

Веб-страница с точки зрения файловой структуры.

Веб-страница – это: HTML-документ (.htm) + файлы мультимедиа + активные компоненты.

Структура сайта:

Логическая – информационная организация.

Физическая – алгоритм размещения файлов по подкаталогам корневой папки сайта.

Имена файлов – html-страниц и папок: только латинские символы, цифры, некоторые спецсимволы, без пробелов и никакой кириллицы! Примеры: index.htm, dom.html, Lelik-2.htm.

Именованная главная (первая) страница сайта: index.html, index.htm.

Абсолютный и относительный путь:

xyz.html – файл HTML с именем xyz.html находится в текущем каталоге;

abc/xyz.html – файл HTML с именем xyz.html находится в подкаталоге abc текущего каталога;

http://myDomain.by/abc/xyz.html – файл HTML с именем xyz.html находится в каталоге abc на сервере MyDomain.by;

../abc/xyz.html – файл HTML с именем xyz.html находится в подкаталоге abc каталога, лежащего на один уровень выше текущего;

../../abc/xyz.html – файл HTML с именем xyz.html находится в подкаталоге abc каталога, лежащего на два уровня выше текущего.

Элементы веб-страницы.

Заголовок. Текст, отображающийся в строке заголовка браузера при просмотре страницы. Первым появляется при загрузке, служит закладкой на страницу, имеет больший вес при индексации.

Текст.

Фон. Фоновый цвет. Фоновый звук. Фоновый рисунок.

Графика. В HTML-документе присутствует лишь адрес файла с графическим изображением:

D:\site\image\ris.gif (ошибка: абсолютная адресация на файл локального компьютера)

D:\мои рисунки\ris.gif (еще одна ошибка: кириллица)

..\..\image\ris.gif (правильно)

Форматы графических файлов в Интернет: растровые: gif, png, jpg и векторные: svg.

Размер изображения задается в пикселях или % от ширины окна браузера.

Альтернативный текст – отображается в режиме отключения графики или неграфическими браузерами.

Функции графики: фон, миниатюры, графические карты, визуал, навигация.

Элементы веб-страницы: гиперссылки. Гиперссылка – это средство установки связей в веб-пространстве.

Элементы гиперссылки: указатель ссылки, адресная часть.

Указатель гиперссылки: выделенный фрагмент текста, графические изображение, кнопка, изображение-карта.

Адресная часть гиперссылки: ведет на определенную область страницы (закладка), иную страницу того же сайта или документ другого типа на данном компьютере, страницу внешнего сайта (URL), другой сервис Интернет (адрес e-mail, телефон).

Элементы веб-страницы: таблицы, задаются в пикселях или % окна, выполняют функции метода представления данных и средства компоновки страниц.

Динамические компоненты: счетчик посещений, рекламный баннер, формы, бегущая строка, анимационные эффекты при переходе к другой странице, полиморфная кнопка и т.д.

Фрейм – область окна браузера, в которой осуществляется просмотр отдельного HTML-документа.

Установочный файл фреймов – отдельный документ HTML, задающий способ раскладки и форматирования фреймов, и альтернативное представление для браузеров, не поддерживающих фреймы.

Зачем изучать HTML: гибкость; глубина понимания; упрощение отладки; цена; независимость.

Синтаксис языка HTML.

Тег (дескриптор) – инструкция языка HTML. Тег может быть открывающий (<тег>) и закрывающий (</тег>), причем открывающий тег может иметь атрибуты (параметры), влияющие на его поведение.

Контейнер (блок)– структура, состоящая из открывающего или открывающего и закрывающего тегов.

Примеры:

<ТЕГ параметр1=значение1 параметр2=значение2 ...>

текст

другие конструкции

</ТЕГ>

<ТЕГ параметр1=значение1...> текст другие конструкции</ТЕГ>

<ТЕГ параметр1=значение1...> текст другие конструкции

Кодировки текста – однобайтовые – 256 символов: ASCII; KOI8 (код обмена информацией) (KOI8-R); CP1251; CP866; ISO 8859-5; двухбайтовые – 65636 символов: UNICODE; четырехбайтовые: ISO 10646; другие: UTF-8 – является комбинацией ISO 10646 и UNICODE.

Структура документов HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3//DTD HTML 4.0//EN"
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>
```

...заголовок

```
</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

... тело документа

```
</BODY>
```

```
</HTML>
```

Документ HTML является блочным элементом и сам состоит из блоков. Два основных блока – это HEAD (определяет свойства документа) и BODY (определяет тело документа). Каждый из них включает другие блоки. HTML-документ состоит из стандартных элементов разметки: заголовки, списки, таблицы, параграфы и т.д., – которые разделены на два типа: строчные и блочные. К блочным относятся параграф, список, таблица. К строчным относятся начертание, текст гипертекстовых ссылок.

Содержимое области HEAD. В области заголовка документа могут содержаться как теги, описывающие свойства документа, так коды программ (сценариев), вызываемых в теле документа. Примеры тегов:

TITLE – определяет текст, отображаемый в заголовке браузера;

META – определяет различные свойства документа, например, тип кодировки, ключевые слова, описание документа, автор и т.д.;

base href – указывает адрес документа;

link – определяет ссылку на другой файл.

Пример организации области HEAD:

```
<HEAD>
```

```
<TITLE>Моя страничка</TITLE>
```

```
<META name="keywords" content="список терминов">
```

```

<META name="description" content="краткое описание содержания...">
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=windows-1251">
  <META NAME="Author" CONTENT="Svetlana Hancharova">
  <META NAME="GENERATOR" CONTENT="Mozilla/6.05 [en]
(WinNT; I) [Netscape]">
<base href="http://www.buk.by/itk.html">
<link rel="StyleSheet" type="text/css" href="http://www.buk.by/site.css">
</HEAD>

```

Элемент BODY. Тег <BODY> определяет тело документа, имеет различные атрибуты для определения цвета фона документа, цвета текста, цвета ссылок и т.д. Пример записи:

```

<BODY параметр1=данные1, параметр2=данные2...>
текст документа
</BODY>

```

Параметры элемента BODY:

bbgcolor – определяет цвет фона для тела документа;

text – определяет цвет, используемый при выводе на экран текста из данного документа;

background – определяет адрес URL, откуда будет братья изображение для фона текущего документа;

link – определяет цвет, который будет использоваться при выводе на экран текста из еще не выбранных вами гипертекстовых связей;

vlink – определяет цвет, который будет использоваться при выводе на экран текста из уже проверенных вами гипертекстовых связей;

alink – задает цвет, которым будут выделяться в тексте гипертекстовые связи в тот момент, когда пользователь щелкает по ним клавишей мыши.

Элементы, задающие шрифт:

```
<FONT size=... color=...> текст </FONT>
```

size – устанавливает размер шрифта. Изменяется в пределах от 1 до 7. Размер по умолчанию – 3. Может быть указан относительный размер, например **size="+1"** или **size="-2"**;

color – указывает цвет, которым будет выделен данный текст. Цвета задаются в виде RGB-значения с шестнадцатеричной нотацией, либо выбирается один из 16 стандартных цветов, описанных в элементе BODY при рассмотрении атрибута BGCOLOR.

Элементы форматирования на уровне блоков:

Параграф <p align=(left|center|right|justify)>

Заголовок <H1|H2|H3|H4|H5|H6 align=...>

Горизонтальная линия <hr align=... size=... width=...>

Перевод каретки (новая строка)

Элементы, задающие начертание:

<TT> телетайпный текст

<I> *стиль с наклонным шрифтом*

 стиль с жирным шрифтом

<U> стиль с подчеркиванием текста

<BIG> печать текста шрифтом увеличенного размера

<SMALL> печать текста шрифтом уменьшенного размера

<SUB> печать текста со сдвигом вниз (нижний индекс)

<SUP> печать текста со сдвигом вверх (верхний индекс)

<STRIKE> ~~стиль с перечеркиванием текста~~

Маркированные списки:

 ... первый пункт списка

 ... второй пункт списка

...

Нумерованные списки:

 ... первый пункт списка

 ... второй пункт списка

...

Таблица

<TABLE BORDER=... WIDTH=... >

<TR>

<TD параметры=... > 1-я клетка 1-ой строки </TD>

<TD параметры=... > 2-я клетка 1-ой строки </TD>

</TR>

<TR>

<TD> 1-я клетка 2-ой строки </TD>

...

</TR>

...

</TABLE>

Типы связей. Связь, устанавливаемая между гипертекстовыми документами, называется ссылкой (link) или гиперссылкой (hyperlink). Существуют следующие типы связей:

– якорь (anchor, target) или закладка (bookmark) – установка метки перед определенным блоком документа;

- ссылка на закладку (якорь) – обращение к установленной метке;
- относительная ссылка – путь к файлу относительно корневого каталога узла;
- абсолютная ссылка – полный путь к файлу с указанием доменного адреса узла и пути к файлу на данном узле.

Ссылки:

` текст или указатель на объект `

Примеры:

`Щелкните по этой ссылке`

` Белый кораблик!`

`Письмо`

Якорь (anchor).

Определение якоря – `Maryia`

Обращение к якорю – `Maryia`

Изображения:

`<img src=... alt=... align=... width=... height=... border=... hspace=...
vspace=...>`

``

Описание атрибутов тега `img`:

`src` – путь к файлу;

`alt` – текст подписи;

`align` – выравнивание;

`width` – ширина;

`height` – высота;

`border` – толщина контура;

`hspace` – отступ текста по горизонтали от изображения;

`vspace` – отступ текста по вертикали от изображения.

Формы:

`<FORM METHOD=POST ACTION=«email»>`

`<INPUT TYPE=CHECKBOX>` – в `<form>` создается элемент ввода «опция»;

`<INPUT TYPE=FILE>` – в `<form>` создается элемент ввода «выбор файла»;

`<INPUT TYPE=HIDDEN>` – в форме создается скрытый элемент;

`<INPUT TYPE=IMAGE>` – создается элемент ввода «изображение»;

`<INPUT TYPE=PASSWORD>` – создается элемент ввода текста с обеспечением защиты содержимого;

`<INPUT TYPE=RADIO>` – создается элемент ввода «селекторная кнопка»;

`<INPUT TYPE=RESET>` – создается кнопка сброса;

`<INPUT TYPE=SUBMIT>` – создается кнопка передачи;

`<INPUT TYPE=TEXT>` – создается элемент ввода текста;

тег OPTION – с помощью конструкции тегов <option> определяется меню внутри элемента <select>;

тег SELECT – начальный и конечный теги определяют меню с несколькими вариантами выбора или список;

тег TEXTAREA – многострочная область ввода текста в форме;
</FORM>

Технология CSS (Cascading Style Sheets).

Говоря о дизайне html-документов, основной упор делается на то, что элементы документа должны точно размещаться относительно друг друга в окне браузера. Элементами документа являются текстовые блоки, графика и другие объекты, для каждого из которых определяются свои настройки. Чтобы избежать повторения настроек для различных блоков с одинаковым оформлением, была предложена технология CSS (Cascading Style Sheets – каскадные таблицы стилей). Технология CSS позволяет определять форму представления документа. Спецификация CSS level 1 (CSS1) стала стандартом в 1996 г., CSS2 – в 1998 г., а CSS3 – в 2014.

Использование элемента <STYLE> – один из основных способов внедрения таблицы стилей в HTML-документ. Элемент <STYLE> управляет как отображением элементов разметки, так и описывает стилевые свойства элементов. Позволяет определить стиль отображения для: стандартных элементов разметки (тегов); произвольных классов; объектов HTML.

Пример:

```
<style><!--
```

TAG {параметр:значение; ...} – определение стиля для конкретного тега;

TAG.Class {параметр:значение;} – определение класса элементов, которые будут отображаться одинаково для группы элементов разметки;

.Class {параметр:значение;} – определение класса элементов, которые будут отображаться одинаково;

```
--></style>
```

XML (eXtensible Markup Language). Язык XML (eXtensible Markup Language – расширяемый язык разметки) разделяет содержание и представление документа. Позволяет пользоваться любыми тегами, создаваемыми автором в процессе разработки документа. В рамках XML разрабатываются фиксированные системы разметки документов, например, MathML – язык разметки математических текстов, ChemML – язык разметки химических текстов. Об XML можно сказать, что это HTML+семантическая разметка.

Методы реализации динамических страниц:

JavaScript – модули интегрируются в файл HTML как подпрограмма;

Java – модули существуют как самостоятельные приложения (апплеты .class);

PHP (Personal Home Page) – сценарии вставляются в тот участок документа, где необходимо разместить интерактивный элемент; позволяет организовать счетчик посещений, статистику обращений к разделам сайта, защищать доступы к файлам паролями и т.д.

ASP (Active Server Page) – скрипт выполняется на сервере, пользователю отсылается готовый HTML-документ с результатами работы ASP. Программное обеспечение пользователя не имеет значения.

DHTML – расширение стандарта HTML – использование сценариев («скриплетов»), обрабатываемых браузером совместно с кодом HTML. Позволяет создать движущиеся объекты, выпадающие меню, подсвечивающиеся кнопки и др.

XML (eXtensible Markup Language). Язык XML (eXtensible Markup Language – расширяемый язык разметки) разделяет содержимое и представление документа. Позволяет пользоваться любыми тегами, создаваемые автором в процессе разработки документа. В рамках XML разрабатываются фиксированные системы разметки документов, например, MathML – язык разметки математических текстов, ChemML – язык разметки химических текстов. Об XML можно сказать, что это HTML + семантическая разметка.

Средства создания веб-страниц:

- простейшие текстовые редакторы – Блокнот, Far Editor, Norton Editor;
- специализированные HTML-редакторы – AceHTML или Notepad++;
- редакторы визуального проектирования WYSIWYG – MS VisualStudio.NET., Adobe Dreamweaver.
- системы управления сайтом (веб-контентом).

Content management systems (CMS) – программный комплекс для создания, публикации, редактирования и организации содержимого, настройки и администрирования сайта.

Среди основных функций:

- хранение содержимого сайта в БД и/или в файловой системе, наличие средств для управления информационным содержанием;
- стандартизация представления информации на сайте, использование шаблонов, позволяющих централизованное редактирование;
- масштабируемость по функциональности и по нагрузке;
- управление пользователями, разделение ролей.

