

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

Кафедра систем автоматизированного проектирования и информационных
систем

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ
ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ «МЕТОДЫ
РАЗМЕТКИ НА WEB-ДОКУМЕНТОВ НА ОСНОВЕ HTML5»**

*для обучающихся по направлению 09.03.02 «Информационные системы и
технологии», профиль «Разработка web-ориентированных
информационных систем» всех форм обучения*

Воронеж 2022

УДК 658.512:621(07)

ББК 30.18:85.1:34.5я7

Составители: Э. Р. Саргсян, Н. А. Рындин

Методические рекомендации по выполнению лабораторных работ по дисциплине «Методы разметки на Web-документов на основе HTML5» для обучающихся по направлению 09.03.02 «Информационных системы и технологии», профиль «Разработка web-ориентированных информационных систем» всех форм обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост.: Э. Р. Саргсян, Н. А. Рындин – Воронеж: Изд-во ВГТУ, 2021. – 33 с.

Приводится описание выполнения лабораторных работ по курсу «Методы разметки на Web-документов на основе HTML5» для студентов обучающихся по направлению 09.03.02 «Информационных системы и технологии», профиль «Разработка web-ориентированных информационных систем» всех форм обучения.

УДК 658.512:621(07)
ББК 30.18:85.1:34.5я7

Рецензент -

Рекомендовано методическим семинаром кафедры САПРИС и методической комиссией ФИТКБ Воронежского государственного технического университета в качестве методических материалов

ВВЕДЕНИЕ

Историю развития языка HTML можно сравнить с детективным рассказом в том, что в ней есть свой неожиданный поворот событий, в результате которых появилась новая версия языка — HTML5.

Предполагалось, что язык HTML уйдет в небытие, не дожив до XXI столетия. Организация W3C (World Wide Web Consortium - Консорциум Всемирной паутины), которая занимается разработкой и внедрением официальных стандартов Всемирной паутины, забросила язык HTML в далеком 1998 г., считая его не способным на дальнейшее выживание.

Свои надежды на будущее консорциум W3C возлагал на модернизированного наследника HTML — язык XHTML. Но язык HTML не умер. Его "подобрала" группа программистов-аутсайдеров и не только возвратила его к жизни, но и заложила основу для новых возможностей, которые мы с вами и исследуем.

Как вы знаете, HTML — это язык для создания веб-страниц. Ключевая идея языка HTML - организация содержимого с помощью элементов, не претерпела никаких изменений с самых ранних времен Всемирной паутины. Более того, даже очень старые веб-страницы без проблем обрабатываются в наиболее современных браузерах (включая и те, которые не существовали на момент создания этих страниц, например Firefox или Chrome).

Но успех и почтенный возраст также несут с собой определенные существенные угрозы. Что и случилось с языком HTML — в 1998 г. консорциум W3C прекратил его поддержку и попытался заменить его языком на основе языка XML - XHTML 1.0.

XHTML 1.0

В стандарте XHTML используются те же синтаксические соглашения, что и HTML, но в нем ужесточены требования к следованию установленным правилам. Большая часть отступлений от правил разметки, которые сходят с

рук в традиционном HTML, попросту неприемлемы в XHTML.

Например, допустим, что вы хотите выделить курсивом последнее слово заголовка следующим образом:

```
<h1>История языка <i>HTML5</i></h1>
```

Но при этом вы случайно поменяли местами два последних тега:

```
<h1>История языка <i>HTML5</h1></i>
```

Когда браузер сталкивается с этой слегка подпорченной разметкой, он в состоянии "понять", что вы действительно имели в виду, и без малейших претензий выделяет последнее слово курсивом. Но несовпадающие теги нарушают официальные правила XHTML. Если проверить эту страницу в валидаторе формата XHTML (или открыть ее в какой-либо программе для разработки веб-сайтов, например Dreamweaver), то будет выведено сообщение, указывающее на эту ошибку.

В отношении разработки веб-сайтов это полезно, т. к. позволяет отловить небольшие ошибки, вследствие которых страница может отображаться по-разному в различных браузерах, или которые могут вызвать более серьезные проблемы при попытке отредактировать страницу с целью ее улучшения.

Поначалу XHTML пользовался большим успехом. Профессиональные веб-разработчики, раздосадованные индивидуальными особенностями отображения страниц браузерами и вседозволенностью в области веб-разработки, скопом переходили на XHTML. При этом они были вынуждены совершенствовать свои навыки работы и отказываться от использования значительного количества полусырых возможностей форматирования, предоставляемых HTML.

