

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
БОРИСОГЛЕБСКИЙ ФИЛИАЛ
(БФ ФГБОУ ВО «ВГУ»)

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
Компьютерные сети, Интернет и мультимедиа технологии

Методические указания для обучающихся по освоению дисциплины

Приступая к изучению учебной дисциплины, прежде всего обучающиеся должны ознакомиться с учебной программой дисциплины. Электронный вариант рабочей программы размещён на сайте БФ ВГУ.

Знание основных положений, отраженных в рабочей программе дисциплины, поможет обучающимся ориентироваться в изучаемом курсе, осознавать место и роль изучаемой дисциплины в подготовке будущего педагога, строить свою работу в соответствии с требованиями, заложенными в программе.

Основными формами контактной работы по дисциплине являются лекции и лабораторные занятия, посещение которых обязательно для всех студентов (кроме студентов, обучающихся по индивидуальному плану).

В ходе лекционных занятий следует не только слушать излагаемый материал и кратко его конспектировать, но очень важно участвовать в анализе примеров, предлагаемых преподавателем, в рассмотрении и решении проблемных вопросов, выносимых на обсуждение. Необходимо критически осмысливать предлагаемый материал, задавать вопросы как уточняющего характера, помогающие уяснить отдельные излагаемые положения, так и вопросы продуктивного типа, направленные на расширение и углубление сведений по изучаемой теме, на выявление недостаточно освещенных вопросов, слабых мест в аргументации и т.п.

Не следует дословно записывать лекцию, лучше попытаться понять логику изложения и выделить наиболее важные положения лекции в виде опорного конспекта. Рекомендуется использовать различные формы выделения наиболее сложного, нового, непонятного материала, который требует дополнительной проработки: можно пометить его знаком вопроса (или записать на полях сам вопрос), цветом, размером букв и т.п. – это поможет быстро найти материал, вызвавший трудности, и в конце лекции (или сразу же, попутно) задать вопрос преподавателю (не следует оставлять непонятый материал без дополнительной проработки, без него иногда бывает невозможно понять последующие темы). Материал уже знакомый или понятный нуждается в меньшей детализации – это поможет сэкономить усилия во время конспектирования.

В ходе выполнения лабораторных работ необходимо не просто внимательно читать методические указания к работам и аккуратно выполнять все задания и упражнения, но и обращать внимание на сложные моменты (в тексте они выделены и снабжены отдельным примечанием), внимательно анализируя текст примечания и приведённые примеры, при необходимости экспериментировав и обращаясь к справочникам.

При подготовке к промежуточной аттестации необходимо повторить пройденный материал в соответствии с учебной программой, примерным перечнем вопросов, выносившихся на зачет. Рекомендуется использовать конспекты лекций и источники, перечисленные в списке литературы в рабочей программе дисциплины, а также ресурсы электронно-библиотечных систем. Необходимо обратить особое внимание на темы учебных занятий, пропущенных по разным причинам. При необходимости можно обратиться за консультацией и методической помощью к преподавателю.

План лекций по дисциплине

План лекций по дисциплине «Компьютерные сети, Интернет и мультимедиа технологии»

Тема №1. Общие сведения о компьютерных сетях

- Понятие и назначение компьютерных сетей
- История появления и развития компьютерных сетей.
- Классификация компьютерных сетей.

Тема №2. Организация компьютерных сетей

- Принципы организации компьютерных сетей.
- Стандартизация компьютерных сетей. Модель OSI. Стек TCP/IP. Стандарты IEEE 802.

Тема №3. Физическая среда передачи данных

- Сетевые кабели.
- Беспроводная среда.

Тема №4. Сетевые устройства

- Сетевые устройства, работающие на различных уровнях модели OSI.

Тема №5. Технологии локальных сетей

- Технологии на разделяемой среде.
- Проводные локальные сети.
- Беспроводные локальные сети.

Тема №6. Адресация в сетях TCP/IP

- Аппаратные адреса.
- Сетевые адреса.
- Символьные адреса

Тематика рефератов

1. Новые возможности HTML5;
2. Новые возможности CSS3;
3. Обзор бесплатных хостингов;
4. Обзор ресурсов для веб-разработчика
5. Обзор онлайн-инструментов для веб-разработчика.
6. Сравнение разных видов вёрстки веб-страниц (с примерами реальных сайтов).
7. Инструменты разработчика в современных браузерах.
8. Использование CSS3 для создания веб-презентаций (с созданием собственного примера веб-презентации +3 балла).
9. «Облачные» сервисы.

Методические рекомендации по выполнению лабораторной работы «Основы HTML»

HTML (англ. HyperText Markup Language) — язык разметки гипертекста — универсальный язык разметки документов во *Всемирной паутине* (англ. World Wide Web). При помощи языка HTML (или XHTML) созданы большинство документов Всемирной паутины (*веб-страниц*).

HTML реализует средства, позволяющие публиковать документы, содержащие текст, таблицы, списки, изображения и другие элементы, переходить от одного документа к другому по гиперссылкам, включать видео и аудиоклипы и другие приложения прямо в документы.