Преимущества CMS:

- уменьшение трудозатрат на создание веб-сайтов, т.е. сроков и стоимости разработки;
- возможность концентрироваться в ходе разработки на обеспечении удобства для целевых пользователей;
- повышение качества информационного продукта;

- снижение трудоемкости и стоимости поддержки информации, снижение требований к квалификации персонала – с системами может работать и неспециалист в сфере ИТ (контент-менеджер, вебмастер, дизайнер);

- улучшение возможностей дальнейшего развития продукта – за счёт модульной архитектуры, разделения данных и их представления.

Особенности и недостатки CMS:

- необходимость работать с чужими кодом и архитектурой системы;
- требовательность к компетенциям программиста (специализация);
- более низкая производительность по сравнению со специализированными программными решениями (универсальность не бывает «бесплатной»);

- требовательность к программным ресурсам: PHP; MySQL, PostgreSQL; ASP.NET, C#, Java, VB.NET, Python и др.; MSSQL, Oracle

- необходимость настройки и доработки под конкретные задачи, которая не всегда осознаётся неспециалистами (CMS легко принять за готовое решение).

Лекция 3 Этапы анализа и проектирования сайта

Цель: Изучение этапов предпроектного анализа и проектирования сайтов. Знакомство с методами поискового продвижения: поисковой оптимизацией и маркетингом в социальных сетях, особенностями поискового продвижения веб-ресурсов культуры и искусства. Ознакомление с законодательным регулированием байнета.

Основные вопросы

- 1. Этапы анализа сайта.*
- 2. Этапы проектирования сайта.*
- 3. Основы поисковой оптимизации,.*
- 4. Особенности поискового продвижения ресурсов культуры и искусства в Интернет.*
- 5. Законодательное регулирование белорусского сегмента сети интернет.*

Этапы анализа сайта:

- определение назначения и критериев успешной разработки сайта;
- определение состава пользователей и способов использования ими сайта;
- определение и организация информационных тем;
- анализ ограничений и определение методов их преодоления;
- уточнение назначения, критериев работы и состава информации узла с учетом ограничений.

- анализ сайтов-«конкурентов».

Цели и задачи сайта. Главные и второстепенные цели. Критерии достижения цели (как оценивать степень успешной реализации поставленных целей).

Примеры задач:

- публикация информации об организации;
- расширение круга пользователей;
- предоставление информации о своих продуктах и услугах;
- обеспечение возможности пользователям присылать отзывы и предоставлять дополнительные сведения;
- прием заказов на продукты и услуги в оперативном режиме;
- в рамках интрасети (внутренней сети компании) предоставление доступа к административным распоряжениям, документации и переписке компании;
- в рамках экстрасети (внешней сети с ограниченным кругом пользователей) предоставление сторонним организациям доступа к части интрасети компании, например, для электронной коммерции.

Анализ состава пользователей сайта. Кто входит в состав пользователей сайта? Каковы потребности пользователей? Какая информация требуется пользователям? Могут ли обращаться к сайту конкуренты?

Проанализировать каждую категорию пользователей по следующим критериям:

- интересы, потребности, навыки, способности и предпочтения;
- платформа, браузер, быстродействие коммутируемого соединения и степень подготовки и опыта работы в Интернет;
- описания платформ потенциальных пользователей, включая изготовителя оборудования, модели, оперативную память, жесткие диски и другое оборудование.

Определение информационной тематики. Организация информационных тем по категориям. Состояние разработки информационных тем.

Ограничения сайта, оценка их значимости:

- низкая пропускная способность;
- применение пользователями браузеров, предназначенных только для текста;
- недостаточная пропускная способность для применения графических и мультимедийных файлов:
- недостаточные знания разработчиков компании в области HTML и мультимедиа;
- большие затраты и трудоемкость разработки многоязычных версий.

Корректировка проекта веб-сайта с учетом ограничений.

Этапы проектирования веб-сайта.

- определение концептуальной модели сайта (описания назначения, темы и стиля);
- построение логической структуры сайта (блок-схемы навигации, на которой показана организация и пути навигации по сайту):
 - проектирование средств навигации;
 - проектирование информационного наполнения;
 - построение или обновление прототипа (прототипов),
 - разработка дизайна и программирование сайта;
 - тестирование сайта.

Основы поискового продвижения.

Чтобы увеличить аудиторию пользователей, ежедневное количество посетителей необходимо популяризировать ресурс (сайт), то есть произвести его оптимизацию и продвижение в ведущих поисковых системах Интернет.

Под продвижением сайта понимается комплекс мер по обеспечению посещаемости сайта целевыми посетителями. Целевые посетители – это посетители, которые заинтересованы в информации товаре или услуге, представленной на продвигаемом сайте.

Одним из важнейших этапов продвижения сайта является поисковая оптимизация (Search engine optimization), которая представляет собой комплекс мер по повышению позиций сайта в поисковых системах, и, таким образом, позволяет увеличить его целевую посещаемость. Это длительный и комплексный процесс сбора всех возможных усилий на популяризацию конкретного ресурса.

Аудитория радио, телевидения, печатных СМИ, наружной рекламы и Интернета качественно отличается. Отличие не только в том, что Интернет является общей средой, но и в том, что продвижение сайта, в отличие от роликов, магистральных щитов и модулей в газетах – именно целевая реклама.

Существуют белые, серые и черные методы продвижения.

«Белые» методы (также называют «Белое SEO») являются законными с точки зрения поисковых систем, но продвижение сайта белыми методами занимает больше времени, и требует большего денежного вложения ввиду своей трудоемкости.

«Серые» методы можно трактовать как что-то среднее между черными и белыми методами продвижения, так как они условно законные. В сущности, эти способы продвижения сайта являются полулегальными и поэтому не лишены риска, что продвигаемый сайт не «забанят».

Под «черными» методами продвижения сайта понимают применение

методов, запрещенных правилами поисковых систем, в частности, нарушающие лицензию Яндекса и Google, а также нарушающие правила участия в рейтинге сайтов (например, TOP-100). Использование таких методов – прямая дорога сайту в «бан» (исключение из индекса поисковой системы).

Методы продвижения сайта делятся на две категории:

1. Внутренняя оптимизация (работа с сайтом: выбор ключевых слов, прописание мета-тэгов, оптимизация html-кода, перелинковка страниц, оригинальные тексты, регулярное обновление контента, RSS-рассылки, файлы robots.txt и sitemap.xml и др.).

2. Работа с внешними факторами (внешние ссылки, упоминания на других интернет-ресурсах, регистрация в поисковых системах и каталогах, реклама ресурса на тематических форумах, участие сайта в рейтингах, продвижение сайта в социальных медиа, продвижение сайта в закладочных сервисах, контекстная реклама, обмен банерами, создание рассылок и др.).

Особенности продвижения сайтов общественных и некоммерческих организаций.

Некоммерческие цели – когда сайты созданы не для того, чтобы приносить прибыль, предлагать, продвигать, рекламировать или продавать товары, продукты, товары и услуги. Такие интернет-проекты называются некоммерческими. Тематические ресурсы, сайты государственных учреждений сферы культуры и искусства, общественных организаций, благотворительных фондов и др., занимающиеся различными социальными проектами, различные клубы по интересам, образовательные учреждения.

Основная цель некоммерческих сайтов или сайтов общественных организаций – информирование определенной группы людей, следовательно, деятельность по продвижению таких сайтов должна быть направлена на обновление контента сайта (добавление новостей, событий, объявлений, касающихся самой организации).

Одной из особенностей продвижения таких сайтов является широкое использование методов офлайн-рекламы. Организации, которые занимаются активной общественной деятельностью, проводя конференции, семинары, акции, мероприятия, могут продвигать свой сайт в офлайн режиме, распространяя брошюры, листовки, бюллетени с адресом на собственный веб-сайт, а также указывая его на баннерах и растяжках в помещениях, где проводят свои мероприятия.

Достаточно эффективным методом продвижения являются социальные сети и размещение пресс-релизов. Пресс-релизы представляют собой короткие (одна-две страницы) официальные информационные сообщения для прессы, рассказывающие о различных мероприятиях, событиях в жизни общественной организации или учреждения. Их можно размещать на

собственном ресурсе либо на специализированных сайтах, как правило, бесплатно. Размещение пресс-релизов дает многосторонний эффект. Во-первых, их могут прочитать посетители тех ресурсов, где вы их размещаете. Во-вторых, сайтами, где есть пресс-релизы, пользуются многие журналисты. Ваш пресс-релиз может стать основой для публикаций в сетевых и печатных изданиях. В-третьих, разместив в пресс-релизах гиперссылки на свои ресурсы, вы сможете повысить их цитируемость.

Для некоммерческого сайта в Интернет можно предложить следующие методы продвижения: внутренняя оптимизация; регистрация в поисковых системах; обмен баннерами с другими сайтами; обмен ссылками с другими сайтами; участие в рейтинговых системах, создание рассылки сайта (RSS-рассылки), рекламирование на форумах, продвижение в социальных сетях.

Законодательное регулирование белорусского сегмента сети интернет:

ЗАКОН РЕСПУБЛИКИ БЕЛАРУСЬ 10 ноября 2008 г. № 455-З Об информации, информатизации и защите информации. Источник: <https://pravo.by/document/?guid=3871&p0=h10800455> – Национальный правовой Интернет-портал Республики Беларусь

УКАЗ ПРЕЗИДЕНТА РЕСПУБЛИКИ БЕЛАРУСЬ 1 февраля 2010 г. № 60 О мерах по совершенствованию использования национального сегмента сети Интернет. Источник: <https://pravo.by/document/?guid=3871&p0=r31000060> – Национальный правовой Интернет-портал Республики Беларусь.

3. ПРАКТИЧЕСКИЙ РАЗДЕЛ

Методические указания к практическим и лабораторным работам

Темы практических и лабораторных занятий, предусмотрены в рамках дисциплины «Основы фронтенд разработки» высшего образования по учебной дисциплине для студентов специальности 1-21 04 01 Культурология (по направлениям), направления специальности 1-21 04 01-02 Культурология (прикладная) специализации 1-21 04 01-02 04 Информационные системы в культуре. Предложенные для работы темы практических и лабораторных занятий взаимосвязаны и требуют от студентов последовательного изучения содержания дисциплины. Основными материалами, используемыми студентами в ходе подготовки к лабораторным занятиям, являются конспекты лекций, печатные и электронные документальные источники информации, приведенные в учебно-методическом комплексе. Объем заданий рассчитан на выполнение студентами в рамках учебных занятий. При отсутствии студента на занятии задание должно быть выполнено самостоятельно и передано преподавателю на проверку.

3.1. Описание лабораторных работ

Лабораторная работа №1

Методы оптимизации графики для веб

Тема 2. Язык HTML. Технология CSS

Цель: Уменьшение размера графического изображения до 30Кб и сохранение с минимальными потерями качества в формате JPEG

Теоретические сведения:

Размер файла изображения (его «вес») в Кб напрямую зависит от нескольких параметров:

1. Цветовое решение картинки - играет самую большую роль. Чем больше оттенков и мелких деталей на картинке, тем сложнее будет уменьшить это изображение. Предмет, сфотографированный на однотонном матовом фоне, лучше оптимизируется, чем тот же предмет на пестром фоне.

2. Геометрические размеры изображения также определяют размер файла в Кб. Оптимальные размеры изображения для просмотра на среднестатистическом мониторе – не более 800 x 600 пикселей (точек).

3. Глубина цвета (битовое разрешение). По умолчанию в AdobePhotoshop установлена 8-битная глубина цвета, не стоит ее изменять.

4. Перед оптимизацией рекомендуется сделать предварительное редактирование фотографии (а именно, кадрирование и удаление лишних деталей фона) можно добиться высокого качества компрессии.

5. Каждое последующее пересохранение отредактированного файла формата JPEG вызывает повторное сжатие изображения, приводя к невосполнимым потерям его качества, и в первую очередь именно резкости. Если после сохранения вы не удовлетворены качеством, лучше начать сначала, а не переделывать сохраненный файл.

Отсюда вывод – всегда работайте с копиями, оставляя оригинал нетронутым!

Задание 1. Оптимизация изображений в формате JPG

Методические рекомендации выполнения:

1.1. Откройте изображение в программе APhotoshop.

1.2. По необходимости отредактируйте изображение.

1.3. Выберите File (Файл) → SaveForWeb(Сохранить для Web).

1.4. В центре открывшегося окна размещается оптимизируемая картинка. Щелкните на заголовок Optimized (Оптимизация) в верхней части окна (зачастую окно с заголовком открывается Optimized по умолчанию).

1.5. Уменьшим геометрические размеры изображения. Щелкните на заголовке ImageSize (Размер Изображения) в правой части экрана. В полях Width (Ширина) или Height (Высота) изменяем значения. По самой длинной стороне выставляете цифру 600 или 800. Включаете (или если включено, оставляете включенной) галку около ConstrainProportions для сохранения пропорций изображения. Установив размер, щелкните кнопку Apply (Применить) чтобы применить новый размер. Если размеры ширины или высоты картинки уже меньше 800-600 точек, то менять ничего не нужно.

Окно Save for Web сохраняет пропорции изображения.

1.6. В поле формат из выпадающего списка выберем тип выходного файла JPEG.

В поле ввода Quality (Качество) ставим цифру 50. Чем эта цифра больше, тем лучше будет фотография.

1.7. Выберем качество сжатия (компрессии) изображения: щелкните на круглую кнопку со стрелкой. Затем в появившемся меню нужно выбрать строку Optimizeforfilesize (Оптимизировать по размеру файла). Откроется диалоговое окно. В поле DesiredFileSize введите с клавиатуры цифру 30. Щелкните кнопку ОК. Программа автоматически оптимизировала изображение до 30Кб.

В левом нижнем углу окна SaveforWeb указан наиболее важный параметр всего процесса уменьшения изображения – размер файла, который не должен превышать установленный лимит в 30 Кб. После манипуляций

обращайте на этот параметр, возможно, что на каком-то этапе вы уже добились требуемого значения, без существенной потери качества.

1.8. Если качество изображения значительно «упало», а размер еще велик, уменьшим геометрические размеры картинки еще раз (п. 3). Уменьшите большую из сторон сначала на 100 пикселей (вместо "800" поставьте сначала 700, а если тоже будет большой размер, то поставьте 600) и повторите пункт 5. Повторите его необходимое число раз, контролируя «вес» файла и качество изображения.

Если «вес» файла не превышает 30Кб и качество вполне приемлемо, то щелкните на кнопке Ok (справа вверху окна SaveforWeb) и, выбрав имя файла и путь к нему, сохраните его на диск.

