

Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ ФГБОУ ВО «КемГУ»
Дата и время: 2024-02-21 00:00:00
471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436
Федеральное государственное бюджетное
образовательное учреждение высшего образования
"Кемеровский государственный университет"
Новокузнецкий институт (филиал)

Факультет информатики, математики и экономики
Кафедра математики, физики и математического моделирования

А.Д. Ульянов

ИНТЕРНЕТ-ТЕХНОЛОГИИ

*Методические рекомендации по выполнению внеаудиторной самостоятельной работы
для обучающихся по направлениям подготовки*

*01.03.02 Прикладная математика и информатика, профиль «Математическое
моделирование и информационные технологии»*

*02.03.03 Математическое обеспечение и администрирование информационных систем,
профиль «Программное и математическое обеспечение информационных технологий»*

Часть 1

Новокузнецк

2020

Ульянов А.Д.

Интернет-технологии: методические рекомендации по выполнению внеаудиторной самостоятельной работы для студентов факультета информатики, математики и экономики, обучающихся по направлениям подготовки 01.03.02 Прикладная математика и информатика (уровень бакалавриата); 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата): в 2 ч. Ч 1. / А.Д. Ульянов. - Новокузнецкий ин-т (фил.) Кемеров. гос. ун-та. – Новокузнецк: НФИ КемГУ, 2020 – 11 с.

В настоящих методических указаниях для студентов представлены материалы по выполнению внеаудиторной самостоятельной работы по дисциплине «Интернет-технологии»: основные теоретические сведения в форме конспектов лекций с примерами решения типовых задач по темам «Информационные ресурсы коллективного пользования: производство и распространение, в том числе ресурсы для лиц с ограниченными возможностями здоровья», «Язык гипертекстовой разметки HTML», «Каскадные таблицы стилей CSS», «Введение в клиентское программирование. Язык Javascript», «Программирование на стороне сервера. Язык PHP», «Работа с данными. СУБД MS SQL», «Авторизация и безопасность», «Построение насыщенных веб-приложений», «Принципы проектирования веб-приложений», «Развёртывание приложений. Хостинг и поисковая оптимизация. Командная разработка». Также представлены банк задач для контрольных работ, методические рекомендации по решению и оформлению, оценивание работ в балльно-рейтинговой системе и список основной и дополнительной литературы.

Методические рекомендации предназначены для наиболее рациональной организации внеаудиторной самостоятельной работы студентов при подготовке к выполнению контрольных работ и теста.

Рекомендовано
на заседании кафедры
математики, физики и математического
моделирования
22 октября 2020г.
Заведующий кафедрой

 / Е.В. Решетникова

Ульянов А.Д., 2020
Федеральное государственное
бюджетное образовательное
учреждение высшего образования
«Кемеровский государственный
университет», Новокузнецкий
институт (филиал), 2020

Текст представлен в авторской редакции

Оглавление

ПРЕДИСЛОВИЕ	4
1. ЯЗЫК ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ HTML	5
1.1. Теоретические сведения.....	5
1.2. Банк заданий к контрольной работе №1 по разделу «Язык гипертекстовой разметки HTML»...	6
1.3. Особенности оценивания контрольной работы в балльно-рейтинговой системе.....	6
2. КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ CSS	7
2.1. Теоретические сведения.....	7
2.2 Банк заданий к контрольной работе №2 по разделу «Каскадные таблицы стилей CSS»	7
2.3. Особенности оценивания контрольной работы в балльно-рейтинговой системе.....	8
3. ВВЕДЕНИЕ В КЛИЕНТСКОЕ ПРОГРАММИРОВАНИЕ. ЯЗЫК JAVASCRIPT	9
3.1. Теоретические сведения.....	9
3.2 Банк заданий к контрольной работе №3 по разделу «Введение в клиентское программирование. Язык Javascript»	9
3.3. Особенности оценивания контрольной работы в балльно-рейтинговой системе.....	10
4. СПИСОК РЕКОМЕНДУЕМОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ.....	11

