

Рассмотрено на

Методическом совете

Протокол № 01/23

« 17 » февраля 2023 год



« Утверждаю »

Директор БТК КГТУ

К.К.Келебаев

102

2023 год

## РАБОЧАЯ ПРОГРАММА по организации курса «Веб-верстка»

Разработана: инженером-программистом

Эстебес уулу Адилетом

## Программа учебной дисциплины «Веб-вёрстка»

Автор	Эстебес уулу Адилет
Учебная нагрузка (всего)	72 час
Курс, ОП	Веб-вёрстка
Формат изучения дисциплины	Оффлайн

### 1. Цель, результаты освоения дисциплины и пререквизиты

Цель освоения дисциплины «Веб-вёрстка» – научить слушателей применять базовые навыки на верстке веб-сайта и веб-предложении.

Обучающие познакомятся со стандартами Web 2.0, научатся работать с макетами и форматировать код, освоят адаптивную верстку, и по итогу — смогут создавать быстрые и удобные сайты, которые точно понравятся пользователям.

В результате освоения дисциплины обучающийся должен **уметь**:

- создавать и редактировать Интернет страницы;
- работать с HTML - кодом в программе Macromedia Dreamweave или аналогичной, используя стандартные объекты, рисование, редактирование уже созданных Интернет страниц
- создавать стили (CSS)
- работать со скриптами (JavaScript)
- создавать веб-сайты средней сложности;
- проводить формализацию лингвистических знаний, анализ и синтез лингвистических структур, квантитативный анализ текстовых данных с использованием математических знаний и методов;
- гибко адаптироваться к различным профессиональным ситуациям, проявлять творческий подход, инициативу и настойчивость в достижении целей профессиональной деятельности и личных.

- самостоятельно приобретать и использовать в практической деятельности новые знания и умения, стремиться к саморазвитию

В результате освоения дисциплины обучающийся должен **знать**:

- основные понятия HTML — кода;
- основные понятия CSS — кода;
- основные понятия JavaScript - кода;
- существующие способы построения Интернет страниц;
- основных средства создания и редактирования Интернет страниц с помощью средств операционной системы или специализированных программ.
- знать преимущества той или иной технологии / языка, используемого при разработке веб-приложений
- принципы работы современных веб-приложений и сети интернет в целом

В результате освоения дисциплины обучающийся должен **обладать**:

- основные понятия HTML — кода;
- основные понятия CSS — кода;
- основные понятия JavaScript - кода;
- навыками практического использования алгоритмов и структур данных при решении конкретных задач;
- навыками практического применения концепций объектно-ориентированного программирования;
- практической работы по созданию и разработке web-приложений;

Изучение дисциплины не требует специальных пререквизитов.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении дисциплин:

1. Научно-исследовательский семинар
2. Подготовка и защита выпускной квалификационной работы

## 2. Содержание учебной дисциплины

Тема (раздел дисциплины)	Объем в часах	Планируемые результаты обучения (ПРО), подлежащие контролю	Формы контроля
Тема 1. Основы вёрстки.	4 ч.	Узнаете основы веб-верстки: как разрабатывать статические веб-страницы и задавать стили элементам. Будете использовать редакторы кода с расширениями, помогающими в разработке. Например, Emmet	Домашнее задание
Тема 2. Язык гипертекстовой разметки HTML.	4 ч.	Урок посвящен HTML верстке с нуля. Говорим о роли атрибутов и изучаем общую схему описания HTML тегов. Основная цель любой HTML-вёрстки — передача смысла блоков. В этом уроке мы рассмотрим возможности последнего стандарта HTML5 в области семантики и узнаем о доступности в веб	Домашнее задание
Тема 3. Каскадные таблицы стилей. CSS.	4 ч.	Для визуального оформления веб-страницы создан язык CSS. Изучаем базовые возможности языка, разбираемся, как их использовать вместе с HTML. Учимся подключать CSS файлы и знакомимся с базовыми типами селекторов. Что такое каскадность, и как она работает в CSS? Урок посвящен различиям в приоритетах селекторов и умению использовать это в своих проектах.	Домашнее задание
Тема 4. Формы.	4 ч.	Формы — важный интерактивный элемент веб-страницы. Как и ссылки, формы обеспечивают взаимодействие пользователя и страницы, позволяя отправлять данные. Для этого в HTML существуют специальные конструкции, которые говорят браузеру, какие поля может использовать пользователь и как их обрабатывать. Для отделения формы от остальных участков вёрстки используется специальный тег <form>. Если его не использовать, то браузер не поймёт, какие именно данные нужно отправить.	Домашнее задание