Задание 2. Оптимизация изображений в формате GIF

Методические рекомендации:

2.1. Загрузите изображение в Photoshop. Выберите File (Файл) — >SaveForWeb(Сохранить для Web).

2.2. В диалоговом окне нажмите на закладки вверху окна для просмотра изображения в одном из типов:

- Original – просмотр Оригинала;
- Optimized – оптимизированное изображение с учетом заданных параметров оптимизации;
- 2-Up – одновременный просмотр Оригинала и оптимизированного изображения;
- 4-Up – одновременный просмотр Оригинала и трех оптимизированных изображений с разными параметрами.

Окошечко подсвечивается синей рамкой, чтобы узнать какое из них активно в данный момент.

2.3. Выберите в списке Формат файла – GIF. Ознакомьтесь с настройками формата (рисунок 1).

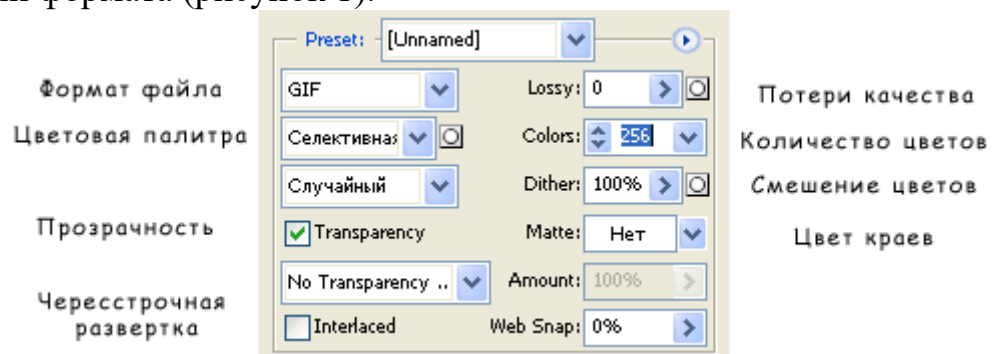


Рисунок 1 – Основные параметры настройки формата gif

2.4. Переместите ползунок Lossy или введите число для удаления части пикселей из изображения, тем самым уменьшится размер файла. Вы можете изменять этот параметр от 5% до 30%, не боясь сильно ухудшить

качество рисунка. Размер файла при этом уменьшается существенно, в зависимости от выбранных опций.

Нельзя использовать этот параметр с включенной опцией Interlaced(Чересстрочная развертка).

2.5. Задайте палитру цветов. Формат GIF сохраняет прямо в файле используемые в изображении цвета, вот только их количество ограничено числом 256. Зато это могут быть любые цвета. Чаще используется Perceptual (Перцепционная).

2.6. Параметр Colors. Его значение показывает максимальное количество цветов в изображении. Если у вас файл содержит полноцветное изображение с большим количеством цветов, то значение должно быть максимальным. Если укажите меньше, чем требуется, то возможны искажения в цветопередаче.

2.7. Для создания чересстрочной развертки изображения отметьте опцию Interlaced. При загрузке изображения оно будет появляться постепенно, вначале показывая рисунок с плохим качеством, а потом постепенно улучшая.

2.8. Если вам необходимо изображение с прозрачным фоном, то установите галочку Transparency (Прозрачность). Изображение готово к сохранению! Нажмите ОК.

На вкладке ColorTable (Таблица цветов) отображается палитра всех цветов, которые содержатся в изображении. Внизу находятся кнопки.

Если дважды щелкнуть на каком-либо цвете палитры, то откроется окно Палитра цветов. В меню палитры цветов можно выбрать команды изменения, сортировки цветов, загрузки и сохранения палитры.

Задание 3. Параметры оптимизации PNG-файлов

В случае формата PNG-8 (количество цветов не более 256) параметры оптимизации те же, что и для формата GIF, за исключением того, что параметра Lossy нет. Для формата PNG-24 (24-битная глубина цвета) вы можете лишь указать значения параметров Transparency (Прозрачность) и Interlaced (Чересстрочный).

Вкладка ColorTable (Таблица Цветов) может использоваться только для PNG-8 также, как для формата GIF. На вкладке ImageSize (Размер изображения) можно изменить размер или разрешение картинки.

3.1. Откройте подготовленные изображения в AdobePhotoshop.

3.2. Оптимизируйте изображение (размер файла не более 300 кб) для дальнейшего размещения в Internet в формате png.

3.3. Сохраните 2 варианта выбранного изображения, не превышая заданные размеры: 300*300 px и 800*800 px (рисунок 2).

3.4. Для изображения размера 300*300 px можно сделать дополнительную копию, например, добавить фильтр или перевести в черно-белое изображение (рисунок 3).

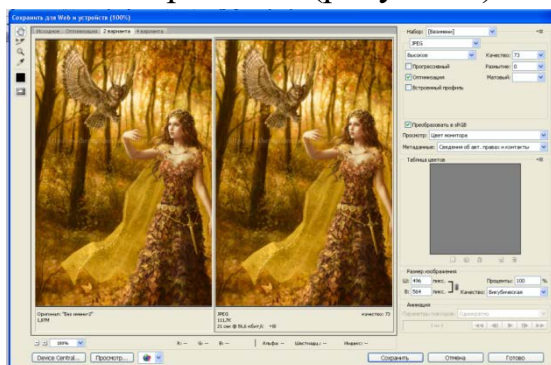


Рисунок 2 – Сохранение в формате gif

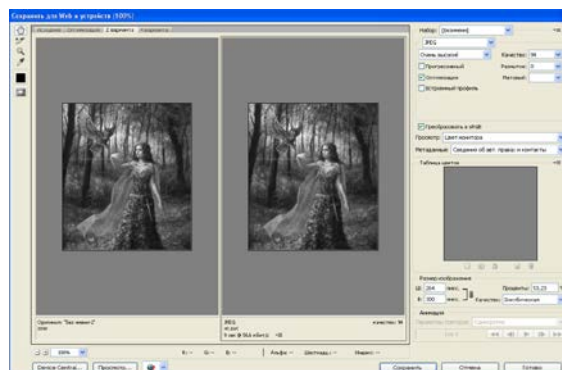


Рисунок 3 – Изменение изображение

Задание 4. Сохранение фрагмента PNG-файлов

4.1. Откройте изображение.

4.2. Отделите изображение от фона и сохраните файлы с прозрачным фоном (png).

Лабораторная работа №2

Элементы программирования на языке HTML.

Структура документа. Основные теги

Тема 2. Язык HTML. Технология CSS

Цель: освоение студентами синтаксиса HTML, принципов разделения содержимого и оформления веб-страницы.

Задание 1. Знакомство с HTML.

1.1. Создание и сохранение файла в проекте: создать в папке html-документ, для этого открываем Sublime Text, нажимаем File / New File. В появившемся окне, в поле «Имя файла» написать index.html (рисунок 4).

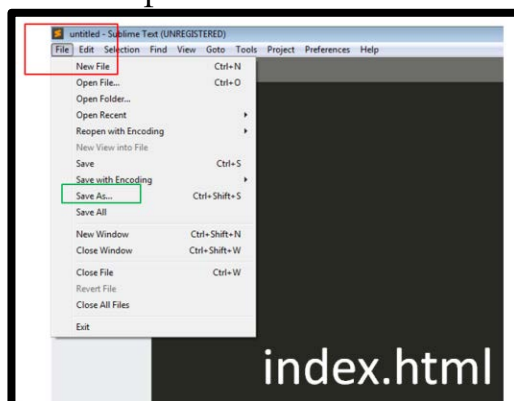


Рисунок 4 - Создание файла index.html

1.2. Ознакомьтесь со структурой документ HTML и основными тегами (рисунок 5).

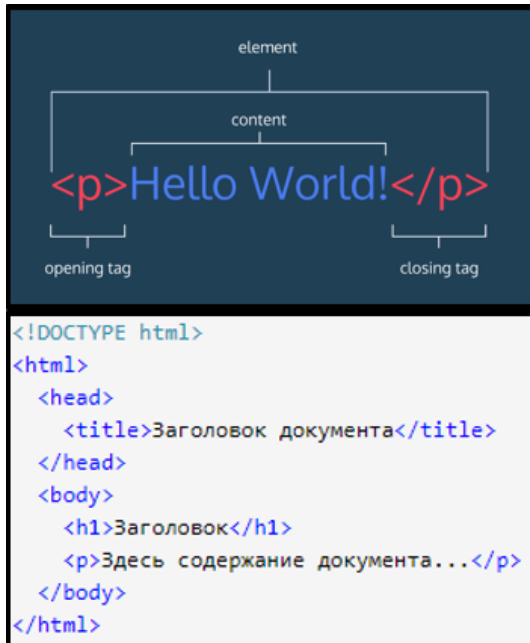


Рисунок 5 – Элемент и структура документа

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Заголовок документа</title>
5   </head>
6   <body>
7     <h1>Заголовок 1</h1>
8     <h2>Заголовок 2</h2>
9     <h3>Заголовок 3</h3>
10    <h4>Заголовок 4</h4>
11    <h5>Заголовок 5</h5>
12    <h6>Заголовок 6</h6>
13
14    <p> Hello World!</p>
15
16  </body>
17 </html>

```

Рисунок 6 – Теги заголовков и параграфа

<!DOCTYPE html> - определяет тип документа и версию HTML.

Тег <html> определяет начало HTML-файла, внутри него хранится заголовок (<head>) и тело документа (<body>).

Заголовок документа, как еще называют блок <head>, может содержать текст и теги, но содержимое этого раздела не показывается напрямую на странице, за исключением <title>(отображается на вкладке).

Тело документа <body> предназначено для размещения тегов и содержательной части веб-страницы, все, что будет отображено на сайте.

<h1></h1>.....<h6></h6>— теги заголовков, от самого большого к самому маленькому (рисунок 6).

<p></p> – параграф (рисунок 6).

 – расставление акцентов в тексте путём выделения его фрагментов полужирным начертанием.

 – выделение текста курсивом.

 – обрыв строки.

<!--...--> – тег для добавления комментариев в документ. Помещённые внутри него теги не интерпретируются браузером.

1.3. Ознакомьтесь с тегами для добавление списков.

Существует два вида списков – упорядоченные (нумерованный – тег) и неупорядоченные (маркированный – тег). Каждый отдельный пункт в списке размечается с помощью элемента (рисунок 7, 8).

```

<h3 id="list">Список продуктов</h3>
<ul>
  <li>Батон</li>
  <li>Масло</li>
  <li>Икра красная или черная</li>
  <li>Петрушка</li>
</ul>

<h3 id="recipe">Пошаговый рецепт</h3>
<ol>
  <li>Нарезать батон</li>
  <li>Намазать масло</li>
  <li>Положить икру</li>
  <li>Украсить петрушкой</li>
</ol>

```

Рисунок 7 – Списки в HTML

Список продуктов

- Батон
- Масло
- Икра красная или черная
- Петрушка

Пошаговый рецепт

1. Нарезать батон
2. Намазать масло
3. Положить икру
4. Украсить петрушкой

Рисунок 8 – Списки в веб

1.4. Ознакомиться с тегами для добавление атрибутов.

Открывающие теги могут содержать атрибуты: менять свойства объекта, которому соответствует тег. Атрибуты записываются внутри открывающего тега через пробел в виде ключевого слова (имя), знака “=” и параметра (значения атрибута). Порядок следования атрибутов в теге не важен. Разделителем атрибутов также является пробел.

1.5. Ознакомиться с тегами для добавление изображений

HTML-изображения добавляются на веб-страницы с помощью тега `` (самозакрывающийся) (рисунок 9), который имеет обязательный атрибут `src`, значением которого является абсолютный или относительный путь к изображению, атрибут `alt`="название картинки".

```



```

Рисунок 9 – Вставка картинки

1.6. Ознакомиться с тегами для добавление ссылок

Атрибут `href` определяет URL – адрес документа, на который следует перейти, а содержимое контейнера `<a>` является ссылкой (рисунок 10). Текст, расположенный между тегами `<a>` и ``, по умолчанию становится синего цвета и подчеркивается, атрибут `target` со значением «`_blank`» – открывает документ в новом окне.

```

<p><a href="https://www.youtube.com/watch?v=eMd3nRezhcA"
target="_blank">Рецепт</a></p>

```

Рисунок 10 – Вставка ссылки

1.7. Ознакомиться с тегами для создания форм

Основу любой формы составляет элемент `<form>...</form>` (рисунок 11).