ПРЕДИСЛОВИЕ

Настоящие методические рекомендации адресованы студентам, получающим квалификацию бакалавр по направлениям подготовки: 01.03.02 Прикладная математика и информатика, профиль «Математическое моделирование и информационные технологии» и 02.03.03 Математическое обеспечение и администрирование информационных систем, профиль «Программное и математическое обеспечение информационных технологий» и направлены на оказание помощи студентам в подготовке к выполнению контрольных работ по темам «Язык гипертекстовой разметки HTML», «Каскадные таблицы стилей CSS» и «Введение в клиентское программирование. Язык Javascript» дисциплины «Интернет-технологии».

Современный мир сложно представить без сети интернета и его информационного поля. Основы этого явления были заложены еще во второй половине XX века, но развитие не останавливается и сегодня. Беспрепятственный доступ к большому объему информации из каждого дома обуславливает ускорение развития всего общества и каждого человека в отдельности.

Инфраструктура сети получила мощный толчок с удешевлением домашних персональных компьютеров (ПК), серверных компьютеров, активным развитием мобильных устройств. Вместе с тем, значительную популярность получила часть сети, ориентированная на рядовых пользователей и предоставляемые им цифровые услуги. Следствием этого стала возрастающая потребность в специалистах, имеющих навыки в проектировании и разработке программного обеспечения, реализующего доступ к информации и сервисам сети интернет, а именно - в разработчиках интерактивных веб-сайтов.

Наибольшая доля веб-сайтов данного сегмента сегодня реализуется посредством языка программирования PHP, в связке с языком гипертекстовой разметки HTML, каскадными таблицами стилей CSS и интерпретируемым языком Javascript. Для хранения и доступа к данным чаще всего используется СУБД MySQL. Следует отметить простоту изучения основ данных технологий, что позволяет быстро начинать разрабатывать новые веб-сайты.

Данные методические материалы позволяют студенту подготовиться к практическим занятиям и выполнению контрольных работ по соответствующим темам. Методические рекомендации могут оказаться полезными при написании курсовых и выпускных квалификационных работ.

1. ЯЗЫК ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ HTML

1.1. Теоретические сведения

Информация в интернете чаще всего представляется в виде текста. Для лучшего восприятия текста с экрана его принято форматировать и окрашивать, вставлять мультимедийный контент и т.д. Для формирования каркаса форматирования текста используется язык HTML. HTML не является языком программирования, так как на нем невозможно описать алгоритмы. В первую очередь это язык разметки текста. Иными словами это средство описания представления контента в окне браузера посредством тегов и их атрибутов.

Тегами называют символьные конструкции, представленные в виде текста, взятого в угловые скобки. Отметим, что бывают теги открывающиеся и закрывающиеся, включающие содержимое (`<ИМЯ_ТЕГА>содержимое</ИМЯ_ТЕГА>`) и закрытые теги, не включающие содержимого (`<ИМЯ_ТЕГА/>`). Любой тег может включать именованные свойства (атрибуты) в своем объявлении (`<ИМЯ_ТЕГА атрибут1="значение атрибута">`). Атрибуты могут быть обязательными и необязательными.

Отметим, что совокупность тегов документа формирует древовидную структуру, также называемую DOM-деревом (document object model). Для страниц существуют требования к верхним узлам дерева: тег 'html' является корневым для всего дерева, тег 'html' включает теги 'head' и 'body', тег 'head' включает тег 'title'. Тег 'body' включает визуальное содержимое страницы. Содержимое тега 'head' не включено в процесс построения визуальной части страницы, но служит для хранения мета-данных о странице (например: 'title' - заголовок страницы).

Теги могут иметь различные названия и атрибуты, но для большинства задач описания текста достаточно стандартных имен тегов и их атрибутов. Для лучшего понимания приведем примеры.

Существуют двойные теги форматирования шрифта текста: 'b' (жирный), 'i' (наклонные), 'u' (подчеркивание). Текст, заключенный в данные теги будет представлен соответствующим образом.