		Ведь на странице может быть не одна, а сразу несколько различных форм.	
Тема 4. Flexbox.	6 ч.	CSS модуль раскладки Flexible Box, обычно называемый флексбокс или flexbox или просто flex (флекс), был разработан как модель одномерного-направленного макета и как один из методов распределения пространства между элементами в интерфейсе, с мощными возможностями выравнивания. В этой статье даётся краткое описание основных функций flexbox, которые мы рассмотрим более подробно в остальных руководствах.	Домашнее задание
Тема 5. CSS Grid.	6 ч.	Макет сетки (CSS Grid Layout) — это двумерная система сетки в CSS. Сетки подходят для размещения ключевых областей страницы и небольших элементов пользовательского интерфейса. В этой статье представлен обзор макета сетки и новой терминологии, которая является частью спецификации CSS Grid Layout Level 1. Эти функции будут описаны более подробно позже в приложении.	Домашнее задание
Тема 6. Анимация.	6 ч.	CSS3-анимация придаёт сайтам динамичность. Она оживляет веб-страницы, улучшая взаимодействие с пользователем. В отличие от CSS3-переходов, создание анимации базируется на ключевых кадрах, которые позволяют автоматически воспроизводить и повторять эффекты на протяжении заданного времени, а также останавливать анимацию внутри цикла.	Домашнее задание
Тема 7. Адаптивность и медиа элементы.	6 ч.	Регулировка разрешения экрана Частичное решение: делаем всё гибким Гибкие изображения Еще один способ: отзывчивые изображения Интересная фишка для iPhone Настраиваемая структура макета страницы Адаптивная вёрстка с помощью медиазапросов CSS3	Домашнее задание

Тема 7. JavaScript для верстальщика.	12 ч.	Основы, Выражения в JavaScript, Переменные, функция Циклы	Домашнее задание
Тема 8. Мини-проект в портфолио.	12 ч.	Итоге создайте свое собственное портфолио с этими изученными технологиями	Домашнее задание
<b>Итого часов:</b>	<b>72</b>		

### 3. Содержание разделов дисциплины

Тема (раздел дисциплины)	Планируемые результаты обучения (ПРО), подлежащие контролю
Тема 1. Основы вёрстки.	Узнаете основы веб-верстки: как разрабатывать статические веб-страницы и задавать стили элементам. Будете использовать редакторы кода с расширениями, помогающими в разработке. Например, Emmet
Тема 2. Язык гипертекстовой разметки HTML.	Урок посвящен HTML верстке с нуля. Говорим о роли атрибутов и изучаем общую схему описания HTML тегов. Основная цель любой HTML-вёрстки — передача смысла блоков. В этом уроке мы рассмотрим возможности последнего стандарта HTML5 в области семантики и узнаем о доступности в веб
Тема 3. Каскадные таблицы стилей. CSS.	Для визуального оформления веб-страницы создан язык CSS. Изучаем базовые возможности языка, разбираемся, как их использовать вместе с HTML. Учимся подключать CSS файлы и знакомимся с базовыми типами селекторов. Что такое каскадность, и как она работает в CSS? Урок посвящен различиям в приоритетах селекторов и умению использовать это в своих проектах.
Тема 4. Формы.	Формы — важный интерактивный элемент веб-страницы. Как и ссылки, формы обеспечивают взаимодействие пользователя и страницы, позволяя отправлять данные. Для этого в HTML существуют специальные конструкции, которые говорят браузеру, какие поля может использовать пользователь и как их обрабатывать. Для отделения формы от остальных участков вёрстки используется специальный тег <form>. Если его не использовать, то браузер не поймёт, какие именно данные нужно отправить. Ведь на странице может быть не одна, а сразу несколько различных форм.