```
<form>
  <label for="meal">Любимое блюдо:</label>
  <input type="text" name="food" id="meal">
  <input type="submit" value="Submit">
</form>
```

Любимое блюдо:

Рисунок 11 – Создание формы

Создать поле ввода в форме – элемент `<input>`. Элемент с `type="text"` отображает текстовое поле, `name=""` – определяет имя формы. Без атрибута `name` информация не будет отправлена.

Метки - `<label>` с атрибутом `for`. При щелчке по этому элементу соответствующий элемент `<input>` выделяется/выбирается.

Для отправки введенных данных на сервер используем кнопку (`submit`).

Задание 2. Создать текстовый блок в документе по образцу (рисунок 12).

О блоге и обо мне

Я — Катя Арметта, журналист, блогер и радио-ведущая. Веду жизнь *нетипичной сицилийской домохозяйки*. В моем блоге и при личной встрече я расскажу об истории острова, покажу места, знакомые многим по древнегреческим мифам и голливудским фильмам, проведу в церкви и музеи, поделюсь информацией о пляжах, на которых отдыхала и о ресторанах, в которых ела, научу готовить **типичные сицилийские блюда**. С удовольствием дам вам все известные мне **явки и адреса**.

Когда видишь Сицилию с моря, кажется, что это мираж. Мир под оливами и цветущий миндаль — это **Сицилия**. А ещё: золотые долины citrusовых и поля лучшей в мире пшеницы; великолепные виноградники и старинные сыроварни, где до сих пор все делается вручную. *Этна и Стромболи*. Три моря с чудесной россыпью островков; красивые пляжи и природные заповедники с уникальными птицами.

Рисунок 12 – Образец оформления текста

Задание 3. Вставить изображение и оформить списки в документе (рисунок 13).

Рецепт бутерброда

Список продуктов

- Батон
- Масло
- Икра красная или черная
- Петрушка

Пошаговый рецепт

1. Нарезать батон
2. Намазать масло
3. Положить икру
4. Украсить петрушкой

Рецепт бутерброда



Список продуктов

- Батон
- Масло
- Икра красная/ черная
- Петрушка

Рисунок 13 – Образец оформления списков и изображений

Лабораторная работа №3

Элементы программирования на языке HTML: таблицы, стили

Тема 2. Язык HTML. Технология CSS

Цель: рассмотреть возможности применения таблиц, рассмотреть используемые кодировки для кириллицы, назначение каскадной таблицы стилей (CSS).

Последовательность лабораторной работы представлена на рисунке 14.

The figure consists of 12 numbered slides, each illustrating a step in the lab work:

- Slide 1: ТАБЛИЦА** - Definition of a table as a structured set of data. Includes a sample table with columns 'Имя', 'Возраст', 'Пол', 'Рост' and rows of names and ages.
- Slide 2: ШАГ 1.** - Initial HTML code: `<table border="2"></table>`. Step 2: Adding cells: `<table border="2"><tr><td>1 ячейка</td><td>2 ячейка</td><td>3 ячейка</td></tr><tr><td>4 ячейка</td><td>5 ячейка</td><td>6 ячейка</td></tr></table>`.
- Slide 3: ШАГ 3.** - Adding rows. Code: `<table border="2"><tr><td>1 ячейка</td><td>2 ячейка</td><td>3 ячейка</td></tr><tr><td>4 ячейка</td><td>5 ячейка</td><td>6 ячейка</td></tr></table>`. Notes: `<tr>` forms a row, `<td>` forms a cell.
- Slide 4: ШАГ 4. ОБЪЕДИНЕНИЕ ЯЧЕЕК** - Merging cells. Example: a table with 2 columns and 2 rows, where the first row has 2 cells and the second row has 1 cell with colspan=2.
- Slide 5: Рассмотрите пример** - Another example of merging cells. Code: `<table border="2"><tr><td colspan="2">1 ячейка + 2 ячейка</td><td colspan="3">3 ячейка + 6 ячейка</td></tr><tr><td colspan="4">4 ячейка</td><td colspan="2">5 ячейка</td></tr></table>`.
- Slide 6: Задание 1** - Task to create a table in 'index' with specific data. Includes a table with columns 'Месяц', 'Январь', 'Февраль', 'Март' and rows 'Машина', 'Квартира', 'Получен'. Also includes a table with columns 'Возраст', '1979', '1989', '1979', '1989' and rows 'Александр Сидоров', 'Анна', 'Евгений', 'Ольга', 'Светлана Степановна', 'Антон', 'Анна'.
- Slide 7: Кодировка** - HTML meta tags for encoding: `<!DOCTYPE html><html><head><meta charset="utf-8"><title>Task 4</title></head></html>`. Task: add meta tags for Cyrillic.
- Slide 8: CSS** - Steps: 1. Open folder in Sublime text. 2. Open 'index.html'. 3. Create 'style.css'. 4. Save changes. 5. Open 'index.html' in browser. Note: CSS allows linking styles to documents.
- Slide 9: CSS Link** - Code: `<link href="./style.css" type="text/css" rel="stylesheet">`. Note: link attribute points to CSS file.
- Slide 10: 1. Селектор = Тег** - Example: HTML `<p>The world is full of fascinating places. Planning the perfect vacation involves packing up, leaving home, and experiencing something new.</p>` and CSS `p { }`. Note: CSS selector corresponds to the HTML tag.
- Slide 11: CSS Property** - Example: `p { color: gray; }`. Note: Color can be assigned with a name or a numeric value.
- Slide 12: Задание 2** - Task to add text to a page using different tags. Example CSS: `font-family: Georgia; шрифт; font-size: 18px; размер; font-weight: bold; начертание; text-align: right; выравнивание`.

Рисунок 14 – Этапы выполнения лабораторной работы

Жирный шрифт, курсив, малые прописные буквы

Жирность шрифта - атрибут CSS font-weight

- normal - обычный
- bold - жирный
- bolder - более жирный
- 100, 200, 300, 400, 500, 600, 700, 800, 900 - значения шрифта, 100 - самый тонкий шрифт

Атрибут text-decoration (украшение текста):

- underline - подчеркивание (Е)
- overline - надчеркивание (Е)
- line-through - зачеркивание (Е)
- none - без изменений
- underline wavy - волнистая линия
- underline wavy red - волнистая линия красного цвета

Атрибут text-align (выравнивание шрифта):

- left - выравнивание по левому краю
- center - выравнивание по центру
- right - выравнивание по правому краю
- justify - выравнивание по ширине

Атрибут text-indent (выравнивание отступа):

- normal - обычный отступ
- length - отступ в заданном количестве
- percentage - отступ в заданном проценте

КЛАССЫ

```
html
<h1 class="green"> ... </h1>

css
.green {
  color: green;
}
```

- Элементы HTML могут иметь атрибуты class (классы).
- В CSS классы выбираются через точку.
- Классы применяются при одинаковом оформлении нескольких элементов под разными тегами.
- Или наоборот, если нужно одинаковых тегов, и нужно изменить только несколько их элементов. В таком случае CSS применит оформление ко всем элементам с заданным классом.

Назначение несколько классов

```
.green {
  color: green;
}
.bold {
  font-weight: bold;
}

<h1 class="green bold"> ... </h1>
```

- В CSS можно применить несколько классов к одному элементу.
- Для этого имена классов в HTML нужно разделять пробелом.
- Поскольку добавлены несколько классов они могут содержать одинаковые стили, но с разными значениями, то будет значение у класса, который описан в CSS коде ниже.

13

14

15

ID Name

```
html
<h1 id="large-title"> ... </h1>

css
#large-title {
}
```

Идентификатор элемента - id атрибут - уникальная стилизация. Значение id всегда должно быть одно. В CSS идентификатор обозначается через #.

Специфичность

```
h1 { color: red; }
h2 { color: blue; }
h3 { color: green; }
h4 { color: purple; }
```

- Специфичность** - это способ, с помощью которого браузеры определяют, какое значение свойства CSS наиболее соответствует элементу на следующей странице. Будут применены. Специфичность основана на порядке следствия, состоящих из селекторов и селекторов CSS различных типов.
- Специфичность** представляет собой вес, присваиваемый конкретному правилу CSS. Это правило определяется количеством каждого из типов селекторов в данном правиле. Если у нескольких правил специфичности одинаковы, то к элементу и применяется последнее по порядку правило CSS. Специфичность имеет значение только в том случае, если один элемент соответствует нескольким правилам. Согласно спецификации CSS, правило для множественно совпадающих селекторов и селекторов будет иметь большую приоритет, чем правило, соответствующее одному.

Несколько селекторов

```
h1 { color: #FF0000; }
h2 { color: #FF0000; }
```

```
h1, h2 { color: #FF0000; }
```

- В CSS можно группировать селекторы, если их атрибуты одинаковы.
- В этом случае два разных селектора нужно разделить запятой. Тогда эти значения будут применяться и к абзацу, и к заголовку 1 уровня.

16

17

18

Цепочка селекторов

```
div > p.class="red-text">Красный параграф</p>

p.red-text {
  color: #FF0000;
}
```

- К одному элементу можно применить несколько классов и идентификаторов.
- Пример: данный стиль будет применен ТОЛЬКО элементу с тегом <p> и классом red-text.

Задание 3

- Используя полученные знания, оформите ваш текст.
- Используйте все варианты: селекторы, классы и id.

Пример файла HTML

```
<h1 id="main">Пример файла HTML</h1>
<h2 class="sub">Пример файла CSS</h2>
<p>Пример файла CSS</p>
```

Пример файла CSS

```
h1 { color: red; }
h2 { color: blue; }
p { color: green; }
```

19

20

21

Рисунок 14 (продолжение) – Этапы выполнения лабораторной работы

Лабораторная работа №4

Блочная система. Позиционирование

Тема 3. Верстка сайта

Цель: рассмотреть особенности использования блочной верстки, параметры блока и его настройки, порядок отображения элементов на странице (позиционирование).

Последовательность лабораторной работы представлена на рисунке 15.

1. Блочная модель

- Первое, что нужно понять для эффективного использования CSS, — это, что вы видите на странице, состоит из прямоугольников. Каждый сайт представляет собой нечто более, чем набор прямоугольных блоков, расположенных рядом, сверху, слева и внутри друг друга.
 - Задвижки - Прямоугольники.
 - Блочная панель - Прямоугольники.
 - Изображения - Прямоугольники.
- Прямоугольники = HTML-элементам, которые бывают разных типов.
- Чаще всего в HTML-элементах — это обозначается как «блок».
- Если два типа HTML-элементов: строчные и блочные. Между ними есть разительное отличие, которое также влияет на использование блочной модели CSS.

2. Блочная модель

3. Блочная модель - слои

Шрифты — ширина блочного содержимого элемента, для блочных элементов значение по умолчанию равно 100%, у строчных элементов значением по умолчанию. Высота — определяет высоту элемента, как правило, она зависит от содержания содержимого, но при желании можно задать конкретную высоту. Цвет, не работает только с блочными элементами. Границы — границы есть у каждого элемента, даже если вы не видите, у них может быть разный размер, цвет и оформление. Отступы — они определяют расстояние между границей элемента и его содержанием, но можно использовать, например, для того, чтобы текст внутри элемента оставался читаемым. Поля — они определяют расстояние между границей элемента и тем, что его окружает. Так же у каждого элемента есть свойство «z-index», CSS-свойство, которое определяет основные способы управления размером, положением и внешним видом HTML-элемента.

4. Использование блочной модели – упражнение 1

1. Создать новый файл в своей папке «html»
2. Сохранить под именем «block1.html»

5.

1. Прописать название страницы, установить язык и т.д.
2. Создать структуру документа
3. Присоединить файл css к новой странице

6.

1. Ввести 3 блочных элемента «div»

Задать параметры (одинаковые) для всех «div»:

1. Ширина,
2. Границы (цвет, размер, тип)
3. Цвет

7. Сброс настроек по умолчанию

