

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Усынин Максим Валерьевич
Должность: Ректор
Дата подписания: 28.04.2025 16:47:29
Уникальный программный ключ:
f498e59e83f65dd7c3ce7bb8a25cbbabb33ebc58

**Частное образовательное учреждение высшего образования
«Международный Институт Дизайна и Сервиса»
(ЧОУВО МИДиС)**

Кафедра математики и информатики

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ
И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ
ПО ОБЩЕОБРАЗОВАТЕЛЬНОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ
СОО 03.01 ОСНОВЫ ПРОФЕССИОНАЛЬНОГО РАЗВИТИЯ**

Специальность: 09.02.07 Информационные системы и программирование
Направленность (профиль): Разработка веб и мультимедийных приложений
Квалификация выпускника: Разработчик веб и мультимедийных приложений
Уровень базового образования обучающихся: Основное общее образование

Форма обучения: Очная

Год набора: 2025

Автор – составитель: Хаятова Л.Р.

Челябинск 2025

СОДЕРЖАНИЕ

1. Паспорт фонда оценочных средств	3
1.1. Область применения	3
1.2. Планируемые результаты	3
1.3. Показатели оценки результатов обучения	4
2. Задания для контроля и оценки результатов	14
3. Критерии оценивания.....	38

1. Паспорт фонда оценочных средств

1.1. Область применения

Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся (далее – Фонд оценочных средств) предназначен для проверки результатов освоения общеобразовательного учебного предмета СОО.03.01 Основы профессионального развития основной профессиональной образовательной программы среднего профессионального образования (далее – образовательная программа) по специальности 09.02.07 Информационные системы и программирование, направленность Разработка веб и мобильных приложений.

Форма промежуточной аттестации по семестрам.

Семестр	Форма аттестации
первый	Контрольная работа
второй	Зачет с оценкой

1.2. Планируемые результаты

В результате освоения программы общеобразовательного учебного предмета СОО.03.01 Физика учитываются планируемые результаты освоения образовательной программы:

Код и наименование формируемых компетенций	Планируемые результаты освоения дисциплины	
	Общие	Дисциплинарные
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	<p>В части трудового воспитания:</p> <ul style="list-style-type: none"> - готовность к труду, осознание ценности мастерства, трудолюбие; - готовность к активной деятельности технологической и социальной направленности, способность инициировать, планировать и самостоятельно выполнять такую деятельность; - интерес к различным сферам профессиональной деятельности, Владение универсальными учебными познавательными действиями: <p>а) базовые логические действия:</p> <ul style="list-style-type: none"> - самостоятельно формулировать и актуализировать проблему, рассматривать ее всесторонне; -устанавливать существенный признак или основания для сравнения, классификации и обобщения; - определять цели деятельности, задавать параметры и критерии их достижения; - выявлять закономерности и противоречия в рассматриваемых явлениях; 	<ul style="list-style-type: none"> - уметь критически анализировать данные для решения познавательной задачи; - уметь устанавливать причинно-следственные связи при работе с информацией различных типов; - уметь выбирать способы решения задач, адекватные имеющимся ресурсам.

	<ul style="list-style-type: none"> - вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности; - развивать креативное мышление при решении жизненных проблем б) базовые исследовательские действия: <ul style="list-style-type: none"> - владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем; выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу ее решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения; - анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях; - уметь переносить знания в познавательную и практическую области жизнедеятельности; - уметь интегрировать знания из разных предметных областей; - выдвигать новые идеи, предлагать оригинальные подходы и решения; - способность их использования в познавательной и социальной практике. 	
<p>ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности</p>	<p>В области ценности научного познания:</p> <ul style="list-style-type: none"> - сформированность мировоззрения, соответствующего современному уровню развития науки и общественной практики, основанного на диалоге культур, способствующего осознанию своего места в поликультурном мире; - совершенствование языковой и читательской культуры как средства взаимодействия между людьми и познания мира; 	<ul style="list-style-type: none"> - уметь осуществлять с соблюдением правил информационной безопасности поиск необходимой информации в справочной литературе, сети Интернет, средствах массовой информации для решения познавательных задач; оценивать полноту и достоверность информации; - уметь объяснять критерии поиска и отбора

	<ul style="list-style-type: none"> - осознание ценности научной деятельности, готовность осуществлять проектную и исследовательскую деятельность индивидуально и в группе; - Овладение универсальными учебными познавательными действиями: в) работа с информацией: <ul style="list-style-type: none"> - владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления; - создавать тексты в различных форматах с учетом назначения информации и целевой аудитории, выбирая оптимальную форму представления и визуализации; - оценивать достоверность, легитимность информации, ее соответствие правовым и морально-этическим нормам; - использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности; - владеть навыками распознавания и защиты информации, информационной безопасности личности. 	<p>информации, оценки полученных данных; учитывать при работе специфику современных источников социальной, экономической и личной информации; приобретение опыта осуществления учебно-исследовательской и практической деятельности</p>
<p>ОК 04. Эффективно взаимодействовать и работать в коллективе и команде</p>	<ul style="list-style-type: none"> - готовность и способность к образованию и саморазвитию, самостоятельности и самоопределению; - овладение навыками учебно-исследовательской, проектной и социальной деятельности; Овладение универсальными коммуникативными действиями: б) совместная деятельность: 	<ul style="list-style-type: none"> - приобретать опыт осуществления проектной деятельности в форме участия в подготовке учебных бизнес-проектов, в том числе – на региональном материале; - приобретать опыт взаимодействия с людьми

	<ul style="list-style-type: none"> - понимать и использовать преимущества командной и индивидуальной работы; - принимать цели совместной деятельности, организовывать и координировать действия по ее достижению: составлять план действий, распределять роли с учетом мнений участников обсуждать результаты совместной работы; - координировать и выполнять работу в условиях реального, виртуального и комбинированного взаимодействия; - осуществлять позитивное стратегическое поведение в различных ситуациях, проявлять творчество и воображение, быть инициативным <p>Овладение универсальными регулятивными действиями:</p> <p>г) принятие себя и других людей:</p> <ul style="list-style-type: none"> - принимать мотивы и аргументы других людей при анализе результатов деятельности; - признавать свое право и право других людей на ошибки; <p>развивать способность понимать мир с позиции другого человека.</p>	<p>на основе ценностей современного российского общества: идеалов гуманизма, демократии, мира и взаимопонимания между народами, людьми разных культур.</p>
--	--	--

Код и формулировка компетенции	Показатели освоения компетенции
<p>ПК 8.1. Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика</p>	<p>Практический опыт: Разрабатывать эскизы веб-приложения. Разрабатывать схемы интерфейса веб-приложения. Разрабатывать прототип дизайна веб-приложения. Разрабатывать дизайн веб-приложений в соответствии со стандартами и требованиями заказчика. Разрабатывать интерфейс пользователя для веб-приложений с использованием современных стандартов.</p> <p>Умения: Создавать дизайн с применением промежуточных эскизов, прототипов, требований к эргономике и технической эстетике. Учитывать существующие правила корпоративного стиля. Придерживаться оригинальной концепции дизайна проекта и улучшать его визуальную привлекательность. Разрабатывать интерфейс пользователя для веб-приложений с использованием современных стандартов.</p>

	<p>Знания: Нормы и правила выбора стилистических решений. Способы создания эскиза, схем интерфейса и прототипа дизайна по предоставляемым инструкциям и спецификациям. Правила поддержания фирменного стиля, бренда и стиливых инструкций. Стандарт UIX - UI & UX Design. Инструменты для разработки эскизов, схем интерфейсов и прототипа дизайна веб-приложений.</p>
--	--

Личностные результаты реализации программы воспитания

Личностные результаты реализации программы воспитания	Код личностных результатов реализации программы воспитания
Осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности;	ЛР 7
Демонстрирующий умение эффективно взаимодействовать в команде, вести диалог, в том числе с использованием средств коммуникации	ЛР 13
Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения (в ред. Приказа Минпросвещения России от 17.12.2020 N 747)	ЛР 17
Активно применять полученные знания на практике.	ЛР 22

1.2. Планируемые результаты освоения компетенций

В результате освоения программы общеобразовательной учебной дисциплины СОО 03.01 основы профессионального развития учитываются планируемые результаты освоения общих (ОК) и профессиональных (ПК) компетенций.

