

МИНОБРНАУКИ РОССИИ



Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
(ФГБОУ ВО «РГГУ»)

ФАКУЛЬТЕТ «ИНФОРМАЦИОННЫХ СИСТЕМ И БЕЗОПАСНОСТИ»
Кафедра «Информационных технологий и систем»

WEB-ПРОГРАММИРОВАНИЕ

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Направление подготовки 09.03.03 «Прикладная информатика»
Направленность (профиль): Информационно-коммуникационные технологии цифровой
трансформации

Уровень высшего образования: бакалавриат

Форма обучения: очная

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2024

WEB-ПРОГРАММИРОВАНИЕ

Рабочая программа дисциплины

Составитель:

к.х.н., с.н.с., доцент А.М. Подорожный

Ответственный редактор:

к.с-х.н., доцент, зав. кафедрой

Информационных технологий и систем Н.Ш. Шукенбаева

УТВЕРЖДЕНО

Протокол заседания кафедры

Информационных технологий и систем

№ 8 от 04.03.2024 г.

ОГЛАВЛЕНИЕ

1	Пояснительная записка	4
1.1.	Цель и задачи дисциплины	4
1.2.	Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций	4
1.3.	Место дисциплины в структуре образовательной программы.	4
2	Структура дисциплины	5
3	Содержание дисциплины	5
4	Образовательные технологии	6
5	Оценка планируемых результатов обучения	6
5.1.	Система оценивания	6
5.2.	Критерии выставления оценки по дисциплине	7
5.3.	Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине	8
6	Учебно-методическое и информационное обеспечение дисциплины	10
6.1.	Список литературы	10
6.2.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет»	11
6.3.	Профессиональные базы данных и информационно-справочные системы	11
7	Материально-техническое обеспечение дисциплины	11
8	Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов	11
9	Методические материалы	13
9.1.	Планы практических заданий	13
9.2.	Иные материалы. Планы самостоятельной работы.	23
	Приложение 1. Аннотация дисциплины	25

1 Пояснительная записка

1.1. Цель и задачи дисциплины

Цель дисциплины: «Web-программирование в гуманитарной сфере» имеет целью теоретическое и практическое освоение методов и технологий, используемых в дизайне и верстке современных Web-документов.

Задачи: выработка у студентов системного подхода к решению задач дизайна и верстки интернет ресурсов, приобретение практических навыков построения различных web-систем на основе выбора наименее трудоемкой и, вместе с тем, адекватной поставленной задаче методологии разработки.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-2. Способен разрабатывать, тестировать и адаптировать прикладное программное обеспечение.	ПК-2.1. Знает основные среды для разработки программного обеспечения, методы тестирования и адаптации прикладного программного обеспечения.	Знать: принципы работы и основные протоколы сети интернет, языки программирования и разметки, подходы, используемые при создании сайтов, методы тестирования и адаптации программных компонентов интернет-ресурсов, технологии поисковой оптимизации и продвижения сайтов.
	ПК-2.2. Умеет разрабатывать, тестировать и адаптировать прикладное программное обеспечение.	Уметь: определять методы решения задач предметной области, с средствами сети интернет, разрабатывать структуру базы данных, адаптированную к использованию в интернет проектах, тестировать компоненты программного обеспечения интернет-ресурсов, оценивать эффективность использования конкретного языка и технологии при решении поставленных задач.
	ПК-2.3. Владеет современными языками программирования и методиками разработки, тестирования и адаптации прикладного программного обеспечения.	Владеть: навыками работы с инструментальными средствами разработки в сети интернет (HTML, CSS), языками PHP, MySQL и JavaScript, методами оптимизации, построения и продвижения сайтов. практическими навыками разработки и тестирования программных компонентов интернет-ресурсов.

1.3. Место дисциплины в структуре образовательной программы.

Дисциплина «Web-программирование» является дисциплиной вариативной части учебного плана по направлению подготовки 09.03.03 Прикладная информатика, профиль: Информационно-коммуникационные технологии цифровой трансформации. Дисциплина реализуется на факультете Информационных систем и безопасности кафедрой Информационных технологий и систем. Для освоения дисциплины необходимы знания,

умения и владения, сформированные в ходе изучения следующих дисциплин: «Архитектура вычислительных систем», «Информационные системы», «Алгоритмы и структуры программирования», «Компьютерная графика в задачах цифровой трансформации».

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин: «Мировые информационные ресурсы», «Анализ данных социальных сетей», «Сервис ориентированная архитектура информационных систем», «Методы информационного поиска», «Информационно-поисковые системы и машины».

2 Структура дисциплины

Общая трудоёмкость дисциплины составляет 4 з.е., 144 академических часа.

Структура дисциплины для очной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
4	Лекции	24
4	Практические занятия	32
Всего:		56

Объем дисциплины в форме самостоятельной работы обучающихся составляет 88 академических часов.

3 Содержание дисциплины

№	Наименование раздела дисциплины	Содержание
1.	Основы работы сети интернет	<p>Условия работы информационных сетей: протоколы передачи данных, уровни сетевого соединения, клиент-серверные технологии. Новации Интернета: распределённая архитектура и динамическая маршрутизация, пакетная передача данных (фрагментация, инкапсуляция), модель OSI.</p> <p>Протоколы прикладного уровня, адрес URL. Уровень представления (формат, процессы), сеансовый уровень. Транспортный уровень: протоколы TCP и UDP. Сетевой уровень: адреса IPv4 и IPv6, публичные и частные, динамические и постоянные. Канальный уровень: MAC-адрес, состав кадра, контрольная сумма, виды модуляции.</p> <p>Коммуникации интернета: скорость перемещения, виды. Кабельная и беспроводная связь: сравнение применение. Оптоволокно: достоинства, применение. Неполная обработка сигналов.</p>
2.	World Wide Web и другие прикладные	<p>Ресурс WWW: основные понятия, история развития. Другие ресурсы: почта web-mail и e-mail, FTP, терминальное управление, DNS, торрент, мессенджеры. Строение адреса URL: протоколы</p>

	сервисы	<p>доступа к ресурсу, доменное имя, адреса порта и выводимого файла, вывод данных по запросу.</p> <p>Средства создания Web-страниц. Программные: HTML, CSS, JavaScript, MySQL, PHP, DOM, DHTML, пакет XAMPP. Конструкторы сайтов: технологии, виды, рейтинг, WordPress, Bitrix, упрощенные CMS и др. Виртуальные Web-редакторы их применение. Размещения сайтов в Сети: бесплатный домен, платный домен, собственный домен, хостинг.</p>
3.	Юзабилити сайта	<p>Понятие «Юзабилити». Постановка задачи на создание сайта, эволюция прототипов, этапы создания. Изучение аудитории, профилирование. Разработка семантического ядра. Учет психофизических характеристик людей при проектировании сайта. Структурирование информации, меры по повышению доступности сайта. Дизайн сайта: эстетическая и функциональная стороны, цвета, текст и фон, стили, элементы навигации, анимация, шрифты. Топологическая и файловая структура сайтов. Статичные и интерактивные сайты, средства, придающие интерактивность. Оптимизация крупных сайтов.</p>
4.	Верстка сайтов средствами HTML, CSS, JavaScript.	<p>Развитие, характеристика HTML 5. Правила записи тегов, контейнеры, автономные теги, атрибуты и пр. Структура документа HTML. Раздел Head: теги Meta, Link, Script, Style и др. Теги физического и логического форматирования текста, создание заголовков, списков, таблиц, вставка иллюстраций, блоки DIV, семантические теги, глобальные атрибуты.</p> <p>Назначение и синтаксис записи стилей CSS, способы помещения на веб-страницу. Наследование, каскадирование стилей. Стили форматирования документа, текста, шрифтов, графики, цвета и др.</p> <p>Основы JavaScript: способы подключения, помещения на странице. Возможности языка JavaScript: типы данных, операции, операторы, функции, объекты, массивы. Примеры использования JavaScript для верстки Web-страниц.</p> <p>Объектная модель документа DOM, назначение, структура, примеры использования.</p>

4 Образовательные технологии

Для проведения учебных занятий по дисциплине используются различные образовательные технологии. Для организации учебного процесса может быть использовано электронное обучение и (или) дистанционные образовательные технологии.

