

УДК 004.43+004.738.5

ББК 32.973.26–018.1

Д75

Дронов В. А.

Д75 HTML и CSS: 25 уроков для начинающих. — СПб.: БХВ-Петербург, 2020. — 400 с.: ил.

ISBN 978-5-9775-4070-4

В книге 25 иллюстрированных уроков, 100 практических упражнений на тему разработки простых веб-сайтов и более 50 заданий для самостоятельной работы. Дано введение в веб-разработку, раскрыты основы HTML, CSS и работы в WWW. Объяснено, как с помощью HTML структурировать и форматировать текст, работать с графикой и мультимедиа, таблицами, гиперссылками, веб-формами, фреймами и метаданными. Рассказано, как средствами CSS оформлять веб-страницы, использовать блоки, списки и таблицы, изображения и гиперссылки, элементы управления, фон и колонки, располагать, выводить и скрывать элементы, делать макеты веб-страниц, выполнять преобразования, анимацию, медиазапросы и др.

Электронное приложение-архив на сайте издательства содержит коды всех примеров и пяти учебных веб-сайтов.

Для начинающих веб-разработчиков

УДК 004.43+004.738.5

ББК 32.973.26–018.1

Группа подготовки издания:

Руководитель проекта	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Сависте</i>
Компьютерная верстка	<i>Людмилы Гауль</i>
Дизайн обложки	<i>Карины Соловьевой</i>

Подписано в печать 30.04.20.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 32,25.

Тираж 1000 экз. Заказ №

«БХВ-Петербург», 191036, Санкт-Петербург, Гончарная ул., 20.

Отпечатано с готового оригинал-макета

ООО «Принт-М», 142300, М. О., г. Чехов, ул. Полиграфистов, д. 1

ISBN 978-5-9775-4070-4

© ООО «БХВ», 2020

© Оформление. ООО «БХВ-Петербург», 2020

Оглавление

ПРЕДИСЛОВИЕ.....	11
Типографские соглашения.....	13
ЧАСТЬ I. ВВЕДЕНИЕ В ВЕБ-РАЗРАБОТКУ	15
УРОК 1. HTML.....	17
1.1. Упражнение. Создаем нашу первую веб-страницу	17
1.2. Теги. Язык HTML.....	20
Как набирать HTML-код? — часть 1	23
1.3. Упражнение. Продолжаем изучать теги	24
Как набирать HTML-код? — часть 2.....	25
1.4. Упражнение. Помещаем на веб-страницу изображения, знакомимся с одинарными тегами и атрибутами тегов.....	25
Не забывайте ставить закрывающие теги!	29
1.5. Структурирование HTML-кода и метаданные.....	29
1.6. Комментарии HTML.....	31
Как еще можно использовать комментарии?	31
1.7. Самостоятельные упражнения.....	32
УРОК 2. CSS.....	33
2.1. Оформление по умолчанию.....	33
2.2. Упражнение. Задаем оформление для веб-страницы	34
2.3. Таблицы стилей, стили, атрибуты стилей и наследование. Язык CSS	36
2.4. Внешние таблицы стилей и их привязка к веб-страницам	39
2.5. Упражнение. Используем стилевые классы и знакомимся с правилом каскадности.....	40
2.6. Упражнение. Используем важные атрибуты стилей.....	44
2.7. Упражнение. Применяем составные селекторы	45
2.8. Внутренние таблицы стилей	46
2.9. Встроенные стили.....	47

2.10. Комментарии	48
2.11. Самостоятельные упражнения	48
УРОК 3. КАК РАБОТАЕТ WWW?	51
3.1. Клиенты и серверы.....	51
3.2. Упражнение. Публикуем наш веб-сайт на веб-сервере	52
3.3. Интернет-адрес и его составные части	56
Протокол	56
Адрес хоста	57
Протокол IP, IP-адреса и доменные имена	58
Путь к файлу.....	58
Примеры интернет-адресов и их обработка веб-обозревателем.....	58
Пути к файлам в HTML-тегах	59
3.4. Упражнение. Используем веб-страницу по умолчанию	59
3.5. Ошибка 404.....	59
Notepad++	60
ЧАСТЬ II. HTML	61
УРОК 4. СТРУКТУРА	63
4.1. Абзацы и заголовки	63
Разделитель	64
4.2. Упражнение. Создаем списки	65
4.3. Список описаний.....	68
4.4. Упражнение. Используем семантическую разметку.....	68
Тег <section>	70
4.5. Адрес и блочная цитата	70
4.6. Блочные элементы.....	72
4.7. Правила вывода текста в блочных элементах	73
4.8. Текст фиксированного формата	74
4.9. Упражнение. Применяем блочный контейнер	74
4.10. Самостоятельные упражнения	76
УРОК 5. ТЕКСТ	77
5.1. Теги выделения.....	77
5.2. Встроенные элементы	78
5.3. Упражнение. Применяем встроенный контейнер.....	79

5.4. Переносы строк.....	79
5.5. Упражнение. Вставляем специальные символы посредством литералов.....	80
5.6. Самостоятельные упражнения.....	82
УРОК 6. ГРАФИКА И МУЛЬТИМЕДИА.....	83
6.1. Вставка графических изображений.....	83
6.2. Форматы графики, применяемые в WWW.....	84
Растровые форматы.....	84
Векторный формат.....	86
6.3. Ссылки на файлы и их указание.....	86
6.4. Встроенно-блочные элементы.....	88
Абсолютные пути к файлам и веб-сервер.....	88
6.5. Упражнение. Помещаем на веб-страницу видеоролик.....	89
Формат HTML 5.....	92
6.6. Вставка аудиоролика.....	92
Содержимое тегов <video> и <audio>.....	93
6.7. Семантическая иллюстрация.....	93
6.8. Упражнение. Помещаем на веб-страницу видео YouTube.....	94
Лицензия Creative Commons.....	96
6.9. Самостоятельные упражнения.....	96
УРОК 7. ТАБЛИЦЫ.....	99
7.1. Упражнение. Создаем простую таблицу.....	99
7.2. Упражнение. Выполняем слияние ячеек.....	102
7.3. Упражнение. Делаем таблицу со сложным содержимым.....	104
7.4. Упражнение. Используем секции таблицы.....	107
Заголовок таблицы.....	108
7.5. Самостоятельные упражнения.....	109
УРОК 8. ГИПЕРССЫЛКИ.....	111
8.1. Упражнение. Создаем гиперссылки.....	111
Графическая гиперссылка.....	114
Почтовая гиперссылка.....	115
8.2. Загрузочные гиперссылки.....	115
Форматы файлов, поддерживаемые веб-обозревателями.....	116
Пустая гиперссылка.....	116
8.3. Гиперссылки, ссылающиеся на фрагменты веб-страниц. Якоря..	117