Код компетенций	Содержание компетенции	Планируемые результаты освоения компетенций
<p>ОК 01 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам</p>	<p>Умения: распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы; владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)</p>	<p>Знания: актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности</p>
<p>ОК 02 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности</p>	<p>Умения: определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска, применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение; использовать различные цифровые средства для решения профессиональных задач</p>	<p>Знания: номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации, современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности в том числе с использованием цифровых средств</p>
<p>ОК 04 Эффективно взаимодействовать и работать в коллективе и команде</p>	<p>Умения: организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности</p>	<p>Знания: психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности</p>

<p>ПК 8.1. Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика</p>	<p>Умения: Создавать дизайн с применением промежуточных эскизов, прототипов, требований к эргономике и технической эстетике. Учитывать существующие правила корпоративного стиля. Придерживаться оригинальной концепции дизайна проекта и улучшать его визуальную привлекательность. Разрабатывать интерфейс пользователя для веб-приложений с использованием современных стандартов.</p>	<p>Знания: Нормы и правила выбора стилистических решений. Способы создания эскиза, схем интерфейса и прототипа дизайна по предоставляемым инструкциям и спецификациям. Правила поддержания фирменного стиля, бренда и стилевых инструкций. Стандарт UIX - UI & UX Design. Инструменты для разработки эскизов, схем интерфейсов и прототипа дизайна веб-приложений.</p>
--	--	--

1.3. Показатели оценки результатов обучения

Содержание общеобразовательной учебной дисциплины	Результаты обучения (ОК, ПК, ЛР)	Вид контроля	Наименование оценочного средства/форма контроля
1 семестр			
2. Языки для верстки сайтов	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
3. Структура html-документа. Теги HTML	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
4. Инструменты для создания HTML кода	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
5. Visual Studio Code. Плагины и расширения	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
6. Разметка текста. Ссылки и изображения. Таблицы. Формы	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
7. Основы CSS	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
8. Оформление текста. Подключение шрифтов	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
9. Базовая стилизация	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
10. Валидность кода	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
11. Верстка первого сайта	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ

12. Верстка сайта с боковой панелью	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
13. Верстка многостраничного сайта	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
Темы 2-13	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Промежуточный	Контрольная работа
2 семестр			
1. Основные свойства flex-контейнеров и flex-элементов	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
2. Поток вывода элементов на страницу	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
3. Адаптивная верстка сайтов. Синтаксис написания медиа запросов	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
4. Построение сеток. Фреймворк Bootstrap	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
6. Адаптивная верстка сайта	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Текущий	Проверка практических работ
Темы 1-6	ОК 01, 02, 04, ПК 8.1 ЛР 7, 13, 17, 22	Промежуточный	Зачет с оценкой

2. Задания для контроля и оценки результатов

2.1. Задания для текущего контроля успеваемости

Тема 2. Языки для верстки сайтов

Практическая работа № 1 Создайте план разработки простого веб-сайта. Определите цели, целевую аудиторию, основные функции и этапы разработки.

Цель: Знакомство с основами веб-разработки. Научиться планировать создание веб-сайта, определяя его цели, целевую аудиторию и основные функции. Развить навыки работы в команде и управления проектом.

ХОД РАБОТЫ:

План разработки простого веб-сайта

1. Определение целей веб-сайта

- ****Основная цель****: Создание простого веб-сайта для представления информации о продукте/услуге (например, сайт для кафе, блога или портфолио).

- ****Дополнительные цели****:

- Привлечение новых клиентов.
- Увеличение узнаваемости бренда.
- Предоставление информации о товарах/услугах.
- Обеспечение возможности обратной связи с пользователями.

2. Определение целевой аудитории

- ****Целевая аудитория****:

- Потенциальные клиенты (возраст, интересы, местоположение).
- Текущие клиенты (для получения информации о новинках и акциях).
- Пользователи, интересующиеся тематикой сайта (например, любители кулинарии для сайта кафе).

3. Основные функции веб-сайта

- ****Информационные страницы****:
 - Главная страница с кратким описанием.
 - Страница "О нас" с информацией о компании/авторе.
 - Страница "Услуги/Продукты" с описанием и ценами.
 - Контактная страница с формой обратной связи и контактной информацией.
- ****Дополнительные функции****:
 - Галерея изображений (фото продукции, интерьера и т.д.).
 - Блог или раздел новостей для публикации актуальной информации.
 - Интеграция с социальными сетями.

4. Этапы разработки

- ****Этап 1: Исследование и планирование****
 - Сбор информации о целевой аудитории и конкурентах.
 - Определение структуры сайта (карта сайта).
- ****Этап 2: Дизайн****
 - Создание макета (wireframe) страниц.
 - Выбор цветовой схемы и шрифтов.
 - Разработка логотипа и других графических элементов.
- ****Этап 3: Разработка****
 - Выбор технологий (HTML, CSS, JavaScript, возможно, фреймворки).
 - Создание структуры сайта (разработка страниц).
 - Наполнение сайта контентом (тексты, изображения).
- ****Этап 4: Тестирование****
 - Проверка функциональности всех элементов сайта.
 - Тестирование на различных устройствах и браузерах.
 - Исправление ошибок и улучшение пользовательского опыта.
- ****Этап 5: Запуск****
 - Размещение сайта на хостинге.
 - Настройка доменного имени.
 - Проведение рекламной кампании для привлечения пользователей.
- ****Этап 6: Поддержка и обновление****
 - Регулярное обновление контента.
 - Мониторинг работы сайта и исправление возможных ошибок.
 - Сбор и анализ отзывов пользователей для улучшения сайта.

Тема 3. Структура html-документа. Теги HTML

Практическая работа №2: Создайте HTML-документ с использованием различных тегов: заголовки, параграфы, списки, ссылки и изображения.

Цель: Ознакомить студентов с основами HTML и его структурой. Научить студентов использовать различные HTML-теги для создания веб-страницы. Развить навыки работы с текстовым редактором и браузером для просмотра созданного HTML-документа. Понять, как правильно структурировать информацию на веб-странице с помощью заголовков, параграфов, списков, ссылок и изображений.

ХОД РАБОТЫ:

Задание— создать простой HTML-документ, который будет включать различные элементы: заголовки, параграфы, списки, ссылки и изображения. Это поможет вам освоить основы разметки HTML и научиться структурировать информацию на веб-странице.

Шаги выполнения задания

1. ****Откройте текстовый редактор****:

- Вы можете использовать любой текстовый редактор, например, Notepad, Visual Studio Code, Sublime Text или другой.

2. ****Создайте новый файл****:

- Сохраните его с расширением `.html`, например, `my_first_webpage.html`.

3. ****Структура HTML-документа****:

Вставьте следующий код в ваш файл:

```

`html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Моя первая веб-страница</title>
</head>
<body>
  <h1>Добро пожаловать на мою веб-страницу!</h1>
  <p>Это мой первый HTML-документ, созданный в рамках практической работы.</p>
  <h2>Мои увлечения</h2>
  <p>Я люблю заниматься следующими вещами:</p>
  <ul>
    <li>Чтение книг</li>
    <li>Программирование</li>
    <li>Путешествия</li>
  </ul>
  <h2>Мои любимые сайты</h2>
  <p>Вот несколько ссылок на мои любимые сайты:</p>
  <ol>
    <li><a href="https://www.wikipedia.org" target="_blank">Wikipedia</a></li>
    <li><a href="https://www.github.com" target="_blank">GitHub</a></li>
    <li><a href="https://www.stackoverflow.com" target="_blank">Stack Overflow</a></li>
  </ol>
  <h2>Мое любимое изображение</h2>
  <p>Вот изображение, которое мне нравится:</p>
  
  <footer>
    <p>© 2023 Моя первая веб-страница</p>
  </footer>
</body>
</html>
`

```

4. ****Настройка изображения****:

- Замените `https://example.com/image.jpg` на URL изображения, которое вы хотите использовать. Убедитесь, что изображение доступно по указанной ссылке.

5. ****Открытие документа в браузере****:

- Сохраните изменения и откройте ваш HTML-файл в веб-браузере (двойной щелчок по файлу или открытие через меню браузера).

6. ****Проверка результата****:

- Убедитесь, что все элементы отображаются корректно: заголовки, параграфы, списки, ссылки и изображение.

Обсуждение результатов

После завершения задания, обсудите с однокурсниками, какие теги вы использовали, и как они помогают структурировать информацию на веб-странице. Обратите внимание на важность правильного использования HTML-тегов для обеспечения удобства чтения и навигации.

Тема 4. Инструменты для создания HTML кода

Практическая работа №3: «Установка и настройка текстовых редакторов (например, Notepad++)»

Цель:

1. Ознакомить студентов с процессом установки текстового редактора.
2. Научить студентов настраивать текстовый редактор для удобной работы с кодом.
3. Развить навыки работы с текстовыми редакторами, которые будут полезны в веб-разработке.

ХОД РАБОТЫ:

1. ****Скачивание Notepad++****:
 - Перейдите на официальный сайт Notepad++: notepad-plus-plus.org.
 - Нажмите на кнопку «Download» (Скачать).
 - Выберите последнюю версию для вашей операционной системы (обычно это версия для Windows) и скачайте установочный файл.
2. ****Установка Notepad++****:
 - Запустите скачанный установочный файл.
 - Следуйте инструкциям мастера установки:
 - Примите условия лицензионного соглашения.
 - Выберите папку для установки (по умолчанию это будет `C:\Program Files\Notepad++`).
 - Выберите дополнительные компоненты, которые вы хотите установить (рекомендуется оставить все по умолчанию).
 - Нажмите «Install» (Установить) и дождитесь завершения установки.
3. ****Запуск Notepad++****:
 - После завершения установки запустите Notepad++.
 - Вы можете найти ярлык на рабочем столе или в меню «Пуск».
4. ****Настройка Notepad++****:
 - ****Выбор языка программирования****:
 - Откройте новый файл (Файл → Новый).
 - Перейдите в меню «Язык» и выберите «HTML» (или другой язык, с которым вы собираетесь работать).
 - ****Настройка цветовой схемы****:
 - Перейдите в меню «Настройки» → «Стиль конфигурирования».
 - Выберите цветовую схему, которая вам нравится (например, «Monokai» или «Solarized»).
 - Нажмите «Сохранить и закрыть».
 - ****Включение автозавершения****:
 - Перейдите в меню «Настройки» → «Параметры».
 - В разделе «Редактирование» установите галочку на «Включить автозавершение» и «Включить подсветку синтаксиса».
 - Нажмите «Закреть».
5. ****Создание и сохранение первого файла****:
 - Создайте новый файл (Файл → Новый).
 - Напишите простой HTML-код, например:

```
```html
```

```

<!DOCTYPE html>
<html lang="ru">
<head>
 <meta charset="UTF-8">
 <title>Моя первая страница</title>
</head>
<body>
 <h1>Добро пожаловать!</h1>
 <p>Это мой первый HTML-документ, созданный в Notepad++.</p>
</body>
</html>
```

```

- Сохраните файл (Файл → Сохранить как) с расширением `.html`, например, `my_first_page.html`.