```

* {
margin: 0;
padding: 0;
}
    
```

- Обнулить отступы и поля у всех элементов на странице можно с помощью универсального селектора «*».

8. Границы (border)

1. Свойства border: 3 значения: width, style и color.
 - width — толщина границы (обычно в пикселях),
 - style может принимать значения solid, dashed, dotted и т.д.,
 - color — прописывать цвет словом «red», так и безымянным цветом.

9.

1. Последовательно добавляйте новые параметры для блоков.
2. Просмотрите изменения в браузере.

10. Разные значения для разных сторон

Для разных сторон можно установить свои значения с помощью «двухсторонних» свойств, например:

- 1) Для разных сторон можно установить равные значения границ с помощью свойств:
 - 1) border-top,
 - 2) border-right,
 - 3) border-bottom,
 - 4) border-left.
- 2) Также для разных сторон равными отступы:
 - 1) padding-top,
 - 2) padding-right,
 - 3) padding-bottom,
 - 4) padding-left.
- 3) Для разных сторон разные поля:
 - margin-top,
 - margin-right,
 - margin-bottom,
 - margin-left.

11. Разные значения для разных сторон

1. Разные значения могут записываться в 1 строку по примеру.
 - * Значение выставляется от открытого по часовой стрелки (верх-право-ниж-лево).
2. При одинаковых значениях параллельных параметров: сверху-слева = 10; слева-справа = 200; снизу-справа = 200; снизу-слева = 10px

12. Позиционирование

Порядок отображения элементов на странице называется потоком документа. Блочные элементы отображаются как прямоугольные области, идущие друг за другом сверху вниз, а строчные элементы располагаются сверху вниз и слева направо и при необходимости переносятся на новую строку. Элементы можно вкладывать друг в друга. Чем раньше в коде расположен элемент, тем выше он расположен на странице. Некоторые типы элементов могут вести себя в потоке блочного слона. Например, они могут располагаться в несколько колонок, как блочно строчные элементы. Любой элемент, который находится в потоке, занимает на странице свою собственную площадь и, если надо, отталкивает от себя соседя с помощью полей.

Рисунок 15 – Этапы выполнения лабораторной работы

Позиционирование

Некоторые свойства, которые изменяют то, как браузер позиционирует элементы, указывают, где находится элемент на странице, могут ли элементы накладываться друг на друга:

- position
- display
- z-index
- float
- clear

Каждое из этих свойств позволит размещать и просматривать элементы на веб-странице. Они могут использоваться в сочетании с любыми другими стиливыми свойствами.

2-й способ

1. Измените значения.
2. Посмотрите результат.

Упражнение 2 – вложенные блоки

1. Измените содержание в файле .html
2. Добавьте стили
3. Посмотрите результат
4. Добавьте значение margin
5. Посмотрите результат

Свойство display: flex

CSS правило предназначено для выравнивания flexbox сетки, которая строится из дочерних HTML-тегов, расположенных внутри некоторого родительского контейнера.

Для задания данной сетки необходимо установить правило display в значение:

display: flex; — для блочного отображения;

display: inline flex; — для inline-блочного отображения.

1. Внесите изменения по образцу.
2. Посмотрите результат.
3. Внесите изменения по образцу.
4. Посмотрите результат.

1. По очереди вводите разные значения (1, 2, 3).
2. Посмотрите результат.

*Тема (стрелка, по умолчанию) *column (распределит в колонку, но не выводит за границы родителя)

Свойство flex-direction устанавливает направление главной и поперечной осей или, говоря другими словами, располагает элементы ряд или в колонку. Применяется к родительскому элементу для flex-блока.

Свойство justify-content

"flex-end" - расположение справа

"flex-start" - расположение слева (по умолчанию)

- Свойство justify-content выравнивает элементы, когда они не используют все доступное пространство по горизонтали. Оно работает подобно всем остальным свойствам контейнера. Свойство контролирует выравнивание элементов, когда они выйдут за границы.
- Свойство justify-content должно быть использовано вместе со свойством display, которое имеет значение "flex".

Свойство justify-content

1. background-color: yellow; display: flex; justify-content: center;
2. background-color: yellow; display: flex; justify-content: space-around;
3. background-color: yellow; display: flex; justify-content: space-between;

1. По очереди вводите разные значения
2. Посмотрите результат

Свойство: align-items

"flex-end" - расположение по нижней границе

"flex-start" - расположение по верхней границе (по умолчанию)

Свойство align-items выравнивает flex-элементы внутри контейнера в перпендикулярном направлении.

Свойство: align-items

1. align-items: center;
2. align-items: space-between;
3. align-items: stretch;

1. По очереди вводите разные значения
2. Посмотрите результат

Для значения stretch, обязательно убрать высоту дочерних блоков!!

Практическое задание 1

- Подобрать правильные значения.
- Сохранить результат (файлы «block1.html» и «styleblock.css»)

Примеры использования блочной модели – Упражнение 2

1. Создать новый файл в своей папке «html»
2. Сохранить под именем «block2.html»

1. Создать новый CSS-файл в папке «css»
2. Имя файла «styleblock2.css»

- Выполнить по образцу

Позиционирование.

- position
- display
- z-index
- float
- clear

- CSS свойство position указывает тип позиционирования элемента на веб-странице.
- Позиционирование определяет относительно чего, при смещении, будет позиционироваться элемент: относительно окна браузера, других элементов или текущего местоположения на веб-странице.

Каждый элемент уровня блока занимает свое собственное линейное пространство и может перекрывать друг друга. Это зависит от значения для блочных элементов.

Данное свойство может принимать одно из четырех значений:

- static
- relative
- absolute
- fixed

position: static; - по умолчанию!

Рисунок 15 (продолжение) – Этапы выполнения лабораторной работы

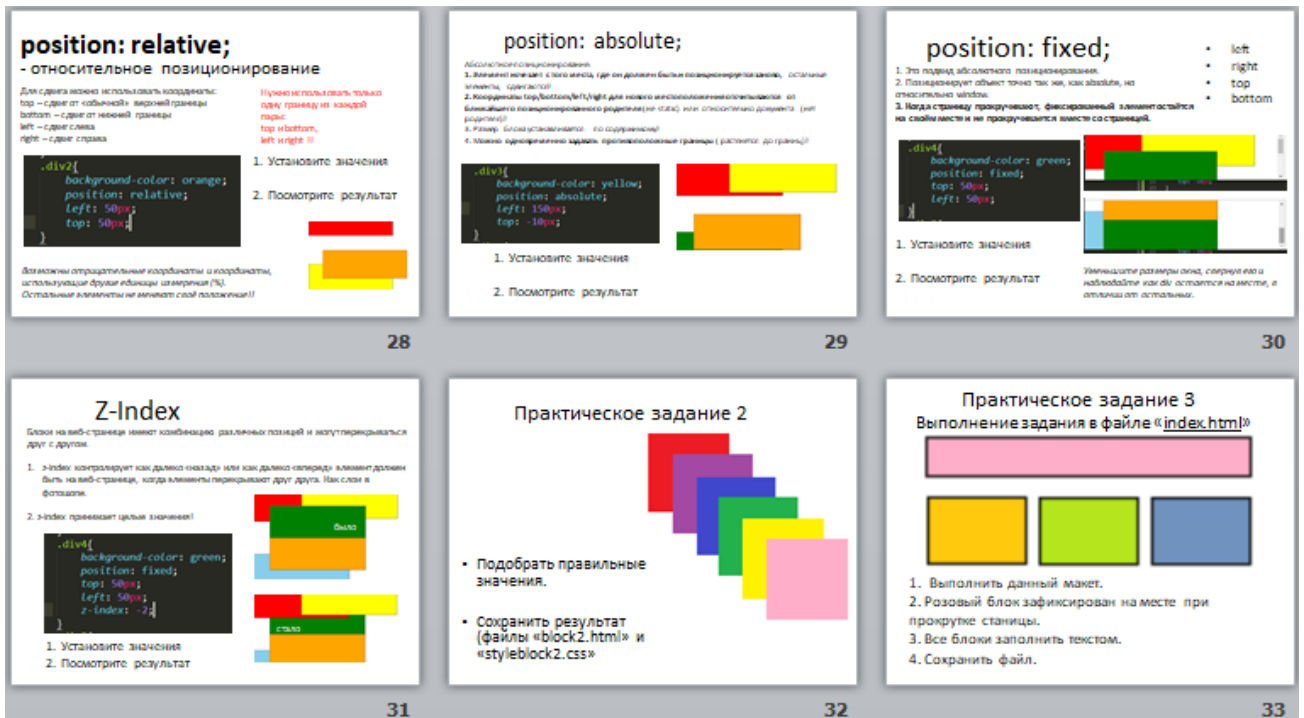


Рисунок 15 (продолжение) – Этапы выполнения лабораторной работы

Лабораторная работа №5 Адаптивный шаблон веб-ресурса Тема 3. Верстка сайта

Цель: рассмотреть основные методы достижения адаптивности сайта, свойства модуля flexbox и особенности верстки на его основе.

Последовательность лабораторной работы представлена на рисунке 16.

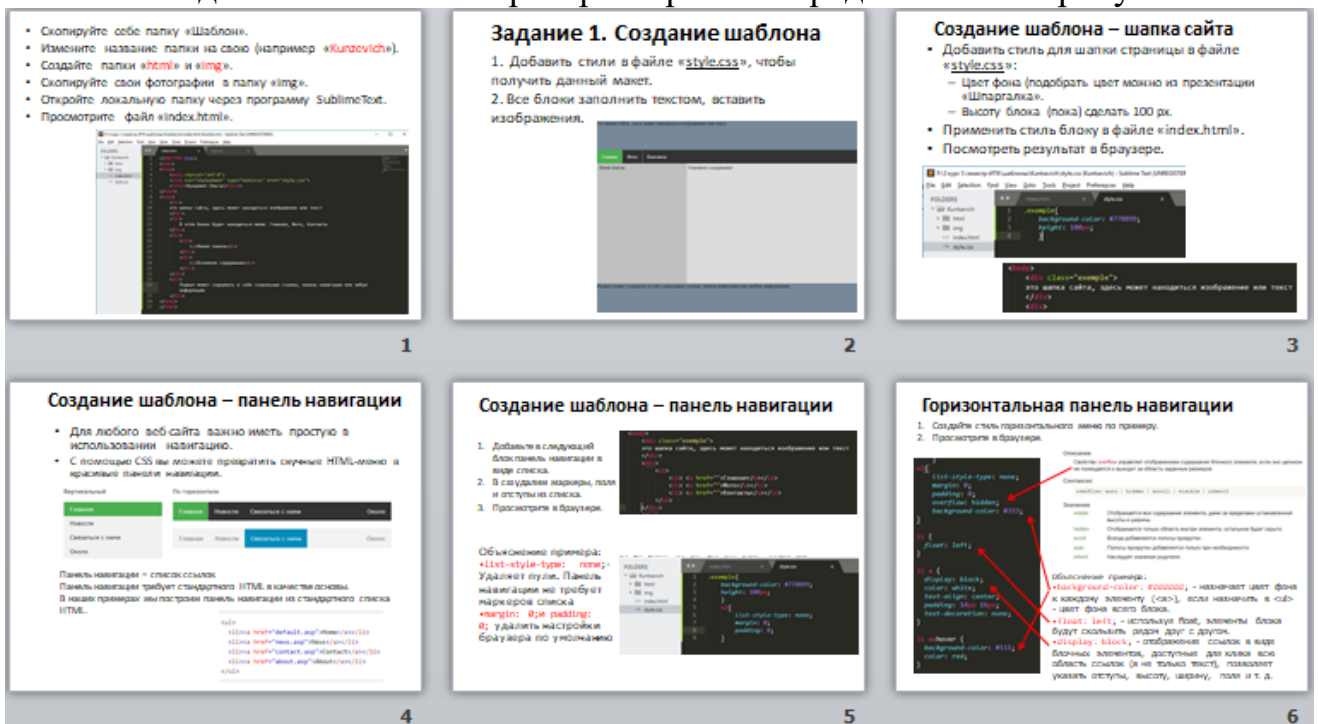


Рисунок 16 – Этапы выполнения лабораторной работы

Горизонтальная панель навигации

Активная / текущая ссылка

- Добавьте к текущей ссылке «активный класс», чтобы пользователь знал, на какой странице находится.
- Добавьте `border-right` свойство в `li`, чтобы создать разделители ссылок.
- Дополните активный класс (`li.active`) - чтобы разделить его было.
- Посмотрите результат в браузере.

Горизонтальная панель навигации

Выпадающее меню (раскрывающийся список)

- Создайте раскрывающийся список, который появляется, когда пользователь наводит указатель мыши на элемент.
- Каждый элемент из этого списка будет отдельной страницей.

CSS

- Дополните файл `style.css` по примеру.

Было

Стало

HTML

- Измените файл `index.html` по примеру.

Было

Стало

В примере мы использовали элемент контейнера (блок `div`), чтобы создать раскрывающийся содержимое и добавить в него новый список (выпадающее меню, может быть представлен только с использованием тега `ul`).

HTML

- Измените файл `index.html` по примеру.

Было

Стало

Самостоятельно рассмотрите способ создания вертикальной панели навигации.

- Добавьте на своей странице такой вариант меню (по желанию).

Вертикальная панель навигации

- Создайте новый список для вертикальной панели навигации.
- Примените стили к элементам `li` внутри списка по предыдущему примеру.

пример

```
li a {
  display: block;
  width: 60px;
}
```

Вы также можете установить ширину `li` и удалить ширину `ul`, так как они будут занимать всю ширину, доступную при отображении как элементам блока.

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 60px;
}
```

Результат будет тот же, что и в предыдущем примере:

```
li a {
  display: block;
}
```

Создайте базовую панель навигации с серым цветом фона и измените цвет фона ссылки, когда пользователь наводит на них указатель мыши:

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  background-color: #cccccc;
}
li a {
  display: block;
  color: #000;
  padding: 5px 10px;
  text-decoration: none;
}
li a:hover {
  background-color: #fff;
  color: #000;
}
```

Центрировать ссылки и добавить границы

Добавьте `text-align:center` в `li` или `ul`, чтобы центровать ссылки. Добавьте `border` свойство к `li`, чтобы добавить рамку вокруг панели навигации. Если вам нужны границы вокруг навигационной панели, добавьте в `li` `border-bottom` ко всем элементам `li`, кроме последнего:

```
ul {
  border: 1px solid #555;
}
li {
  text-align: center;
  border-bottom: 1px solid #555;
}
li:last-child {
  border-bottom: none;
}
```

Фиксированная вертикальная навигационная панель в полную высоту

Создайте полноэкранную «левую» базовую навигацию

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 20%;
  background-color: #f0f0f0;
  height: 100%; /* Full height */
  position: fixed; /* Fix it in place, even in scroll */
  overflow: auto; /* Enable scrolling if the sidebar has too much content */
}
```

Создание шаблона: левая панель и основное содержание

- Создадим макет с 2 столбцами с разной шириной.
- Содержание - это самая большая и важная часть сайта.
- Дополнительный контент можно использовать в качестве альтернативной навигации.

CSS

- Повторяем стили на примерах.
- Добавим соответствующие классы и элементы.
- Посмотрим в браузере.

Классу `row` надать свойство `flex` - гибкая адаптивная структура элементов.

Классу `col` включить себе общие настройки для 2х колонок:

- `flex-grow: 1` - общее увеличение;
- `flex-shrink: 1` - при уменьшении экрана, и последующем его увеличении;
- `padding` - отступы/поля внутри элементов;
- Цвет фона и ширина для каждого столбца.

Media-запрос - адаптируем макет для маленьких экранов.

Адаптивный веб-дизайн – The Viewport (область просмотра)

После того как добавили мета-запрос в файл `style.css`, необходимо прописать мета данные в файле `html` в части `head`.

```
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

- Область просмотра - это видимая пользователем область веб-страницы.
- Область просмотра зависит от устройства и на небольшом экране будет меньше, чем на экране компьютера.
- В начале веб-страницы были разработаны только для экранов компьютеров - статический дизайн и фиксированный размер.
- Веб-страницы фиксированного размера плохо открываются при работе с Интернетом с помощью планшета и мобильного телефона.
- Чтобы исправить это, необходимо уменьшить всю веб-страницу до размера экрана этих устройств.

Дополнительные правила

- Используйте больше элементов фиксированной ширины - например, если изображение отображается с шириной, большей, чем область просмотра, это может привести к горизонтальной прокрутке просмотра. Вы можете настроить это содержание так, чтобы оно соответствовало ширине области просмотра.
- Используйте `max-width` для ограничения ширины области просмотра для правильной визуализации - поскольку размеры экрана и ширина в CSS сильно различаются между устройствами.
- Используйте `min-width` для ограничения ширины элементов для области просмотра на меньших устройствах. Есть возможность использование значений относительной ширины, например `width: 100%`. Будьте осторожны при использовании больших абсолютных значений позиционирования. Это может привести к тому, что элемент выйдет за пределы области просмотра на небольших устройствах.

Настройка области просмотра

- Включите следующий `<meta>` элемент области просмотра на все веб-страницы в разделе `head`.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link href="style.css" rel="stylesheet" type="text/css" />
<link href="main.html" rel="stylesheet" type="text/css" />
```

`width=device-width` - задает ширину страницы, чтобы соответствовать ширине устройства (который будет вычисляться в зависимости от устройства).

`initial-scale=1.0` - устанавливает начальный уровень масштабирования при первой загрузке страницы в браузере.

Пример веб-страницы

- без мета-тега области просмотра
- с мета-тегом области просмотра

1

2

Без мета-тега области просмотра

С мета-тегом области просмотра

Создание шаблона - Подвал

- Добавьте последнему блоку стиль `clear: both` (то же что и шпала сайта).
- По желанию измените самостоятельно его высоту или фоновый цвет:
 - в самом теге или через создание отдельного стиля.
- Посмотрите изменение в браузере.

```
<div class="clear" style="clear: both; height: 20px; background-color: #cccccc; width: 100%; text-align: center; padding: 5px 0 5px 0;">
```

Общие настройки шаблона

- Примените данные параметры.
- Заполните страницу информацией.
- Займитесь стилизацией отображения текста.
- Посмотрите в браузере.