5 Оценка планируемых результатов обучения

5.1. Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль:		
- опрос-коллоквиум	5 баллов	20 баллов
- практические задания	5 баллов	20 баллов
- самостоятельные работы	5 баллов	20 баллов
Зачет с оценкой по билетам		40 баллов
Итого за семестр		100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала		Шкала ECTS
95 – 100	Отлично	зачтено	A
83 – 94			B
68 – 82	Хорошо		C
56 – 67	Удовлетворительно		D
50 – 55			E
20 – 49	Неудовлетворительно	не зачтено	FX
0 – 19			F

5.2. Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A,B	«отлично»	<p>Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения.</p> <p>Свободно ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ C	«хорошо»	<p>Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей.</p> <p>Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами.</p> <p>Достаточно хорошо ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с</p>

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
		<p>учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D,E	«удовлетворительно»	<p>Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами.</p> <p>Демонстрирует достаточный уровень знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F,FX	«неудовлетворительно»	<p>Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами.</p> <p>Демонстрирует фрагментарные знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

Вопросы к опросам-коллоквиумам

Тема 1.

1. Понятия: протокол передачи данных, уровни сетевого соединения. (ПК-2.1)

2. Понятия: сервер и клиент, пакетная передача данных OSI. (ПК-2.1)
3. Распределенная архитектура и динамическая маршрутизация в интернете. (ПК-2.1)
4. Уровни прикладной, представления, сеансовый в модели OSI. (ПК-2.1)
5. Транспортный и канальный уровни в модели OSI. (ПК-2.1)
6. Сетевой уровень в модели OSI. (ПК-2.1)
7. Цифровые кабельные и беспроводные коммуникации интернета. (ПК-2.2)
8. Оптоволоконные коммуникации, прием и обработка сигналов в OSI. (ПК-2.2)

Тема 2.

1. World Wide Web как гипертекст. Основные определения. (ПК-2.2)
2. История развития World Wide Web. (ПК-2.2)
3. Электронная почта e-mail и web-mail. Сравнение, применение. (ПК-2.2)
4. Протоколы FTP, удаленного доступа и другие прикладные сервисы. (ПК-2.2)
5. Структура адреса URL. (ПК-2.2)
6. Программные средства создания Web-страниц, пакет ХАМРР. (ПК-2.2)
7. Конструкторы сайтов, их виды, возможности, визуальные Web-редакторы. (ПК-2.2)
8. Размещение в Интернете бесплатного и платного доменов. (ПК-2.2)
9. Создание собственного Web-домена. Хостинг. (ПК-2.2)

Тема 3.

1. Термин «Юзабилити», постановка задачи при создании сайта. (ПК-2.2)
2. Сегментация и профилирование аудитории Web-сайта. (ПК-2.2)
3. Проектирование ядра ключевых слов. (ПК-2.2)
4. Специфика создания Web-контента. (ПК-2.1)
5. Дизайн сайта: цвет и форма. (ПК-2.2)
6. Дизайн сайта: композиция. (ПК-2.2)
7. Дизайн сайта: функциональная сторона. (ПК-2.2)
8. Топологические и файловая структуры Web-сайтов. (ПК-2.2)
9. Интерактивные средства Web-сайтов. (ПК-2.2)
10. Личный кабинет, оптимизация крупных сайтов. (ПК-2.2)

Тема 4.

1. Создание и развитие языка HTML. (ПК-2.3)
2. Правила записи тегов HTML. (ПК-2.3)
3. Содержание раздела HEAD. (ПК-2.3)
4. Физическое и логическое форматирование текста в HTML. (ПК-2.3)
5. Функции, преимущества, синтаксис записи CSS. (ПК-2.3)
6. Связывание стилей CSS с HTML документом. (ПК-2.3)
7. Способы записи стилевых свойств. (ПК-2.3)
8. Классы стилей, наследование стилей. (ПК-2.3)

Из вопросов коллоквиумов к зачету составляются билеты, по два вопроса в билете, обязательно из разных разделов.

6 Учебно-методическое и информационное обеспечение дисциплины

6.1. Список литературы

Основная

1. Гуриков, С. Р. Интернет-технологии : учебное пособие / С.Р. Гуриков. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2024. — 174 с. — (Высшее образование). — DOI 10.12737/1044018. - ISBN 978-5-16-019834-7. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2137802>
2. Тиге, Д. К. DHTML и CSS : практическое руководство / Д. К. Тиге ; пер. с англ. М. И. Талачевой. - 2-е изд. - Москва : ДМК Пресс, 2023. - 560 с. - (Быстрый старт). - ISBN 978-5-89818-455-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2106224>
3. Г.А. Лисьев, П.Ю. Романов, Ю.И. Аскерко. Программное обеспечение компьютерных сетей и web-серверов : учебное пособие / — Москва : ИНФРА-М, 2020. — 145 с. Режим доступа: <https://new.znanium.com/catalog/product/1068576>.
4. Фаррелл, Б. Веб-компоненты в действии : практическое руководство / пер. с англ. Д. А. Беликов. - Москва : ДМК Пресс, 2020. - 462 с. - ISBN 978-5-97060-856-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1210671>
5. Т.И. Немцова, Т.В. Казанкова, А.В. Шнякин. Компьютерная графика и web-дизайн : учебное пособие / под ред. Л.Г. Гагариной. – Москва : ИД «ФОРУМ» : ИНФРА-М, 2024. – 400 с. + Доп. материалы Режим доступа: <https://new.znanium.com/catalog/document?id=344430>.

Дополнительная

1. Цыгулин, А. А. Основы веб-программирования : учебное пособие / А. А. Цыгулин. - Новосибирск : Изд-во НГТУ, 2020. - 64 с. - ISBN 978-5-7782-4197-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1866934>
2. Энж Э. SEO - искусство раскрутки сайтов: Самоучитель. - СПб:БХВ-Петербург, 2017. - 816 с.: ISBN 978-5-9775-3686-8 - Режим доступа: <https://new.znanium.com/catalog/product/978493>.
3. Локхарт, Д., Рагимов Р.Н. Современный PHP. Новые возможности и передовой опыт. - Москва : ДМК Пресс, 2016. - 304 с. - ISBN 978-5-97060-184-6. Электронный ресурс. Режим доступа: <https://new.znanium.com/catalog/product/1028044>.
4. Дронов В.А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов: Пособие / Дронов В.А. - СПб:БХВ-Петербург, 2016. - 688 с. ISBN 978-5-9775-3529-8 - Режим доступа: <http://znanium.com/catalog/product/944562>

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Smashing Magazine – крупнейший ресурс, посвященный Веб-разработкам. Статьи, книги, сведения о вакансиях, заказах и пр. <https://www.smashingmagazine.com>.
2. Хабр – крупнейший русскоязычный ресурс, посвященный Веб-разработкам. Разработки, администрирование, дизайн, менеджмент, маркетинг и др. <https://habr.com/ru/>
3. <https://www.scopus.com> – Международная реферативная наукометрическая БД, Scopus, доступна по подписке.
4. <http://www.consultant.ru/> - КонсультантПлюс компьютерная справочная правовая система.

6.3. Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: <https://liber.rsuh.ru/ru/bases>

Информационные справочные системы:

1. Консультант Плюс
2. Гарант

7 Материально-техническое обеспечение дисциплины

Для реализации программы дисциплины требуется:

- компьютеры, с предустановленным базовым программным обеспечением в составе ОС Windows 10 и MS Office;
- объединение компьютеров в локальную сеть компьютерного класса с высокоскоростным выходом в интернет;
- для лекционного курса – посадочные места по количеству обучающихся, доска, мультимедиа проектор с экраном;
- рабочее место преподавателя, оснащенное компьютером с выходом на проектор.