6. ****Открытие файла в браузере****:

- Найдите сохраненный файл на вашем компьютере и откройте его в веб-браузере, чтобы увидеть результат.

Тема 5. Visual Studio Code. Плагины и расширения

Практическая работа №4: Установка Visual Studio Code и добавление полезных плагинов (например, Live Server и Emmet).

Цель:

1. Ознакомить студентов с процессом установки Visual Studio Code (VS Code).
2. Научить студентов устанавливать и настраивать полезные плагины для повышения продуктивности.
3. Развить навыки работы с редактором кода, который широко используется в веб-разработке.

ХОД РАБОТЫ:

1. ****Скачивание Visual Studio Code****:

- Перейдите на официальный сайт Visual Studio Code:

code.visualstudio.com.

- Нажмите на кнопку «Download» (Скачать) и выберите версию для вашей операционной системы (Windows, macOS или Linux).

2. ****Установка Visual Studio Code****:

- Запустите скачанный установочный файл.

- Следуйте инструкциям мастера установки:

- Примите условия лицензионного соглашения.

- Выберите папку для установки (по умолчанию это будет `C:\Program Files\Microsoft VS Code`).

- Выберите дополнительные параметры установки (например, создание ярлыка на рабочем столе).

- Нажмите «Install» (Установить) и дождитесь завершения установки.

3. ****Запуск Visual Studio Code****:

- После завершения установки запустите Visual Studio Code.

- Вы можете найти ярлык на рабочем столе или в меню «Пуск».

4. ****Установка полезных плагинов****:

- ****Live Server****:

- Откройте панель расширений, нажав на иконку «Extensions» (или используйте комбинацию клавиш `Ctrl + Shift + X`).

- В строке поиска введите «Live Server».
- Найдите расширение от Ritwick Dey и нажмите на кнопку «Install» (Установить).
- **Emmet**:
 - Emmet обычно уже встроен в VS Code, но если вы хотите проверить его настройки:
 - В панели расширений введите «Emmet» и убедитесь, что оно установлено (обычно оно включено по умолчанию).

5. **Настройка Live Server**:

- После установки Live Server, откройте HTML-файл (или создайте новый файл с расширением `.html`).
- Чтобы запустить сервер, щелкните правой кнопкой мыши на редакторе и выберите «Open with Live Server» (Открыть с помощью Live Server).
- Ваш файл откроется в браузере, и любые изменения, которые вы вносите, будут автоматически обновляться.

6. **Использование Emmet**:

- В файле HTML вы можете использовать Emmet для быстрого создания кода. Например, введите `!` и нажмите `Tab`, чтобы сгенерировать базовую структуру HTML-документа.

- Попробуйте также использовать другие сокращения Emmet для создания элементов, например, `ul>li*5` для создания нумерованного списка с пятью элементами.

Обсуждение результатов

После завершения задания обсудите с однокурсниками, как плагины Live Server и Emmet могут улучшить ваш рабочий процесс и какие другие расширения вы могли бы использовать в будущем.

Тема 6. Разметка текста. Ссылки и изображения. Таблицы. Формы

Практическая работа № 5. Создайте веб-страницу, которая включает текстовые элементы, ссылки, изображения, таблицы и формы.

Цель:

1. Научить студентов создавать полноценные веб-страницы с использованием HTML.
2. Ознакомить студентов с основными элементами разметки, такими как текст, ссылки, изображения, таблицы и формы.
3. Развить навыки работы с HTML-кодом для создания интерактивных и информативных веб-страниц.

ХОД РАБОТЫ:

1. **Создание HTML-файла**:
 - Откройте Visual Studio Code (или другой текстовый редактор).
 - Создайте новый файл и сохраните его с расширением `.html`, например, `my_web_page.html`.

2. **Структура HTML-документа**:

- Начните с создания базовой структуры HTML-документа:

```

<<<html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Моя веб-страница</title>
</head>

```

```

<body>
  <h1>Добро пожаловать на мою веб-страницу!</h1>
  <p>Это пример веб-страницы, созданной с использованием HTML.</p>
</body>
</html>
...

```

3. ****Добавление текстовых элементов**:**

- Добавьте различные текстовые элементы, такие как заголовки, абзацы и списки:

```

```html
<h2>Обо мне</h2>
<p>Меня зовут [Ваше имя], и я изучаю веб-разработку.</p>

 Интересы:
 Программирование
 Дизайн
 Музыка

...

```

### 4. **\*\*Добавление ссылок\*\*:**

- Вставьте ссылки на внешние ресурсы:

```

```html
<h2>Полезные ссылки</h2>
<p>Вот некоторые ресурсы, которые я рекомендую:</p>
<ul>
  <li><a href="https://www.w3schools.com/" target="_blank">W3Schools</a></li>
  <li><a href="https://developer.mozilla.org/" target="_blank">MDN Web
Docs</a></li>
</ul>
...

```

5. ****Добавление изображений**:**

- Вставьте изображение:

```

```html
<h2>Мое любимое изображение</h2>

...

```

\*(Замените `https://example.com/image.jpg` на URL изображения, которое вы хотите использовать.)\*

### 6. **\*\*Создание таблицы\*\*:**

- Создайте таблицу с данными:

```

```html
<h2>Моя таблица</h2>
<table border="1">
  <tr>
    <th>Название</th>

```

```

        <th>Год</th>
    </tr>
    <tr>
        <td>HTML</td>
        <td>1993</td>
    </tr>
    <tr>
        <td>CSS</td>
        <td>1996</td>
    </tr>
    <tr>
        <td>JavaScript</td>
        <td>1995</td>
    </tr>
</table>
...

```

7. ****Добавление формы****:

- Вставьте форму для сбора данных:

```

...html
<h2>Контактная форма</h2>
<form action="#" method="post">
    <label for="name">Имя:</label><br>
    <input type="text" id="name" name="name" required><br><br>

    <label for="email">Электронная почта:</label><br>
    <input type="email" id="email" name="email" required><br><br>

    <input type="submit" value="Отправить">
</form>
...

```

8. ****Завершение документа****:

- Закройте теги `body` и `html` в конце вашего документа:

```

...html
</body>
</html>
...

```

Тема 7. Основы CSS

Практическая работа №6: «Напишите CSS-стили для ранее созданной HTML-страницы. Используйте селекторы, свойства и значения для стилизации элементов»

Цель:

1. Научить студентов применять CSS для стилизации HTML-элементов.
2. Ознакомить студентов с различными селекторами, свойствами и значениями CSS.
3. Развить навыки работы с внешними и внутренними стилями.

ХОД РАБОТЫ:

1. ****Создание CSS-файла****:

- В Visual Studio Code создайте новый файл и сохраните его с расширением `.css`, например, `styles.css`.

2. ****Подключение CSS к HTML**:**

- В вашем HTML-файле, внутри тега ``, добавьте ссылку на созданный CSS-файл:

```
```html
<link rel="stylesheet" href="styles.css">
```
```

3. ****Стилизация элементов**:**

- Теперь вы можете добавлять стили в ваш CSS-файл. Вот несколько примеров стилей, которые можно использовать:

```
```css
/* Стили для всего документа */
body {
 font-family: Arial, sans-serif; /* Шрифт для всего текста */
 line-height: 1.6; /* Межстрочный интервал */
 margin: 0;
 padding: 20px;
 background-color: #f4f4f4; /* Цвет фона */
}

/* Стили для заголовков */
h1 {
 color: #333; /* Цвет текста заголовка */
 text-align: center; /* Центрирование заголовка */
}

h2 {
 color: #007BFF; /* Цвет текста подзаголовка */
}

/* Стили для параграфов */
p {
 color: #555; /* Цвет текста параграфа */
}

/* Стили для ссылок */
a {
 color: #007BFF; /* Цвет ссылок */
 text-decoration: none; /* Убираем подчеркивание */
}

a:hover {
 text-decoration: underline; /* Подчеркивание при наведении */
}

/* Стили для изображений */
img {
 display: block; /* Отображение изображения как блока */
 margin: 20px auto; /* Центрирование изображения */
 border: 2px solid #ccc; /* Рамка вокруг изображения */
 border-radius: 10px; /* Скругление углов */
}
```
```

```

}

/* Стили для таблицы */
table {
  width: 100%; /* Ширина таблицы */
  border-collapse: collapse; /* Убираем двойные границы */
  margin-top: 20px; /* Отступ сверху */
}

th, td {
  padding: 10px; /* Отступы внутри ячеек */
  text-align: left; /* Выравнивание текста */
  border: 1px solid #ddd; /* Граница ячеек */
}

th {
  background-color: #007BFF; /* Цвет фона заголовков таблицы */
  color: white; /* Цвет текста заголовков таблицы */
}

/* Стили для формы */
form {
  margin-top: 20px; /* Отступ сверху */
  background-color: #fff; /* Цвет фона формы */
  padding: 20px; /* Отступы внутри формы */
  border: 1px solid #ddd; /* Граница формы */
  border-radius: 5px; /* Скругление углов формы */
}

input[type="text"],
input[type="email"] {
  width: 100%; /* Ширина полей ввода */
  padding: 10px; /* Отступы внутри полей ввода */
  margin: 10px 0; /* Отступы сверху и снизу */
  border: 1px solid #ccc; /* Граница полей ввода */
  border-radius: 5px; /* Скругление углов полей ввода */
}

input[type="submit"] {
  background-color: #007BFF; /* Цвет фона кнопки */
  color: white; /* Цвет текста кнопки */
  border: none; /* Убираем границу */
  padding: 10px 15px; /* Отступы внутри кнопки */
  border-radius: 5px; /* Скругление углов кнопки */
  cursor: pointer; /* Курсор при наведении */
}

input[type="submit"]:hover {
  background-color: #0056

```

Практическая работа №7: «Подключите веб-шрифты с помощью Google Fonts к вашей HTML-странице»

Цель:

1. Научить студентов подключать веб-шрифты с помощью Google Fonts.
2. Ознакомить студентов с изменением шрифтов и стилей текста на веб-странице.
3. Развить навыки стилизации текста с использованием различных шрифтов.