Теги семейства 'h' ('h1', 'h2', 'h3', 'h4') представляют текст заголовка страницы или раздела соответственного уровня.

Теги семейства 'table' ('table', 'tr', 'td', 'th', 'thead', 'tbody') служат для формирования таблиц на странице. До появления разнообразия устройств с различным размером экрана это семейство активно использовалось для позиционирования контента на странице. Вместе с активным внедрением смартфонов и планшетных ПК разработчиком пришлось отказаться от использования таблиц, так как их размер фиксирован. На смену таблицам в форматировании и позиционировании (верстке) пришли блоки (блочная верстка).

Тег 'div' служит для разделения контента на блоки. Внутренняя верстка блоков реализуется атрибутами тега.

Рассмотрим тег 'a'. Его назначение - предоставление ссылки на страницу сайта или якорь на странице. Основным атрибутом тега 'a' является 'href' - ссылка на страницу с возможными GET параметрами в ней. Тег может включать содержимое между открывающим и закрывающим частями тега. Содержимое будет представлено на итоговой странице в браузере в виде ссылки. В базовом виде такое представление выражено в изменении курсора мыши и выделении синим цветом (если тело ссылки это текст).

Важное значение для веб-страниц имеет тег 'form'. Он существует для интерактивного ввода данных и их отправки на серверную часть приложения для обработки. Внутри тега 'form' содержатся теги 'input' различного вида. Вид (тип данных) определяется атрибутом 'type'. Наиболее часто значения могут быть следующими:

- 1) text - текстовое поле ввода,
- 2) hidden - скрытое текстовое поле ввода,
- 3) file - поле для ввода файла.

Теги 'input' могут быть включены во вложенные теги объекта 'form' для позиционирования, согласно задумке разработчика.

Важной частью тега 'form' является атрибуты 'action' и 'method'. Значением атрибута 'action' является относительная или абсолютная ссылка на ресурс, обрабатывающий запросы http, отправляемые данной формой (конечная точка запроса). Значение атрибута 'method' служит для определения 'метода' запроса к серверу. Наиболее часто встречаются методы 'get' и 'post'.

При работе с тегом 'form' следует остановить внимание на атрибуте 'enctype'. Его значение может быть 3х видов: 'application/x-www-form-urlencoded', 'multipart/form-data', 'text/plain'. Каждый вид определяет способ кодирования запроса при отправке на сервер браузером:

- 1) text/plain - введенный в поля текст не подлежит преобразованиям,
- 2) application/x-www-form-urlencoded - текст полей преобразуется как параметры из url (все символы, кроме латиницы будут заменены на соответствующий код),
- 3) multipart/form-data - все пересылаемые данные не будут преобразованы, но будут отправляться частями.

Значение multipart/form-data используется, когда необходимо отправить файл на сервер. Тогда загрузка будет производится порциями байт без автоматической трансформации.

1.2. Банк заданий к контрольной работе №1 по разделу «Язык гипертекстовой разметки HTML»

1. Разработать HTML документ, включающий табличные данные.
2. Разработать HTML документ, включающий форму отправки данных.
3. Разработать HTML документ, включающий блочную структуру (с использованием тегов div).

1.3. Особенности оценивания контрольной работы в балльно-рейтинговой системе

Контрольная работа по разделу «Язык гипертекстовой разметки HTML» является промежуточной формой контроля знаний студентов и представляет собой письменное выполнение определенных заданий. Она предназначена для проверки знаний студентов по учебной дисциплине «Интернет-технологии», а также служит для закрепления полученных знаний, умений и навыков. В контрольной работе студентам предлагаются задачи, сформулированные на основании материала, изложенного в лекциях проработанного на практических занятиях или самостоятельно изученного студентами. Перед тем как приступить к выполнению контрольной работы, студентам следует ознакомиться с теоретическим материалом и разобраться с разобранными в нем типовыми задачами.