По своей сути HTML-файл — это простой текстовый документ, содержащий специальные структурные и семантические команды — *теги* (*дескрипторы*). Размеченный с их помощью файл интерпретируется специальными программами — *браузерами* (*веб-обозревателями* — от англ. Web browser) и отображается в виде документа, в удобной для человека форме.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). Согласно современной идеологии веб-разработки, содержание документа должно быть отделено от его представления.

Чтобы документы HTML одинаково отображались в разных браузерах и на разных устройствах, необходимо придерживаться общепринятого стандарта. Разработкой и внедрением стандартов языка HTML (а также других стандартов Всемирной паутины) занимается *Консорциум Всемирной паутины* (англ. World Wide Web Consortium, W3C). Первая версия языка HTML была разработана британским учёным Тимом Бернерсом-Ли приблизительно в начале девяностых годов XX века.

Долгое время основной версией языка являлась спецификация HTML 4.01, принятая ещё 24 декабря 1999 года. Работа над новой версией языка, получившей название HTML 5, официально начата W3C в 2007 году. К этому моменту действующая

спецификация языка уже сильно отставала от требований развивающейся Всемирной паутины, и новая версия языка стала широко использоваться и поддерживаться основными браузерами ещё до окончания её разработки. Официально стандарт HTML 5 утвержден 28 октября 2014 года. С 1 ноября 2016 года рекомендована к применению версия HTML 5.1.

В рамках лабораторных работ будет рассматриваться версия HTML 5.1.

1. Создание простейшего HTML-документа

Ресурс Всемирной паутины — *веб-сайт* — по сути, представляет собой структурированный набор веб-страниц, связанных гиперссылками.

Прежде чем публиковать ресурс во Всемирной паутине, его, как правило, создают локально (на собственном компьютере), уже на этапе разработки организуя желаемую структуру сайта. Все файлы будущего сайта обычно хранятся в отдельном каталоге.

1.1. Создание первой web-страницы

1.1.1. Создадим директорию для разрабатываемого сайта.

Создайте в «Моих документах» свой каталог (например, названный вашей фамилией), а в него поместите папку, в которой вы будете хранить файлы вашего сайта (например, «Site»).

Обратите внимание! Для именования папки (а также всех файлов и папок, которые вы будете создавать в дальнейшем) используйте только латиницу и цифры, и не используйте кириллицу и пробелы.

Как было отмечено выше, HTML-документ — это простой текстовый файл, поэтому для его создания можно воспользоваться любым простым текстовым редактором — например, поставляемым с операционной системой Windows редактором Блокнот. Первую простейшую HTML страницу создадим именно с его помощью.

1.1.2. Откройте программу Блокнот (как правило, располагается в главном меню в разделе «Программы» → «Стандартные»).

Как уже было упомянуто, в языке HTML для разметки текста используются дескрипторы (теги). *Теги* представляют собой команды, записанные с помощью латиницы (и, иногда, цифр) и заключенные в угловые скобки «<» и «>» (например, тег абзаца: <p>).

Большинство используемых тегов являются *парными*, т.е. текст, на который действует тег, располагается между *открывающим* (начальным) и *закрывающим* (конечным) тегом, который отличается от открывающего наличием знака «/» перед именем. Например, парный тег, определяющий *элемент* страницы «абзац»: <p>Привет!</p>.

В некоторых случаях употребление закрывающего тега не обязательно, а иногда вообще не требуется — такие теги называются *одиночными* (автономными). Заметим, что в XHTML (расширение HTML на основе XML) все теги обязательно должны быть закрыты.

Регистр букв, используемых для записи HTML тегов, не важен (в отличие от XHTML), но рекомендуется использовать строчные буквы.

Обратите внимание! Если используется несколько парных тегов один внутри другого, то они должны быть строго вложенными друг в друга, «перехлест» элементов недопустим!

1.1.3. Создайте заготовку для HTML-страницы.

Все содержимое HTML-страницы должно размещаться между тегами <html> и </html>. Наберите в Блокноте:

```
<html>
</html>
```

Документ HTML состоит из *заголовочной части* («шапки»), задаваемой тегами <head> и </head> и *тела документа*, заключенного между тегами <body> и </body>. Название документа, а также другая информация о документе содержится в заголовочной части, а содержимое документа находится в его теле.

Добавьте эти элементы в вашу заготовку. Должна получиться следующая минимальная структура HTML документа:

```
<html>
  <head>

  </head>
  <body>

  </body>
</html>
```

Обратите внимание! Вложенные элементы набраны с отступами. Сделано это для лучшей читаемости кода. Браузер при интерпретации страницы игнорирует пробелы и переводы строки, поэтому при создании страницы в редакторе старайтесь набирать код так, чтобы он легче читался.

1.1.4. Чтобы приведённая выше заготовка превратилась в полноценную HTML-страницу, добавим содержимое в разделы заголовка и тела документа.

В секцию `<head>` добавьте заголовок страницы «Мой первый HTML-документ» с помощью тега `<title>` (этот заголовок будет отображаться в браузере в заголовке окна или вкладки).