Но многие из предполагаемых преимуществ XHTML — такие как функциональная совместимость с инструментами XML, облегчение обработки страниц автоматизированными программами, переносимость на мобильные платформы, а также расширяемость самого языка XHTML — так никогда и не материализовались.

Тем не менее XHTML стал стандартом для большинства серьезных веб-разработчиков. Но в то время, как все казались достаточно довольными этим языком разметки, у него был один скелет в шкафу: хотя браузеры понимали разметку XHTML, они не обеспечивали строгую проверку отступлений от правил, требуемую этим стандартом. Это означает, что страница может быть создана с нарушением правил XHTML, но браузеры и глазом не моргнут при ее обработке.

Более того ничто не могло помешать веб-разработчику бросить в одну кучу небрежно написанную разметку и устаревшее HTML-содержимое и назвать все это страницей XHTML. Ни один браузер на планете не имел бы никаких претензий при обработке такой страницы. И это обстоятельство вызывало у людей, ответственных за стандарт XHTML, глубокое чувство тревоги.

XHTML 2: неожиданный провал

Эта проблема должна была, по идее, быть решенной в следующей версии — XHTML 2. В ней нужно было ужесточить правила обработки ошибок, которые бы заставляли браузеры не обрабатывать страницы, не отвечающие стандарту XHTML 2. В XHTML 2 также избавились от многих странностей и соглашений, унаследованных от HTML.

Например, система нумерации заголовков (`<h1>`, `<h2>`, `<h3>` и т. д.) была заменена одним элементом `<h>` с уровнем обозначаемого им заголовка, зависящего от местонахождения этого элемента в веб-странице. Подобным образом элемент `<a>` был заменен возможностью, позволяющей веб-разработчикам преобразовывать любой элемент в ссылку, а вместо атрибута

alt элемента `` был предложен новый способ предоставления альтернативного содержимого.

Изменения такого рода были типичными для XHTML 2. С теоретической точки зрения, они делали язык более аккуратным и были более понятными. Но с практической стороны, они вынуждали всех изменить свой подход к созданию веб-страниц (не говоря об обновлении уже созданных веб-страниц), не добавляя при этом никакой новой функциональности, чтобы оправдать всю эту работу. Попутно из XHTML 2 было удалено несколько удобных элементов, которые все еще пользовались популярностью среди веб-разработчиков, таких как `` (полужирный текст), `<i>` (текст курсивом) и `<iframe>` (для вложения одной веб-страницы в другую).

Но, возможно, худшей проблемой оказалась чрезвычайно медленная скорость внесения изменений. Разработка XHTML 2 тащилась в течение пяти лет, и интерес разработчиков к этому стандарту медленно угасал.

HTML5: возвращение к жизни

Приблизительно в то же самое время (начиная с 2004 г.) группа разработчиков начала рассматривать будущее Всемирной паутины в другом ракурсе. Вместо того чтобы попытаться разобраться, что было неправильным (или просто "грязным" с философической точки зрения) в HTML, они сфокусировались на том, чего в нем не хватало, что хотели бы иметь веб-разработчики для воплощения своих идей.

В конце концов, HTML зародился как инструмент для отображения документов. С добавлением языка сценариев JavaScript, HTML преобразовался в систему для разработки веб-приложений, таких как поисковые движки, онлайн-магазины, картографические инструменты, средства чтения электронной почты и многие другие. Но в то время как искусное веб-приложение может делать много чего впечатляющего, создать такое приложение — задача не из легких. Большинство разработчиков использует для этого мешанину кода JavaScript, один или несколько популярных

инструментариев JavaScript, а также веб-приложение, исполняемое на веб-сервере. Добиться правильного и единообразного взаимодействия всех этих составляющих на разных браузерах — сложная задача.

Такая ситуация вызывала особенную озабоченность среди разработчиков браузеров, поэтому группа дальновидных разработчиков из компании Opera Software (создатели браузера Opera) и компании Mozilla Foundation (создатели браузера Firefox) начали агитировать за включение в XHTML больше возможностей, ориентированных на разработчиков. Когда их попытки не увенчались успехом, компании Opera, Mozilla и Apple создали группу WHATWG (Web Hypertext Application Technology Working Group - рабочая группа по технологии гипертекстовых веб-приложений) с целью работы над новыми решениями.