8.4. Упражнение. Создаем панель навигации	118
Полоса навигации	121
Карта-изображение.....	122
8.5. Самостоятельные упражнения	123
УРОК 9. ВЕБ-ФОРМЫ И ЭЛЕМЕНТЫ УПРАВЛЕНИЯ	125
9.1. Обработка пользовательских данных. Веб-формы и веб-приложения	125
9.2. Упражнение. Создаем веб-форму для заказа столика.....	127
9.3. Как работает веб-форма? Параметры веб-формы	132
9.4. Параметры, поддерживаемые всеми элементами управления....	135
9.5. Поля ввода/выбора.....	135
9.6. Упражнение. Используем флажки и переключатели.....	138
9.7. Упражнение. Создаем список.....	139
9.8. Область редактирования	143
9.9. Кнопки.....	144
9.10. Поле выбора файла	145
Регулятор	146
9.11. Самостоятельные упражнения	147
Группа элементов управления	147
УРОК 10. ФРЕЙМЫ	149
10.1. Упражнение. Создаем фотогалерею на основе фрейма.....	149
10.2. Самостоятельные упражнения	153
УРОК 11. МЕТАДААННЫЕ.....	155
11.1. Упражнение. Задаем метаданные для веб-страницы	155
Популярные в России поисковые службы.....	156
11.2. Самостоятельные упражнения.....	157
ЧАСТЬ III. CSS.....	159
УРОК 12. СЕЛЕКТОРЫ	161
12.1. Основные селекторы и комбинированный селектор	161
Привязка нескольких стилевых классов.....	162
12.2. Упражнение. Используем селекторы атрибутов	163
12.3. Упражнение. Применяем псевдоклассы	166
Стили с несколькими селекторами	168

12.4. Псевдоклассы, поддерживаемые CSS	168
12.5. Упражнение. Используем псевдоэлементы.....	173
12.6. Разделители	174
12.7. Приоритет селекторов и его вычисление	175
12.8. Самостоятельные упражнения	176
УРОК 13. ШРИФТ И ТЕКСТ.....	179
13.1. Гарнитура, кегль, насыщенность и начертание шрифта.....	179
13.2. Упражнение. Используем загружаемые шрифты.....	181
Бесплатные и платные шрифты	183
Форматы загружаемых шрифтов.....	185
13.3. Оформление, преобразование текста и форма строчных букв....	186
Значения initial и inherit	187
13.4. Цвет и уровень прозрачности	188
13.5. Отступы и интервалы	189
Единицы измерения CSS.....	191
13.6. Упражнение. Используем генерируемое содержание.....	191
13.7. Упражнение. Применяем счетчики.....	193
13.8. Выравнивание по горизонтали и вертикали	195
13.9. Упражнение. Создаем тень у текста.....	197
Указание сразу всех параметров шрифта	198
13.10. Самостоятельные упражнения	198
УРОК 14. БЛОКИ.....	201
14.1. Упражнение. Используем внутренние и внешние просветы для создания стильного заголовка.....	201
14.2. Упражнение. Выводим рамки у элементов	205
14.3. Рамки со скругленными углами	210
14.4. Упражнение. Задаем размеры элементов	212
14.5. Размеры элементов, внутренние просветы и рамки. Режим установки размеров	215
14.6. Упражнение. Создаем тень у блочного элемента.....	216
14.7. Самостоятельные упражнения	218
УРОК 15. СПИСКИ И ТАБЛИЦЫ	219
15.1. Упражнение. Задаем параметры списков.....	219
15.2. Параметры таблиц.....	221

15.3. Параметры ячеек таблиц.....	224
15.4. Самостоятельные упражнения	224
УРОК 16. ИЗОБРАЖЕНИЯ И ГИПЕРССЫЛКИ.....	227
16.1. Параметры графических изображений	227
16.2. Упражнение. Оформляем гиперссылки, применяя псевдоклассы гиперссылок.....	228
УРОК 17. ЭЛЕМЕНТЫ УПРАВЛЕНИЯ	231
17.1. Упражнение. Оформляем веб-формы и элементы управления ..	231
17.2. Псевдоклассы и псевдоэлементы элементов управления	233
17.3. Самостоятельные упражнения	234
УРОК 18. ФОН.....	237
18.1. Сплошной фон	237
18.2. Упражнение. Создаем линейный градиент	239
18.3. Упражнение. Создаем радиальный градиент	244
18.4. Повторяющиеся градиенты	249
18.5. Упражнение. Указываем графический фон.....	250
18.6. Упражнение. Накладываем несколько графических фонов.....	257
18.7. Самостоятельные упражнения	259
УРОК 19. КОЛОНКИ	261
19.1. Многоколоночная верстка текста.....	261
19.2. Дополнительные параметры колонок	262
УРОК 20. ПРЕДСТАВЛЕНИЕ ЭЛЕМЕНТОВ	265
20.1. Упражнение. Делаем «легкую» панель навигации	265
20.2. Упражнение. Используем таблицы без таблиц	267
Управление видимостью элементов.....	271
Изменение курсора мыши	272
20.3. Самостоятельные упражнения	273
УРОК 21. МЕСТОПОЛОЖЕНИЕ ЭЛЕМЕНТОВ.....	275
21.1. Упражнение. Используем плавающие элементы и семантические врезки	275
Центрирование элемента страницы по горизонтали	279
21.2. Упражнение. Создаем адаптивную полосу навигации с помощью упорядочивающего элемента	280