Тема 4. Flexbox.	CSS модуль раскладки Flexible Box, обычно называемый флексбокс или flexbox или просто flex (флекс), был разработан как модель одномерного-направленного макета и как один из методов распределения пространства между элементами в интерфейсе, с мощными возможностями выравнивания. В этой статье даётся краткое описание основных функций flexbox, которые мы рассмотрим более подробно в остальных руководствах.
Тема 5. CSS Grid.	Макет сетки (CSS Grid Layout) — это двумерная система сетки в CSS. Сетки подходят для размещения ключевых областей страницы и небольших элементов пользовательского интерфейса. В этой статье представлен обзор макета сетки и новой терминологии, которая является частью спецификации CSS Grid Layout Level 1. Эти функции будут описаны более подробно позже в приложении.
Тема 6. Анимация.	CSS3-анимация придаёт сайтам динамичность. Она оживляет веб-страницы, улучшая взаимодействие с пользователем. В отличие от CSS3-переходов, создание анимации базируется на ключевых кадрах, которые позволяют автоматически воспроизводить и повторять эффекты на протяжении заданного времени, а также останавливать анимацию внутри цикла.
Тема 7. Адаптивность и медиа элементы.	<p>Регулировка разрешения экрана</p> <p>Частичное решение: делаем всё гибким</p> <p>Гибкие изображения</p> <p>Еще один способ: отзывчивые изображения</p> <p>Интересная фишка для iPhone</p> <p>Настраиваемая структура макета страницы</p> <p>Адаптивная верстка с помощью медиазапросов CSS3</p>
Тема 7. JavaScript для верстальщика.	<p>Основы, Выражения в JavaScript, Переменные, функция</p> <p>Циклы</p>
Тема 8. Мини-проект в портфолио.	Итоге создайте свое собственное портфолио с этими изученными технологиями

## 4. Оценивание

Оценки выставляются по десятибалльной шкале.

**Оценка 10** ставится за безошибочно выполненное задание, полностью соответствующее условию, то есть программу на языке Python3, которая при выполнении решает поставленную задачу, не выдает ошибок и предупреждений, учитывает все возможные варианты входных данных.

**Оценка 9** ставится за программу, содержащую незначительные недостатки.

**Оценка 8** ставится за программу, содержащую незначительные недостатки, способные повлиять на время выполнения задачи в худшую сторону.



**Оценка 7** ставится за программу, содержащую значительные недостатки кода.

**Оценка 6** ставится за программу, частично решающую поставленную задачу.

**Оценка 5** ставится за код, содержащий множественные ошибки в реализации и решающий только незначительную часть поставленной в условии задачи.

**Оценка 4** ставится за программу, которая не решает поставленную в условии задачу.

**Оценка 3** ставится за код, выполнение которого невозможно из-за допущенных в нем синтаксических ошибок.

**Оценки 2 и 1** ставятся за код, написание которого не закончено и выполнение которого невозможно из-за допущенных в нем синтаксических ошибок.

**Оценка 0** ставится за отсутствие решенного задания или за обнаруженный случай плагиата.

Итоговая оценка за курс состоит из оценок в рамках текущего контроля (домашние работы) и промежуточной аттестации (экзамен в виде показа самостоятельно выполненного индивидуального проекта) по следующей формуле:

$$\text{Одидциплина} = 0,4 \cdot \text{Оэкзамен} + 0,6 \cdot \text{Осам. работа}$$

Самостоятельная работа оценивается по выполненным домашним заданиям, которые студенты загружают в свои репозитории на веб-сервисе <https://github.com/>. Домашние задания, если явно не указано иное, необходимо выложить в репозиторий до 23:59 дня, предшествующего следующему семинару.