Группа не ставила перед собой задачу заменить HTML, ее целью было плавное расширение языка, и причем такое, чтобы расширения были обратно совместимыми. Надо сказать, что самая ранняя версия работы этой группы включала две спецификации расширений — Web Applications 1.0 и Web Forms 2.0. В конечном итоге эти стандарты эволюционировали в HTML5.

Предполагается, что число 5 в названии HTML5 означает: данный стандарт является продолжением стандарта HTML (последней версией стандарта HTML перед XHTML была версия 4.01). Это, конечно же, не совсем верно, т. к. HTML5 поддерживает все разработки, существовавшие в области создания веб-страниц в течение десяти лет после выпуска HTML 4.01, включая строгий синтаксис в стиле XHTML (если разработчики желают использовать его), а также множество инноваций для JavaScript. Тем не менее это название делает ясным следующее: язык HTML5 может поддерживать соглашения XHTML, но требует следования правилам HTML.

В 2000-х все интересное происходило в лагере группы WHATWG. После некоторого периода болезненных размышлений организация W3C решила распустить работающую над XHTML 2 группу и работать вместо этого над формализацией стандарта HTML5. На этом этапе первоначальный стандарт

HTML5 был разделен на более управляемые части, и многие из его функциональных возможностей стали отдельными стандартами.

Что входит в состав HTML5?

Браузеров, "поддерживающих" HTML5, не существует. Вместо этого, каждый браузер поддерживает постепенно расширяющееся подмножество возможностей HTML5. Этот подход является как хорошим, так и плохим. Хорош он потому, что браузеры могут быстро реализовать готовые части стандарта HTML5, пока другие возможности продолжают развиваться. А плох он тем, что заставляет веб-разработчиков беспокоиться о проверке, поддерживает ли конкретный браузер каждую функциональную возможность, которую они хотят использовать.

Далее приводится список и короткое описание основных категорий функциональных возможностей, охватываемых HTML5:

Ядро HTML5

Эта часть HTML5 составляет официальную версию спецификации организации W3C. Она содержит новые семантические элементы, новые и усовершенствованные элементы управления для веб-форм, поддержку аудио и видео, а также холст для рисования с помощью JavaScript. В эту категорию входит большинство функциональных возможностей, которые наилучшим образом поддерживаются браузерами.

Ранние возможности HTML5

Это возможности, которые были реализованы в первоначальной спецификации HTML5, подготовленной группой WHATWG. Большинство из них — это спецификации для возможностей, требующих JavaScript и поддерживающих развитые веб-приложения. Наиболее важными являются локальное хранение данных, приложения, работающие в автономном режиме, и обмен сообщениями.

Возможности, иногда называемые HTML5

Это возможности следующего поколения, которые часто считаются частью HTML5, хотя они никогда не входили в стандарт HTML5. Эта категория включает спецификацию CSS3 и геолокацию.

В результате перехода поддержки HTML сначала от организации W3C к группе WHATWG, а потом обратно, возникла довольно необычная ситуация. Технически организация W3C отвечает за определение, что является официальным HTML5, а что — нет. Но в то же самое время группа WHATWG продолжает свою работу, придумывая будущие возможности HTML. Только теперь она называет его не HTML5, а просто HTML, объясняя это тем, что HTML будет продолжать существовать, как живой язык.

Так как HTML является живым языком, то HTML-страница никогда не устареет и не перестанет работать. Для HTML-страниц никогда не потребуется номер версии (даже в блоке указания типа документа `<doctype>`), а веб-разработчикам никогда не понадобится обновлять свою разметку от одной версии языка к другой, чтобы она работала на новых браузерах.

Так как HTML — это живой язык, новые возможности (и новые элементы) можно добавлять к стандарту HTML в любое время. Некоторые разработчики могут решить использовать эти возможности в своих веб-страницах, а некоторые разработчики веб-браузеров — поддерживать их в своих продуктах. Но возможности никогда не будут привязаны к какой-либо конкретной версии стандарта.