21.3. Упражнение. Используем элементы-перечни	287
21.4. Параметры потомков упорядочивающего элемента.....	293
Полное центрирование элемента страницы	293
21.5. Упражнение. Используем свободно позиционируемые элементы.....	294
21.6. Упражнение. Применение относительно позиционируемых элементов.....	302
21.7. Самостоятельные упражнения.....	305
УРОК 22. МАКЕТЫ ВЕБ-СТРАНИЦ.....	307
22.1. Классические макеты.....	307
22.2. Упражнение. Создаем двухколоночный макет на основе плавающего элемента.....	308
22.3. Упражнение. Делаем табличный макет	310
22.4. Упражнение. Делаем рамочный макет на фиксированных элементах.....	314
22.5. Упражнение. Создаем прокручивающийся элемент страницы ..	317
Формулы.....	320
Переполнение и плавающие элементы.....	320
Макет с фиксированной шириной.....	321
22.6. Самостоятельные упражнения	321
УРОК 23. ПРЕОБРАЗОВАНИЯ	323
23.1. Упражнение. Применяем преобразования к элементам	323
23.2. Упражнение. Применяем сложные преобразования.....	329
23.3. Самостоятельные упражнения	329
УРОК 24. АНИМАЦИЯ	331
24.1. Упражнение. Применяем анимацию с двумя состояниями	331
24.2. Сложная анимация с двумя состояниями.....	337
24.3. Упражнение. Создаем анимацию с несколькими состояниями ..	337
24.4. Самостоятельные упражнения.....	344
УРОК 25. МЕДИАЗАПРОСЫ.....	347
25.1. Упражнение. Адаптируем веб-страницы под мобильные устройства.....	347
25.2. Упражнение. Создаем печатную редакцию веб-сайта	352
25.3. Самостоятельные упражнения	356

ЗАКЛЮЧЕНИЕ	359
ПРИЛОЖЕНИЕ 1. В КОПИЛКУ ВЕБ-ВЕРСТАЛЬЩИКА.....	361
П1.1. Фотография в стиле Polaroid.....	361
Внешний вид (приведены три элемента)	361
HTML: приведен код, создающий один такой элемент	361
CSS.....	362
П1.2. Фотогалерея в стиле Polaroid	362
Внешний вид	362
HTML	363
CSS.....	363
П1.3. Круглая виньетка.....	364
Внешний вид	364
HTML	364
CSS.....	364
П1.4. Спойлер	364
Внешний вид	365
HTML	365
CSS.....	366
П1.5. Меню-гамбургер.....	368
Внешний вид	368
HTML	368
CSS.....	369
П1.6. Аккордеон.....	370
Внешний вид	370
HTML	370
CSS.....	372
ПРИЛОЖЕНИЕ 2. РАЗРАБОТКА ОДНОСТРАНИЧНЫХ ВЕБ-САЙТОВ	373
П2.1. Упражнение. Создаем одностраничный веб-сайт	373
П2.2. Упражнение. Создаем панель навигации для одностраничного веб-сайта.....	379
П2.3. Самостоятельные упражнения	382
ПРИЛОЖЕНИЕ 3. HTML-ТЕГИ.....	383
ПРИЛОЖЕНИЕ 4. ОПИСАНИЕ ЭЛЕКТРОННОГО АРХИВА.....	387
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	389

Предисловие

Для кого эта книга?
О чем в ней рассказывается?
Современные веб-технологии
Уроки, упражнения, наглядность и практика
Пять учебных веб-сайтов
Электронное приложение
Типографские соглашения

Эта книга — для начинающих разработчиков, пока еще «чайников» в веб-верстке, но желающих узнать на практике, как делаются веб-страницы и веб-сайты. В ней рассмотрены самые простые и эффективные средства, которые необходимо знать каждому и которые используются в большинстве случаев даже самыми «продвинутыми» веб-верстальщиками.

Объем книги небольшой, материал можно освоить за месяц. В итоге читатель научится верстать простые, но надежно работающие веб-сайты с дизайном, который может удовлетворить потенциального заказчика. Читатель получит практический опыт веб-верстки, узнает, как для получения нужного результата использовать средства, предлагаемые современными веб-технологиями, и — кто знает — возможно, начнет карьеру профессионального веб-верстальщика. Более того, в процессе изучения книги читатель самостоятельно создаст пять сайтов, вполне пригодных для публикации в Сети и включения в портфолио.

Практика — вот девиз, которому следовал автор. Практика — и необходимая теория, даваемая по ходу дела.

Вот принципы, на основе которых построена книга:

- ◆ материал разбит на 25 коротких, наглядных, иллюстрированных уроков, в которых практические упражнения совмещаются с теоретическими отступлениями;
- ◆ побольше иллюстраций — «лучше один раз показать, чем сто раз рассказать»;
- ◆ рассказывая о каком-либо инструменте, автор старается показать (на конкретном примере), чем он может быть полезен;

- ◆ описываются только наиболее актуальные инструменты — те, которые могут пригодиться в большинстве случаев. Редко используемые, чересчур сложные в применении и поддерживаемые не всеми веб-обозревателями средства оставлены «за кадром»;
- ◆ рассказ ведется простым человеческим языком, с минимумом специальной терминологии;
- ◆ и никакой «воды» — только по делу!

В книге мы изучим все необходимые для разработки веб-страниц и веб-сайтов технологии:

- ◆ HTML — язык, на котором пишутся сами веб-страницы. Именно с его помощью текст разбивается на абзацы и заголовки, создаются списки и таблицы, на страницы помещаются графические изображения, аудио- и видеоролики;
- ◆ CSS — язык, на котором описывается оформление веб-страниц: каким шрифтом нужно вывести текст абзацев, а каким — заголовков, каким фоном следует залить панель навигации, какой рамкой нужно ее окружить и как сделать так, чтобы при наведении курсора мыши на гиперссылку она плавно меняла цвет (то есть создать анимацию);
- ◆ принципы функционирования Интернета: что такое клиент, сервер, протокол HTTP и страница по умолчанию.

Каждый урок содержит:

- ◆ упражнения — выполняемые под руководством автора. Они представляют собой набор пронумерованных действий, которые необходимо выполнить для достижения поставленного результата, а в конце — теоретические знания в виде краткой сводки;
- ◆ параграфы с одной лишь теорией. Как правило, описываемые в них сведения потребуются при выполнении самостоятельных упражнений;
- ◆ самостоятельные упражнения — их следует выполнить вам, уважаемые читатели.

В процессе учебного курса мы создадим целых пять веб-сайтов:

- ◆ четыре — посвященные придуманному автором суши-бару «Йокогама». Они различаются лишь макетами, на основе которых созданы страницы;
- ◆ пятый — посвящен великой рок-группе «The Beatles».

Книгу сопровождает электронный архив, содержащий необходимые для работы файлы, равно как и результаты выполнения всех упражнений курса (см. приложение 4). Этот архив можно загрузить с адреса <ftp://ftp.bhv.ru/9785977540704.zip>.