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
```

Установить общую/автономную отступы для всех элементов.

Установить общую ширину для всех элементов.

Рисунок 16 (продолжение) – Этапы выполнения лабораторной работы

Лабораторная работа №6

Создание веб-ресурса с помощью CMS

Тема 5. Системы управления содержимым сайта (CMS)

Цель занятия: ознакомиться с этапами проектирования, создать собственный веб-ресурс с помощью CMS WordPress (классический редактор).

Задание 1. Вход в административную часть или консоль своего веб-сайта (рисунок 17).

Посетите страницу входа на свой сайт, написав wp-admin после имени домена (например, <http://example.com/wp-admin>). Теперь войдите в WordPress, используя имя пользователя и пароль (рисунок 18).

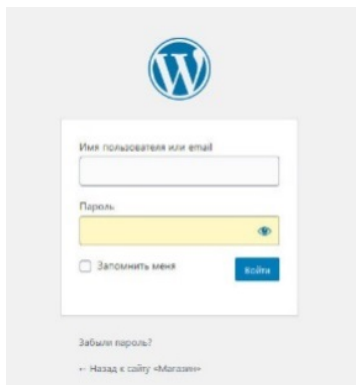


Рисунок 17 – Вход в консоль CMS WordPress

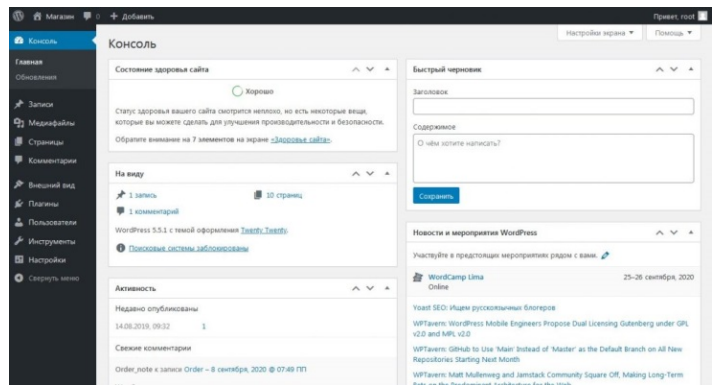


Рисунок 18 –Интерфейс консоли CMS WordPress

После входа в систему вы попадете на главный экран консоли, который называется «Панель управления». Здесь начинается настройка вашего сайта.

Вверху экрана находится область, называемая панелью инструментов. Нажмите на название вашего сайта – это ссылка на главную страницу вашего нового сайта WordPress.

Задание 2. Работа в консоли администрирования WordPress

Первая страница, которая отображается после входа в систему, называется Панель инструментов, содержит список последних действий, которые выполняли на сайте, и помогает находиться в курсе новой и интересной информации из многих ресурсов WordPress.

В левой части экрана находится главное меню навигации, в котором подробно описаны все административные функции, которые можно выполнять. Пункты меню следующие: Консоль, Записи, Медиафайлы, Страницы, Комментарии, Внешний вид, Плагины, Пользователи, Инструменты, Настройки.

Задание 3. Заполнить информацию в пункте «Ваш профиль».

Задание 4. Ознакомиться с вкладкой «Внешний вид».

Задание 5. Создать записи.

Заполнить поле заголовка, написать небольшую тестовую запись в текстовом поле. По завершению, нажать кнопку «Опубликовать», которая находится справа от окна поля публикации. Просмотреть результат можно кликнув по названию сайта на панели инструментов в верхней части экрана.

Задание 6. Комментарии

Данная функция позволяет создать обратную связь между администратором сайта и посетителями. Комментарии к записям бывают самых разных форм: простые (Хорошая работа! Нравится пост!), обширные обсуждения, комментарии, которые дополняют опубликованную информацию. Ответ на комментарии и их модерирование может занять много времени.

Задание 7. Создание рубрик

Сообщения обычно хранятся в рубриках и/или метках, поэтому связанные записи можно хранить вместе. «Добавить новую рубрику» можно через вкладку «Записи – Рубрики»: указать название рубрики, при необходимости указать родительскую рубрику (создать подрубрику), описание.

Сортировать рубрики в WordPress можно двумя способами: по имени (в алфавитном порядке) или по идентификационному номеру, который присваивается при создании. Идентификационный номер необходимо указывать сразу при создании рубрики.

Задание 8. Настройка сайта

Поместить записи в рубрики. Активировать вкладку «Все записи». Навести указатель мыши на заголовок любой записи, под заголовком можно увидеть ссылки Изменить | Свойства | Удалить | Перейти. Нажмите «Изменить», чтобы отредактировать одну из публикаций. В правой части экрана редактирования записи есть строка для выбора рубрики. Нажмите кнопку «Обновить». Повторить эти шаги для всех записей.

Задание 9. Использование плагинов WordPress

Плагины WordPress также известны как дополнения или расширения. Это программные скрипты, которые добавляют на сайт функции и события. Они охватывают весь спектр: от актуальных прогнозов погоды до простой организации записей и рубрик. Установить плагины можно согласно инструкциям, предоставленным автором плагина.

Задание 10. Спроектировать и создать собственный веб-ресурс.

Лабораторная работа №7

Создание веб ресурса с помощью онлайн конструктора

Тема 6. Хостинг и публикация веб-сайта. Онлайн-конструкторы

Цель: рассмотреть особенности создания веб-сайтов с помощью онлайн-конструкторов сайтов.

Конструктор сайтов — это система, позволяющая создавать сайты и управлять ими без каких-либо дополнительных знаний и навыков. Конструктор для сайтов шаг за шагом предложит вам выбрать дизайн для сайта, цвет, стиль и картинки. Нет надобности покупать новое программное обеспечение. Не нужно изучать языки программирования. Конструктор сайтов может строить разнообразные страницы. Скажем, вы хотите создать сайт для вашей фирмы, онлайнное резюме или личную фото галерею. Все делается в простом пошаговом режиме. Сочетание хорошего дизайна и удобной системы создания сайтов приводит к успеху.

Задание 1. Создать ресурс на бесплатном конструкторе tilde.

Последовательность лабораторной работы представлена на рисунке 19.

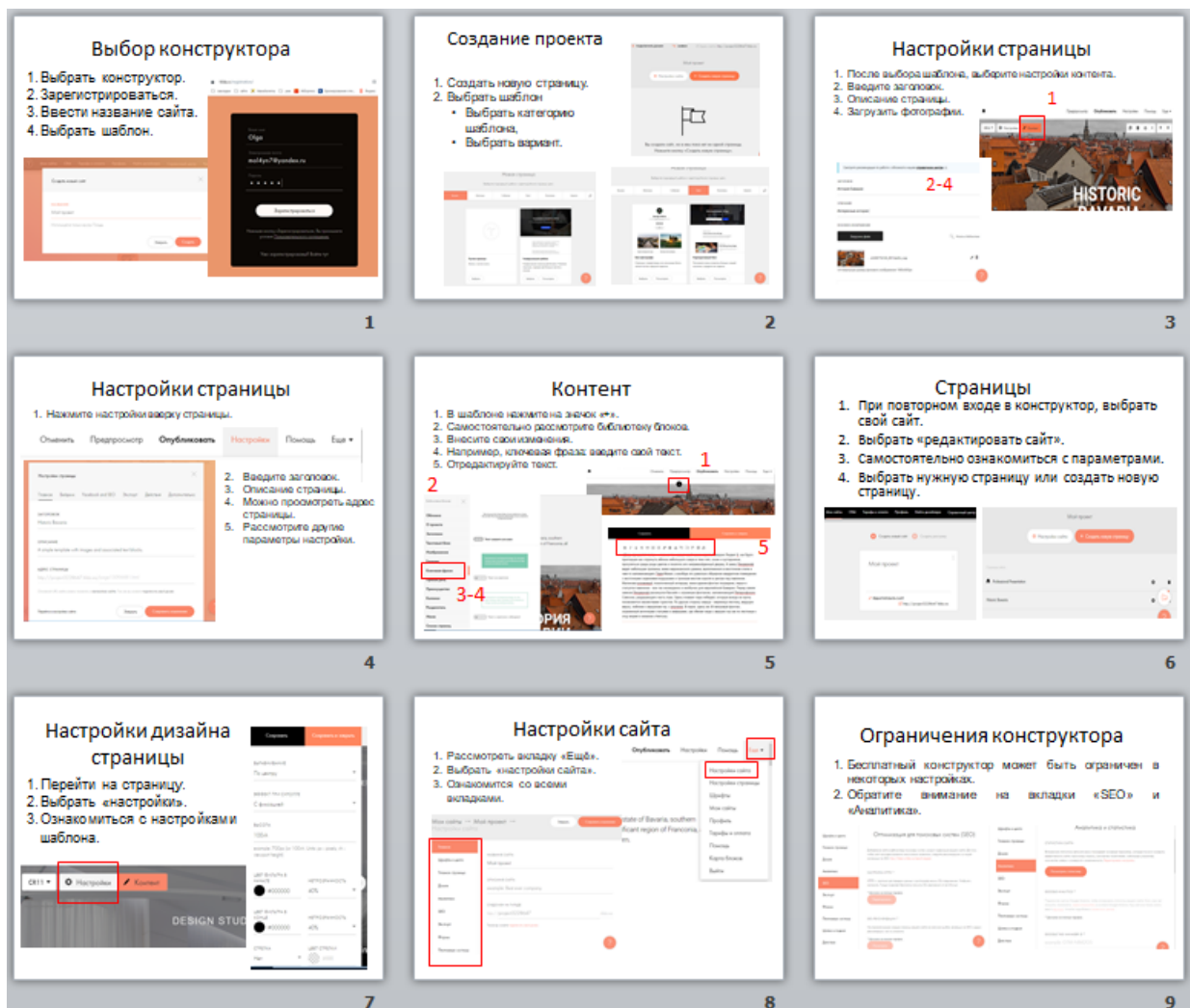


Рисунок 19 – Этапы выполнения лабораторной работы

Задание 2. Самостоятельно ознакомиться с другими возможностями настройки сайта.

Задание 3. Наполнить сайт своим контентом.

Результат показать преподавателю.

Лабораторная работа №8

Создание персонального веб-сайта

Тема 8. Разработка веб-сайта по заданной тематике

Цель: применить полученные знания и навыки по созданию и верстке веб-страниц, тестированию и оптимизации сайта.

Задание 1. Создать персональный сайт

1.1. Определить цель, задачи сайта, целевую аудиторию сайта.

1.2. Разработать структуру страниц, выбрать макет и тип верстки.

1.3. Подготовить контент и графические изображения.

1.4. Выполнить верстку страниц сайта.

1.5. Провести тестирование сайта.

Задание 2. Подготовить презентацию готового проекта.

3.2. Описание практических занятий

Практическая 1. Модели верстки сайтов

Тема 3. Верстка сайта

Цель: закрепление базовых понятий дисциплины, формирование информационной компетенции: обладать методологией использования социальных сетей как коммуникативного пространства, уметь использовать информационные технологии для разработки и создания рекламы учреждения социокультурной сферы.

Задание. Подготовить макеты персональных сайтов по разным типам верстки, используя графические редакторы.

Практическая 2. Сравнительная характеристика CMS

Тема 5. Системы управления содержимым сайта (CMS)

Цель: ознакомиться с основными функциями CMS, специализацией, видами и особенностями применения CMS при создании и ведении различных ресурсов.

Задание. Создать и заполнить таблицу по существующим *системам управления содержимым сайта (CMS)* и их возможностям.

Практическая 3. Методы поискового продвижения

Тема 7. Основы поискового продвижения

Цель: научиться проводить поисковую оптимизацию интернет-ресурса сферы культуры.

Краткие теоретические сведения.

Оптимизация – это ряд мероприятий по изменению сайта, направленных на улучшение позиций ресурса в поисковиках по определенным запросам пользователей. К ним относятся работы над программным кодом (html) и информационным наполнением страниц.

Комплексный анализ сайта, который поможет узнать основные характеристики интересующего вас сайта. Узнать такие параметры, как ТИЦ и PR и многое другое.

Теги.

Title. Содержимое этого тега оказывает ключевое влияние на релевантность страницы по указанным в нем поисковым запросам. Для эффективного поискового продвижения сайта в title задаются уникальные заголовки для каждой страницы.

Заголовки (H, h1, h2 и т.д.). В заголовок первого уровня вносятся ключевые слова, соответствующие содержанию страницы, в следующие

уровни – разделов и абзацев. От логичности форматирования текста также зависит удобство ресурса для пользователя (юзабилити).

Keywords. Содержит перечень ключевых слов.

Description. Служит для описания страницы, которое может показываться как сниппет в выдаче некоторых поисковых систем и помогает пользователям принять решение о переходе по ссылке.

Alt. Содержит описание к картинке. Поисковые роботы не индексируют изображение без заполненного. Если в него добавить ключевые слова, релевантность страницы увеличится, а пользователи смогут найти сайт при просмотре картинок по своему запросу.

Контент. Тексты для успешной раскрутки сайта обязательно должны быть уникальными, т.е. не встречаться на других страницах интернета. Этот факт отслеживается нерегулярно, чтобы избежать попадания под фильтры поисковых систем. Для наилучшей релевантности плотность ключевых слов на странице должна варьироваться в пределах 5%. Для низкочастотных запросов важна общая полезность текста. Чем полнее раскрыта тема по продвигаемым ключам, тем выше в итоге окажется сайт в выдаче поисковых систем.

Внутренняя перелинковка. С помощью ссылок на внутренние разделы можно управлять PR отдельных страниц и повышать вес низкочастотных запросов. Например, линк на новую страницу каталога из раздела с высоким PR, повысит ее видимость и ускорит индексацию поисковыми системами. В тексте ссылки указывается продвигаемый низкочастотный запрос.

Задание. Поисковая оптимизация интернет-ресурса сферы культуры

1. Выбрать ресурс сферы культуры
2. Проанализировать его с помощью <http://pr-cy.ru/>, <http://www.alexa.com/> или [Cy-pr.com](http://www.cy-pr.com)
3. Подобрать ключевые слова к сайту.
3. Зарегистрировать его в каталоге – любом.
4. Проверьте, проиндексирован ли ваш ресурс:
<http://webmaster.yandex.ru/addurl.xml>
<http://www.google.ru/intl/ru/addurl.html>
5. На любом блоге или в твиттере создайте ссылку на интернет-ресурс.
6. Выполнение работы оформить в документе Word и PowerPoint и предоставить для проверки преподавателю.

4. РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ

4.1 Задания для управляемой самостоятельной работы

Самостоятельная работа студентов направлена на обогащение их умений и навыков по учебной дисциплине «Основы фронтенд разработки» в свободное от обязательных учебных занятий время. Целью самостоятельной работы студентов является усвоение в полном объеме содержания учебной дисциплины через систематизацию, планирование и контроль собственной деятельности. Преподаватель дает задания по самостоятельной работе и регулярно проверяет их исполнение.

Дисциплина изучается в течение одного семестра и предусматривает самостоятельную работу студентов. К зачету студент должен представить созданный им сайт-визитку, разработанный брендбук. На семинарских занятиях предусмотрено рассмотрение теоретических вопросов по истории развития графического дизайна, дизайна интернет-ресурсов и пользовательских интерфейсов.

Перечень самостоятельных заданий

1. Создать статический персональный сайт на языке HTML с использованием CSS, содержащий не менее 3 страниц и графические изображения.
2. Создать персональный сайт с помощью систем управления содержимым (CMS).
3. Создать авторский тематический сайт на любом онлайн-конструкторе сайтов.

Примерный список конструкторов сайтов: <https://wix.com/>, <https://wordpress.com/>, <https://www.blogger.com/>, <https://sites.google.com/>, <http://www.ucoz.ru/> и др.

4.2 Перечень вопросов для проведения зачета

1. Пространство www. История. Персоналии.
2. Принципы и перспективы развития всемирной паутины – веб-пространства.
3. Способы активного отображения информации во Всемирной паутине.
4. Дизайн веб-сайтов. Понятие веб-интерфейса. Дизайн интерфейса. Принципы успешной навигации.
5. Классификация сайтов.

6. Критерии качества веб-ресурсов:
7. Кодировки кириллицы в веб-пространстве.
8. Требования к названиям файлов и каталогов в веб-пространстве. Наименование первой страницы сайта.
9. Логическая и физическая структура сайта.
10. Векторные форматы веб-графики swf, svg. Флеш-технологии в веб-дизайне.
11. Растровые форматы веб-графики JPEG, GIF, PNG. Оптимизация размеров изображений.
12. Типы графики: фоновая, выставочная, визуал, логотип, навигационная, рекламная.
13. Язык гипертекстовой разметки текста HTML. История создания, версии. Декларация версии HTML <!DOCTYPE ...>.
14. Принцип разделения содержимого и оформления веб-страницы. Каскадные таблицы стилей (CSS). Подключение стилей.
15. Принцип разделения содержимого и оформления веб-страницы. Каскадные таблицы стилей (CSS). Селекторы (тэг, класс, ID).
16. Синтаксис языка HTML. Тег. Стандартное строение контейнера.
17. Структура HTML-документа. Содержимое области HEAD. Содержимое области BODY.
18. Язык гипертекстовой разметки текста HTML (основные теги, их атрибуты и значения атрибутов).
19. Способы задания URL. Абсолютный и относительный путь.
20. Гиперссылки. Элементы гиперссылок.
21. Формы. Элементы форм.
22. Программные средства создания веб-страниц. Adobe Dreamweaver как средство структурного построения сайта и его публикации.
23. Макет сайта. Основные структурные блоки сайта: шапка, подвал, область контента и др. Способы верстки: фиксированная, резиновая, адаптивная.
24. Верстка страниц с помощью таблиц. Преимущества и недостатки.