ХОД РАБОТЫ:

1. ****Выбор шрифтов на Google Fonts****:

- Перейдите на сайт [Google Fonts](https://fonts.google.com/).

- Выберите один или несколько шрифтов, которые вы хотите использовать на своей странице. Например, вы можете выбрать `Roboto` и `Lora`.

2. ****Подключение шрифтов к HTML****:

- После выбора шрифтов на Google Fonts, вы получите код для подключения.

Скопируйте его.

- Вставьте этот код в секцию ``<head>`` вашего HTML-документа. Пример:

```

<<<html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Моя веб-страница с шрифтами</title>
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&family=Lora:wght@400;700&display=swap" rel="stylesheet">
</head>
<body>
  <h1>Добро пожаловать на мою веб-страницу!</h1>
  <p>Это пример веб-страницы, созданной с использованием HTML и Google
Fonts.</p>
</body>
</html>
<<<

```

3. ****Изменение стилей текста в CSS****:

- Теперь откройте ваш CSS-файл (`styles.css`) и измените шрифты для различных элементов. Например:

```

<<<css
/* Стили для всего документа */
body {
  font-family: 'Roboto', sans-serif; /* Основной шрифт для всего текста */
  line-height: 1.6; /* Межстрочный интервал */
  margin: 0;
  padding: 20px;
  background-color: #f4f4f4; /* Цвет фона */
}

/* Стили для заголовков */
h1 {

```

```

font-family: 'Lora', serif; /* Шрифт для заголовка */
color: #3333; /* Цвет текста заголовка */
text-align: center; /* Центрирование заголовка */
}

h2 {
font-family: 'Roboto', sans-serif; /* Шрифт для подзаголовка */
color: #007BFF; /* Цвет текста подзаголовка */
}

/* Стили для параграфов */
p {
font-family: 'Roboto', sans-serif; /* Шрифт для параграфов */
color: #5555; /* Цвет текста параграфа */
}

/* Стили для ссылок */
a {
font-family: 'Roboto', sans-serif; /* Шрифт для ссылок */
color: #007BFF; /* Цвет ссылок */
text-decoration: none; /* Убираем подчеркивание */
}

a:hover {
text-decoration: underline; /* Подчеркивание при наведении */
}

```

4. ****Просмотр изменений****:

- Сохраните изменения в HTML и CSS файлах.
- Откройте вашу HTML-страницу в браузере, чтобы увидеть изменения.

Обсуждение результатов

После выполнения задания, обсудите с однокурсниками, какие шрифты вы выбрали и как они повлияли на внешний вид вашей страницы. Обсудите, какие шрифты лучше подходят для различных типов контента и как использование веб-шрифтов может улучшить читаемость и эстетику веб-страницы.

Тема 9. Базовая стилизация

Практическая работа №8: «Создайте стиль для вашей веб-страницы, используя цвета, фоны, отступы и границы. Примените стили к различным элементам

Цель:

1. Научить студентов применять различные стили к HTML-элементам с использованием CSS.
2. Ознакомить студентов с основами работы с цветами, фонами, отступами и границами.
3. Развить навыки создания визуально привлекательных веб-страниц.

ХОД РАБОТЫ:

1. ****Создание CSS-файла****:
 - Если у вас еще нет CSS-файла, создайте новый файл и сохраните его с расширением `css`, например, `styles.css`.
2. ****Подключение CSS к HTML****:

- В вашем HTML-файле, внутри тега ``<head>``, добавьте ссылку на созданный CSS-файл:

```
``html
<link rel="stylesheet" href="styles.css">
```
```

### 3. **\*\*Стилизация элементов\*\***:

- Теперь вы можете добавлять стили в ваш CSS-файл. Вот несколько примеров стилей, которые можно использовать:

```
``css
/* Стили для всего документа */
body {
 font-family: Arial, sans-serif; /* Шрифт для всего текста */
 line-height: 1.6; /* Межстрочный интервал */
 margin: 0;
 padding: 20px;
 background-color: #f0f8ff; /* Цвет фона страницы */
}

/* Стили для заголовков */
h1 {
 color: #2c3e50; /* Цвет текста заголовка */
 text-align: center; /* Центрирование заголовка */
 padding: 20px; /* Отступы вокруг заголовка */
 background-color: #3498db; /* Цвет фона заголовка */
 border-radius: 10px; /* Скругление углов */
}

h2 {
 color: #2980b9; /* Цвет текста подзаголовка */
 margin-top: 30px; /* Отступ сверху */
}

/* Стили для параграфов */
p {
 color: #34495e; /* Цвет текста параграфа */
 padding: 10px; /* Отступы внутри параграфов */
 background-color: #ecf0f1; /* Цвет фона параграфов */
 border: 1px solid #bdc3c7; /* Граница вокруг параграфов */
 border-radius: 5px; /* Скругление углов */
 margin-bottom: 20px; /* Отступ снизу */
}

/* Стили для ссылок */
a {
 color: #e74c3c; /* Цвет ссылок */
 text-decoration: none; /* Убираем подчеркивание */
 border-bottom: 1px solid #e74c3c; /* Подчеркивание в виде границы */
 padding: 2px; /* Отступы вокруг ссылок */
}

a:hover {
 text-decoration: underline; /* Подчеркивание при наведении */
 color: #c0392b; /* Цвет ссылок при наведении */
}
```

```

}

/* Стили для кнопок */
button {
 background-color: #3498db; /* Цвет фона кнопки */
 color: white; /* Цвет текста кнопки */
 border: none; /* Убираем границу */
 padding: 10px 15px; /* Отступы внутри кнопки */
 border-radius: 5px; /* Скругление углов кнопки */
 cursor: pointer; /* Курсор при наведении */
 transition: background-color 0.3s; /* Плавный переход цвета фона */
}

button:hover {
 background-color: #2980b9; /* Цвет фона кнопки при наведении */
}

/* Стили для изображений */
img {
 display: block; /* Отображение изображения как блока */
 margin: 20px auto; /* Центрирование изображения */
 border: 2px solid #bdc3c7; /* Рамка вокруг изображения */
 border-radius: 10px; /* Скругление углов */
}

```

#### 4. **\*\*Просмотр изменений\*\***:

- Сохраните изменения в HTML и CSS файлах.
- Откройте вашу HTML-страницу в браузере, чтобы увидеть изменения.

### Тема 10. Валидность кода

#### **Практическая работа №9: «Проверить валидность HTML и CSS кода с помощью W3C Validator.»**

Цель:

1. Научить студентов использовать инструменты для проверки валидности HTML и CSS.
2. Ознакомить студентов с основами исправления ошибок и предупреждений в коде.
3. Развить навыки работы с валидными кодами для улучшения совместимости и доступности веб-страниц.

**ХОД РАБОТЫ:**

#### 1. **\*\*Подготовка кода\*\***:

- Убедитесь, что у вас есть HTML и CSS файлы, которые вы хотите проверить.

#### 2. **\*\*Проверка HTML-кода\*\***:

- Перейдите на сайт [W3C Markup Validation Service](https://validator.w3.org/).
- Выберите один из способов проверки:
  - Вставьте URL вашей веб-страницы, если она опубликована в интернете.
  - Выберите вкладку "Validate by File Upload", чтобы загрузить ваш HTML файл.
  - Или выберите "Validate by Direct Input", чтобы вставить код HTML в текстовое поле.
- Нажмите кнопку "Check" для начала проверки.

#### 3. **\*\*Анализ результатов\*\***:

- После проверки вы получите отчет о найденных ошибках и предупреждениях. Обратите внимание на:

- Ошибки (errors): это критические проблемы, которые необходимо исправить.
- Предупреждения (warnings): это рекомендации по улучшению кода, которые не обязательно исправлять, но желательно.

#### 4. **\*\*Исправление ошибок\*\***:

- Внимательно изучите каждую ошибку и предупреждение. Исправьте их в вашем HTML-коде.
- Например, если вы видите ошибку о неправильно закрытом теге, убедитесь, что все теги правильно открыты и закрыты.

#### 5. **\*\*Проверка CSS-кода\*\***:

- Перейдите на сайт [W3C CSS Validation Service](https://jigsaw.w3.org/css-validator/).
- Выберите один из способов проверки:
  - Вставьте URL вашего CSS файла, если он опубликован.
  - Выберите вкладку "Validate by File Upload", чтобы загрузить ваш CSS файл.
  - Или выберите "Validate by Direct Input", чтобы вставить код CSS в текстовое поле.
- Нажмите кнопку "Check" для начала проверки.

#### 6. **\*\*Анализ результатов CSS\*\***:

- Как и в случае с HTML, вы получите отчет о найденных ошибках и предупреждениях. Исправьте их в вашем CSS-коде.

#### 7. **\*\*Повторная проверка\*\***:

- После внесения исправлений повторно проверьте как HTML, так и CSS код, чтобы убедиться, что все ошибки и предупреждения устранены.

#### Заключение

После завершения всех шагов и исправления ошибок, обсудите с однокурсниками, какие проблемы вы нашли в своем коде и как вы их исправили. Это поможет вам лучше понять важность валидного кода и повысит качество ваших веб-страниц.

### Тема 11. Верстка первого сайта

**Практическая работа №10: «Создайте одностраничный сайт на основе ранее выполненных работ. Включите текст, изображения, ссылки и стили. Опубликуйте его на платформе GitHub Pages или аналогичной.**

Цель:

1. Научиться создавать одностраничные веб-сайты, используя HTML и CSS.
2. Ознакомиться с основами веб-дизайна и структуры страницы.
3. Развить навыки публикации веб-сайтов на платформах, таких как GitHub Pages.

#### ХОД РАБОТЫ:

##### 1. **\*\*Создание структуры проекта\*\***:

- Создайте папку для вашего проекта, например, `my-website`.
- Внутри этой папки создайте два файла: `index.html` и `styles.css`.

##### 2. **\*\*Создание HTML-страницы\*\***:

- Откройте файл `index.html` и добавьте следующий базовый код:

```

`html
<!DOCTYPE html>
<html lang="ru">
<head>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Мой Одностраничный Сайт</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
 <header>
 <h1>Добро пожаловать на мой одностраничный сайт!</h1>
 <nav>

 Обо мне
 Портфолио
 Контакты

 </nav>
 </header>
 <section id="about">
 <h2>Обо мне</h2>
 <p>Я веб-разработчик, который любит создавать красивые и функциональные сайты.</p>

 </section>

 <section id="portfolio">
 <h2>Портфолио</h2>
 <p>Вот некоторые из моих работ:</p>

 Проект 1
 Проект 2
 Проект 3

 </section>
 <section id="contact">
 <h2>Контакты</h2>
 <p>Свяжитесь со мной по электронной почте: your-email@example.com</p>
 </section>

 <footer>
 <p>© 2023 Мой Одностраничный Сайт</p>
 </footer>
</body>
</html>

```

### 3. **\*\*Создание CSS-стилей\*\***:

- Откройте файл `styles.css` и добавьте стили для вашего сайта. Вот пример:

```

```css
body {
  font-family: Arial, sans-serif;
  line-height: 1.6;
  margin: 0;

```

```

padding: 20px;
background-color: #f0f8ff;
}

header {
background-color: #3498db;
color: white;
padding: 20px;
text-align: center;
}

nav ul {
list-style: none;
padding: 0;
}

nav ul li {
display: inline;
margin-right: 20px;
}

section {
margin: 20px 0;
padding: 15px;
background-color: #ecf0f1;
border-radius: 5px;
}

footer {
text-align: center;
margin-top: 20px;
font-size: 0.8em;
}

img {
max-width: 100%;
height: auto;
}

```

4. ****Добавление изображений и ссылок**:**

- Замените `your-image-url.jpg` на URL изображения, которое вы хотите использовать.
- Замените `link-to-your-project1`, `link-to-your-project2`, и `link-to-your-project3` на ссылки на ваши проекты.

5. ****Тестирование локально**:**

- Откройте файл `index.html` в браузере, чтобы убедиться, что все работает

Тема 12. Верстка сайта с боковой панелью

Практическая работа №11: «Создайте веб-страницу с боковой панелью навигации. Используйте Flexbox или Grid для расположения элементов. Добавьте ссылки на другие разделы сайта.».

Цель: Научиться использовать Flexbox или CSS Grid для создания адаптивных макетов.

2. Ознакомиться с основами создания боковой панели навигации.

3. Развить навыки работы с HTML и CSS для создания многостраничных сайтов. ХОД РАБОТЫ:

1. ****Создание структуры проекта****:

- Создайте папку для вашего проекта, например, `sidebar-navigation`.
- Внутри этой папки создайте два файла: `index.html` и `styles.css`.

2. ****Создание HTML-страницы****:

- Откройте файл `index.html` и добавьте следующий код:

```

<<<html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Веб-страница с боковой панелью навигации</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <aside class="sidebar">
      <h2>Навигация</h2>
      <ul>
        <li><a href="#home">Главная</a></li>
        <li><a href="#about">О нас</a></li>
        <li><a href="#services">Услуги</a></li>
        <li><a href="#contact">Контакты</a></li>
      </ul>
    </aside>
    <main class="content">
      <h1>Добро пожаловать на нашу веб-страницу!</h1>
      <section id="home">
        <h2>Главная</h2>
        <p>Это главная страница нашего сайта.</p>
      </section>
      <section id="about">
        <h2>О нас</h2>
        <p>Мы предоставляем различные услуги для наших клиентов.</p>
      </section>
      <section id="services">
        <h2>Услуги</h2>
        <p>Мы предлагаем широкий спектр услуг, включая веб-разработку и дизайн.</p>
      </section>
      <section id="contact">
        <h2>Контакты</h2>
        <p>Свяжитесь с нами по электронной почте: <a href="mailto:info@example.com">info@example.com</a></p>
      </section>
    </main>
  </div>

```

```

</body>
</html>
```

```

### 3. **\*\*Создание CSS-стилей\*\***:

- Откройте файл `styles.css` и добавьте стили для вашего сайта. Вот пример, использующий Flexbox:

```

```css
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
}

.container {
  display: flex;
}

.sidebar {
  width: 250px;
  background-color: #3498db;
  color: white;
  padding: 15px;
}

.sidebar h2 {
  margin-top: 0;
}

.sidebar ul {
  list-style: none;
  padding: 0;
}

.sidebar ul li {
  margin: 10px 0;
}

.sidebar ul li a {
  color: white;
  text-decoration: none;
}

.sidebar ul li a:hover {
  text-decoration: underline;
}

.content {
  flex: 1;
  padding: 20px;
}
```

```

```

background-color: #ecf0f1;
}

section {
margin-bottom: 20px;
padding: 15px;
background-color: white;
border-radius: 5px;
}

h1, h2 {
color: #2c3e50;
}
...

```

#### 4. **\*\*Тестирование локально\*\***:

- Откройте файл `index.html` в браузере, чтобы убедиться, что все работает корректно. Вы должны увидеть боковую панель навигации слева и основной контент справа.

#### 5. **\*\*Добавление дополнительных стилей (по желанию)\*\***:

- Вы можете добавить дополнительные стили, чтобы улучшить внешний вид вашей страницы, например, изменить цвета, шрифты или добавить эффекты

### **Тема 13. Верстка многостраничного сайта**

**Контрольная работа №12:** «Создайте многостраничный сайт, состоящий из главной страницы и нескольких дополнительных страниц. Реализуйте навигацию между страницами и используйте общий стиль для всех страниц.».

Цель: Развитие навыков веб-разработки и дизайна.

Создание многостраничного сайта позволит:

1. **\*\*Освоить основы HTML и CSS\*\***: Понимание структуры веб-страниц и стилей поможет вам научиться создавать и оформлять контент.
2. **\*\*Развить навыки проектирования пользовательского интерфейса\*\***: Вы сможете применять принципы удобства и доступности, чтобы ваш сайт был интуитивно понятным и привлекательным для пользователей.
3. **\*\*Научиться работать с адаптивным дизайном\*\***: Вы получите опыт в создании сайтов, которые хорошо выглядят и функционируют на различных устройствах, включая мобильные телефоны и планшеты.
4. **\*\*Углубить знания в области навигации и пользовательского опыта\*\***: Вы научитесь создавать логичную и удобную навигацию, что является ключевым аспектом успешного веб-дизайна.
5. **\*\*Создать портфолио работ\*\***: Вы сможете продемонстрировать свои навыки и творчество потенциальным работодателям или клиентам, что может помочь в дальнейшем трудоустройстве или фрилансе.

**ХОД РАБОТЫ:**

Вариант 1: Портфолио

**\*\*Задание\*\***: Создайте личное портфолио с использованием HTML и CSS.

**\*\*Страницы\*\***:

1. **\*\*Главная\*\***: Краткое введение о вас, ваше имя и фотография.
2. **\*\*Проекты\*\***: Список ваших проектов с описанием и ссылками на них.
3. **\*\*Контакты\*\***: Форма обратной связи с полями для имени, электронной почты и сообщения.

**\*\*Дополнительные требования\*\*:**

- Используйте адаптивный дизайн, чтобы сайт выглядел хорошо на мобильных устройствах.
- Добавьте стили для кнопок и форм, чтобы они были визуально привлекательными.

**Вариант 2: Блог****\*\*Задание\*\*:** Создайте многостраничный блог на тему, которая вам интересна.**\*\*Страницы\*\*:**

1. **\*\*Главная\*\*:** Список последних постов с кратким описанием и изображениями.
2. **\*\*О блоге\*\*:** Информация о вас и о том, почему вы решили создать блог.
3. **\*\*Посты\*\*:** Страница с полным текстом одного из ваших постов, включая заголовок, дату и авторство.

**\*\*Дополнительные требования\*\*:**

- Используйте различные стили для заголовков и текстов, чтобы выделить важные моменты.
- Добавьте навигацию между постами и возможность вернуться на главную страницу.

**Вариант 3: Веб-сайт компании****\*\*Задание\*\*:** Создайте сайт для вымышленной компании, предлагающей услуги.**\*\*Страницы\*\*:**

1. **\*\*Главная\*\*:** Краткое описание компании, её миссии и ценностей.
2. **\*\*Услуги\*\*:** Подробное описание предлагаемых услуг с ценами и возможностью запроса консультации.
3. **\*\*Контакты\*\*:** Информация о местоположении компании, телефон, электронная почта и форма для обратной связи.

**\*\*Дополнительные требования\*\*:**

- Используйте фирменные цвета и шрифты для создания единого стиля.
- Сделайте акцент на удобстве навигации и простоте использования сайта.

**Тема 14. Основные свойства flex-контейнеров и flex-элементов****Практическая работа №13: «Создайте простой макет с использованием Flexbox.»**

Цель: освоение основ Flexbox для создания адаптивного веб-макета.

**ХОД РАБОТЫ:**

1. **\*\*Создавать структуру веб-страницы с использованием HTML\*\*:** Понимание базовых элементов, таких как заголовок, навигация, основной контент и подвал.
2. **\*\*Применять CSS для стилизации и выравнивания элементов\*\*:** Использование Flexbox для упрощения выравнивания и распределения пространства между элементами.
3. **\*\*Разрабатывать адаптивные макеты\*\*:** Создание макетов, которые автоматически подстраиваются под различные размеры экранов.

1. **\*\*Создание HTML-структуры\*\*:**

- Создайте файл `index.html`.
- Используйте базовые HTML-элементы для создания заголовка, навигационного меню, основного контента и подвала. Включите все необходимые теги и структуру.

2. **\*\*Добавление стилей с помощью CSS\*\*:**

- Создайте файл `styles.css`.
- Используйте селекторы CSS для стилизации элементов. Примените Flexbox для заголовка и основного контента, чтобы обеспечить правильное выравнивание и распределение пространства.

- Настройте цветовую гамму, отступы и шрифты для улучшения внешнего вида макета.

3. **\*\*Тестирование макета\*\***:

- Откройте файл `index.html` в веб-браузере.
- Проверьте, как элементы выглядят и ведут себя при изменении размеров окна браузера, чтобы убедиться в адаптивности макета.

4. **\*\*Анализ и улучшение\*\***:

- Проанализируйте ваш макет. Как он выглядит на разных устройствах? Есть ли возможность улучшить его?
- Попробуйте внести изменения в CSS для улучшения внешнего вида или функциональности.

5. **\*\*Документация\*\***:

- Запишите свои наблюдения и выводы о том, как Flexbox помогает в создании адаптивных макетов.
- Сохраните все файлы и подготовьте их к демонстрации.

**Практическая работа №14: «Создайте контейнер с тремя элементами (например, карточками товара). Используйте свойства `flex-direction`, `justify-content` и `align-items` для настройки их расположения. Измените порядок элементов с помощью свойства `order`.»**

Цель: освоить использование Flexbox для создания контейнера с тремя элементами (карточками товара) и научиться настраивать их расположение с помощью свойств `flex-direction`, `justify-content`, `align-items`, а также изменять порядок элементов с помощью свойства `order`.

ХОД РАБОТЫ

1. **\*\*Создание HTML-структуры\*\***:

- Создайте файл `index.html`.
- Добавьте структуру для контейнера и карточек товара.