Используемое программное обеспечение:

1. Windows 10
2. Microsoft Office 2013 Pro
3. Mozilla Firefox 52.8.1 ESR
4. Кроссплатформенная сборка Web-сервера XAMPP 7.3.0
5. Adobe CS4 Master Collection
6. Kaspersky Endpoint Security
7. Платформа ZOOM

8 Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;

письменные задания выполняются на компьютере со специализированным программным обеспечением или могут быть заменены устным ответом; обеспечивается индивидуальное равномерное освещение не менее 300 люкс; для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; письменные задания оформляются увеличенным шрифтом; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

- для глухих и слабослышащих: лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования; письменные задания выполняются на компьютере в письменной форме; экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

- для лиц с нарушениями опорно-двигательного аппарата: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих: в печатной форме увеличенным шрифтом, в форме электронного документа, в форме аудиофайла.
- для глухих и слабослышащих: в печатной форме, в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата: в печатной форме, в форме электронного документа, в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих: устройством для сканирования и чтения с камерой SARA CE; дисплеем Брайля PAC Mate 20; принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих: автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих; акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата: передвижными, регулируемые эргономическими партами СИ-1; компьютерной техникой со специальным программным обеспечением.

9 Методические материалы

9.1. Планы практических заданий

Практическая работа №1. Создание Web-страниц формата .html в текстовом редакторе Notepad++.

- Общие положения.
- Создание стандартной структуры страницы.
- Разметка Web-страницы с помощью таблиц.
- Форматирование страницы средствами HTML.
- Форматирование текста средствами CSS.
- Добавление графики.
- Создание фона.
- Специальные символы.
- Создание гиперссылок.

Практическая работа №2. Создание коммерческого Web-сайта.

- Определение параметров Web-проекта;
- Создание пустых страниц;
- Импорт и правка текста;
- Работа с таблицами;
- Создание и применение стилей CSS;
- Добавление иллюстраций;
- Создание навигационной панели и гиперссылок;
- Создание интерактивных элементов;
- Финишные операции.

Практическая работа №3. Создание Web-сайта на сервере wordpress.com.

- Общие положения;
- Установка WordPress на XAMPP;
- Начало работы;
- Создание шапки сайта;
- Создание главного меню;
- Создание подвала;
- Дизайн и заполнение главной страницы;
- Блоги новостей и комментариев;
- Страницы с таблицами;
- Статичные и смешанные страницы.

Практическая работа №4. Создание интернет-магазина с использованием .JavaScript.

- Общие положения;
- Главная страница;
- Страница «Контакты» с Яндекс-картой;
- Интерактивные страницы с товарами;
- Страница корзины;
- Проектирование бонусов.

Пример описания практической работы.

Практическая работа №1

Создание Web-страниц формата .html в текстовом редакторе «Notepad++»

1. Общие положения

Набор программного кода в специализированном текстовом редакторе, таком как Notepad++, - это универсальное средство создания Web-страницы. Здесь процесс создания кода полностью подконтролен пользователю.

Конечно, таким способом сейчас создают сайты довольно редко, чаще пользуются готовыми решениями. Но знание основ HTML, CSS и других средств становится необходимым, когда:

- возникают «непреодолимые» трудности, которые знающий человек может разрешить, внося нужные изменения в программный код;
- когда требуется объединять в единое целое готовые блоки;
- когда необходимо создать нечто ценный, уникальный продукт.

Ну и, конечно, в учебных целях обязательно требуется создавать сайты набором программного кода.

В данной работе предлагается создать сайт средствами HTML 5 и CSS 3. Это сейчас рекомендуемые версии. С одной стороны, они современны, с другой – достаточно проверены, и поддерживаются последними версиями ведущих браузеров.

Язык HTML

HTML – это язык разметки Web-страниц. Код HTML форматирует содержание страницы (текст, рисунки, звуки и прочее) с помощью стандартных команд – **тегов** (они также называются **дескрипторы**). При форматировании задаются размер, расположение и много других свойств элементов, из которых состоит страница.

При записи программного кода команды-теги заключают в угловые скобки, например: **<HTML>**. Большинство тегов парные, они подразделяются на открывающие и закрывающие. Открывающий тег определяет действие, которое производится с текстом или другим элементом, стоящим после него. Закрывающий указывает на окончание действия и имеет косую черту: **</HTML>**. Команда распространяется на все, что расположено между открывающим и закрывающим тегом.

Кроме парных тегов есть и одиночные, не имеющие закрывающей части. Например, **МЕТА** теги или команда добавления графики: ****.

Теги могут быть вложены друг в друга. Это происходит в том случае, когда к участку текста или другому объекту одновременно применяется несколько команд. При этом работает **принцип стека**: команда, примененная позже, отменяется раньше (и наоборот – самая ранняя команда отменяется позже всех).

Пример. Запись:

```
<P>Это <Strong>вложенные <Em>теги</Em></Strong></P>
```

устанавливает следующую разметку:

Это вложенные теги

Теги могут иметь **атрибуты**, которые задают различные варианты использования одного и того же тега. Пишется атрибут, затем через знак «=» – свойство атрибута.

Примеры:

- **** – src, это обязательный атрибут, он показывает путь к загружаемому изображению;
- **< IMG src=Images/picture.jpg alt="Подпись к рисунку">** – здесь приведён ещё один, необязательный атрибут alt, текстовая альтернатива рисунку, читаемая поисковой системой.

В HTML 5 число атрибутов сильно уменьшено по сравнению с предыдущими версиями. Свойства устаревших атрибутов теперь устанавливаются средствами CSS. Лучше не употреблять устаревшие средства, поскольку неизвестно: когда и в каких браузерах они будут отменены.

Важно, что к некоторым командам язык HTML нечувствителен.

- В программном коде между разными тегами может задаваться сколько угодно пробелов и переводов строки, это не имеет никакого значения. Например, записи:

```
<P>Internet</P><P>Explorer</P> и
<P>Internet</P>           <P>Explorer</P>
```

полностью равнозначны.
- Теги и атрибуты (но не значения атрибутов) можно набирать как строчными, так и прописными буквами. Например, обозначение тега **TABLE**, **Table** и даже **tAbLe** будут восприниматься браузерами одинаково.

Стили CSS

Основные параметры форматирования текста: шрифт, начертание, размер и др. устанавливаются в помощью стилей CSS. Это даёт много преимуществ, а именно:

- разделение вёрстки и дизайна (кода и оформления, каждую работу можно вести параллельно);
- стили хранятся и редактируются в одном месте, что упрощает создание единого оформления всех документов сайта;
- у CSS очень много возможностей для оформления документов, они касаются не только текста элементов, но и цвета, положения, отображения различных элементов, есть много эффектов, трансформаций, анимаций и пр.

Имеются и другие сильные стороны.

Стиль CSS создаётся следующим образом: **Тег1, Тег2 {свойство1: значение; свойство2: значение; свойство3: значение}**. Пишется тег или теги, к которым относится стиль, далее в фигурных скобках перечисляются свойства и их значения. То есть, не надо каждому тегу задавать оформление, заданный дизайн устанавливается автоматически.

Пример:

H1{font-family: Arial; color: blue; text-align: center}. Запись означает, что для всех заголовков размера H1 будет устанавливаться шрифт Arial, синий цвет, выравнивание по центру.

Существует три варианта помещения стиля в код HTML.

1. Ссылка на внешнюю таблицу стилей. На сайте создается файл с расширением .css, в который собираются стили сайта. А на Web-страницах с помощью тега **Link** создаётся гиперссылка на страницу стилей.
2. Создание таблицы стилей на отдельной Web-странице. Для этого используется тег **<Style></Style>**, помещаемый в раздел HEAD этой страницы. В нем перечисляются используемые стили.
3. Внутритекстовые стили. Применяются тогда, когда для какого-то фрагмента кода хотят создать необычное форматирование. Имеются разные записи для одного тега, для выделенного участка из нескольких тегов, и для участка внутри тега.

2. Создание стандартной структуры страницы

2.1. Откройте программу **Notepad++**, откройте новый файл, поставьте синтаксис HTML: **Синтаксисы → Н → HTML**.