Обратите внимание! Не путайте заголовок (название) страницы (задаваемый тегом `<title>`) и заголовочную часть документа (секция `<head>`)!

В секции `<body>` разместите текст «Hello World Wide Web!», оформив его в виде параграфа (для создания параграфа используется парный тег `<p>`).

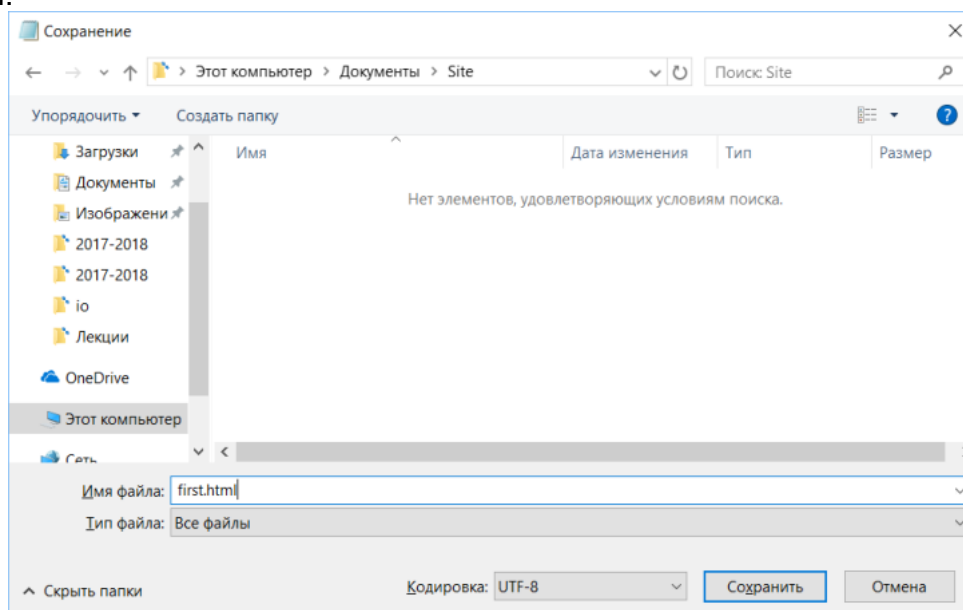
В итоге должен получиться файл следующего содержания:

```
<html>
  <head>
    <title>Мой первый HTML-документ</title>
  </head>
  <body>
    <p>Hello World Wide Web!</p>
  </body>
</html>
```

1.1.5. Чтобы полученный файл было удобно просматривать с помощью браузера нужно его сохранить как HTML-страницу.

Обратите внимание! При первом сохранении нового файла обязательно соблюдайте изложенный ниже порядок действий.

Выберите меню «Файл» → «Сохранить как...», чтобы вызвать диалоговое окно сохранения.



В этом окне переместитесь в каталог, который вы создали для хранения файлов вашего сайта (например, «Site»).

Из выпадающего списка «Кодировка» выберите «UTF-8».

Из выпадающего списка «Тип файла» выберите «Все файлы»

В окне «Имя файла» введите имя нашего файла вместе с расширением: *first.html*, и нажмите кнопку «Сохранить».

Обратите внимание! Все создаваемые в дальнейшем html-файлы также сохраняйте в том же самом каталоге (например, «Site»).

В результате в папке вашего сайта появится html-файл. Зайдите в эту папку с помощью Проводника или любого другого файлового менеджера (Блокнот при этом закрывать не нужно!) и откройте файл с помощью браузера (например, Mozilla Firefox, Google Chrome и др.). Проанализируйте полученный результат.

Обратите внимание! При дальнейшем редактировании файла, чтобы увидеть внесенные изменения, нужно сохранить (Ctrl+S) файл в редакторе (вызывать команду «Сохранить как...» больше не нужно) и обновить страницу (F5) в браузере.

1.2. Валидность web-страниц

Выше было упомянуто, что существуют разные версии языка HTML.

Браузеры, стремясь обеспечить пользователю комфортный просмотр веб-страниц, стараются отображать страницы, созданные с использованием любой версии языка. Более того, даже если вы допустите ошибку в коде, браузер не сообщит вам об этом, а будет пытаться «угадать», что вы имели в виду. Таким образом, ошибка может долго оставаться незамеченной и проявиться в неожиданный момент. При этом поведение разных браузеров может быть довольно непредсказуемым.

В том числе и поэтому так важно придерживаться стандартов при создании HTML-страниц и в web-дизайне это считается хорошим тоном. Чтобы проверить «правильность» (*валидность*) вашего кода, т.е. его соответствие стандартам, существует специальная процедура *валидации*.