Порядок выполнения работы

Задание 1

Составить свое резюме по следующему плану (в общей сложности около десяти строчек). Вместо многоточий и пояснений в скобках вставить свои данные: Я, ... (ФИО), родился ... (дата и год рождения) в городе ... (место, где родились). В 20... году окончил ... школу (лицей, гимназию) с углубленным изучением ..., средний балл аттестата ..., сумма баллов по ЕГЭ ... В 20... году поступил на факультет ... ИХиБТ НИУ ИТМО. Средний балл по первым сессиям (аттестациям) ... В свободное от учебы время увлекаюсь ... (перечислить не менее трех увлечений помимо учебы). Мои приятели в группе: ... (перечислить не менее трех фамилий из группы).

Выполнение

I. Прочтите общие сведения о языке HTML ([1] п.п. 1.1, 1.2).

II. Подготовьте бланк для HTML-кода Вашего резюме. Для этого следует:

1) выполнить команду Пуск Программы Стандартные Блокнот Файл Сохранить как...;

2) в окне Сохранение документа открыть папку своей группы. В поле Имя стереть стандартный шаблон *.txt и ввести Резюме Фамилия.htm (вместо Фамилия, естественно, должна стоять Ваша фамилия), нажать кнопку Сохранить. Если все сделано правильно, то в папке Вашей группы появится этот файл со значком обозревателя, в противном случае – со значком блокнота. Тогда следует удалить созданный файл и повторить все снова;

3) открыть созданный файл и в окне программы Explorer выполнить команду Вид Просмотр HTML-кода. Программа Explorer откроет созданный Вами блокнот.

III. Введите в созданный документ стандартные теги:

Теги набираются в английском регистре. Для ускорения и облегчения набора желательно активно пользоваться командами копирования и переноса информации. Подробнее об этих тегах см. [1] п. 1.5. IV. Внутри контейнера

напечатайте свое резюме. Выполните команду Файл Сохранить. V.

Посмотрите, как выглядит набранный текст без «украшательств». Для этого следует открыть созданный файл из папки своей группы, затем сжать размеры окна Explorer. Обратите внимание на информацию в заголовке окна Explorer и в поле Адрес, а также на то, как компоуются строки текста при изменении размеров окна, в которое он выдается. Вывод. Explorer компоует строки документа в своем окне, не обращая внимание на их компоовку в исходном коде. Он самостоятельно компоует документ так, чтобы его можно было читать без горизонтальной полосы прокрутки. Поэтому при записи кода в блокноте можно клавишей свободно разрывать длинные строки и вставлять пустые строки между смысловыми частями документа 5 так, чтобы было удобно просматривать именно код, а не его отображение в окне Explorer

VI. Проведите первую коррекцию кода. Для этого следует:

- 1) не закрывая Explorer, перейти в окно Блокнота;
- 2) вставить в контейнер ... контейнер (вместо ФИО, естественно, должна стоять Ваша фамилия);
- 3) выполнить команду Файл Сохранить;
- 4) перейти в окно Explorer;
- 5) нажать кнопку Обновить. Посмотрите, как повлиял тег.

Если какой-то атрибут «не сработал», значит, допущены ошибки в написании английских слов или пропущены пробелы между атрибутами. Следует найти ошибки и повторить цикл коррекции кода. Итак, один цикл коррекции кода включает следующие действия: внесение и сохранение исправлений в окне блокнота, переход в окно Explorer и обновление его информации кнопкой Обновить или командой Вид Обновить. В дальнейших пунктах инструкции будут просто указываться изменения, которые следует внести в код, а все остальные действия перечисляться не будут.

VIII. Проведите третью коррекцию кода. Для этого следует:

- 1) после раздела «ФИО, где и когда родился», поставить тег абзаца; \
- 2) после раздела «какую школу кончил, средний балл аттестата, сумма баллов по ЕГЭ» – тег разрыва строки;

3) перед разделом «увлечения помимо учебы» вставить в код тег горизонтальной линии

X. Измените вид шрифтов в разных частях резюме. Для этого следует контейнером <H1> выделить свою фамилию, контейнером <I> – город, где Вы родились, контейнером <H6> – информацию о школе, контейнером с атрибутами SIZE=7 COLOR=RED – информацию о факультете, контейнером с атрибутами SIZE=1 COLOR=GREEN – информацию о друзьях, контейнером выделить какое-нибудь увлечение.

XI. Оформите часть текста в виде списка. Для этого следует контейнером выделить блок об увлечениях, разбить его на пункты тегами. Аналогично контейнером и тегами оформить в виде маркированного списка перечень Ваших друзей. Посмотрите, как выглядят стандартные списки. XII. Измените нумерацию в списке. Для этого с помощью атрибутов TYPE и START поменяйте нумерацию в списке увлечений.