2.2. Первой строкой надо указать, что документ создается средствами HTML 5. Об этом будет свидетельствовать запись:

```
<!DOCTYPE html>
```

2.3. Далее построим обычную структуру Web-документа:

```
<HTML>
<HEAD>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Наберите указанные теги. Рекомендуется для ускорения набора кода максимально использовать буфер обмена.

Тег `<HTML>` указывает на язык программирования HTML. Эта запись нужна потому, что Web-страницы могут создаваться и другими средствами.

В разделе **HEAD** задаются базовые параметры страницы, посетитель при загрузке сайта их не видит.¹

В разделе **BODY** размещается содержание станицы. Здесь все, что находится за пределами угловых скобок `< >`, выводится на экран.

2.4. Сохраните созданную страницу под именем **Index** в отдельную папку (названную латинским шрифтом). В строке **Тип файла** поставьте из списка **Hyper Text Markup Language file**.

По протоколам **HTTP/HTTPS**, при отсутствии в адресе **URL** указаний на имя файла включается главная страница сайта, которая должна иметь имя **Index**.

2.5. Запустите файл, щелкнув по иконке, при запуске включается браузер и демонстрируется содержимое страницы, пока это пустое окно. Щелкните по иконке правой кнопкой и включите **Edit with Notepad++**. В дальнейшем надо будет набирать код в Notepad++, сохранять его (можно **Ctrl + S**) и смотреть изменения в браузере.

2.6. Создайте заголовок страницы, отображаемый в строке заголовка браузера. Для этого в разделе **HEAD** надо создать строку: `<TITLE></TITLE>`, и между этими тегамы написать название страницы, страница будет называться: **Казахская кухня**. Теперь при запуске страницы в строке заголовка, на самом верху экрана, появляется ее название.

Создадим несколько управляющих параметров, реализуемых с помощью непарных управляющих тегов **META**.

2.7. Важным является мета-тег, указывающий на кодировку UTF-8.² Её поддерживают все браузеры, в том числе и устаревшие версии:

```
<meta charset="UTF-8"/>
```

2.8. Запишите непарный тег с атрибутами:

```
<META name="author" contents="Имя Фамилия" >
```

Поставьте свои имя и фамилию. Этот дескриптор создает имя автора для поисковой системы.

2.9. Далее там же запишем

```
<META name="description" contents="Страница создана при выполнении лабораторной работы по предмету ..." >
```

Поставьте название предмета. Здесь после contents пишется то, что поисковая система будет приводить в качестве резюме при помещении вашей странице в список вместе с другими найденными страницами. Легко убедиться, что Meta-теги на экране не отображаются (кроме TITLE).

Между тегамы можно ставить пробелы и переводы строки. HTML их не чувствует, а для чтения программного кода операции удобно разделять переводом строки.

3. Разметка Web-страницы с помощью таблиц

Создание чего-либо обычно начинается с проектирования. Разметку несложной станицы обычно создают с помощью таблиц. Таблицы в Web-программировании имеют более широкое применение, чем в текстовых документах. Создаваемая нами страница будет

¹ Исключение: название страницы выводится на экран в самой ве
Базовые параметры используются: браузерами при загрузке стра
машинами при помещении данной страницы в список ссылок, со
других целей.

² Кодировка UTF-8 является доминирующей в интернете. В этой
последовательностью от одного до четырёх байтов. Если без под
всегда начинается с 0, двухбайтная с 110, трёхбайтная с 1110, чет
устанавливают значения, соответствующие определённым симво
латинской раскладке клавиатуры ASCII, что даёт экономию по с
(поскольку большинство текстов интернета написано в этой раск
здесь экономии нет. Всего же кодировка UTF-8 способна закоди

Шапка	
Строка меню	
Иллюстрации	Текст
Нижняя часть	

иметь структуру, изображенную на рисунке (линии по умолчанию будут невидимы).

Для создания таблиц используются четыре тега:

- `<TABLE>` `</TABLE>` – создает таблицу;
- `<TR>` `</TR>` – создает новую строку;
- `<TH>` `</TH>` – создает ячейку заголовка;
- `<TD>` `</TD>` – создает простую ячейку.

По умолчанию в ячейку заголовка пишется текст полужирного начертания с выравниванием по центру. В обычной ячейке текст не выделяется и имеет выравнивание по левому краю.

Для начинающего создать подобную структуру сложно, можно легко запутаться. Поэтому будем снабжать создаваемый программный код **комментариями** – замечаниями, недоступными для просмотра в браузере. Комментарии вводятся так: `<!--Текст комментария -->`.

3.1. После `<BODY>` напишите:

```
<TABLE border=1></TABLE>3
```

Мы создали таблицу, внутри которой будет помещено все содержание Web-страницы. Надписью `border=1` задали толщину границ ячеек в 1 пиксель. Ширину таблицы задали в 100% экрана монитора. Не имеет значения, каков будет монитор у пользователя, таблица по ширине займет весь экран.

3.2. После `<TABLE>` напишите `<TR><TH></TH></TR>` – это будет шапка, где будет помещен красочный заголовок страницы. Обратите внимание на то, что в строке обязательно надо создать ячейку, даже если она одна. После `</TR>` создадим комментарий: `<!--Шапка -->`. Чтобы ячейка отобразилась в окне браузера, надо в нее вписать какое-либо слово (между открывающим и закрывающим тегом). Сделайте это.

3.3. Создайте еще одну строку, это будет строка меню. В ней создайте три ячейки заголовка, в каждую впишите что-либо. К строке создайте комментарий: `<!--Строка меню -->`, и посмотрите что получилось.

3.4. Создайте еще одну строку для ячеек (простых) иллюстраций и текста, с двумя ячейками, впишите в ячейки что-либо.

3.5. Завершите разметку страницы созданием строки с одной простой ячейкой для нижней части: `<TR><TD></TD></TR>`.

Впишите в ячейку что-либо, сделайте комментарий, посмотрите, что получилось.

По горизонтали получилось не то, что надо. Дело в том, что HTML стремится сделать все ячейки одинаковыми по ширине, причем к тому же примерно одинаковыми. Придать ячейкам разную ширину можно, но это сложный процесс. Проще создавать объединенные ячейки, занимающие несколько столбцов. Это делается командой **COLSPAN**.

3.6. Для ячеек `<TH>` и `<TD>` шапки и низа поставьте атрибут `colspan=3` (т.е. эти ячейки занимают три столбца каждая). Для текста поставьте `colspan=2`. Расположение ячеек должно стать таким, как на рисунке.

3.7. Команду `border=1` уберите, границы ячеек нужны были только для наглядности.

4. Форматирование текста средствами HTML

4.1. В Блокноте откройте файл **Казахская кухня**, выделите начало текста до слов «Деликатесы из конины», и с помощью буфера обмена перенесите этот текст в раздел, отведенный для основного текста (то, что там было написано – удалите). При просмотре видно, что форматирование текста оставляет желать лучшего. По умолчанию HTML выставляет левое выравнивание без абзацев и каких-либо иных параметров.

4.2. Вернемся в Notepad++ и установим абзацы. Для этого текст каждого имеющегося абзаца надо заключить между тегами `<P>` и `</P>`. Одновременно эти теги устанавливают шрифт обычного начертания. Для ускорения процесса можно их вводить через буфер обмена (сначала `<P>`, потом `</P>`).

³ Вообще-то теги `<Table>` и `<Tr>` теперь атрибутов не имеет. Но мы напишем устаревший атрибут временно.

В HTML можно установить 6 уровней заголовков, что логически упорядочивает данные на Web-странице. Кроме того, от умелого использования заголовков зависит эффективность нахождения страницы в интернете поисковыми роботами. Обычный текст соответствует 4 уровню; 1 уровень имеет самый крупный текст; 5 и 6 используются для сносок и примечаний. На странице рекомендуется использовать не более 3-х уровней.

4.3. Примените к заголовку второй уровень: `<H2></H2>`, к 1-му абзацу третий уровень: `<H3></H3>`, это будет подзаголовок. Теги заголовков надо ставить первыми, в самом начале и в конце абзаца.

4.4. Во второй абзац поставим теги `` и `` В браузере этот текст будет отмечен курсивом, а поисковая система расценит его как особенный.

Списки – простой и эффективный способ упорядочивания материала в логической последовательности.

4.5. **Нумерованный (упорядоченный) список** создается с помощью двух пар тегов: `` и `` устанавливают начало и конец нумерованного списка, а `` и `` отмечают отдельные элементы списка. Найдите в тексте какое-либо из перечислений и ограничьте его тегами ``, а каждое из слов – тегами ``. Посмотрите, получился ли список.

4.6. С помощью различных атрибутов можно установить нумерацию римскими цифрами и буквами, задать начало нумерации не с первого номера, пропустить номера в списке. Задайте `<OL type=a>`, посмотрите что получилось.

4.7. **Маркированный (неупорядоченный) список** создается тегами `` и ``, элементы списка ограничиваются теми же `` и ``. Создайте маркированный список на основе какого-либо из перечислений.

4.8. Тег `<blockquote>` указывает на то, что текст внутри является цитатой, он по иному форматируется. Примените к любому абзацу команду `<blockquote></blockquote>`.

4.9. Тег `<mark>` используется для выделения фрагмента текста. Поставьте их до и после слов подзаголовка: **его история, обычаи и традиции**. Эти слова будут отмечены аналогично выделению текста в тетради маркером. Причём, в отличии от команд CSS, это выделение будет учтено поисковыми системами и другими соответствующими программными средствами.⁴

Средства CSS логической нагрузки не несут, они предназначены только для человека.

5. Форматирование текста средствами CSS

Прежде всего, требуется создать файл со стилями CSS.

5.1. Включаем **Notepad++**, создаём новый файл, устанавливаем синтаксис CSS, сохраняем файл под именем **styles.css**.

5.2. В файл записываем стили:

```
H2{font-family: Verdana; color: blue; text-align: center}
H3{font-family: Verdana; text-align: center}
P{font-family: Arial; text-align: justify}
```

5.3. Далее переходим в **Index**, и в разделе **Head** пишем тег связи с файлом стилей:

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

Функции атрибутов: `rel` отражает цель связи с файлом, `href` – адрес, `type` – тип файла.

Однако при выводе страницы в браузер становится заметно, что она нуждается в корректировке.

5.4. Уберите тег из строк с заголовками `<H2>` и `<H3>`, тогда они приобретут центральное выравнивание.

⁴ В HTML имеются команды логического форматирования текста. В результате их применения текст для посетителя страницы становится жирным, курсивным, подчёркнутым и пр. А для поисковых систем и другого специального ПО, он тоже приобретает другое значение: становится важным, особенным, цитатой, программным кодом и др. Причём для программы градаций больше, чем для посетителя. Так, курсивный текст может иметь 4 логических значения, что реализуется разными тегами.

«Цитату» можно оставить прежней, она будет сильнее отличаться от основного текста.

5.5. А списки нуждаются в задании стиля. Нужно распространить стиль основного текста **P** на **Ol** и **Ul**:

```
P,Ol,Ul{font-family: Calibri; text-align: justify}
```

5.6. Но теперь фразы до и после списка надо ограничить тегом **<P></P>**. Иначе стиль этих фраз исчезает.

В процессе вёрстки постоянно надо просматривать текст и вносить нужные корректировки.

Продemonстрируем некоторые расширенные возможности оформления текста с помощью CSS. Для этого применим внутритекстовую запись CSS:

```
<Span style= "свойство: значение; свойство: значение;">участок текста</Span>
```

5.6. Найдите текст: **из овощей, фруктов, рыбы, морских продуктов**, Сделаем его разноцветным и пожирнее:

```
<Span style= "font-weight: 600; color: green">овощей, фруктов,</Span>
```

```
<Span style= "font-weight: 600; color: blue">рыбы, морских продуктов,</Span>
```

```
<Span style= "font-weight: 600; color: orange">печеными, мучными изделиями и сладостями.</Span>
```

5.7. А фразу: **Мясо – основа большинства блюд**, сделаем очень жирной, красной и подчеркнём:

```
<Span style= "color: red; font-weight: 900; text-decoration-line: underline">Мясо – основа большинства блюд,</Span>
```

6. Добавление графики

В интернете применяются 3 растровых графических формата: JPEG, GIF и PNG. Используется также векторная SVG-графика и анимация, создаваемая с помощью векторных редакторов.⁵

5.1. Скопируйте рисунки **KazKuh1**, **KazKuh2**, **KazKuh3**, **KazKuh4** в папку вашего сайта. Теперь вставьте рисунки командами:

```
<IMG src=Images/KazKuh1.gif>
```

.....

```
<IMG src=Images/KazKuh4.jpg.>
```

Должны появиться иллюстрации.

*Src – обязательный атрибут, без него неясно, какой файл надо вставлять.*⁶

5.2. Иллюстрации лучше расположить вертикально, а это удобнее всего сделать, установив для каждой перевод строки непарным тегом **
**.

А далее к рисункам тоже надо будет применить свои стили CSS.

5.3. Раскрываем файл стилей и пишем:

```
IMG{width: 280px; border: 1px solid black; margin: 5px}
```

В данном случае мы задали следующие стили:

- ширина иллюстрации – 280 пикселей, вне зависимости от реальных размеров рисунка;
- рамка толщиной 1 пиксель, сплошная линия, черного цвета;
- установлено расстояние от иллюстрации до других объектов в 5 пикселей.

⁵Используемая в Сети растровая графика проходит специальную подготовку в графических редакторах, чаще всего в Adobe Photoshop. В Photoshop надо применить команду Файл → Сохранить для Web. В открывшемся окне можно менять различные параметры, в том числе формат. Без этой подготовки иллюстрации в Сеть не попадут, даже если будут вставлены в программный код. В процессе подготовки происходит уменьшение размеров графического файла за счет снижения разрешения и глубины цветности.

⁶ На странице можно разместить рисунки из любой папки компьютера, а также те, которые находятся в Сети. Для этого надо правильно прописать путь доступа к ним. Но при указании пути доступа могут встретиться различные трудности, о некоторых разработчик сайта может и не знать. Поэтому лучше все рисунки перенести в папку Image, размещенную вместе с другими компонентами сайта.

5.9. В самой верхней ячейке таблицы, в шапке, в качестве заголовка вставим рисунок **TextKazKuh**.

Заголовок получился неудачным. Размер его слишком мал, и рамка здесь лишняя. Для корректировки можно применить другой класс стиля **IMG**.

5.10. В файле стилей сделайте запись:

```
.logo{width: 100%; border: 0px}
```

А в теге, загружающем **TextKazKuh**, поставьте ещё один атрибут: **class="logo"**. После этого рисунок логотипа займёт всю страницу, рамка исчезнет.

Для одного и того же тега можно создать классы стилей, в которых вносятся изменения. В классе меняются только те свойства, которые в нём поименованы. Остальные свойства исходного стиля остаются без изменений.

5.11. Поскольку поисковые системы с иллюстрациями не работают, необходимо дать текстовую альтернативу рисунку, внесите ещё один атрибут: **alt="Казахская кухня"**. Альтернативный текст заголовка будет прочитан, что повысит рейтинг материала в списке выдачи.

6. Создание фона

В HTML 5 фон можно задать как для всей страницы, так и для различных участков, причём только средствами CSS. В более ранних версиях для этой цели использовались атрибуты к тегам.

Количество возможных цветов соответствует цветовой модели RGB, цвета задаются двумя шестнадцатеричными цифрами, перед числовым кодом всегда ставится знак #.

16 цветов имеют собственные имена, их можно задавать по имени. Например: **red** – красный (**#FF0000**), **blue** – синий (**#0000FF**), **silver** – серебряный, (**#C0C0C0**), **olive** – оливковый (**#808000**). В скобках приведены шестнадцатеричные коды этих цветов.

6.1. Создадим на странице светло-голубой фон:

```
Body {background-color:#C0FFFF}.
```

А часть страницы украсим фоновым рисунком. Чтобы не мешать чтению, фоновый рисунок должен быть светлым, малоконтрастным, не очень сложным. Этим условиям отвечает файл **ForFon1.jpg**.

6.2. Скопируйте фоновый рисунок в папку с графикой, а затем создайте стиль:

```
TD {background-image: url(Images/ForFon1.jpg)}
```

Фоновый рисунок хорошо заметен на основном тексте, в других ячейках он виден хуже.

Интересно, что если фоновый рисунок по размеру меньше Web-страницы, то по умолчанию он копируется на странице в виде плитки. Что и происходит в данном случае. При этом чем меньше размер рисунка, тем меньший вклад он вносит в размер отправляемой страницы. Рисунок загружается один раз, а затем следует команда его тиражирования.

7. Специальные символы

Специальных символов в HTML 5 довольно много, их сотни. К ним относятся малораспространенные знаки пунктуации, математические символы, значки, маркеры, стрелки, знаки валют, зодиака, греческий алфавит и др. То есть, их применение разнообразно.

Команды специальных символов всегда начинаются с **&**, например: **@**, **&int**, **☦**. Для этих команд угловые скобки **<>** не ставятся.⁷

7.1. При импорте текста HTML игнорирует множественные пробелы, между всеми словами остается только один пробел. Команда ** ** вставляет **неразрывные пробелы**, каждый из которых в обязательном порядке присутствует на Web-странице.

⁷ Команды могут иметь именной и числовой код. Именной код присвоен командам вывода более важных символов, например, **>** (>, больше), **¢** (€, евро), **©** (©, знак авторского права). Числовой код имеет любая команда вывода символа, включая буквы различных алфавитов, цифры, служебные знаки и пр. То есть числовой код позволяет вывести любой символ используемой кодировки. Числовой код записывается в следующем виде: **&#номер символа**. Есть ещё CSS код, очевидно, он используется в стилях CSS.

В начале абзаца, отмеченного как цитата (<Blokquote>) поставьте подряд 5 раз символ ** sp**. Получится абзацный отступ первой строки.

7.2. Впрочем, каждый раз вводить таким образом отступ первой строки неразумно. Лучше его прописать в виде стиля. В данном случае придётся разделить абзацы и списки на два стиля. Стил **Ol,Ui** оставьте без изменения, а стил **P** дополните записью: **text-indent: 30px**. В каждой строке основного текста должен появиться абзацный отступ.

7.3. В любом месте страницы поставьте знаки <>, **Br**, **&abcd**. Все эти неправильно введенные команды HTML отобразятся в виде текста.

Все ошибочно введенные команды HTML и других средств создания сайтов отображаются на странице в виде текста.

7.4. В нижней строке страницы наберите: **©Миреев В.А.**, а имеющийся там текст удалите. Отобразится копирайт автора этого текста.

7.5. Введем символ-иллюстрацию из специального шрифта. В файле стилей зададим особо крупный шрифт: **H1 {text-align: center; font-size: 50px}** А в ячейке над рисунками создадим строку текста **<H1>🍖</H1>** (после неё обязательно надо поставить **
**). В результате отобразится иконка в виде мяса на кости.

Среди более чем 2 миллионов символов UTF-8 можно найти спецсимволы на любую тематику. В том числе нашлась серия кулинарных картинок-эмоджи.

Таким образом, на страницу можно загрузить и рисунок, который выглядит как текст (см. пункт 5.9.), и текст в виде рисунка.

8. Создание гиперссылок

Гиперссылки являются важнейшим элементом Web-страниц, они обеспечивают логическую структуру всемирной паутины WWW. Без гиперссылок интернет в его нынешнем виде невозможен.

Гиперссылка создается парными тегами **<A>** и ****, между которыми заключен текст или картинка, служащие гиперссылкой. Адрес, на который ведет гиперссылка, задается обязательным атрибутом **href: **.⁸

8.1. В строке меню создадим гиперссылки на рецепты нескольких блюд. Для этого в каждой ячейке меню напишем соответствующие ссылки: **<A>Рецепт 1**, **<A>Рецепт 2**, **<A>Рецепт 3** (вместо того, что там было). Но пока эти гиперссылки никуда не ведут, надо создать соответствующие страницы.

8.2. Скопируйте файл **Kazyi.html** в ту же папку, где находится ваша главная страница, после чего для ссылки **Рецепт 1** поставьте атрибут адреса гиперссылки: ****. Надпись **Рецепт 1** должна приобрести синий цвет и стать подчеркнутой (так по умолчанию обозначаются текстовые гиперссылки).

Но каждая страница должна тоже иметь ссылки куда-то, иначе она будет тупиком, откуда не выбраться. Для страницы **Kazyi.html** сделаем обратную ссылку на **Index.html**.

8.3. В конце текста сделайте новую строку, придайте ей центральное выравнивание, и напишите: **НА ГЛАВНУЮ**. Поставьте также тег выделения текста ****.⁹ Затем поставьте тег гиперссылки: ****. Не забудьте, что после надписи должны идти завершающие теги, в порядке, обратном по отношению к открывающим.

8.4. Сделаем для этой страницы более подходящий цвет, красный, заданием стиля:

⁸ Гиперссылкой может служить:

- участок текста любого размера, от буквы до нескольких абзацев;
- баннер – графический или анимированный объект, ставший гиперссылкой.

Гиперссылка может вести на:

- закладку на текущей Web-странице;
- другую Web-страницу данного сайта, как правило на том же носителе;
- Web-страницу, находящуюся в интернете, а также в локальной сети;
- мультимедийный ресурс в сети или на данном компьютере;
- документ FTP сервера;
- адрес электронной почты;
- сценарий на языке Java Script.

Есть и другие варианты гиперссылок.

⁹ Для пользователя этот текст имеет жирное начертание, поисковик его оценивает, как особо важный.

.kazyi{background-color:#FFA050}

А на странице в теге `<body>` надо поставить атрибут `class="kazyi"`.

- 8.5. Создайте еще одну страницу-рецепт для любого выбранного вами блюда. Раскройте код страницы **Казы** и сотрите всё содержание, оставив форматирование. Раскройте файл **Казахская кухня** и скопируйте оттуда любой рецепт вместо стертого текста.
- 8.6. Сохраните файл в свою папку под другим именем в виде страницы html, Поменяйте цвет аналогично п. 8.4. Заголовок, ссылка на главную страницу, название в `<title>` должны присутствовать.
- 8.7. Пропишите ссылку **Рецепт 2** на созданный файл. Проверьте, работает ли ссылка.
- 8.8. Аналогично создайте третью страницу для выбранного блюда. Ссылкой на нее будет надпись **Рецепт 3**.