Типографские соглашения

В книге будут часто приводиться различные языковые конструкции, применяемые в HTML и CSS. Для наглядности при их написании использованы следующие типографские соглашения (в реальном коде они недействительны):

- ◆ HTML- и CSS-код набран моноширинным шрифтом:

```
<h1>Суши-бар <em>Йокогама</em></h1>
<p></p>
```

```
p {
    font-size: 14pt;
    text-align: justify;
}
```

- ◆ В угловые скобки (<>) заключаются наименования различных значений, которые дополнительно выделяются курсивом. В реальный код, разумеется, должны быть подставлены реальные значения. Например:

```
url(<ссылка на графический файл>)
```

Здесь вместо подстроки *ссылка на графический файл* должна быть подставлена реальная ссылка на графический файл, который будет загружен и выведен на страницу.

- ◆ В квадратные скобки ([]) заключаются необязательные фрагменты кода. Например:

```
<цвет> [<заполняемая область>]
```

Здесь *заполняемая область* может указываться, а может и не указываться.

- ◆ Вертикальной чертой (|) разделяются доступные для выбора значения, из которых в код можно подставить лишь одно. Например:

```
step(<количество скачков>[, start|end])
```

Здесь после запятой можно подставить либо *start*, либо *end*, но не то и другое одновременно и не какое-либо иное значение.

- ◆ Слишком длинные фрагменты, не помещающиеся на одной строке, автор разрывал на несколько строк и в местах разрывов ставил знаки ↵. Например:

```
background-image: linear-gradient(to right top, ↵
salmon 0%, white 67%, salmon 100%);
```

Приведенный код здесь разбит на две строки, но должен быть набран в одну. Символ ↵ при этом нужно удалить.

- ◆ Троекочием (. . .) помечены фрагменты кода, пропущенные ради сокращения объема текста:

```
<h1>Суши-бар <em>Йокогама</em></h1>
```

```
. . .
```


```
<p>Ждем вас!</p>
```

Здесь весь код между двумя приведенными строками пропущен.

Обычно такое можно встретить в исправленных впоследствии фрагментах кода — приведены лишь собственно исправленные строки, а оставшиеся неизменными пропущены. Также троекочие используется, чтобы показать, в какое место должен быть вставлен вновь написанный код: в начало исходного фрагмента, в его конец или в середину, между уже присутствующими в нем строками.



ВНИМАНИЕ!

Все приведенные здесь типографские соглашения имеют смысл только в примерах написания языковых конструкций HTML и CSS. В реальном коде они не используются, за исключением знака  и троекочия.

ЧАСТЬ I

ВВЕДЕНИЕ

В ВЕБ-РАЗРАБОТКУ

- ⇒ Наша первая веб-страница. Язык HTML.
- ⇒ Оформляем веб-страницы посредством стилей. Язык CSS.
- ⇒ Как работает Всемирная паутина? Клиенты и серверы. Интернет-адреса.

Урок 1. HTML

Создание веб-страниц

Теги

Атрибуты тегов

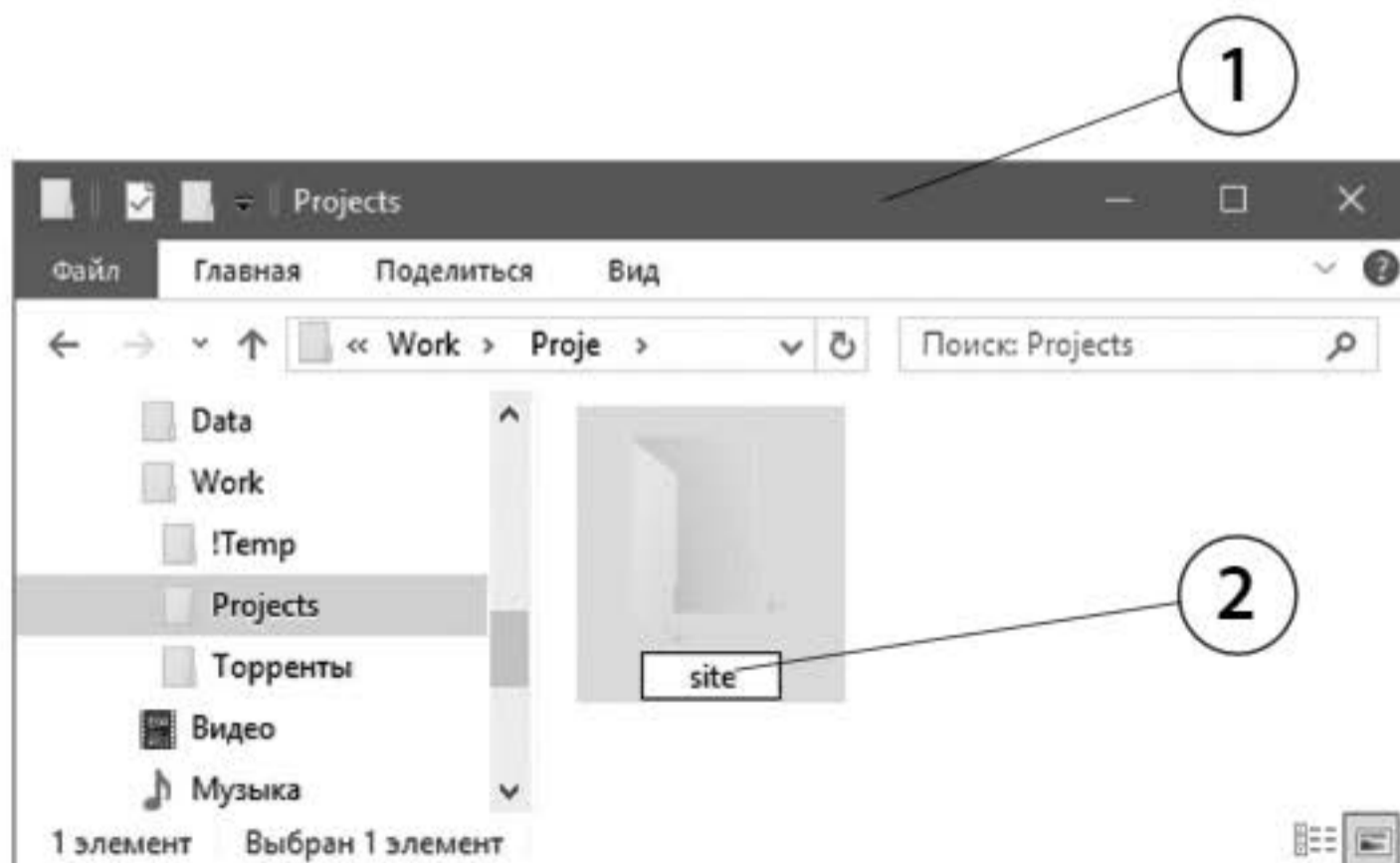
Структурирование кода

Метаданные

1.1. Упражнение. Создаем нашу первую веб-страницу

Сейчас мы создадим нашу первую, совсем простенькую веб-страницу, включающую заголовок и несколько абзацев. На ее примере мы далее будем изучать принципы написания веб-страниц.