Задание 2.

Создать HTML-документ, в котором представлены сведения о гостиницах для размещения участников некоторой конференции. Эти сведения оформить в виде таблицы:

Характеристики гостиниц

Название	Адрес	Тип номера	Цена за день
Центральная	Пр. Мира, д. 4	Люкс одноместный	\$150
		Люкс двухместный	\$250
		Одноместный	\$80
Спортивная	Бульвар Профсоюзов, д. 15	Одноместный	\$50
		Двухместный	\$80
Вокзальная	Привокзальная пл., д. 3	Четырехместный	\$50

Выполнение I.

Подготовьте бланк для HTML-документа. Для этого следует:

1) выполнить команду Пуск Программы Стандартные Блокнот Файл

Сохранить как...;

2) в окне Сохранение документа открыть папку своей группы. В поле Имя ввести Таблица Фамилия.htm (вместо Фамилия, естественно, должна стоять Ваша фамилия), нажать кнопку Сохранить. Если все сделано правильно, то в папке Вашей группы появится этот файл со значком обозревателя, в противном случае – со значком блокнота. Тогда следует удалить созданный файл и повторить все снова;

3) откройте созданный файл и в окне программы EXPLORER выполните команду Вид Просмотр HTML-кода. Программа Explorer откроет созданный Вами блокнот.

II. Введите в созданный документ стандартные теги:

```
<HTML>  
<HEAD>  
<TITLE> Таблица ФИО </TITLE>  
</HEAD>  
<BODY>  
</BODY>  
</HTML>
```

III. Внутри контейнера <BODY> вставьте скелетные теги таблицы:

```
<TABLE>  
<CAPTION>  
    Характеристики гостиниц  
</CAPTION>  
  
</TABLE>
```

IV. Заполните первую строку таблицы. Каждую строку таблицы следует оформлять внутри контейнера <TR>, который вкладывается между тегам </CAPTION> и </TABLE>. Первая строка содержит названия столбцов, для ячеек используйте контейнеры

<TH> ... </TH>. Окончательно контейнер первой строки выглядит так:

```
<TR>  
    <TH> Название </TH> <TH> Адрес </TH>  
    <TH> Тип номера </TH> <TH> Цена за день </TH>  
</TR>
```

Замечание. Для облегчения набора текста можно сначала ввести только первый контейнер Название, затем скопировать его три раза и в копиях заменить только текст внутри контейнера.

V. Заполните вторую строку таблицы. Контейнер второй строки должен располагаться после контейнера первой строки. С этой строки начинается содержание таблицы, для ее ячеек используйте контейнеры `<TD> ... </TD>`. Первая и вторая ячейки по высоте занимают 3 строки таблицы. Чтобы объединить их, добавьте в их контейнеры атрибуты `ROWSPAN=3`. При наборе здесь тоже удобно пользоваться копированием контейнера первой ячейки, а для третьей и четвертой ячеек просто удалить не нужный им атрибут `ROWSPAN=3`. Окончательно контейнер второй строки выглядит так:

```
<TR>
  <TD ROWSPAN=3> Центральная </TD>
  <TD ROWSPAN=3> Пр. Мира, д. 4</TD>
  <TD> Люкс одноместный </TD> <TD> $150 </TD>
</TR>
```

VI. Заполните третью строку таблицы. Контейнер третьей строки должен располагаться после контейнера второй строки. Для ячеек по-прежнему используйте контейнеры `<TD> ... </TD>`. Первая и вторая ячейки входят в объединенные раньше блоки, их объявлять не надо. А третья и четвертая ячейки содержат новую информацию. Скопируйте контейнеры третьей и четвертой ячеек из второй строки и поменяйте в них содержимое. Окончательно контейнер этой строки выглядит так:

```
<TR>
  <TD> Люкс двухместный </TD> <TD> $250 </TD>
</TR>
```

VII. Аналогично п. VI создайте четвертую строку. Сохраните документ и посмотрите его через Explorer. В таблице пока нет разлиновки, но, если все сделано правильно, структура расположения информации должна просматриваться. Если нет – исправляйте ошибки.