```

<<html
<!DOCTYPE html>
<html lang="ru">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Карточки товара</title>
 <link rel="stylesheet" href="styles.css">
</head>
<body>
 <div class="container">
 <div class="card" id="card1">
 <h2>Товар 1</h2>
 <p>Описание товара 1</p>
 <p>Цена: 1000 руб.</p>
 </div>
 <div class="card" id="card2">
 <h2>Товар 2</h2>
 <p>Описание товара 2</p>

```

```

 <p>Цена: 1500 руб.</p>
 </div>
 <div class="card" id="card3">
 <h2>Товар 3</h2>
 <p>Описание товара 3</p>
 <p>Цена: 2000 руб.</p>
 </div>
</div>
</body>
</html>
```

```

2. **Добавление стилей с помощью CSS**:

- Создайте файл `styles.css`.
- Используйте Flexbox для настройки расположения карточек товара.

```

```css
* {
 box-sizing: border-box;
 margin: 0;
 padding: 0;
}

body {
 font-family: Arial, sans-serif;
 background-color: #f4f4f4;
 display: flex;
 justify-content: center;
 align-items: center;
 height: 100vh;
}

.container {
 display: flex;
 flex-direction: row; /* Измените на column для вертикального расположения */
 justify-content: space-around; /* Распределение пространства между карточками */
 align-items: flex-start; /* Выравнивание карточек по верхнему краю */
 width: 80%;
}

.card {
 background: white;
 border: 1px solid #ccc;
 border-radius: 8px;
 padding: 20px;
 width: 30%; /* Ширина карточки */
 box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

/* Изменение порядка карточек */
#card1 {
 order: 2; /* Второй по порядку */
}
```

```

```

}

#card2 {
  order: 1; /* Первый по порядку */
}

#card3 {
  order: 3; /* Третий по порядку */
}
...

```

3. ****Тестирование макета****:
 - Откройте файл `index.html` в веб-браузере.
 - Убедитесь, что карточки отображаются в заданном порядке и правильно расположены.
4. ****Анализ и улучшение****:
 - Проанализируйте макет. Как он выглядит на разных размерах экрана?
 - Попробуйте изменить значение свойства `flex-direction` на `column` и посмотрите, как изменится расположение карточек.
 - Измените порядок карточек, изменяя значения свойства `order` для каждого элемента.
5. ****Документация****:
 - Запишите свои наблюдения о том, как свойства Flexbox влияют на расположение элементов.
 - Сохраните все файлы и подготовьте их к демонстрации.

Тема 15. Поток вывода элементов на страницу

Практическая работа №15: «Создайте страницу с различными элементами, чтобы продемонстрировать поток вывода.»

Цель: изучить и понять, как различные HTML-элементы взаимодействуют друг с другом в потоке вывода. Вы научитесь использовать различные типы элементов, такие как заголовки, параграфы, изображения, списки и таблицы, чтобы создать структурированную и удобочитаемую веб-страницу.

ХОД РАБОТЫ:

1. ****Создание HTML-страницы****:
 - Создайте файл `index.html`.
 - Включите в страницу различные элементы, такие как:
 - Заголовки (`<h1>`, `<h2>`, `<h3>`)
 - Параграфы (`<p>`)
 - Списки (упорядоченные `` и неупорядоченные ``)
 - Изображения (``)
 - Таблицы (`<table>`)
 - Блоки цитат (`<blockquote>`)
2. ****Структурирование контента****:
 - Разделите контент на логические секции, используя заголовки для обозначения каждой секции.
 - Убедитесь, что элементы правильно расположены и читаемы.
3. ****Стилизация с помощью CSS****:
 - Создайте файл `styles.css`.

- Добавьте базовые стили для улучшения внешнего вида страницы, такие как:
 - Шрифты и размеры текста
 - Отступы и поля
 - Цвета фона и текста

4. ****Тестирование страницы****:

- Откройте файл `index.html` в веб-браузере.
- Проверьте, как элементы отображаются и взаимодействуют друг с другом в потоке вывода.
- Убедитесь, что страница корректно отображается на разных устройствах и разрешениях экрана.

5. ****Анализ и улучшение****:

- Проанализируйте, как разные элементы влияют на поток вывода.
- Попробуйте изменить порядок расположения элементов и посмотрите, как это влияет на общий вид страницы.
- Добавьте дополнительные элементы или стили для улучшения дизайна.

6. ****Документация****:

- Запишите свои наблюдения о том, как различные элементы HTML влияют на поток вывода.
- Сохраните все файлы и подготовьте их к демонстрации.

Практическая работа №16: «Создайте блочные и строчные элементы, а также свойства float и clear»

Цель: изучить и понять различия между блочными и строчными элементами в HTML, а также освоить использование свойств CSS `float` и `clear` для управления расположением элементов на веб-странице.

ХОД РАБОТЫ:

1. ****Создание HTML-страницы****:

- Создайте файл `index.html`.
- Включите в страницу различные блочные и строчные элементы, такие как:
 - Блочные элементы: ``<div>``, ``<p>``, ``<h1>``, ``<h2>``, ````, ````, ``<table>`
 - Строчные элементы: ````, ``<a>``, ````, ````, ```

2. ****Структурирование контента****:

- Разделите контент на логические секции, используя заголовки для обозначения каждой секции.
- Включите как блочные, так и строчные элементы, чтобы продемонстрировать их взаимодействие.

3. ****Стилизация с помощью CSS****:

- Создайте файл `styles.css`.
- Добавьте базовые стили для улучшения внешнего вида страницы, такие как:
 - Шрифты и размеры текста
 - Отступы и поля
 - Цвета фона и текста

4. ****Использование свойств float и clear****:

- Примените свойство `float` к некоторым элементам, чтобы изменить их расположение на странице (например, сделайте изображения плавающими).

- Используйте свойство `clear` для управления потоком вывода и предотвращения наложения элементов.

5. ****Тестирование страницы****:

- Откройте файл `index.html` в веб-браузере.
- Проверьте, как элементы отображаются и взаимодействуют друг с другом, особенно при использовании `float` и `clear`.
- Убедитесь, что страница корректно отображается на разных устройствах и разрешениях экрана.

6. ****Анализ и улучшение****:

- Проанализируйте, как свойства `float` и `clear` влияют на расположение элементов.
- Попробуйте изменить порядок применения этих свойств и посмотрите, как это влияет на общий вид страницы.
- Добавьте дополнительные элементы или стили для улучшения дизайна.

7. ****Документация****:

- Запишите свои наблюдения о том, как блочные и строчные элементы, а также свойства `float` и `clear` влияют на поток вывода.
- Сохраните все файлы и подготовьте их к демонстрации.