1. Откроем окно Проводника.
2. Все файлы: веб-страницы, графические изображения, аудио- и видеоролики, входящие в состав веб-сайта, — должны храниться в особой папке или вложенных в нее подпапках. Создадим папку, в которой будут находиться все файлы нашего будущего веб-сайта. Назовем ее site.



3. Запустим Блокнот — простейший текстовый редактор, поставляемый в составе Windows. Найти его можно в группе Стандартные меню Пуск.



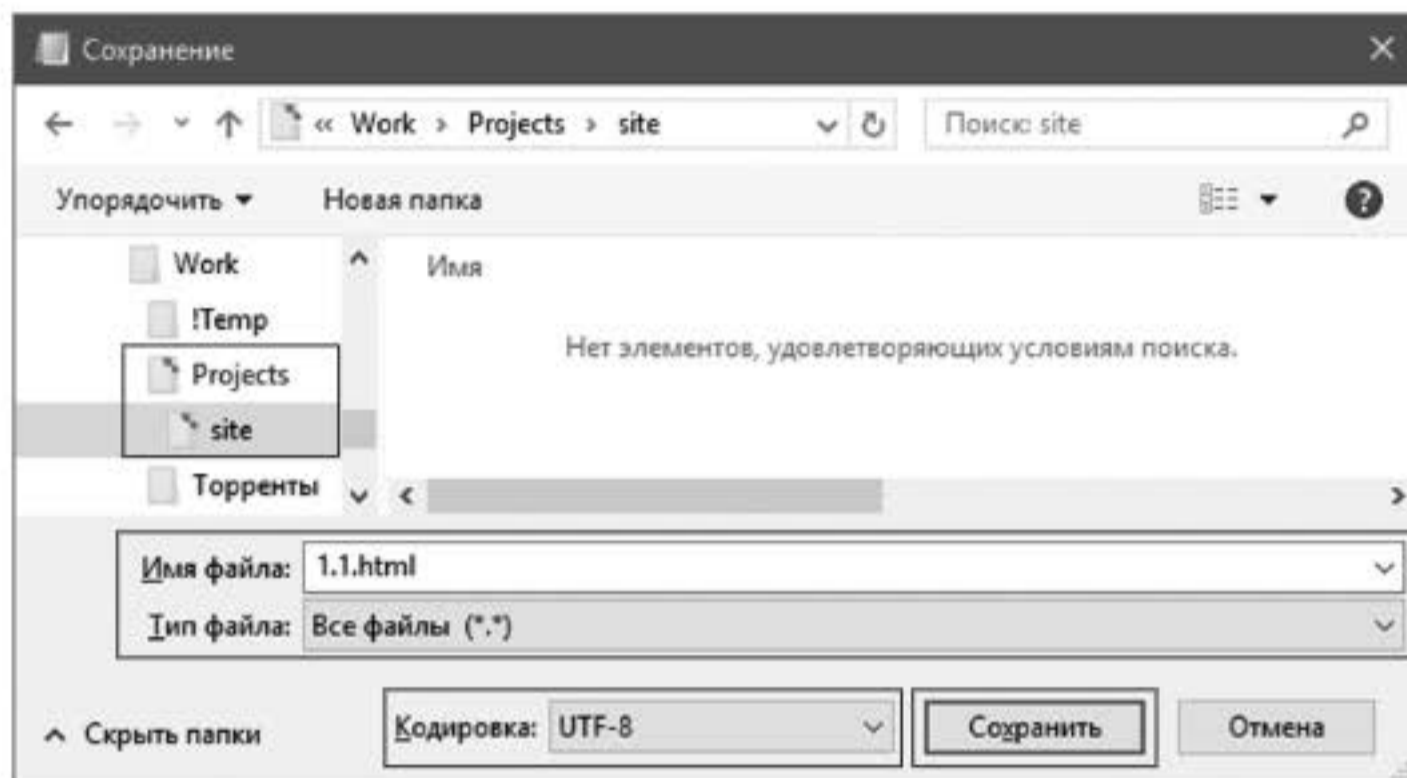
4. В Блокноте наберем следующий текст, стараясь не допускать ошибок:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Йокогама: суши-бар</title>
  </head>
  <body>
    <h1>Суши-бар Йокогама</h1>
    <p>У нас вы найдете: сашими, суши, роллы,
соусы, горячие блюда.</p>
    <p>Также - безалкогольные напитки.</p>
    <p>Ждем вас!</p>
  </body>
</html>
```

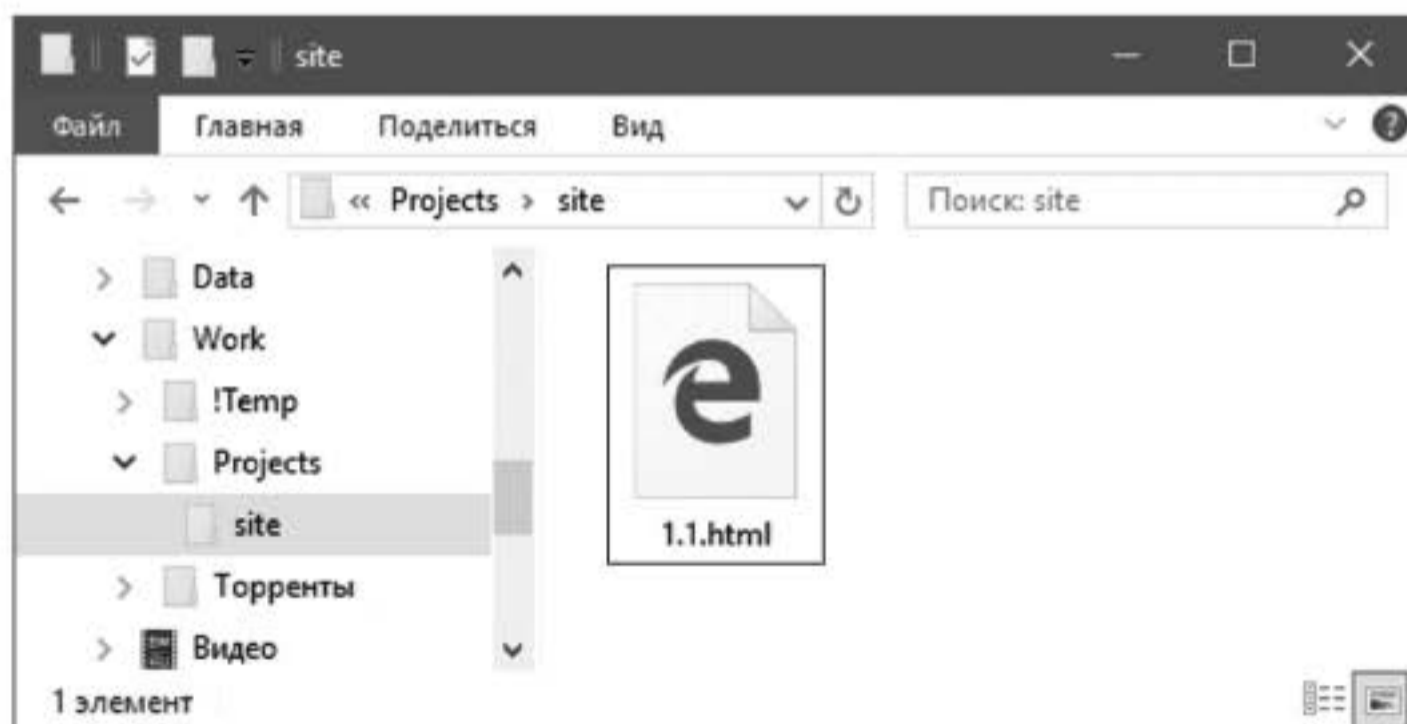
5. Сохраним набранный текст в файле.