Варианты контрольной работы состоят из трех заданий, которые представлены в разделе "Банк заданий к контрольной работе №1 по разделу "Язык гипертекстовой разметки HTML". Варианты отличаются тематикой разрабатываемого HTML-документа. Тематика определяется каждым субъектом уникально, и закрепляется за ним на все оставшиеся контрольные работы.

Система оценивания заданий контрольной работы №1 представлена в таблице 1.1.

Таблица 1.1 Оценивание контрольной работы №1 в БРС.

Критерии оценивания заданий	Количество баллов
Логично и последовательно выполнены все шаги решения, рассуждения имеют четкое обоснование.	15
Ход решения задания верный, но аргументация неполная.	10
Нарушена логическая цепочка рассуждений, решение неполное.	5
Максимальное количество баллов за контрольную работу №1	15

Оформление контрольной работы должно соответствовать Правилам оформления учебных работ студентов¹, принятым в НФИ КемГУ.

2. КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ CSS

2.1. Теоретические сведения

Как уже упоминалось, все теги могут включать атрибуты. Некоторые из них используются для позиционирования и оформления содержимого тегов. Наиболее универсальным атрибутом является 'style' (стиль). Значение атрибута 'style' является структурой данных, которая включает информацию о: цвете, шрифте, видимости и т.д. Отмечу, что значение данного атрибута распространяется на все вложенные теги и текст.

Часто элементы веб страниц могут обладать схожими свойствами. Это приводит к дублированию атрибутов, что осложняет разработку и поддержку страниц и увеличивает интернет трафик. Для решения этих проблем существует CSS (cascade style sheet) - каскадные таблицы стилей. CSS таблица представляет собой отдельный файл, включающий информацию о форматировании тегов в виде именованных блоков - классов css. Каждый css-класс может быть установлен в атрибут 'class' любого тега. Таким образом, все параметры представления описанные в классе будут присвоены неявно свойству style тега.

Именованные наборы свойств могут быть не просто именованными, а сразу быть присвоенными некоторому тегу по его селектору (id, name, rel...). Например, если на всех страницах обязательно присутствует тег с атрибутом 'id' со значением 'container', то имя устанавливается в виде '#container'. Это укажет браузеру, что этот набор свойств применится к тегу, обладающему таким значением атрибута id.

Файлы css подключаются в секции 'head' страницы html следующим образом:

```
<link rel="stylesheet" src="/path/to/file.css"/>.
```

Для такого подключения файла со стилями всеми современными браузерами поддерживается кеширование (локальное сохранение). Поэтому если файл был единожды загружен на одной из страниц, то для других страниц этого же узла, повторная загрузка производится не будет. Это следствие кеширования.

Рассмотрим примеры свойств тегов. Свойство 'color' определяет цвет текста элемента. Свойство 'background-color' определяет цвет или изображение фона элемента. Свойство 'width' определяет ширину элемента в пикселях или процентах от ширины страницы. Отметим, что свойство 'width' продублировано одноименным атрибутом, и может быть установлено именно туда. Но не рекомендуется использовать свойства стиля, так как такой формат удобнее всего.

Пример стиля css представлен ниже:

```
.className {  
width: 100px;  
color: #cccccc;  
}.
```

Символ '.' указывает на то что имя этих атрибутов может быть присвоено атрибуту class. Если префиксом будет являться '#', то имя это id элемента. В случае отсутствия префикса, набор свойств будет присвоен тегу, согласно его имени. Например набор атрибутов:

```
a{  
width: 100px;  
color: #cccccc;  
} будет присвоен каждой ссылке на странице.
```

2.2 Банк заданий к контрольной работе №2 по разделу «Каскадные таблицы стилей CSS»

1. В полученную в результате выполнения контрольной работы №1 html страницу необходимо добавить ссылку на файл css.
2. Файл css необходимо наполнить именованными стилями (классами css).
3. Расставить классы в свойство 'class' тегов.