25. Блочная (дивная) верстка страницы. Блочные и текстовые элементы. Теги <DIV>, . Макетирование с помощью CSS.
26. Браузеры. Задачи браузера.
27. Серверные платформы. Unix и Windows хостинги. Требования к хостингу. Белорусские провайдеры (хостеры).
28. Веб-серверы: Apache, IIS, nginx. Задачи веб-сервера.
29. Типы сайтов – статические и динамические. Понятие систем управления содержимым (контентом) сайта - CMS. Основные функции CMS. Специализация CMS.

30. Язык JavaScript. События JavaScript. Методы реализации динамических страниц. Технологии jQuery, AJAX.
31. CMS WordPress. Создание структуры сайта в WordPress. Рубрики, страницы, меню, виджеты.
32. Создание сайтов с помощью онлайн-конструкторов сайтов.
33. Цели и задачи сайта.
34. Этапы анализа веб-сайта.
35. Этапы проектирования веб-сайта.
36. Понятие онлайн-сервиса. Преимущества и недостатки. Онлайн-сервисы менеджмента и маркетинга в Интернет.
37. Законодательное регулирование национального сегмента сети Интернет.
38. Система доменных имен DNS. Выбор доменного имени ресурса.
39. Поисковое продвижение интернет-ресурса. Понятия SEO, SMM, SMO, ASO.
40. Поисковое продвижение интернет-ресурса. Поисковая оптимизация (SEO). Внутренняя оптимизация сайта. Подбор и размещение ключевых слов.
41. Поисковое продвижение интернет-ресурса. Поисковая оптимизация (SEO). Внутренняя оптимизация сайта. Метатэги.

4.3 Перечень рекомендованных средств диагностики

Для измерения степени соответствия учебных достижений студента требованиям образовательного стандарта рекомендуется использовать проектную деятельность, включающую проблемные, творческие задачи, предполагающие эвристическую деятельность и неформализованный ответ.

В качестве одного из элементов, рекомендуемого для выявления уровня учебных достижений студента, используются критериально-ориентировочные тесты. Они представляют собой совокупность тестовых заданий закрытой формы с одним или несколькими вариантами правильных ответов; заданий на установление правильной последовательности.

4.4 Критерии оценки уровня знаний и умений студентов

В целях подготовки к текущей/промежуточной аттестации, студенту следует просмотреть все имеющиеся и рекомендуемые материалы, представленные в печатном или электронном виде. Промежуточная аттестация проводится с целью оценки качества усвоения студентами всего

объёма содержания дисциплины и определения фактически достигнутых знаний, навыков и умений, а также компетенций, сформированных за время аудиторных занятий и самостоятельной работы студента.

Критерии оценивания ответов студентов

Оценка «отлично» (10-8 баллов) / «зачтено». Ответы на поставленные вопросы излагаются логично, последовательно и не требуют дополнительных пояснений. Делаются обоснованные выводы. Демонстрируются глубокие знания в изучаемой области. Студент демонстрирует владение понятийным аппаратом и научным языком по предмету, умение его использовать в постановке и решении научных и профессиональных задач; способность самостоятельно решать сложные проблемы в рамках учебной программы; усвоение основной и дополнительной литературы, рекомендованной учебной программой; активная самостоятельная работа на лабораторных (практических) занятиях, высокий уровень культуры исполнения заданий, грамотное оформление учебной документации.

Оценка «хорошо» / «зачтено» (7-5 баллов). Ответы на поставленные вопросы излагаются систематизировано и последовательно. Материал излагается уверенно. Демонстрируется умение анализировать материал, однако не все выводы носят аргументированный и доказательный характер. Студент демонстрирует активную самостоятельную работу на практических, лабораторных занятиях, высокий уровень культуры исполнения заданий и оформления учебной документации, периодически участвует в групповых обсуждениях.

Оценка «удовлетворительно» (4 балла) / «зачтено». Допускаются нарушения в последовательности изложения. Имеются упоминания об отдельных базовых нормативно-правовых актах. Демонстрируются поверхностные знания вопроса, с трудом решаются конкретные задачи. Имеются затруднения с выводами. Студент демонстрирует достаточный объем знаний по предмету в рамках образовательного стандарта.

Оценка «неудовлетворительно» (3-1 баллов) / «не зачтено». Материал излагается непоследовательно, сбивчиво, не представляет определенной системы знаний по дисциплине. Не проводится анализ. Выводы отсутствуют. Ответы на дополнительные вопросы отсутствуют. На лабораторных (практических) занятиях студент был пассивен, демонстрировал низкий уровень культуры исполнения заданий и их оформления, отсутствие знаний по предмету в рамках образовательного стандарта или отказ от ответа.

5. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

5.1 Учебная программа

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В условиях современного общества неотъемлемым качеством квалифицированного специалиста сферы культуры и искусства становится высокий уровень владения интернет-технологиями для решения профессиональных задач по использованию, разработке, сопровождению и продвижению интернет-ресурсов в гипермедийном пространстве глобальной сети. Это предполагает знание основных технологий разработки сайтов, методов и инструментов их поискового и социального продвижения, умение применять их интегрировано для решения задач маркетинга и менеджмента.

Учебная дисциплина «Основы фронтенд разработки» является частью модуля «Веб-дизайн и проектирование информационных ресурсов и систем» вместе с учебной дисциплиной «Проектирование информационных ресурсов и систем» и курсовым проектом по учебной дисциплине «Проектирование информационных ресурсов и систем».

Роль учебной дисциплины «Основы фронтенд разработки» состоит в овладении методами и средствами получения, хранения и обработки информации в области веб-дизайна и веб-проектирования, что позволит культурологу-менеджеру качественно выполнять различные задачи, связанные с учебной и в дальнейшем профессиональной деятельностью. Студенты смогут создавать, редактировать и заниматься продвижением веб-ресурсов для организаций во всех сферах культуры и искусства.

Программа обучения предусматривает наличие у студентов базовых знаний по основам информационных технологий; навыков владения основными приемами работы с объектами в операционной среде и офисным пакетом программ, системному анализу и моделированию. Изучение учебной дисциплины «Основы фронтенд разработки» основывается на знаниях и умениях, полученных студентами в процессе освоения таких учебных дисциплин, как «Основы информационных технологий», «Информационная культура специалиста», «Системный анализ и моделирование информационных процессов». Знания, умения и навыки, полученные в рамках изучения данной учебной дисциплины необходимы для дальнейшего усвоения учебных дисциплин в первую очередь «Проектирование информационных ресурсов и систем», а также «Языки и системы программирования», «Технологии компьютерной графики», «Реклама в сфере культуры», «Технологии создания баз данных сферы культуры»,

«Мультимедийная техника и системное программное обеспечение», «Медиакультура специалиста».

Объектом изучения учебной дисциплины «Основы фронтенд разработки» являются веб-ресурсы. Предметом – методы и технологии их создания, публикации, сопровождении и продвижении.

Цель учебной дисциплины «Основы фронтенд разработки» – обеспечить студентов теоретическими знаниями в области веб-дизайна, дать представление о веб-пространстве и веб-технологиях и выработать практические навыки по профессиональному использованию программных и технических средств веб-дизайна в будущей профессиональной деятельности менеджера при создании и сопровождении сайтов в области культуры и искусства.

Целевая направленность дисциплины обуславливает решение следующих *задач*:

- изучение основных понятий и концепций гипертекстового пространства, технологий веб-дизайна;
- изучение основных понятий и особенностей языков HTML, CSS;
- изучение типов верстки сайтов и их особенностей;
- формирование системы базовых знаний принципов и навыков работы с CMS;
- формирование системы базовых знаний этапов анализа и проектирования сайта;
- освоение технологиями создания и публикации веб-ресурсов в среде Интернет;
- освоение эффективных методов и особенностей поискового продвижения веб-ресурсов.;
- изучение законодательного регулирования белорусского сегмента сети интернет.

В результате изучения дисциплины студенты должны *знать*:

- теоретические основы HTML, CSS, понятие о Javascript;
- тенденции развития веб-технологий;
- типы моделей верстки сайтов и их особенности;
- этапы проектирования и реализации веб-сайтов и их содержание;
- принципы и методы создания веб-ресурсов;
- особенности разработки сайта на CMS;
- методы публикации и сопровождения гипертекстовых ресурсов;
- методы поискового продвижения и поисковой оптимизации;
- специфику продвижения ресурсов сферы культуры и искусства в среде Интернет и социальных сетях.

Должны *уметь*:

- использовать при написании сайтов специализированные и визуальные редакторы веб-дизайна;
- применять различные методы верстки веб-страницы;
- применять современные инструментальные средства разработки для создания динамических и интерактивных веб-страниц;
- работать с содержимым веб-страниц и манипулировать стилями, обрабатывать различные события, работать с формой и ее элементами;
- разрабатывать адаптивный дизайн сайтов;
- создавать структуру сайта и его контент в CMS;
- выбирать доменные имена и хостинг;
- выполнять внутреннюю и внешнюю оптимизацию сайта;
- использовать инструменты аналитики при поисковой оптимизации.

Должны владеть:

- практическими навыками работы в CMS;
- создания веб-страниц с применением различных приемов верстки;
- основными навыками создания и продвижения веб-ресурсов;
- творческим подходом к поставленной задаче в процессе практической деятельности.

В соответствии с учебным планом по специальности 1-21 04 01 Культурология (по направлениям), направления специальности 1-21 04 01-02 Культурология (прикладная), специализации 1-21 04 01-02 04 Информационные системы в культуре освоение образовательной программы по учебной дисциплине «Основы фронтенд разработки», должно обеспечить формирование компетенции у студентов следующей специализированной компетенции:

СК-19: использовать технологии фронтенд разработки при создании веб-ресурсов социокультурной сферы.

Основными формами учебной дисциплины являются лекции, лабораторные работы, практические занятия и самостоятельное изучение отдельных вопросов.

Учебным планом на изучение учебной дисциплины «Основы фронтенд разработки» всего предусмотрено 90 часов, в том числе 34 часа аудиторных занятий для студентов дневной формы получения образования, 8 часов аудиторных занятий для студентов заочной формы получения образования. Примерное распределение аудиторных часов по видам занятий для студентов дневной формы получения образования: лекции – 6, лабораторные занятия – 20, практические занятия – 8; для студентов заочной формы получения образования: 2 часа – лекции, 2 часа – практические занятия, 4 часа – лабораторные занятия.

Рекомендованная форма контроля знаний студентов – зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Введение. Технологии веб-дизайна

Цели и задачи дисциплины. Междисциплинарные связи. Понятие гипертекстового пространства www: история создания, инструментальное, техническое, программное и организационное обеспечение. Ключевые принципы Интернета. Глобальная сеть Интернет и информационное пространство WWW: понятие и отличие. Классификация интернет-ресурсов, критерии качества и оценки.

Перспективы развития всемирной паутины. Персоналии веб-дизайна. Понятие веб-клиента и веб-сервера. Браузеры, их виды. Технологии построения веб-приложений. Технологические средства веб: URL-адресация, HTML язык, и HTTP протокол. Клиентские средства разработки веб-приложений: HTML, CSS, JAVASCRIPT. Серверные технологии разработки веб-приложений: PERL, PHP, PYTHON, ASP и др.

Тема 2. Язык HTML. Технология CSS

Язык гипертекстовой разметки текста HTML (история, версии). Структура сайта: логическая, физическая. Требования к именам файлов. Именованная главная (первая) страница сайта.

Декларирование типа веб-документа. Синтаксис языка HTML. Тег. Контейнер (блок). Кодировки текста. Структура документов HTML. Содержимое области HEAD. Тег BODY и его параметры. Элементы, задающие шрифт. Элементы форматирования. Типы связей (ссылки). Формы.

Технология CSS: назначение, история, основные конструкции, версии. Основные понятия и определения. Преимущества каскадных таблиц стилей. Принцип разделения содержимого и оформления веб-страницы. Стандарты CSS. Поддержка возможностей CSS со стороны браузеров. Подключение стилей. Синтаксис CSS. Цвета и фон. Шрифты. Текст. Блоки, границы и отступы.

Возможности HTML5, CSS3, JavaScript и AJAX. Методы реализации динамических страниц.

Средства создания веб-страниц.