При оценивании программы в первую очередь обращается внимание на то, насколько её работа соответствует требованиям, описанным в задании. Программа, не запускающаяся из-за синтаксических ошибок, не может получить оценку выше 3 баллов. Баллы могут сниматься, в частности, за неточное выполнение задания. Во вторую очередь могут оцениваться оптимальность решения (в смысле времени работы программы и количества строк кода) и стиль.

Для получения положительной оценки необходимо решить задачи, написав программу на языке Python.

При обнаружении плагиата в домашнем или контрольном задании это задание получает оценку 0 баллов.

Оценка за домашние задания равна среднему арифметическому оценок за все домашние задания. Итоговая оценка выставляется по накопленной. Способ округления: арифметический.

Экзамен представляет собой показ работы — выполненного на языке Python индивидуального проекта.

Подлежат передаче только элементы контроля по темам 5, 7, 8, 9, 11, 14, 15. Элементы текущего контроля (домашняя работа) – в срок не позднее 14 дней после получения неудовлетворительной оценки. Экзамен – в порядке, установленном факультетом гуманитарных наук. Тематический состав КИМ-ов для передач не отличается от тематического состава КИМ-ов текущего



контроля и промежуточной аттестации.

## 5. Примеры оценочных средств

Примеры заданий:

1. Провести подготовку текстового корпуса для решения исследовательской задачи
2. Преобразовать корпусные данные в векторное представление numpy
3. Визуализировать сложные корпусные данные
4. Предложить средства языка Python, которые можно использовать для решения исследовательской задачи.
5. Провести векторный анализ корпусных данных, продемонстрировать сохранение векторных инвариантов  
Блокирующие элементы не предусмотрены.

## 5. Ресурсы

### 5.1. Рекомендуемая основная литература

б.№п/ п	Наименование
1.	Климов, А. П. JavaScript на примерах [Электронный ресурс] / А.П. Климов . ? 2-е изд., перераб. и доп. ? СПб.: БХВ-Петербург, 2009 ? 336 с.:
2.	Мацяшек, Л. А. Практическая программная инженерия на основе учебного примера [Электронный ресурс] / Л. А. Мацяшек, Б. Л. Лионг ; пер. с англ. - 2-е изд. (эл.). - М.: БИНОМ. Лаборатория знаний, 2012 - 956 с. <a href="http://e.lanbook.com/view/book/8766/">http://e.lanbook.com/view/book/8766/</a>
3.	Овчаренко, А. В. Ajax на примерах [Электронный ресурс] / А. В. Овчаренко. - СПб.: БХВ-Петербург, 2009 - 432 с. <a href="http://znanium.com/bookread.php?book=489766">http://znanium.com/bookread.php?book=489766</a>

### 5.2. Интернет-ресурсы:

Ajax на примерах - <http://znanium.com/bookread.php?book=489766>

JavaScript на примерах - <http://znanium.com/bookread.php?book=489238>

Python. Самое необходимое -

<http://znanium.com/bookread.php?book=354989>

Добавляем Ajax - <http://znanium.com/bookread.php?book=489654>

Практическая программная инженерия на основе учебного примера -

<http://znanium.com/bookread.php?book=477694>

### 5.3. Программное обеспечение

№ п/п	Наименование	Условия доступа/скачивания
1.	HTML, CSS, JavaScript	<i>Свободное лицензионное соглашение</i>

5.4. Профессиональные базы данных, информационные справочные системы, интернет-ресурсы (электронные образовательные ресурсы)

*Нет*

5.5. Материально-техническое обеспечение дисциплины

Компьютерный класс.

## **6. Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов**

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида, могут предлагаться следующие варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

1. *для лиц с нарушениями зрения:* в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

2. *для лиц с нарушениями слуха:* в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

3. *для лиц с нарушениями опорно-двигательного аппарата:* в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.