- 8.9. С помощью атрибута `target` можно поменять способ открытия документа, указанного в гиперссылке. Добавьте в тег ``, тогда при открытии внешнего документа текущая страница из браузера не исчезает, и можно работать с двумя страницами.
- 8.10. Создадим гиперссылку – графический баннер с рисунка пельменей на страницу **Pelmeni**. Сначала эту страницу скопируйте к себе в папку. Затем найдите в HTML-коде страницы тег ``, управляющий вставкой рисунка пельменей и ограничьте его тегами создания гиперссылки `<A ...> `.

Рамка иллюстрации должна поменять свой цвет на сиреневый. По умолчанию все баннеры имеют такую рамку. Курсор на рисунке приобретает вид руки, при щелчке должна открыться страница **Пельмени**.

- 8.11. Однако с этой страницы никуда не ведет гиперссылка, получился тупик. Откройте страницу любого рецепта, вызовите ее код и скопируйте оттуда надпись **НА ГЛАВНУЮ** со всеми управляющими тегами.

Бывает так, что размер изображения велик и его включение в Web-страницу существенно замедлит загрузку документа. Тогда на этот файл создают гиперссылку.

- 8.12. Создайте ссылку на рисунок **Baursaki.jpg**. Скопируйте рисунок в папку Image, после чего создайте гиперссылку из слова: **баурсаками** на рисунок этого кушанья. Атрибуты те же, что и в ранее созданных гиперссылках. При щелчке по слову должен открываться рисунок. Чтобы вызванный рисунок служил иллюстрацией к тексту, дополните тег атрибутом `target=blank`.

Если страница длинная, то в ней можно создать систему навигации с переходом от одного участка к другому. Для создания закладки то место, куда ведет гиперссылка следует обозначить в виде закладки с атрибутом `id="имя закладки"`. Имя закладки должно начинаться с латинской буквы. Сама гиперссылка пишется обычным образом, но добавляется значок #: ``.

Создадим ссылку с начала текста на автора рецептов казахской кухни (тег `<Address>` браузер отображает курсивом, а поисковик читает как контактные данные автора, владельца документа).

- 8.13. В самом конце страницы заключите ©**В.А.Миреев** между тегами `<Address id="chapter">` и `</Address>`. В самом начале заключите **ВВЕДЕНИЕ** между тегами `` и ``. При щелчке по слову **ВВЕДЕНИЕ** должно появляться окончание страницы.
- 8.14. Web-сайт обычно имеет ссылки на внешние ресурсы интернета. Сделайте ссылку на главную страницу поисковой системы **Яндекс**. Для этого скопируйте в папку **Images** значок поисковика **Yandex.gif** и поместите его в конце страницы **Index**, после надписи ©**В.А.Миреев**.
- 8.15. Значок получился слишком крупным. Создайте для него другой класс стиля **IMG** с меньшей шириной. Снабдите этот значок гиперссылкой на **Яндекс**: ``.

Если компьютер подключен к интернету, то при щелчке по значку, появится страница поисковика **Яндекс**. Для возвращения нажмите в меню браузера кнопку возврата.

Можно сделать ссылку на адрес электронной почты. Она запускает окно создания письма по адресу, указанному в этой ссылке.

8.16. После значка **Yandex** наберите **Написать письмо администратору** Поставьте к этой надписи гиперссылку ``. При щелчке запустится приложение для создания письма.

9.2. Иные материалы. Планы самостоятельной работы.

Самостоятельная работа №1 Форматирование

1. Форматирование текста по образцу, используя соответствующие теги и стили.
2. Форматирование заголовка, текста и рисунков собственного содержания, но в соответствии с образцом и с заданными параметрами.
3. Создание гипертекстовой структуры страниц с ссылками на главную страницу в виде текста и баннеров, а с главной страницы ссылки оформляются в виде разных участков одной иллюстрации.
4. Создание страницы в виде таблицы, каждая ячейка которой оформлена либо как текстовая входящая ссылка с главной страницы, либо как ссылка-баннер на главную страницу.
5. Составить таблицу с различным размером ячеек, рамками, фоном, фоновым рисунком.
6. Сделать страницу с бегущими строками заданного шрифта и фона ячеек, затем с бегущими картинками внутри ячеек, затем с символами, произвольно двигающимися внутри ячеек.

Самостоятельная работа № 2 Пользовательские формы

Создать страницу, предназначенную для сбора сведений о посетителях вашего сайта. Сведения должны пересылаться по электронной почте на ваш адрес и никак не кодироваться.

Применить следующие элементы пользовательских форм:

- ✓ текстовая строка;
- ✓ поле для ввода пароля;
- ✓ поле выбора локального файла для загрузки на Web-сервер;
- ✓ опция выбора нескольких вариантов данных;
- ✓ опция переключения между несколькими вариантами данных;
- ✓ текстовое поле;
- ✓ ниспадающее меню;
- ✓ список наименований;
- ✓ кнопка отправления данных;
- ✓ кнопка сброса данных.

Самостоятельная работа №3. Реализовать игру Пятнашки средствами javascript.

Самостоятельная работа №4. Реализовать форум средствами PHP.

Пример задания на самостоятельную работу.

Самостоятельная работа №1 Форматирование

Задание 1

Создайте в текстовом редакторе Блокнот веб-страничку в соответствии с образцом и сохраните ее в папке, названной вашей фамилией, под именем Text.html

В строке заголовка браузера при просмотре должна быть фамилия исполнителя данного задания. Стихотворение напечатано зеленым цветом, а слово "ласточка" должно остаться черным, строки стихотворения отделены символом разрыва строки

Стремитесь к наименьшему объему файла – за лишние теги и атрибуты оценка будет снижаться.

Это заголовок 1

Это заголовок 2

Это заголовок 3

Дни недели (рабочие):

- Понедельник
- Вторник
- Среда
- Четверг
 1. первый урок
 2. второй урок
 3. перемена
- Пятница

Arial Courier Comic Sans MS Tahoma

Все это шрифт размера 5

Мы умеем использовать ^{верхний} и _{нижний} регистр (шрифт Tahoma размер 3)

Травка зеленеет,
Солнышко блестит,
Ласточка с весною
В сени к нам летит.

Задание 2

Создайте веб-страничку по прилагаемому образцу. в заголовке которой укажите "Размещение рисунков", и сохраните ее в вашей папке под именем Pictures.html. Используя подходящие рисунки (они также должны быть сохранены непосредственно в вашей папке либо в специально созданном в ней каталоге Image), создайте страничку, представленную на рисунке.