6. Выберем для этого файла созданную ранее папку `site`. Дадим файлу имя `1.1.html`. Укажем кодировку: **UTF-8**. В раскрывающемся списке **Тип файла** выберем **Все файлы (*.*)**, иначе Блокнот при сохранении добавит к имени файла расширение `txt`.

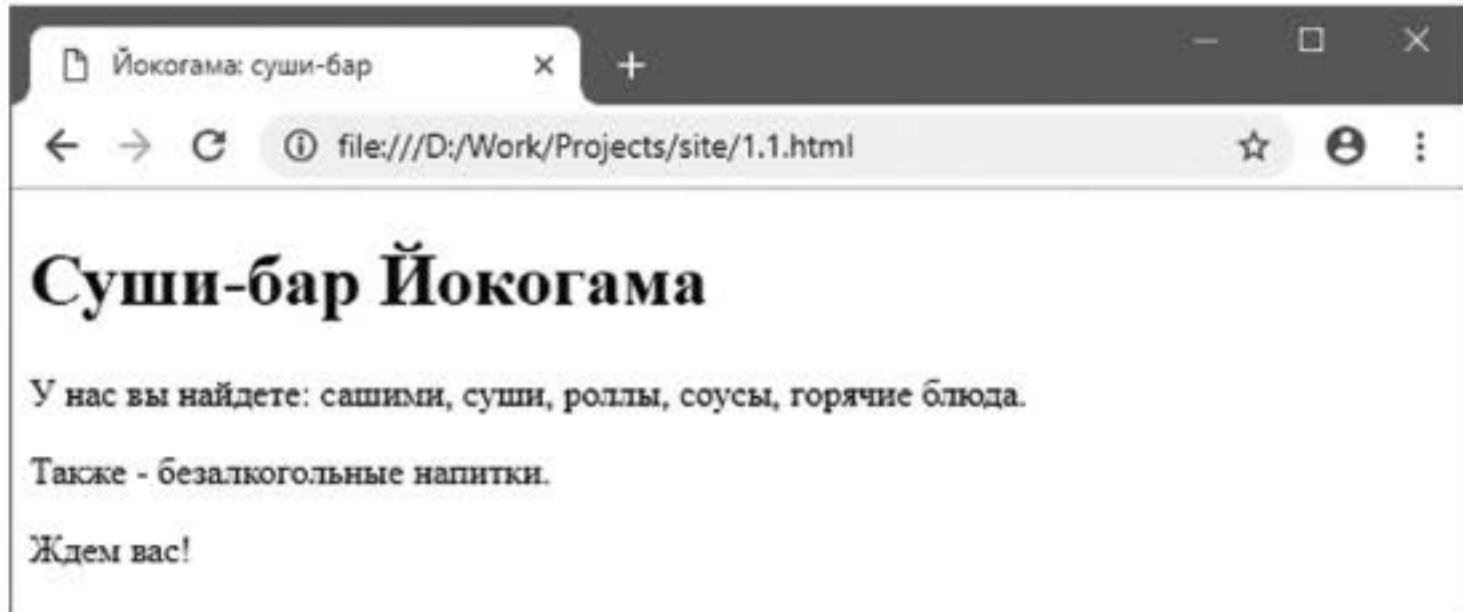


7. Откроем только что созданный файл `1.1.html`, щелкнув на нем двойным щелчком.



Результат

Веб-страница будет открыта в веб-обозревателе, зарегистрированном в системе как программа для открытия HTML-файлов по умолчанию. У автора книги это Google Chrome.

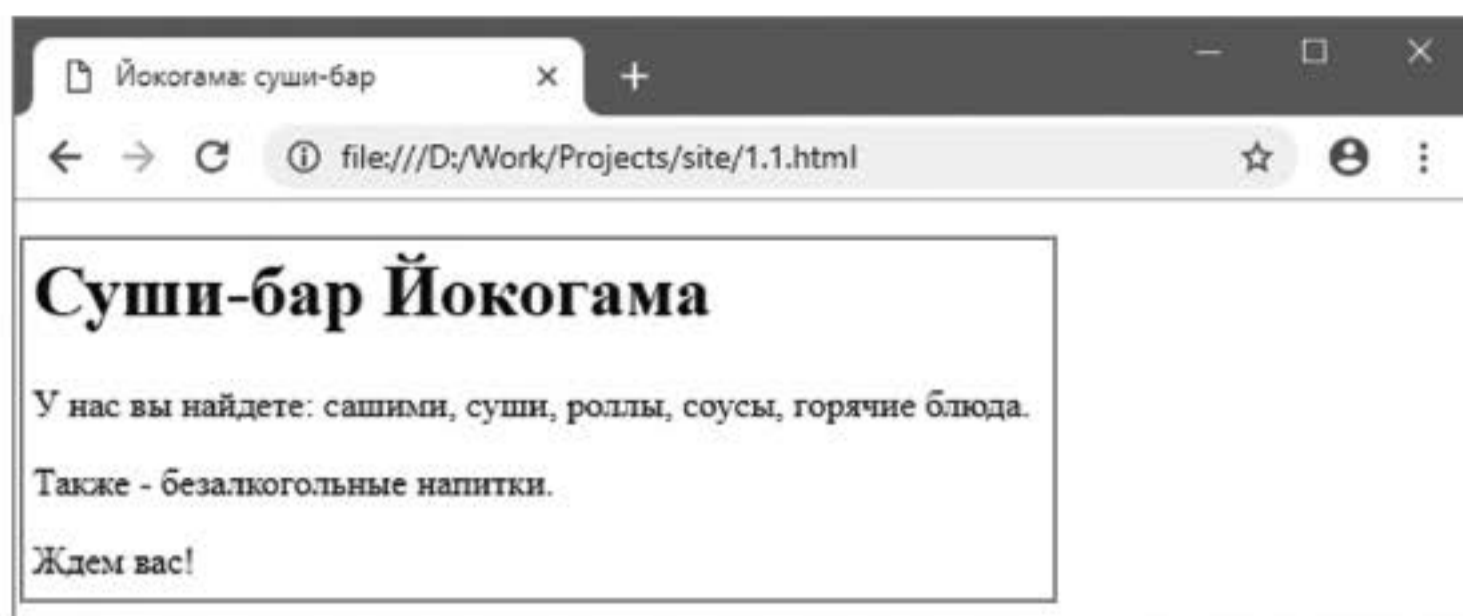


1.2. Теги. Язык HTML

Если рассмотреть набранный нами в Блокноте текст...

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Йокогама: суши-бар</title>
  </head>
  <body>
    <h1>Суши-бар Йокогама</h1>
    <p>У нас вы найдете: сашими, суши, роллы,
соусы, горячие блюда.</p>
    <p>Также - безалкогольные напитки.</p>
    <p>Ждем вас!</p>
  </body>
</html>
```