Практическая работа №17: «Создайте макет, где текст обтекает изображение, и добавьте элементы с различными позициями (absolute, relative). Измените порядок отображения элементов с помощью position и z-index»

Цель: изучить и освоить создание макетов, где текст обтекает изображение, а также использовать различные свойства позиционирования (absolute, relative) для управления расположением элементов на веб-странице. Вы научитесь изменять порядок отображения элементов с помощью свойств `position` и `z-index`, что поможет вам создавать более сложные и визуально привлекательные веб-дизайны.

ХОД РАБОТЫ:

1. ****Создание HTML-страницы****:

- Создайте файл `index.html`.
- Включите в страницу текстовый контент, который будет обтекать изображение. Используйте элементы, такие как `

`, `

`, ` `, и добавьте изображение с помощью тега ``.

2. ****Структурирование контента****:

- Разделите контент на логические секции, используя заголовки и абзацы.
- Убедитесь, что текст правильно обтекает изображение.

3. ****Стилизация с помощью CSS****:

- Создайте файл `styles.css`.
- Добавьте стили для улучшения внешнего вида страницы:
 - Установите размеры и отступы для изображения.
 - Настройте шрифты и цвета текста.
 - Используйте свойство `float` для обтекания текста изображением.

4. ****Использование позиционирования****:

- Примените различные свойства позиционирования к элементам:
 - Используйте `position: relative;` для одного из блоков, чтобы задать его положение относительно его обычного потока.

- Используйте `position: absolute;` для другого блока, чтобы разместить его в определенном месте на странице.

- Измените порядок отображения элементов, используя свойство `z-index`. Убедитесь, что элементы с более высоким значением `z-index` отображаются поверх элементов с более низким значением.

5. ****Тестирование страницы****:

- Откройте файл `index.html` в веб-браузере.

- Проверьте, как текст обтекает изображение, и как различные позиции влияют на расположение элементов.

- Убедитесь, что страница корректно отображается на разных устройствах и разрешениях экрана.

6. ****Анализ и улучшение****:

- Проанализируйте, как свойства `position` и `z-index` влияют на порядок отображения элементов.

- Попробуйте изменить порядок или значения `z-index` и посмотрите, как это влияет на общий вид страницы.

- Добавьте дополнительные элементы или стили для улучшения дизайна.

7. ****Документация****:

- Запишите свои наблюдения о том, как обтекание текста изображением и свойства позиционирования влияют на макет страницы.

- Сохраните все файлы и подготовьте их к демонстрации.

Тема 16. Адаптивная верстка сайтов. Синтаксис написания медиа запросов
Практическая работа №18: Напишите HTML и CSS для страницы, которая будет адаптироваться под разные размеры экранов. Используйте медиа запросы для изменения стилей (например, изменение размера шрифтов, скрытие/показ элементов) при ширине экрана менее 768px и 480px.

Цель: изучить и освоить принципы создания адаптивных веб-страниц с использованием HTML и CSS. Вы научитесь применять медиа-запросы для изменения стилей в зависимости от ширины экрана, что позволит вашему контенту корректно отображаться на различных устройствах, таких как смартфоны, планшеты и настольные компьютеры

ХОД РАБОТЫ:

1. ****Создание HTML-страницы****:

- Создайте файл `index.html`.

- Включите в страницу основные элементы, такие как заголовки, абзацы, изображения и кнопки. Используйте семантические теги, такие как ``, ``, ``, `

`, `` для структурирования контента.

2. ****Стилизация с помощью CSS****:

- Создайте файл `styles.css`.

- Добавьте базовые стили для улучшения внешнего вида страницы, включая:

- Шрифты и размеры текста.

- Отступы и поля.

- Цвета фона и текста.

- Убедитесь, что элементы имеют разумные размеры и отступы для отображения на больших экранах.

3. ****Использование медиа-запросов****:

- Добавьте медиа-запросы в файл `styles.css` для изменения стилей при различных ширинах экрана:

- Для экранов шириной менее 768px:
 - Измените размеры шрифтов для заголовков и абзацев.
 - Скрывайте или показывайте определенные элементы (например, навигационное меню или кнопки).
 - Измените отступы и размеры изображений для лучшего отображения.
- Для экранов шириной менее 480px:
 - Дополнительно измените стили, чтобы обеспечить оптимальное отображение на мобильных устройствах.
 - Убедитесь, что текст остаётся читабельным, а элементы не перекрываются.

4. ****Тестирование страницы****:

- Откройте файл `index.html` в веб-браузере.
- Проверьте, как страница адаптируется при изменении ширины окна браузера.
- Убедитесь, что все элементы отображаются корректно и удобно для чтения на различных устройствах.

5. ****Анализ и улучшение****:

- Проанализируйте, как изменения, внесенные с помощью медиа-запросов, влияют на внешний вид и функциональность страницы.
- Попробуйте добавить дополнительные медиа-запросы для других размеров экранов или улучшить существующие стили.

6. ****Документация****:

- Запишите свои наблюдения о том, как медиа-запросы помогают создавать адаптивный дизайн.
- Сохраните все файлы и подготовьте их к демонстрации.

Тема 17. Построение сеток. Фреймворк Bootstrap

Практическая работа №19: Создайте многостраничный сайт с использованием Bootstrap. (Используйте сетку Bootstrap для создания главной страницы и нескольких дополнительных страниц. Включите навигационное меню, заголовки, текст и изображения, используя классы Bootstrap для адаптивности.) Добавьте компоненты Bootstrap, такие как карточки, кнопки и модальные окна.

Цель: изучить и освоить принципы создания многостраничного сайта с использованием фреймворка Bootstrap. Вы научитесь использовать сетку Bootstrap для создания адаптивной структуры страниц, а также добавлять различные компоненты, такие как навигационное меню, карточки, кнопки и модальные окна, для улучшения пользовательского интерфейса.

ХОД РАБОТЫ:

1. ****Создание структуры проекта****:

- Создайте папку для вашего проекта и внутри нее создайте следующие файлы:
 - `index.html` (главная страница)
 - `about.html` (дополнительная страница)
 - `contact.html` (страница контактов)
 - `styles.css` (файл для ваших собственных стилей, если необходимо)

2. ****Подключение Bootstrap****:

- Включите Bootstrap в ваши HTML-файлы. Вы можете использовать CDN (Content Delivery Network) для подключения CSS и JS файлов Bootstrap. Добавьте ссылки в ``<head>`` вашего HTML.

3. ****Создание навигационного меню****:

- Используйте компоненты Bootstrap для создания навигационного меню, которое будет присутствовать на всех страницах. Убедитесь, что оно адаптивно и корректно отображается на мобильных устройствах.

4. ****Создание главной страницы (`index.html`)****:

- Используйте сетку Bootstrap для размещения заголовков, текста и изображений на главной странице.

- Добавьте карточки с информацией о вашем сайте или услугах. Используйте классы Bootstrap для стилизации карточек.

- Включите кнопки, которые ведут на дополнительные страницы (например, "Узнать больше", "Контакты").

5. ****Создание дополнительных страниц****:

- На странице `about.html` добавьте информацию о вас или вашей компании. Используйте заголовки, текст и изображения.

- На странице `contact.html` создайте форму для обратной связи с полями для имени, email и сообщения. Используйте компоненты Bootstrap для стилизации формы.

6. ****Добавление модальных окон****:

- На любой из страниц добавьте кнопку, которая будет открывать модальное окно с дополнительной информацией или формой. Используйте компоненты Bootstrap для создания модальных окон.

7. ****Тестирование сайта****:

- Откройте каждую страницу в веб-браузере и проверьте, как они выглядят и функционируют.

- Убедитесь, что навигация работает корректно и все элементы адаптивны.

8. ****Анализ и улучшение****:

- Проанализируйте, как использование Bootstrap улучшает адаптивность и внешний вид вашего сайта.

- Попробуйте добавить дополнительные компоненты Bootstrap, такие как слайдеры, карусели или другие элементы, чтобы улучшить функциональность сайта.

9. ****Документация****:

- Запишите свои наблюдения о том, как Bootstrap помогает в создании адаптивного дизайна и улучшении пользовательского интерфейса.

- Сохраните все файлы и подготовьте их к демонстрации.

Тема 18. Верстка макета страницы с использованием Flexbox

Практическая работа №20: Верстка макета страницы на основе предоставленного дизайна. По макету (например, в формате PSD или Figma) создайте страницы с помощью HTML и CSS, используя Flexbox для расположения элементов. Добавьте интерактивные элементы, такие как кнопки и ссылки, и проверьте их функциональность

Цель: научиться верстать страницы на основе предоставленного дизайна, используя HTML и CSS, а также применять Flexbox для расположения элементов. Вы освоите процесс преобразования графического макета в адаптивную веб-страницу, а также добавление интерактивных элементов, таких как кнопки и ссылки.

ХОД РАБОТЫ:

1. ****Верстка макета страницы****:
 - Получите макет страницы в формате PSD или Figma.
 - Проанализируйте дизайн, обратите внимание на структуру, цвета, шрифты и расположение элементов.
2. ****Создание страниц с использованием HTML и CSS****:
 - Создайте HTML-файл для вашей страницы и подключите к нему CSS-файл для стилизации.
 - Используйте Flexbox для расположения элементов на странице, чтобы обеспечить адаптивность и гибкость дизайна.
 - Разделите страницу на семантические блоки (например, заголовок, основное содержимое, боковая панель, подвал) и используйте соответствующие HTML-теги.
3. ****Добавление интерактивных элементов****:
 - Включите интерактивные элементы, такие как кнопки и ссылки, на вашу страницу.
 - Убедитесь, что кнопки имеют соответствующий стиль и функциональность (например, переход на другую страницу или выполнение какого-либо действия).
 - Проверьте работоспособность всех интерактивных элементов в браузере.
4. ****Тестирование и отладка****:
 - Откройте вашу страницу в различных браузерах и на разных устройствах, чтобы убедиться в корректности отображения и функциональности.
 - Проверьте, что элементы правильно адаптируются к различным размерам экрана.
5. ****Документация****:
 - Подготовьте файлы к демонстрации, включая HTML, CSS и любые используемые изображения.