2.3. Особенности оценивания контрольной работы в балльно-рейтинговой системе

Контрольная работа по разделу «Каскадные таблицы стилей CSS» является промежуточной формой контроля знаний студентов и представляет собой письменное выполнение определенных заданий. Она предназначена для проверки знаний студентов по учебной дисциплине «Интернет-технологии», а также служит для закрепления полученных знаний, умений и навыков. В контрольной работе студентам предлагаются задачи, сформулированные на основании материала, изложенного в лекциях проработанного на практических занятиях или самостоятельно изученного студентами. Перед тем как приступить к выполнению контрольной работы, студентам следует ознакомиться с теоретическим материалом и разобраться с разобранными в нем типовыми задачами.

Варианты контрольной работы состоят из трех заданий, которые представлены в разделе "Банк заданий к контрольной работе №2 по разделу " Каскадные таблицы стилей CSS". Варианты отличаются тематикой разрабатываемого HTML-документа. Тематика определяется каждым субъектом уникально, и закрепляется за ним на все оставшиеся контрольные работы.

Система оценивания заданий контрольной работы №2 представлена в таблице 2.1.

Таблица 2.1 Оценивание контрольной работы №2 в БРС.

Критерии оценивания заданий	Количество баллов
Логично и последовательно выполнены все шаги решения, рассуждения имеют четкое обоснование.	15
Ход решения задания верный, но аргументация неполная.	10
Нарушена логическая цепочка рассуждений, решение неполное.	5
Максимальное количество баллов за контрольную работу №1	15

Оформление контрольной работы должно соответствовать Правилам оформления учебных работ студентов¹, принятым в НФИ КемГУ.

3. ВВЕДЕНИЕ В КЛИЕНТСКОЕ ПРОГРАММИРОВАНИЕ. ЯЗЫК JAVASCRIPT

3.1. Теоретические сведения

Согласно сложившейся традиции, разработчики веб сайтов различными способами стараются сократить объем трафика между клиентом и сервером. Один из широко применяемых способов состоит в переносе части бизнес-логики и логики представления данных на сторону клиента (в браузер пользователя). Этот подход реализован посредством встроенного интерпретатора языка Javascript в браузер. Возможности языка не ограничены вычислениями и обработкой данных. Реализована интеграция с DOM-деревом страницы (DOM - document object model). Таким образом, функции языка Javascript, скрипты которого пересылаются вместе со всем документом, могут быть привязаны к событиям на форме страницы.

Код Javascript содержится внутри тега 'script' в двух видах: прямое включение кода в тело тега, или включение файла скрипта (*.js) посредством атрибута 'src'. Отметим, что при использовании второго варианта включения браузеры автоматически кешируют получаемый скрипт.

Выполнение скрипта происходит последовательно сверху вниз, даже если он разделен и определяется сразу в нескольких тегах 'script'. Важной особенностью выполнения является то, что оно происходит сразу, по мере загрузки документа в браузер. Следствием этого является то, что некоторые фрагменты dom-дерева могут быть еще не загружены в момент выполнения скрипта, что может привести к ошибкам времени выполнения. Поэтому скрипты инициализации элементов страницы следует располагать в нижней части страницы.

Скрипт может включать не только прямые инструкции, которые будут выполнены последовательно, но и функции, код которых может быть выполнен в процессе жизненного цикла страницы. Для задания событий пользовательского ввода используются атрибуты тегов, имена которых начинаются с 'on'. Наприме: 'onchange' (изменение значения), 'onclick' (клик левой кнопкой мыши), 'onblur' (потеря фокуса элементов), 'onsubmit' (отправка формы) и т.д. Для этого в соответствующий атрибут необходимо записать имя функции, которую требуется вызвать при соответствующем событии.

Как в любом языке программирования, функция должна быть определена выше того места, где используется. Кроме того, если функция вызывается по событию, то внутри ее тела будет доступно имя this. Эта переменная указывает на элемент dom-дерева, событие которого вызывает данную функцию. Следует быть внимательным при использовании this, так как если вызывать функцию не через событие а в произвольном коде, то имя this будет неопределено.