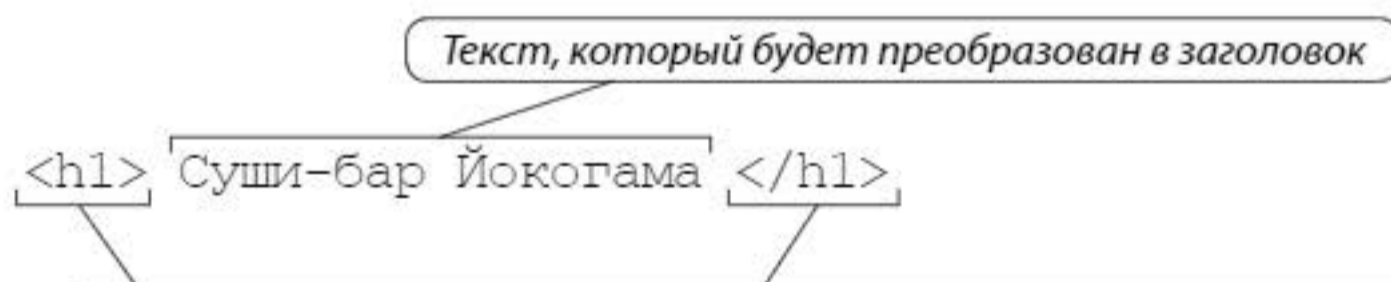
...нетрудно найти часть, создающую «видимое» содержание веб-страницы: заголовок и три абзаца (в приведенном коде и на следующем рисунке она обведена рамкой).



Видно, что она состоит из фрагментов обычного текста и пометок в виде латинских букв и цифр, взятых в угловые скобки (выделены полужирным шрифтом):

```
<h1>Суши-бар Йокогама</h1>  
<p>У нас вы найдете: сашими, суши, роллы, соусы,  
горячие блюда.</p>  
<p>Также - безалкогольные напитки.</p>  
<p>Ждем вас!</p>
```

Рассмотрим первую строку, создающую заголовок:



Тег

Пометка, указывающая веб-обозревателю преобразовать находящийся в нем текст в определенный элемент страницы (в нашем случае — в заголовок).

Пробелы между угловыми скобками (символами «меньше» и «больше») и расположенным между ними именем тега не допускаются.

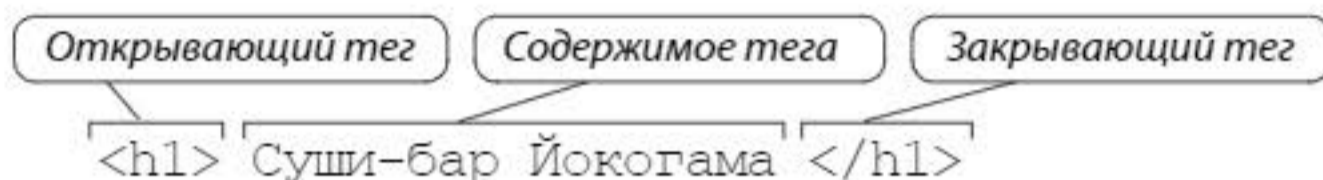
Символ слеша, если он есть, также не отделяется пробелами.

Тег `<h1>` создает заголовок первого уровня, которым предваряются самые крупные фрагменты текста (в нашем случае это сама страница).

Посмотрим на следующую строчку набранного нами кода:

```
<p>У нас вы найдете: сашими, суши, роллы, соусы,  
горячие блюда.</p>
```

Здесь используется тег `<p>`, который выводит на страницу обычный абзац. В последующем коде присутствуют еще два тега `<p>`, создающие остальные абзацы.



Тег `<h1>` — *парный*, так же, как и тег `<p>`.

Парный тег

Включает открывающий тег, содержимое и закрывающий тег.

А что же остальной код (здесь он также обведен рамкой)?

```
<!doctype html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>Йокогама: суши-бар</title>  
  </head>  
  <body>
```

```
    <h1>Суши-бар Йокогама</h1>  
    <p>У нас вы найдете: сашими, суши, роллы,  
соусы, горячие блюда.</p>  
    <p>Также - безалкогольные напитки.</p>  
    <p>Ждем вас!</p>
```

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

Это тоже теги. Они задают сведения о самой странице, используемые веб-обозревателем при ее выводе, или же структурируют код. Мы рассмотрим их в *разд. 1.5*.