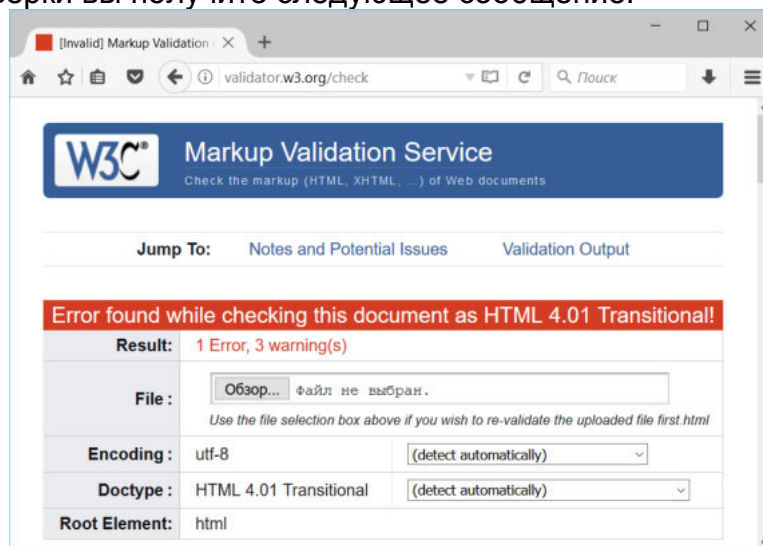
1.2.1. Проверьте, удовлетворяет ли созданная вами страничка действующим стандартам.

Откройте браузер и введите в адресную строку адрес <https://validator.w3.org>. Загрузится страница валидатора консорциума W3C.

На этой странице есть три опции: «Validate by URI» (проверить страницу по заданному адресу), «Validate by File Upload» (проверить файл) и «Validate by Direct Input» (проверить HTML-код, введенный непосредственно в поле ввода).


Вам нужно проверить созданный файл, поэтому щёлкните по вкладке «Validate by File Upload» и нажмите кнопку «Browse...» («Обзор...»), в появившемся диалоге выберите свой файл *first.html* и нажмите кнопку «Открыть». После закрытия диалога выбора файла на странице валидатора нажмите кнопку «Check» («Проверить»).

В результате проверки вы получите следующее сообщение:



Сообщение означает, что ваш файл валидацию не прошёл: была обнаружена одна ошибка (а также сделано 3 предупреждения).

Прокрутите страницу вниз и найдите описание ошибки:

 **Line 1, Column 1:** no document type declaration; implying "<!DOCTYPE HTML SYSTEM>"

Валидатор сообщает, что в файле отсутствует описание типа документа.

1.2.2. Исправьте эту ошибку.

Согласно современным стандартам, используемая версия языка должна быть указана в начале документа с помощью специального элемента <!DOCTYPE>, предназначенного для указания типа текущего документа — DTD (англ. document type definition).

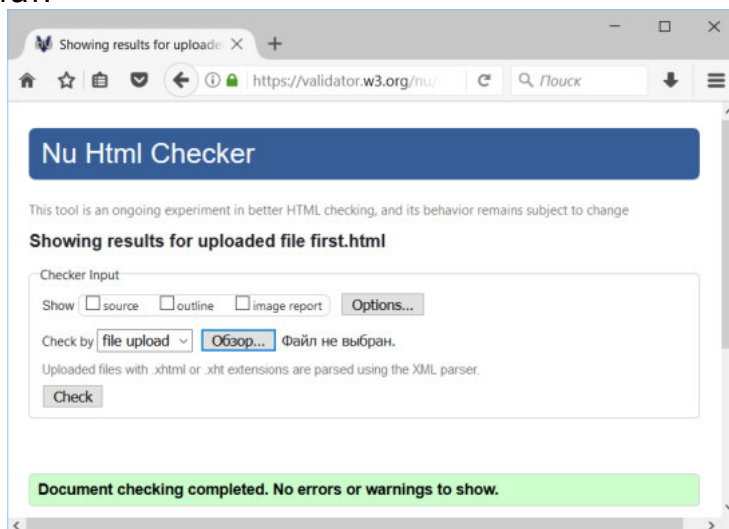
Мы будем использовать синтаксис HTML 5.1. Чтобы сообщить об этом браузеру, добавьте в начало вашего документа следующую строку:

```
<!DOCTYPE html>
```

Обратите внимание! В элементе <!DOCTYPE> мы указали просто html без номера версии. Такой подход используется только начиная с HTML 5. До этого <!DOCTYPE> выглядел гораздо сложнее. Самостоятельно найдите в Интернете и ознакомьтесь с типами <!DOCTYPE> для HTML 4.01.

Сохраните страницу и просмотрите результат в браузере. Визуально разницы пока не видно. Однако проверьте ваш исправленный файл на валидность ещё раз.

В браузере на странице валидатора вновь выберите файл first.html с помощью кнопки «Browse...» («Обзор...») и нажмите кнопку «Revalidate». Вы должны получить следующий результат:



Валидатор сообщает, что ваш файл успешно прошёл валидацию без ошибок и предупреждений («No errors or warnings to show») и удовлетворяет стандарту HTML 5.

1.3 Метатеги

В простейшем случае HTML тег состоит из имени, заключенного в угловые скобки (например,
). Однако многие теги могут (или даже должны) содержать один или несколько *атрибутов*, модифицирующих или определяющих его действие.