Тема 19. Адаптивная верстка сайта

Практическая работа №21: Разработайте сайт, который будет корректно отображаться на мобильных, планшетных и десктопных устройствах.

Цель: разработать адаптивный веб-сайт, который будет корректно отображаться на мобильных, планшетных и десктопных устройствах. Вы научитесь применять современные подходы к веб-разработке, включая использование адаптивного дизайна, CSS Flexbox и медиа-запросов, чтобы обеспечить удобство и доступность контента для пользователей на различных устройствах.

ХОД РАБОТЫ:

Вариант 1: Задание

1. ****Планирование и проектирование****:
 - Определите цель вашего сайта и целевую аудиторию.
 - Создайте прототип или макет сайта, используя инструменты для дизайна (например, Figma или Adobe XD), учитывая адаптивность интерфейса.
2. ****Создание структуры сайта****:
 - Создайте HTML-страницы для вашего сайта, включая главную страницу, страницу "О нас", страницу услуг и страницу контактов.
 - Используйте семантические HTML-теги для улучшения структуры и SEO.
3. ****Стилизация с помощью CSS****:

- Разработайте CSS-стили для вашего сайта, используя Flexbox для расположения элементов.
 - Примените медиа-запросы, чтобы адаптировать стили под различные размеры экранов (мобильные, планшеты и десктопы).
4. ****Добавление интерактивных элементов****:
- Включите интерактивные элементы, такие как кнопки, формы и ссылки, которые будут работать на всех устройствах.
 - Убедитесь, что все элементы имеют правильный стиль и функциональность.
5. ****Тестирование и отладка****:
- Проверьте отображение вашего сайта на различных устройствах и браузерах.
 - Убедитесь, что все элементы корректно адаптируются к разным размерам экранов и работают без ошибок.

Вариант 2: Задание

1. ****Исследование и анализ****:
- Изучите примеры адаптивных сайтов и определите лучшие практики для их разработки.
 - Определите ключевые функциональные элементы вашего сайта и создайте список необходимых страниц.
2. ****Разработка макета****:
- Создайте макет вашего сайта в графическом редакторе, учитывая различные размеры экранов.
 - Определите, как будет выглядеть навигационное меню, заголовки, текст и изображения на разных устройствах.
3. ****Верстка сайта****:
- Напишите HTML-код для всех страниц вашего сайта, следуя структуре, определенной в макете.
 - Используйте CSS для стилизации элементов и Flexbox для обеспечения гибкости и адаптивности.
4. ****Интеграция функциональности****:
- Добавьте интерактивные элементы, такие как слайдеры, модальные окна или формы, чтобы улучшить пользовательский опыт.
 - Убедитесь, что все элементы работают корректно на всех устройствах.
5. ****Финальное тестирование****:
- Проведите тестирование на различных устройствах и браузерах, чтобы убедиться, что сайт отображается и функционирует правильно.
 - Соберите отзывы от пользователей и внесите необходимые изменения для улучшения сайта.

2.2 Задания для промежуточного контроля

Контрольная работа

Варианты тем одностраничного сайта:

1. ****Лендинг для стартапа****: Сайт, который представляет новый продукт или услугу, включает описание, преимущества, отзывы пользователей и кнопку для регистрации или покупки.
2. ****Событие или фестиваль****: Одностраничный сайт для анонса мероприятия с информацией о дате, месте, программе и формой регистрации участников.
3. ****Персональное портфолио****: Сайт для представления работ фрилансера или специалиста, включающий разделы о себе, проекты и контактные данные.
4. ****Кулинарный рецепт****: Одностраничный сайт, посвященный одному рецепту, с пошаговыми инструкциями, фотографиями и возможностью оставить комментарии.
5. ****Промо-страница для приложения****: Сайт, который описывает функции и преимущества мобильного приложения, включает скриншоты и кнопку для загрузки.
6. ****Сайт для благотворительной акции****: Информация о проекте, цели сбора средств, способы пожертвования и отзывы тех, кто уже получил помощь.
7. ****Курс по саморазвитию****: Лендинг, который предлагает онлайн-курс, включает описание курса, отзывы участников и кнопку для регистрации.
8. ****Рекламная страница для нового продукта****: Одностраничный сайт, который фокусируется на одном продукте, с описанием, характеристиками и кнопкой для покупки.
9. ****Сайт для свадебного планировщика****: Информация о услугах свадебного организатора, примеры работ, отзывы клиентов и форма для запроса консультации.
10. ****Сайт для фитнес-тренера****: Одностраничный сайт, где тренер предлагает свои услуги, включает расписание тренировок, отзывы клиентов и возможность записаться на занятия.

Практические задания к дифференцированному зачёту:

Варианты тем для верстки сайта

1. ****Портфолио дизайнера****: Создайте сайт для представления работ графического или веб-дизайнера, включая галерею проектов, раздел "О себе" и контактную форму.
2. ****Блог о путешествиях****: Разработайте блог, где автор делится своими путешествиями, включает фотографии, советы и маршруты, а также возможность комментирования постов.
3. ****Интернет-магазин****: Создайте адаптивный сайт для интернет-магазина, где пользователи могут просматривать товары, добавлять их в корзину и оформлять заказы.
4. ****Кулинарный сайт****: Разработайте сайт с рецептами, где пользователи могут искать блюда по ингредиентам, оставлять отзывы и делиться своими рецептами.
5. ****Сайт для фитнес-клуба****: Создайте сайт для фитнес-клуба с расписанием тренировок, информацией о тренерах, ценами на абонементы и формой для записи на занятия.
6. ****Образовательная платформа****: Разработайте сайт для онлайн-курсов, где пользователи могут просматривать курсы, регистрироваться и оставлять отзывы о пройденных занятиях.
7. ****Сайт для некоммерческой организации****: Создайте сайт для НКО, который включает информацию о миссии, проектах, возможности пожертвований и раздел для волонтеров.
8. ****Музыкальный сайт****: Разработайте сайт для музыкальной группы или исполнителя, включающий биографию, дискографию, даты концертов и возможность покупки билетов.
9. ****Сайт для агентства недвижимости****: Создайте сайт для агентства, где пользователи могут просматривать объявления о продаже и аренде недвижимости, фильтровать по параметрам и запрашивать информацию.
10. ****Сайт для мероприятий****: Разработайте сайт для организации мероприятий (например, конференций или фестивалей), включающий расписание, информацию о спикерах, регистрацию и новости.

3. Критерии оценивания

Критерии оценивания выполнения заданий практических занятий и лабораторных работ

оценка «отлично» выставляется студенту, если он выполнил работу на 90%, при защите работы студент показывает глубокое знание вопросов темы

оценка «хорошо» выставляется студенту, если он выполнил работу на 75%, при защите работы студент без затруднений отвечает на вопросы

оценка «удовлетворительно» выставляется студенту, если он выполнил работу на 60% с незначительными ошибками; при защите показывает слабое знание вопросов темы, не всегда дает исчерпывающие и обоснованные вопросы на заданные вопросы, допускает существенные ошибки

оценка «неудовлетворительно» работа не выполнена.

Критерии оценивания контрольной работы

оценка «отлично» выставляется студенту, если он выполнил работу на 90%, при защите работы студент показывает глубокое знание вопросов темы

оценка «хорошо» выставляется студенту, если он выполнил работу на 75%, при защите работы студент без затруднений отвечает на вопросы

оценка «удовлетворительно» выставляется студенту, если он выполнил работу на 60% с незначительными ошибками; при защите показывает слабое знание вопросов темы, не всегда дает исчерпывающие и обоснованные вопросы на заданные вопросы, допускает существенные ошибки

оценка «неудовлетворительно» работа не выполнена.

Критерии оценивания промежуточной аттестации (дифференцированный зачет)

Оценка "отлично" –

1. Глубокое и прочное усвоение программного материала.
2. Точность и обоснованность выводов.
3. Безошибочное выполнение практического задания.
4. Точные, полные и логичные ответы на дополнительные вопросы.

Оценка "хорошо" –

1. Хорошее знание программного материала.
2. Недостаточно полное изложение теоретического вопроса экзаменационного билета.
3. Наличие незначительных неточностей в употреблении терминов, классификаций.
4. Неполнота представленного иллюстративного материала.
5. Точность и обоснованность выводов.
6. Логичное изложение вопроса, соответствие изложения научному стилю.
7. Негрубая ошибка при выполнении практического задания.

Оценка "удовлетворительно" –

1. Поверхностное усвоение программного материала.
2. Недостаточно полное изложение теоретического вопроса экзаменационного билета.
3. Затруднение в приведении примеров, подтверждающих теоретические положения.

4. Наличие неточностей в употреблении терминов, классификаций.
5. Неумение четко сформулировать выводы.
6. Отсутствие навыков научного стиля изложения.
7. Грубая ошибка в практическом задании.
8. Неточные ответы на дополнительные вопросы.

Оценка *"неудовлетворительно"* –

1. Незнание значительной части программного материала.
2. Неумение выделить главное, сделать выводы и обобщения.
3. Грубые ошибки при выполнении практического задания.
4. Неправильные ответы на дополнительные вопросы.