Пример скрипта и его использования:

```
<script>
let a = 0;
function incAndShowA(){
    alert(++a);
}
</script>
<input type="text" onblur="incAndShowA();"/>
```

3.2 Банк заданий к контрольной работе №3 по разделу «Введение в клиентское программирование. Язык Javascript»

1. В полученную в результате выполнения контрольной работы №1 html страницу необходимо добавить ссылку на файл js.
2. Файл js необходимо наполнить функциями Javascript.
3. Расставить в обработчики событий вызовы определенных функций Javascript.

3.3. Особенности оценивания контрольной работы в балльно-рейтинговой системе

Контрольная работа по разделу «Введение в клиентское программирование. Язык Javascript» является промежуточной формой контроля знаний студентов и представляет собой письменное выполнение определенных заданий. Она предназначена для проверки знаний студентов по учебной дисциплине «Интернет-технологии», а также служит для закрепления полученных знаний, умений и навыков. В контрольной работе студентам предлагаются задачи, сформулированные на основании материала, изложенного в лекциях проработанного на практических занятиях или самостоятельно изученного студентами. Перед тем как приступить к выполнению контрольной работы, студентам следует ознакомиться с теоретическим материалом и разобраться с разобранными в нем типовыми задачами.

Варианты контрольной работы состоят из трех заданий, которые представлены в разделе "Банк заданий к контрольной работе №3 по разделу " Введение в клиентское программирование. Язык Javascript". Варианты отличаются тематикой разрабатываемого HTML-документа. Тематика определяется каждым субъектом уникально, и закрепляется за ним на все оставшиеся контрольные работы.

Система оценивания заданий контрольной работы №3 представлена в таблице 3.1.

Таблица 3.1 Оценивание контрольной работы №3 в БРС.

Критерии оценивания заданий	Количество баллов
Логично и последовательно выполнены все шаги решения, рассуждения имеют четкое обоснование.	15
Ход решения задания верный, но аргументация неполная.	10
Нарушена логическая цепочка рассуждений, решение неполное.	5
Максимальное количество баллов за контрольную работу №1	15

Оформление контрольной работы должно соответствовать Правилам оформления учебных работ студентов¹, принятым в НФИ КемГУ.

4. СПИСОК РЕКОМЕНДУЕМОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ

Основная учебная литература:

1. Селина, Е. Г. Организация интерактивного взаимодействия в HTML-документах : учебно-методическое пособие / Е. Г. Селина. — Санкт-Петербург : НИУ ИТМО, 2018. — 35 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/136467> (дата обращения: 05.10.2020). — Режим доступа: для авториз. пользователей.
2. Беликова, С.А. Основы HTML и CSS: проектирование и дизайн веб-сайтов: учебное пособие по курсу «Web-разработка» : [16+] / С.А. Беликова, А.Н. Беликов ; Южный федеральный университет. — Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2020. — 176 с. : ил. — Режим доступа: по подписке. — URL: <http://biblioclub.ru/index.php?page=book&id=598663> — Библиогр. в кн. — ISBN 978-5-9275-3435-7. — Текст : электронный.

Дополнительная учебная литература:

1. Титов, В.А. Разработка WEB-сайта средствами языка HTML : учебное пособие / В.А. Титов, Г.И. Пещеров ; Институт мировых цивилизаций. — Москва : Институт мировых цивилизаций, 2018. — 184 с. : ил., табл. — Режим доступа: по подписке. — URL: <http://biblioclub.ru/index.php?page=book&id=598475> — Библиогр. в кн. — ISBN 978-5-9500469-3-3. — Текст : электронный.

Литература для оформления учебных работ:

Правила оформления учебных работ студентов : учебно-методическое пособие / И.А. Жибинова, А.Е. Аракелян, О.В. Соколова, Ю.Н. СоинаКутищева. — Новокузнецк : НФИ КемГУ, 2018. — 124 с. — Текст : непосредственный