Атрибуты представляют собой пары «атрибут=значение», записываемые через пробел после имени тега, до закрывающей угловой скобки. Значения (за исключением числовых) принято записывать в двойных (прямых) кавычках «"..."».

Например, одиночный тег <meta>, размещаемый в секции <head> определяет так называемые «метатеги», которые используются для хранения информации, предназначенной для браузеров и поисковых систем (таких тегов в заголовке документа может быть несколько).

Использование метатегов считается хорошим тоном и может быть полезным. Добавьте их на вашу страницу.

1.3.1. Во-первых, нужно сообщить браузеру, какую кодировку вы используете при создании страницы (в нашем случае — это UTF-8). Для этого добавьте в секцию <head> следующий метатег:

```
<meta charset="utf-8">
```

Обратите внимание! Внимательно изучите этот пример и разберитесь, что такое «атрибут тега». Это очень важно для дальнейшей работы! В примере у тега <meta> есть атрибут charset (что говорит о том, что этот метатег предназначен для указания кодировки документа), которому установлено значение "utf-8" (собственно указание кодировки документа: utf-8).

1.3.2. Кроме того, метатеги часто используются, чтобы сообщить поисковым системам сведения об авторе и содержимом документа. Для этого используются метатеги с атрибутом name, принимающим, в частности, одно из следующих значений: author (имя автора документа), description (описание содержания документа), keywords (список ключевых слов). Атрибут content при этом задаёт соответствующее содержание.

Например, чтобы указать сведения об авторе страницы, добавьте следующий метатег (подставьте свои имя и фамилию):

```
<meta name="author" content="Фамилия Имя">
```

1.3.3. Самостоятельно добавьте метатеги с описанием страницы и списком ключевых слов.

Сохраните файл и просмотрите результат в браузере (визуальных изменений не произошло), а также проверьте страницу с помощью валидатора.

Обратите внимание! В дальнейшем не забывайте проверять валидность своих страниц после внесения изменений!

1.4. Инструментарий для создания веб-страниц

Как вы уже убедились, HTML-код можно писать и в Блокноте, однако это не очень удобно.

При веб-разработке обычно используют более мощные редакторы: с подсветкой синтаксиса, автодополнением кода и другими инструментами.

Одним из наиболее популярных таких редакторов является бесплатный редактор Notepad++.

1.4.1. Проверьте, установлен ли редактор Notepad++ на вашем рабочем компьютере.

1.4.2. Если редактор Notepad++ ещё не установлен, установите его, выполнив следующие действия.

Загрузите Notepad++ с сайта разработчика. Для этого зайдите на сайт <https://notepad-plus-plus.org> и нажмите кнопку «download» (загрузить).

На странице загрузки выберите ссылку «Notepad++ zip package» или «Notepad++ 7z package».

Создайте папку «Notepad++» и разархивируйте скачанный архив в неё (папку предпочтительно разместить в «Program Files» или «Моих документах»).

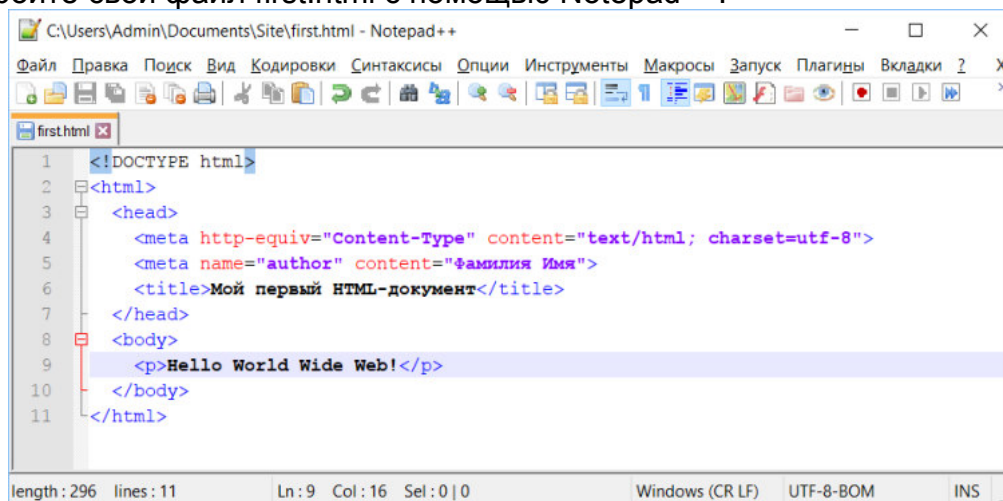
Чтобы редактором было удобнее пользоваться, вынесите ярлык для него на «Рабочий стол» (для этого найдите файл notepad++.exe, щёлкните по нему правой клавишей и выберите «Отправить» → «Рабочий стол (создать ярлык)»).

1.4.3. Запустите Notepad++ с помощью ярлыка и ознакомьтесь с интерфейсом программы.