VIII. Добавьте в таблицу сведения о гостинице «Спортивная». Они

располагаются в пятой и шестой строках. Похожие строки находятся во второй и третьей строках таблицы, и теги, создающие их, уже сформированы. Скопируйте теги из второй и третьей строк соответственно в контейнеры пятой и шестой строк, замените содержание ячеек. Так как здесь первая и вторая ячейки по высоте занимают только 2 строки, поменяйте значение атрибута ROWSPAN. Получаем для новых строк:

```
<TR>
  <TD ROWSPAN=2> Спортивная </TD>
  <TD ROWSPAN=2> Бульвар Профсоюзов, д. 15
</TD> <TD> одноместный </TD> <TD> $50 </TD>
</TR>
<TR>
  <TD> двухместный </TD> <TD> $80 </TD>
</TR>
```

IX. Добавьте в таблицу сведения о гостинице «Вокзальная». Они располагаются в седьмой строке. В ней нет объединения ячеек, все четыре ячейки содержат новую информацию. Можно скопировать теги второй строки, убрать в них атрибуты ROWSPAN и просто заменить информацию в контейнерах ячеек:

```
<TR>
  <TD > Вокзальная </TD>
  <TD> Привокзальная пл., д. 3 </TD>
  <TD> Четырехместный </TD> <TD> $50 </TD>
</TR>
```

X. Форматирование таблицы. Добавьте в тег атрибуты BORDER=3, WIDTH=70%, ALIGN=CENTER, BGCOLOR=YELLOW, BORDERCOLOR=MAROON (можно взять свои значения атрибутов). Посмотрите через Explorer, как действуют эти атрибуты на вид таблицы. Добавьте атрибут CELLSPACING=0, потом измените его на CELLSPACING=10. Посмотрите через браузер, определите, за что отвечает этот атрибут.

XI. Форматирование ячеек. Выделите сведения, относящиеся к разным гостиницам, разной заливкой. Выровняйте по центру ячеек цены номеров. Необходимые для этого сведения – в [1] п. 1.10 и Приложении.

СПИСОК ЛИТЕРАТУРЫ

1. Айзенменгер, Р. HTML 3.2/4.0. Справочник / Р. Айзенменгер. - М.: Бином, 1998. - 368 с.
2. Гаевский, А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2008. - 464 с.
3. Гудман, Д. JavaScript и DHTML. Сборник рецептов. Для профессионалов / Д. Гудман. - М.: Питер, 2004. - 523 с.
4. Дебольт HTML и CSS. Совместное использование / Дебольт, Вирджиния. - М.: ИТ Пресс, 2006. - 512 с.
5. Дунаев, В.В. HTML, скрипты и стили / В.В. Дунаев. - М.: СПб: БХВ, 2006. - 832 с.
6. Закарян, И. Что такое Internet, www и html. Первое знакомство / И. Закарян, В Рафалович,. - М.: ИНФРА-М, 1998. - 212 с.
7. Исси, Коэн Полный справочник по HTML, CSS и JavaScript / Коэн Исси , Лазаро; Исси Коэн, Джозеф. - М.: ЭКОМ Паблшерз, 2007. - 695 с.
8. Кисленко, Н.П. HTML. Самое необходимое / Н.П. Кисленко. - М.: БХВ-Петербург, 2008. - 352 с.
9. Петюшкин, А. HTML экспресс-курс / А. Петюшкин. - М.: СПб: БХВ-Петербург, 2004. - 250 с.
10. Пфаффенбергер HTML, XHTML и CSS. Библия пользователя / Пфаффенбергер и др. - М.: Вильямс; Издание 3-е, 2007. - 752 с.
11. Хольцшлаг, Молли Э. Использование HTML 4. Специальное издание (+ CD - ROM) / Хольцшлаг Молли Э.. - М.: Вильямс, 2001. - 190 с.

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ
ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ «Методы разметки на
Web-документов на основе HTML5»**

*для обучающихся по направлению 09.03.02
«Информационные системы и технологии», профиль
«Разработка web-ориентированных информационных
систем» всех форм обучения*

Составители:

Саргсян Эрик Ромович

Рындин Никита Александрович

Подписано в печать

Формат 60x84 1/8 Бумага для множительных аппаратов

Уч.-изд. л. 3,3 Усл. печ. л. 3,0.

ФГБОУ ВО «Воронежский государственный технический
университет» 396026 Воронеж, Московский просп., 14

Участок оперативной полиграфии издательства
ВГТУ 396026 Воронеж, Московский просп., 14