Рисунок в верхней части страницы состоит из трех одинаковых картинок с границей по 5 пкс, размещенных по центру. Заголовок выделен красным. Вторая картинка размещена два раза, но имеет при этом различные размеры и свободное пространство со всех сторон по 20 пкс. Основной текст разделяется тегом разрыва строки, применено выравнивание по ширине.

Добавьте на страничку еще 2-3 подходящих рисунка, разместив их по своему усмотрению.



История земли Вятской

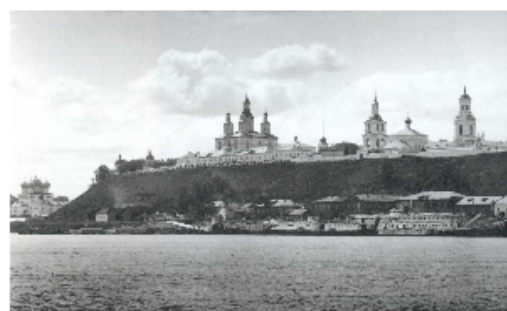
Вятская земля имеет богатую историю. Она стала заселяться еще в глубокой древности, очевидно, уже в верхнепалеолитическое время (50-15 тыс. лет назад). На территории области известны археологические памятники эпохи мезолита, неолита, бронзового века. В VII в. до н.э. в бассейне Вятки начался железный век. Ранний железный век здесь представлен памятниками ананьинской культуры. Ананьинцы принадлежали к финно-угорской этнической группе. Есть

предположение, что они назывались тиссагетами, о которых упоминает древнегреческий историк Геродот, помещавший их к северо-востоку от скифов и сарматов.

В конце XII-нач.XIII вв. в бассейн Вятки стали проникать русские, они селились на свободных землях среди удмуртов и марийцев.

Вятка впервые упоминается в летописях под 1374 годом в связи с походом новгородских ушкуйников на Волжскую Болгарию, входившую в то время в состав Золотой Орды. "В лето 6882 (1374) идоша на низ рекою Вяткою ушкунцы разбойницы, 90 ушкунцев, и пограбиша Вятку и шедше взяша Болгары".

В 70-е гг. XIV в. Вятская земля входила в состав Нижегородского княжества. В 1393 г. это княжество было присоединено к Москве. Нижегородские князья после долгой борьбы вынуждены были покориться и получили в удел Вятскую землю.



Внимание! Все рисунки необходимо оптимизировать (например, в PhotoShop) до объема не более 20-30 килобайт!

Задание 3

0

214

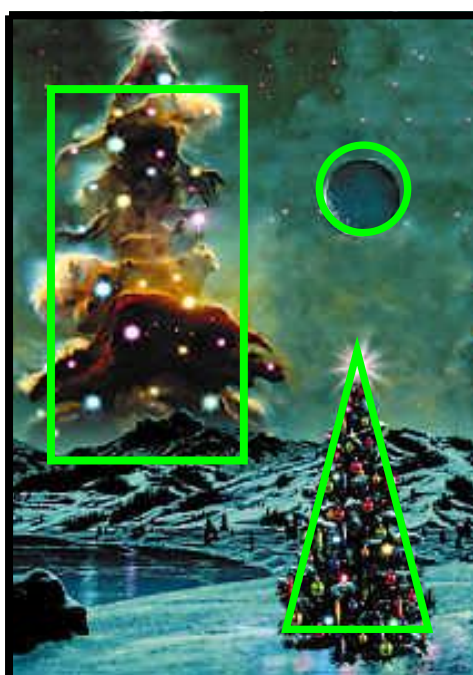
1. Создайте в своей папке на сервере главную веб-страничку **Index.html** еще три дополнительных странички с именами **Page1.html**, **Page2.html**, **Page3.html**, поместите их в специально созданную папку **Pages**.

2. Разместите на каждой дополнительной странице по одной картинке (любой) и текст «Возврат на главную страницу». Картинки при необходимости оптимизируйте и разместите в папке **Image**.

3. Превратите эти картинки и текст в гиперссылку на главную страницу – **Index.html**.

4. На главной странице расположите ссылки-картинки и текстовые ссылки на дополнительные странички.

5. Также на главной странице



300

разместите карту на основе картинки M16.jpg (ее размер 214x300 пкс). Выделенные на рисунке области сделайте гиперссылками все на те же три новых странички.

HTML-код создаваемой карты может выглядеть примерно так:




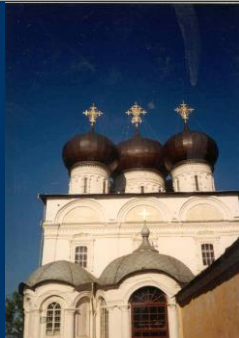
```
<IMG SRC="Image/M16.jpg" WIDTH="214" HEIGHT="300"
BORDER=0 USEMAP="#mymap">
<MAP NAME="mymap">
<AREA SHAPE="circle" COORDS="150,75,20"
HREF="Pages/Page1.html">
<AREA SHAPE="poly" COORDS="150,150,200,280,120,280"
HREF="Pages/Page2.html">
<AREA SHAPE="rect" COORDS="20,30,100,200"
HREF="Pages/Page3.html">
</MAP>
```

6. Оформите страницы, используя изменение цвета фона, гиперссылок. Примените фоновые рисунки. Наполните страницы содержанием.

Задание 4

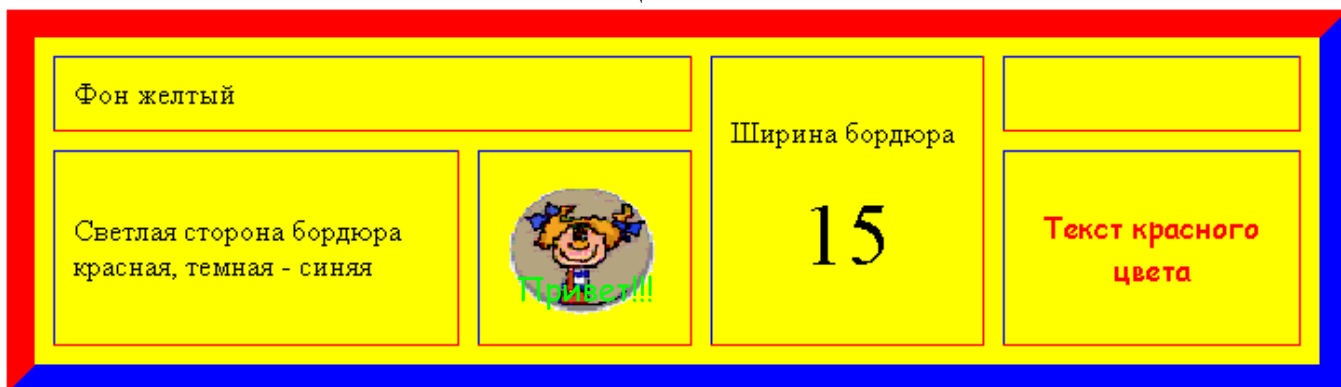
Эти таблицы создайте на новой страничке Table.html. Сделайте ссылку любого типа на эту страницу с Index.html. Обратную гиперссылку сделайте из картинки, вставленной во вторую табличку.

Таблица №1

<p>Ротонда (архитектор А.Л. Витберг)</p>		<p>Александро- Невский собор (архитектор А.Л. Витберг)</p>	
<p>Жемчужины старой Вятки</p>			
	<p>Свято- Серафимовская церковь</p>		<p>Успенский собор Трифонова монастыря</p>

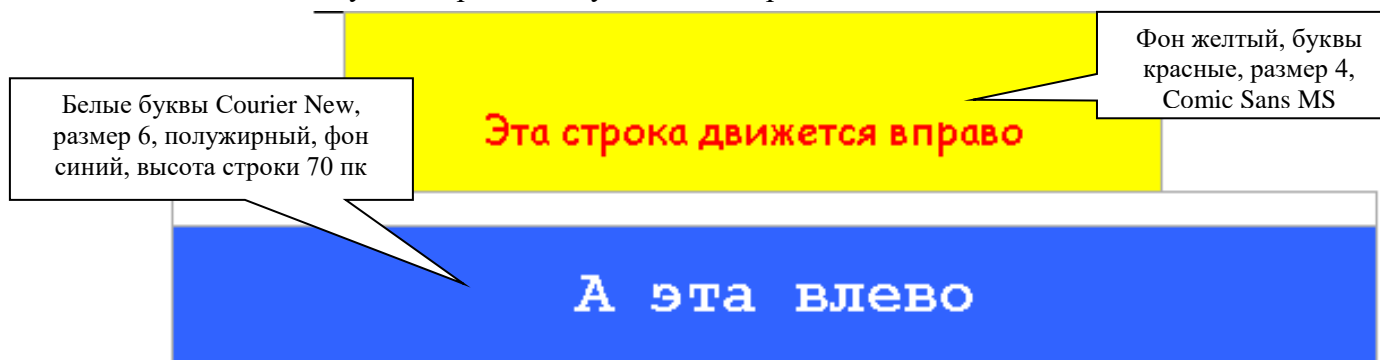
Каждый архитектурный памятник расположен в ячейке индивидуального фоновом цвета, тот же цвет имеет ячейка с названием памятника (цвета подберите по своему вкусу). Гарнитуру, цвет и начертание текста "Жемчужины старой Вятки" также подберите самостоятельно. Картинку во второй таблице вставьте как фоновый рисунок, чтобы разместить на его фоне текст.

Таблица №2



Задание 5

1. Создайте в папке, где хранятся Ваши предыдущие работы, веб-страничку Marquee.html.
2. Сделайте бегущие строки следующего содержания:



3. А эти бегущие строки, размещенные в ячейках невидимой таблицы, вместо текста содержат картинки, которые двигаются, «отражаясь» от краев окна и меняя направление движения, причем с разными скоростями. Лягушка двигается плавно, а тыква – скачками.



4. А эти буквы "живые" – они имеют размер 7, разный цвет, начертание и двигаются в разные стороны, не "уползая" за края ячеек.

		И				!
	Р		В		Т	
П				Е		

5. Дополните страничку несколькими горизонтальными линиями, употребив при этом различные атрибуты тега <HR>

АННОТАЦИЯ ДИСЦИПЛИНЫ

Цель дисциплины: «Web-программирование» имеет целью теоретическое и практическое освоение методов и технологий, используемых в дизайне и верстке современных Web-документов.

Задачи: выработка у студентов системного подхода к решению задач дизайна и верстки интернет ресурсов, приобретение практических навыков построения различных web-систем на основе выбора наименее трудоемкой и, вместе с тем, адекватной поставленной задачи методологии разработки.

В результате освоения дисциплины обучающийся должен:

Знать: принципы работы и основные протоколы сети интернет, языки программирования и разметки, подходы, используемые при создании сайтов, методы тестирования и адаптации программных компонентов интернет-ресурсов, технологии поисковой оптимизации и продвижения сайтов.

Уметь: определять методы решения задач предметной области, с средствами сети интернет, разрабатывать структуру базы данных, адаптированную к использованию в интернет проектах, тестировать компоненты программного обеспечения интернет-ресурсов, оценивать эффективность использования конкретного языка и технологии при решении поставленных задач.

Владеть: навыками работы с инструментальными средствами разработки в сети интернет (HTML, CSS), языками PHP, MySQL и JavaScript, методами оптимизации, построения и продвижения сайтов. практическими навыками разработки и тестирования программных компонентов интернет-ресурсов.