1.4.4. Установите русский язык локализации для Notepad++ (если это ещё не сделано). Для этого выберите пункт меню «Settings» → «Preferences», перейдите на вкладку «General» и в выпадающем списке «Localization» выберите «Русский».

1.4.5. Отключите открытие в Notepad++ файлов из прошлой сессии (если вы работаете на компьютере, которым пользуется несколько человек, это поможет вам не перепутать ваши файлы): выберите пункт меню «Опции» → «Настройки» («Settings» → «Preferences» в английской версии), перейдите на вкладку «Резерв. Копирование» («Backup») и снимите «флажок» с опции «Запоминать текущую сессию для следующего запуска» («Remember current session for next launch»).

1.4.6. Откройте свой файл first.html с помощью Notepad++.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
5 <meta name="author" content="Фамилия Имя">
6 <title>Мой первый HTML-документ</title>
7 </head>
8 <body>
9 <p>Hello World Wide Web!</p>
10 </body>
11 </html>
```

Сохраните ваш файл под другим именем: выберите пункт меню «Файл» → «Сохранить как» и сохраните файл в ту же папку под именем template.html. Это будет ваш шаблон для создания последующих страниц (чтобы каждый раз не набирать исходную структуру документа).

1.4.7. Удалите из своего каталога файл first.html. Проверьте на валидность файл template.html (валидация должна проходить успешно).

В дальнейшем для удобства рекомендуется работать в Notepad++.

1.5. Комментарии

Самостоятельно найдите в справочнике как записываются комментарии в HTML-коде (текст, поясняющий какие-то участки исходного кода страницы и не отображаемый при просмотре документа) и добавьте в свой файл template.html комментарии, поясняющие назначение разделов документа и/или отдельных тегов (на ваш выбор).

Обратите внимание! В HTML (как и в языках программирования) разумное использование комментарием считается хорошим тоном. Используйте их при необходимости.

1.6. Язык текста

Для страницы (а также для её отдельных элементов) с помощью атрибута lang можно указать язык текста (код английского языка: en, для русского: ru).

Самостоятельно задайте для своего файла-шаблона русский язык (для этого нужно добавить атрибут lang с соответствующим значением тегу html).

2. Теги для разметки текста и создания гиперссылок

Рассмотрим основные теги, позволяющие размечать текст на веб-странице.

Прежде всего, создайте главную страницу вашего будущего сайта. Главную страницу принято называть index.html.

Скопируйте (средствами Проводника) файл template.html и сохраните копию этого файла в директории своего сайта под именем index.html. Откройте полученный файл index.html с помощью Notepad++ или любого другого редактора.

Скорректируйте заголовок страницы и содержание метатегов так, чтобы они соответствовали главной странице сайта.

2.1. Абзацы и заголовки

Вы уже знаете, что для разбивки текста на абзацы в HTML используется тег <p> (у вас один абзац с текстом уже есть на странице).

2.1.1. Добавьте ещё один абзац, содержащий некий текст о вашем сайте. Например: «Это мой первый Web-сайт!». Обратите внимание, что между абзацами браузер автоматически добавил интервал.

2.1.2. Разместите на главной странице сведения об авторе сайта.

Для этого добавьте в конец тела документа ещё один абзац, содержащий текст: «Сайт разработан: Фамилия Имя» (подставьте свои имя и фамилию).

2.1.3. Кроме разбиения на абзацы, в HTML-документе можно выделять озаглавленные разделы. Для этого служат теги заголовков, которые могут быть шести уровней: <h1>, <h2>, ... , <h6>. При этом <h1> — заголовок самого высокого уровня, а <h6> — самого низкого.

Добавьте на свою страницу заголовок верхнего уровня, содержащий приветствие: «Добро пожаловать!». Заголовок оформляется с помощью парного тега <h1> и должен располагаться в самом начале страницы (сразу после открывающего тега <body>). Сохраните файл и оцените результат, просмотрев страницу в браузере.

Обратите внимание! Не путайте заголовки разделов <h1>, <h2>, ... , <h6> с заголовочной частью документа <head> и заголовком страницы <title>!

2.2. Разрыв строки и горизонтальная линия

Допустим, вы хотите расположить информацию об авторе сайта в две строки: фамилию и имя разместить на следующей строке после текста «Сайт разработан». Как это сделать? Нажатие клавиши Enter в редакторе будет проигнорировано браузером при отображении страницы, а если использовать новый абзац <p>, то между строками будет слишком большой интервал.

2.2.1. Для перехода на новую строку без создания нового абзаца в HTML существует одиночный (не требующий закрывающего) тег
. Используйте этот тег, чтобы перенести свои имя и фамилию в подписи на новую строку.

2.2.2. Еще один способ отделить информацию на странице — использовать горизонтальные линии («линейки»). «Линейка» позволяет визуально отделить абзацы текста друг от друга. Задается она с помощью одиночного тега <hr>.

Добавьте на главную горизонтальную линию перед сведениями об авторе.

Обратите внимание! Горизонтальная линия <hr> должна располагаться между абзацами, а не внутри одного из них!

2.3. Специальные символы