Элементы гиперссылки: указатель ссылки, адресная часть. Абсолютный и относительный путь.

Элементы веб-страницы: заголовок, текст, фон, графика, гиперссылки, таблицы.

Оптимизация размеров изображения. Форматы графических файлов в Интернет: растровые: gif, png, jpg и векторные: swf, svg. Функции графики:

фон, логотип, визуал, выставочная графика, навигационная и рекламная графика. Интерактивная графика и анимация в веб-пространстве.

Динамические компоненты: счетчик посещений, рекламный баннер, формы, бегущая строка, анимационные эффекты, полиморфная кнопка и т.д.

Звук в веб-пространстве. Форматы звуковых файлов. Сжатие звуковых файлов. Видео в веб-пространстве. Форматы видео. Стандарты сжатия. Способы внедрения звука и видео в документ.

Тема 3. Верстка сайта

Модели верстки: фиксированная, резиновая, адаптивная и смешанная. Типовые макеты.

Табличная верстка сайта. Таблицы как средство компоновки веб-страниц. Принципы табличной верстки. Свойства CSS, описывающие табличную модель. Достоинства и недостатки верстки веб-страниц с помощью таблиц.

Блочная верстка сайта. Особенности поддержки блочной модели различными браузерами. Абсолютное и относительное позиционирование блоков. Особенности использования различных моделей позиционирования для верстки веб-страниц. Конструктивное использование тегов `<div>`, `` и стилей. Свойства CSS, описывающие блочную модель. Макетирование веб-страниц с помощью CSS.

Основные методы достижения адаптивности сайта. Принципы построения адаптивного сайта. Техники адаптивной верстки. Свойства модуля flexbox и особенности верстки на его основе. Семантичность кода страницы: применение HTML-элементов, именование элементов, комбинация именованных элементов.

Тема 4. Этапы анализа и проектирования сайта

Этапы анализа веб-ресурса. Стратегии проектирования. Анализ предметной области, определение назначения, задач сайта, целевой аудитории и информационной тематики. Пользователи сайта, их характеристика.

Основные этапы проектирования сайтов, дизайн интерфейсов, структурные схемы и дизайн-макеты страниц. Проектирование логической структуры сайта. Принципы навигации. Проектирование информационного наполнения. Типы контента сайта. Выбор информационных шаблонов сайта. Концепция графического дизайна сайта. Средства прототипирования. Программирование, наполнение и тестирование сайта.

Типы навигации на сайте. Разработка интерактивных прототипов веб-страниц. Важнейшие принципы информационной архитектуры и usability веб-ресурсов. Особенности подготовки текстов для веб-страниц.

Тема 5. Системы управления содержимым сайта (CMS)

Понятие систем управления содержимым (контентом) – CMS. Основные функции CMS: создание, редактирование и управление контентом. Специализация CMS. Виды CMS и особенности применения при создании и ведении различных ресурсов. Состав CMS. Модульная архитектура CMS.

Работа с CMS. Установка CMS на локальном компьютере, выбор хостинга, домена, основные настройки. Веб-серверы. Локальные веб-серверы. Понятие и работа с базой данных MySQL.

Блочные элементы CMS: компоненты, модули, виджеты. Плагины. Шаблоны (темы) для CMS, подключение и настройка. Создание структуры информационного наполнения сайта. Создание навигации по сайту. Типы меню: одноуровневое, многоуровневое, выпадающее. Создание, редактирование и публикация страницы в среде CMS. Работа с медиаконтентом. Расширение возможностей сайта. Разработка собственных компонентов. Администрирование CMS. Защита CMS. Файл дополнительной конфигурации веб-сервера. Создание пользователей и разграничение доступа.

Тема 6. Хостинг и публикация веб-сайта. Онлайн-конструкторы

Понятие онлайн-конструкторов сайтов. Создание веб-сайтов с помощью онлайн-конструкторов сайтов. Обзор конструкторов сайтов.

Законодательное регулирование белорусского сегмента сети интернет.

Требования к хостингу сайта. Серверные платформы. Unix и Windows хостинги. Веб-серверы: Apache, IIS, nginx. Задачи администрирования веб-сервера. Белорусские провайдеры (хостеры). Стоимость хостинга. Специфика бесплатного хостинга.

Система доменных имен: DNS (Domain Name System). Выбор доменного имени сайта. Регистрация доменов. Регламентация доменных имен государственных органов и организаций Республики Беларусь.

Способы публикации веб-ресурса. FTP-клиенты.

Тема 7. Основы поискового продвижения.

Понятие поискового продвижения интернет-ресурса – SEO (Search Engine Optimization). Онлайн и офлайн продвижение. Поисковая оптимизация сайта. Белые, серые и черные методы поисковой оптимизации. Бан поисковых систем.

Внутренняя оптимизация сайта. Внешняя оптимизация сайта. Выбор ключевых слов. Регистрация в каталогах и поисковых системах. Обмен ссылками. Социальные закладки. Написание уникального контента. Комментирование в блогах. Контекстная и банерная реклама. RSS-рассылки. Файлы robots.txt и sitemap.xml.

Инструменты аналитики веб-разработчика (оценки посещаемости веб-сайтов и анализа поведения пользователей): Яндекс.Метрика, Google Analytics и др. Инструменты аналитики хостинга.

Тема 8. Разработка веб-сайта по заданной тематике

Определение цели, задач сайта, целевой аудитории сайта. Разработка структуры. Выбор макета и типа верстки. Определение общей структуры страницы. Подготовка контента и графических изображений. Верстка страниц сайта. Тестирование сайта. Анализ и оптимизация кода. Презентация готового проекта.

5.2 Учебно-методическая карта учебной дисциплины для дневной формы получения высшего образования

Название темы	Количество аудиторных часов			Количество часов УСР	Форма контроля знаний
	Лекции	Практические занятия	Лабораторные занятия		
Тема 1. Введение. Технологии веб-дизайна	2				
Тема 2. Язык HTML. Технология CSS	2		6		
Тема3. Верстка сайта		2	4		
Тема 4. Этапы анализа и проектирования сайта	2				
Тема 5. Системы управления содержимым сайта (CMS)		2	2	2	проект
Тема 6. Хостинг и публикация веб-сайта. Онлайн-конструкторы			2	2	проект
Тема 7. Основы поискового продвижения		2			
Тема 8. Разработка веб-сайта по заданной тематике			2	2	проект
Всего:	6	6	16	6	

5.3 Учебно-методическая карта учебной дисциплины для заочной формы получения высшего образования

Название темы	Количество аудиторных часов		
	Лекции	Практические занятия	Лабораторные занятия
Тема 1. Введение. Технологии веб-дизайна	0,5		
Тема 2. Язык HTML. Технология CSS	0,5		1
Тема3. Верстка сайта		1	1
Тема 4. Этапы анализа и проектирования сайта	1		
Тема 5. Системы управления содержимым сайта (CMS)		0,5	1
Тема 6. Хостинг и публикация веб-сайта. Онлайн-конструкторы		0,5	1
Всего:	2	2	4

Темы для самостоятельного изучения:

1. Основы поискового продвижения.
2. Разработка веб-сайта по заданной тематике.

5.4 Список основной литературы

1. Батенькина, О.В. Юзабилити информационных систем: учебное пособие / О.В. Батенькина, О.Н. Ткаченко. – Омск: ОмГТУ, 2015. – 144 с. – ISBN 978-5-8149-2095-9. – Текст : электронный // Лань : электронно-библиотечная система. – URL:<https://e.lanbook.com/book/149059>
2. Никулова, Г. А. Проектирование и реализация Web-интерфейса : учебно-методическое пособие / Г. А. Никулова. – Липецк : Липецкий ГПУ, 2020. – 66 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/156075>
3. Побединский, Е. В. Проектирование веб-сайтов с использованием технологий PHP, HTML, CSS и WordPress : учебное пособие / Е. В. Побединский, В. В. Побединский. – Екатеринбург : УГЛТУ, 2018. – 115 с. – ISBN 978-5-94984-651-3. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/142518>
4. Саблина, Н.А. Основы Web-дизайна : учебно-методическое пособие / составитель Н.А. Саблина. – Липецк : Липецкий ГПУ, 2018. – 50 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/115017>

5.5 Список дополнительной литературы

1. Аграновский, А. В. Тестирование веб-приложений : учебное пособие / А. В. Аграновский. – Санкт-Петербург : ГУАП, 2020. – 155 с. – ISBN 978-5-8088-1515-5. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/216533>
2. Асалханов, П. Г. Web-программирование: JavaScript : учебное пособие / П. Г. Асалханов. – Иркутск : Иркутский ГАУ, 2020. – 123 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/183488>
3. Беликова, С. А. Основы HTML и CSS: проектирование и дизайн веб-сайтов: учебное пособие по курсу «Web-разработка» : [16+] / С. А. Беликова, А. Н. Беликов ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2020. – 176 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=598663>
4. Ветцель, К. Я. Интернет-маркетинг : учебное пособие / К. Я. Ветцель. – Красноярск : СФУ, 2018. – 176 с. – ISBN 978-5-7638-4010-0. –

Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/157667>

5. Дзялошинский, И. М. Современный медиатекст. Особенности создания и функционирования : учебник для вузов / И. М. Дзялошинский, М. А. Пильгун. – 2-е изд., испр. и доп. – Москва : Издательство Юрайт, 2022. – 345 с. – (Высшее образование). – ISBN 978-5-534-11621-2. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/495400>

6. Дунаев, В.В. HTML, скрипты и стили. – 4-е изд., перераб. и доп. / В. Дунаев. – Спб.: БХВ-Петербург, 2015. – 816 с.: ил.

7. Жильцова, О.Н. Интернет-маркетинг : учебник для вузов / О. Н. Жильцова [и др.] ; под общей редакцией О. Н. Жильцовой. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2022. – 335 с. – (Высшее образование). – ISBN 978-5-534-15098-8. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/489043>

8. Журавлева, И.А. Технология разработки интернет ресурсов: курс лекций : учебное пособие : [16+] / авт.-сост. И. А. Журавлёва. – Ставрополь : Северо-Кавказский Федеральный университет (СКФУ), 2018. – 171 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=562579>

9. Кабанов, А. Эффективная эксплуатация сайта. Практическое пособие / А. Кабанов. – М. : Дашков и К. – 2015. – 176 с.

10. Кирсанов, Д. Веб-дизайн / Д. Кирсанов. – СПб: Символ, 2015. – 368 с.

11. Кукарцев, В. В. Проектирование и архитектура информационных систем : учебник / В. В. Кукарцев, Р. Ю. Царев, О. А. Антамошкин. – Красноярск : СФУ, 2019. – 192 с. – ISBN 978-5-7638-3620-2. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/157581>

12. Лоусон, Б. Изучаем HTML5. Библиотека специалиста. 2-е изд./ Б. Лоусон Б., Р. Шарп – СПб.: Питер, 2012, – 304 с.

13. Макарова, Т. В. Веб-дизайн : учебное пособие / Т. В. Макарова. – Омск : ОмГТУ, 2015. – 148 с. – ISBN 978-5-8149-2075-1. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/149129>

14. Можаров М. С. Проектирование и разработка информационных систем с web-интерфейсом: Учебное пособие / М.С.Можаров, А.Э.Можарова; М-во науки и высшего образования Рос.Федерации. – Новокузнецк: НФИ

КемГУ, 2019. – 135 с.// Лань : электронно-библиотечная система. – URL: <https://reader.lanbook.com/book/169625#2>

15. Нейл, Т. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / Т. Нейл. – СПб.: БХВ-Петербург, 2014.– 416 с.

16. Нильсен, Я. Веб-дизайн / Я. Нильсен. – Пер. с англ. – СПб.: Символ, 2015. – 512 с.

17. Овчинникова, Р.Ю. Дизайн в рекламе: основы графического проектирования / Р. Ю. Овчинникова ; ред. Л. М. Дмитриева. – Москва : Юнити-Дана, 2017. – 239 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=684750>

18. Роббинс, Дженнифер Нидерст. Веб-дизайн для начинающих. HTML, CSS, JavaScript и веб-графика / Дженнифер Нидерст Роббинс. – 5-е изд. – Санкт-Петербург : БХВ-Петербург, 2021. – 912 с., [22] л. фот. : рис., табл.

19. Сидерхолм, Д. CSS3 для веб-дизайнеров / Д. Сидерхолм. – Москва, Издательство «Манн, Иванов и Фербер», 2013, – 144с.

20. Скотт Б., Нейл, Т. Проектирование веб-интерфейсов./ Б. Скотт, Т. Нейл. – СПб.: Символ-плюс, 2014. – 352 с.

21. Смоленцева, Т. Е. Базовые и прикладные информационные технологии. Разработка Web-приложений : учебно-методическое пособие / Т. Е. Смоленцева. – Москва : РТУ МИРЭА, 2021. – 78 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/218702>

22. Тузовский, А. Ф. Проектирование и разработка web-приложений : учеб. пособие для академического бакалавриата / А. Ф. Тузовский. – М. : Издательство Юрайт, 2016. – 218 с. – Серия : Университеты России.

23. Уильямс, Робин. Дизайн. Книга для недизайнеров. Принципы оформления и типографики для начинающих / Робин Уильямс ; [пер. с англ. В. Черник]. – 4-е изд. – Санкт-Петербург ; Москва ; Минск : Питер, 2022. – 239 с. : цв. ил. – ISBN 978-5-4461-1127-5 : 61-56.

24. Флэнаган, Д. JavaScript. Карманный справочник / Д. Флэнаган. – М. – Вильямс, 2015. – 320с.

25. Фрэйнд, Бен. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств = Responsive Web Design with HTML5 and CSS3 / Бен Фрэйнд ; [пер. с англ. Н. Вильчинский]. – 2-е изд. – Санкт-Петербург : Питер, 2017. – 272 с. : ил. ; 24x17 см. – (Серия "Библиотека программиста"). – ISBN 978-5-496-02271-2. – ISBN 978-1784398934 (англ.) : 28-12.

26. Хоган, Б. Книга веб-программиста: секреты профессиональной разработки веб-сайтов. / Б. Хоган, К. Уоррен, М. Уэбер, К. Джонсон, А. Годин – СПб.: Питер, 2013. – 288 с.: ил.