Иногда требуется вставить на веб-страницу некоторые специальные символы, которых нет на клавиатуре (например, «©», «®», «≥» и др.), или которые нельзя напрямую использовать в HTML коде (например, знаки «<», «>»). Кстати, почему символы «<» и «>» нельзя использовать напрямую?

Для вставки таких символов в HTML можно использовать так называемые *мнемоники* (*символьные примитивы*) или воспользоваться Unicode-кодом этих символов.

2.3.1. Запись специальных символов начинается со знака «&», за которым следует мнемоника, или запись вида «#код». Завершается запись символа точкой с запятой. Например, при указании авторства сайта часто используется символ авторского права («копирайт») «©», мнемоникой для которого в HTML является «©», или, если воспользоваться кодом символа: «©». Знаку больше («>») соответствует мнемоника «>», знаку меньше («<») — «<», двойным кавычкам («"») — «"», а амперсанду («&») — «&».

2.3.2. Добавьте после своей фамилии и имени знак копирайта и год разработки.

2.4. Самостоятельное задние

2.4.1. Создайте страницу, содержащую более подробные сведения об авторе сайта. Для этого вновь скопируйте файл template.html и сохраните копию этого файла под именем about.html.

Откройте файл about.html с помощью редактора. Удалите текст из секции <body>. Скорректируйте заголовок страницы и содержание метатегов.

2.4.2. Добавьте на страницу about.html заголовок первого уровня «Сведения об авторе сайта» и (ниже) заголовок второго уровня «Обо мне», и напишите краткий текст, сообщающий сведения о вас: как вас зовут, сколько вам лет, где вы учитесь и т. п. (2-3 абзаца).

Проконтролируйте правильность отображения и валидность документа.

2.5. Преформатированный текст

Как было отмечено выше, браузер игнорирует пробелы (когда их более одного) и переводы строки, сделанные в редакторе. Однако иногда бывает нужно, чтобы текст отображался в точности так, как он был отформатирован при редактировании. Для указания браузеру, что такое форматирование следует учитывать, служит парный тег `<pre>`. Текст, заключённый в этот тег, будет набран моноширинным шрифтом с учётом всех пробелов, табуляций и переводов строки.

Обратите внимание! Использовать тег `<pre>` следует только при необходимости. Никогда не используйте его для выравнивания текста и т. п.

Частый случай применения этого тега — вывод на HTML-страницу кода (программного, HTML-кода и др.). Отметим, что для вывода кода также существует тег `<code>`, однако при его использовании пробелы и переводы строки браузером игнорируются.

2.5.1. Самостоятельно добавьте ниже раздела «Обо мне» второй заголовок второго уровня «Мои знания и умения». В этом разделе будут перечислено, чему вы научились при выполнении лабораторных работ по HTML.

После заголовка добавьте абзац с текстом «Вот что я знаю об HTML».

2.5.2. Добавьте первый подраздел в раздел «Мои знания и умения», называющийся «Структура HTML-страницы» (используйте заголовок третьего уровня).

С помощью тега `<pre>` добавьте в этот подраздел код, содержащий структуру простейшей HTML-страницы (код отформатируйте с использованием отступов).

Как добиться, чтобы теги отображались на странице, а не обрабатывались браузером?

2.6. Теги физической и логической разметки

В HTML существует возможность изменить внешний вид участков текста (разметить, форматировать текст), подобно тому, как это делается в текстовых редакторах. Теги, предназначенные для такой разметки, делятся на две группы.

Теги *физической разметки*, активно использовавшиеся в старых версиях HTML, прямо указывают браузеру, как должен выглядеть текст. Например, тег `` указывает, что текст должен быть выведен полужирным шрифтом, тег `<i>` — курсивным и т. д.

В связи с провозглашённым принципом отделения содержания HTML-документа от его представления, использование таких тегов не рекомендуется.

Вместо них предпочтительно пользоваться стилями (работа с которыми будет рассмотрена позже) или тегами *логической разметки*, которые лишь информируют браузер, что заключённый в них текст имеет специальное значение, контекст или употребление. При этом решение о том, как именно этот текст будет отображён, остаётся за браузером. Следует, однако, заметить, что на практике все наиболее распространённые браузеры отображают текст, форматированный с помощью логических тегов, примерно одинаково и достаточно предсказуемо.

Как теги физической разметки, так и теги логической разметки можно комбинировать с использованием вложенности.

2.6.1. Рассмотрим несколько тегов логической разметки:

Уже упоминавшийся выше парный тег `<code>` предназначен для отображения фрагментов исходного кода программ (как правило, для этого используется моноширинный шрифт).

Тег `` сообщает браузеру, что текст необходимо визуально выделить. На практике для этого обычно используется курсив.

Тег `` используется для акцентирования текста. Браузеры, как правило, отображают такой текст полужирным начертанием.

С остальными тегами логической разметки ознакомьтесь самостоятельно с помощью справочника.

2.6.2. Самостоятельно отформатируйте текст, содержащий информацию об авторе сайта на странице `about.html`, используя теги логической разметки (перечисленные выше или изученные вами самостоятельно).

Обратите внимание! Рассмотренными тегами разметки не следует злоупотреблять! Выделяйте только те слова, которые действительно этого требуют.

2.7. Гиперссылки

Итак, у вашего сайта есть две странички: index.html и about.html, но они между собой не связаны. А ведь наличие *гиперссылок* (возможности переходить от одного документа к другому посредством щелчка по специальным связующим элементам) — основное преимущество гипертекста.

Для создания гиперссылки в HTML используется парный тег <a>. Между открывающим и закрывающим тегом <a> располагается элемент, по нажатию на который произойдет переход по ссылке (как правило, текст или изображение). Адрес назначения ссылки задается атрибутом href. Например, чтобы поставить ссылку на главную страницу поисковой системы Google можно использовать код Поисковая система Google.

2.7.1. Добавьте в начало раздела «Мои знания и умения» после текста «Вот что я знаю об HTML» абзац, содержащий текст: «А если я чего-то не знаю — поможет Гугл», где слово «Гугл» должно быть активной ссылкой на поисковую систему Google. Проверьте работоспособность созданной ссылки.

2.7.2. Ссылка, которую вы только что создали, называется *абсолютной* (она содержит полный адрес целевого ресурса). Такие ссылки удобны для перехода с вашего сайта на другие. Для создания ссылок между страницами собственного сайта удобнее использовать ссылки *относительные* (путь указывается от расположения страницы, с которой идёт ссылка, или от корня сайта). В частности, если страницы находятся в одном каталоге (как у нас), достаточно указать только имя файла (разумеется, вместе с расширением).

Откройте в редакторе файл index.html и сделайте ваши фамилию и имя (в подписи внизу страницы) ссылкой на страницу about.html. Откройте файл в браузере и проконтролируйте работоспособность ссылки.

Обратите внимание! Гиперссылка не обязательно должна вести на html-страницу. Она может указывать и на другие типы документов (например, простой текст, PDF, архив и др.), которые можно просмотреть или загрузить с помощью браузера.

2.7.3. Бывают ситуации, когда нужно сослаться не просто на страничку, а на какое-то определенное место на этой странице — при переходе по такой ссылке страница будет открыта не на начале, а на нужном месте, если это возможно (типичный пример — оглавление). Такую возможность HTML тоже предоставляет.

Чтобы сослаться на конкретное место документа, нужно это место пометить. Для этого можно воспользоваться универсальным атрибутом id. Этот атрибут задаёт так называемый *идентификатор* элемента и может присутствовать практически у всех HTML-тегов. При этом идентификатор элемента должен быть уникален в пределах страницы.

2.7.4. Добавьте на странице about.html оглавление.

Для этого тегу, задающему заголовок раздела «Обо мне», добавьте атрибут id со значением about_me (здесь используется знак подчёркивания, пробелы в именах идентификаторов использовать запрещено).

После этого выше, сразу после заголовка верхнего уровня «Сведения об авторе сайта» добавьте ссылку вида: Обо мне.

В качестве адреса ссылки здесь указано значение идентификатора id, которому предшествует символ «#».

Обратите внимание! В примере ссылки адрес страницы не указан, т. к. ссылка ведёт на ту же страницу, на которой находится. Если бы ссылка была расположена на другой странице (например, index.html), то она должна была бы содержать адрес целевой страницы перед значением id, и выглядела бы следующим образом: Обо мне.

2.7.5. Самостоятельно добавьте в оглавление остальные разделы вашей страницы about.html. Протестируйте работу всех ссылок.

2.8. Самостоятельное задание

2.8.1. В разделе 1.2.2 вы узнали как задаётся описание типа документа (DTD) в HTML 5: `<!DOCTYPE html>`. Очень просто и лаконично. Однако, как уже было отмечено в том же разделе, так было не всегда: в HTML 4.01 было определено несколько допустимых DTD, каждый из которых выглядел значительно сложнее.

Добавьте в раздел «Мои знания и умения» подраздел «Типы документа в HTML 4.01». В этот раздел поместите (предварительно найдя эту информацию в Интернете или в справочнике) возможные виды DTD (типа документа) для HTML версии 4.01, снабжённые кратким описанием. Для оформления HTML кода используйте тег `<pre>`.

Добавьте ссылку на новый подраздел в оглавление.

2.8.2. Добавьте ссылку на ваш адрес электронной почты в раздел «Обо мне» (это может быть ссылка, привязанная в вашем фамилии и имени, или отдельная ссылка, например с текстом «Пишите мне»).

Чтобы поставить ссылку на адрес электронной почты нужно указать для тега `<a>` значение атрибута `href` в следующем формате: `"mailto:name@domain.com"`.

2.8.3. Дополните раздел «Сведения об авторе сайта» подразделом (заголовок третьего уровня) «Мои любимые сайты», в котором разместите не менее трёх ссылок на разные сайты, снабдив их кратким описанием. Не забудьте добавить ссылку на этот подраздел в оглавление.

2.8.4. Проверьте работоспособность всех ссылок и валидность ваших веб-страниц.