Свод описаний различных тегов и правил их применения составляет *язык HTML*.

HTML

HyperText Markup Language (язык гипертекстовой разметки), служащий для написания самого содержания веб-страниц.

Существует несколько разновидностей, или *версий*, этого языка, разработанных в разное время.

HTML 5

Последняя на данный момент версия языка HTML, которую рекомендуется применять в веб-разработке в настоящее время.

Как набирать HTML-код? — часть 1

Обычно теги, создающие абзацы, заголовки и подобные им элементы (мы рассмотрим их на *уроке 4*), размещаются на отдельных строках — ради наглядности:

```
<h1>Суши-бар Йокогама</h1>
<p>Также - безалкогольные напитки.</p>
<p>Ждем вас!</p>
```

Короткие заголовки и абзацы можно набрать в одну строку, чтобы сделать код компактнее:

```
<p>И безалкогольные напитки.</p><p>Ждем вас!</p>
```

Но даже в таком случае каждый заголовок и абзац будут выведены с новой строки.

Слишком длинный текст в заголовках и абзацах можно переносить по строкам:

```
<p>У нас вы найдете: сашими, суши, роллы,
соусы, горячие блюда.</p>
```

В любом случае веб-обозреватель при выводе выполняет перенос строк самостоятельно.

1.3. Упражнение. Продолжаем изучать теги

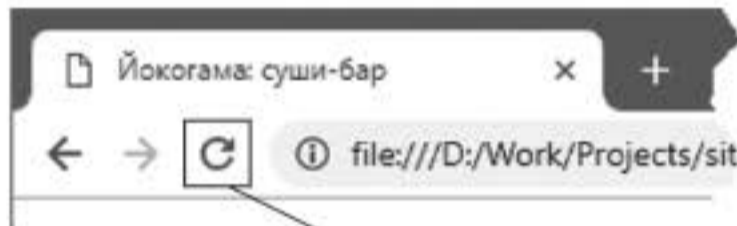
Парный тег `` помечает свое содержимое как очень важный текст. Веб-обозреватель выводит такой текст полужирным шрифтом.

Парный тег `` помечает свое содержимое как просто важный текст. Веб-обозреватель выводит такой текст *курсивом*.

1. Пометим название суши-бара в заголовке как важный текст, а фрагмент «горячие блюда» из первого абзаца — как очень важный. Добавленные теги выделены здесь полужирным шрифтом:

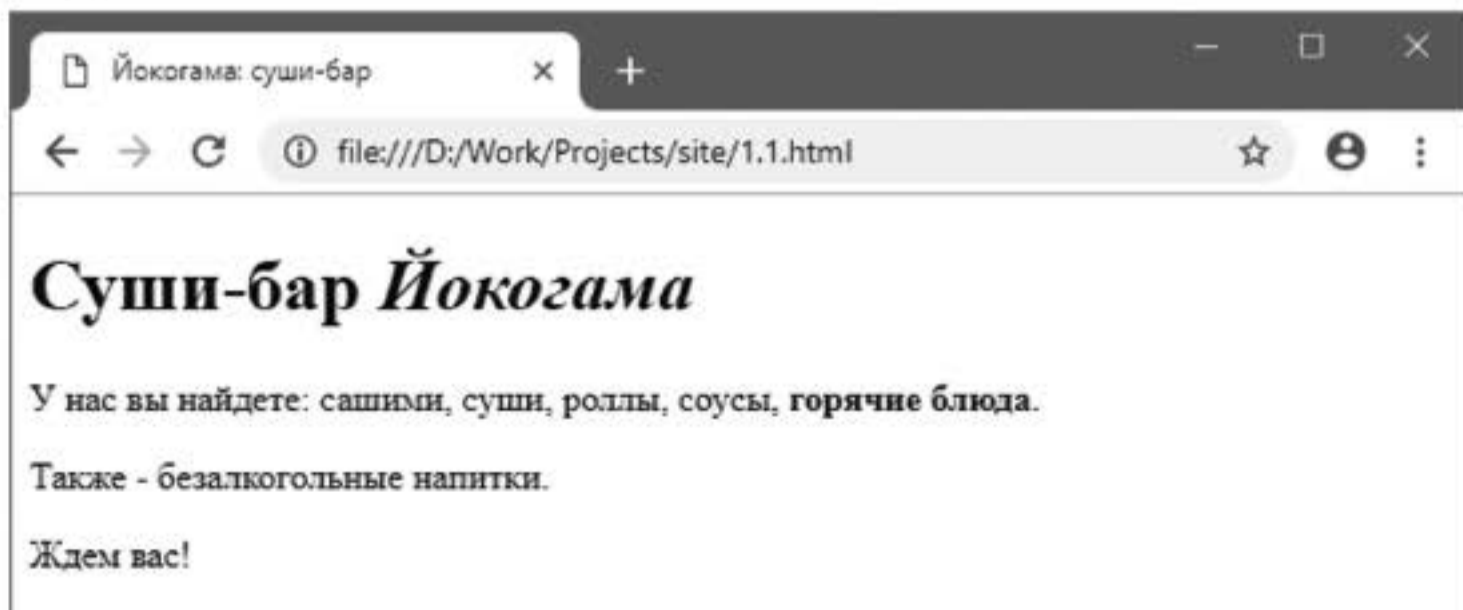
```
<h1>Суши-бар <em>Йокогама</em></h1>
<p>У нас вы найдете: сашими, суши, роллы,
соусы, <strong>горячие блюда</strong>.</p>
```

2. Сохраним исправленный код.
3. Обновим открытую в веб-обозревателе страницу.



Для этого щелкнем расположенную в панели инструментов кнопку **Обновить** (у разных программ она имеет разный вид и разные названия) или просто нажмем клавишу `<F5>`.

Результат



Как набирать HTML-код? — часть 2

Обычно теги, вложенные в какой-либо другой тег и являющиеся его содержимым, для наглядности выделяются увеличенными отступами слева. В следующем примере таким образом выделены теги `<h1>` и `<p>`, вложенные в тег `<body>` (с которым мы познакомимся в *разд. 1.5*):

```
<body>
  <h1>Суши-бар Йокогама</h1>
  <p>У нас вы найдете: сашими, суши, роллы,
  соусы, горячие блюда.</p>
  <p>Также - безалкогольные напитки.</p>
  <p>Ждем вас!</p>
</body>
```

Традиционно такой отступ создается четырьмя пробелами или символом табуляции.

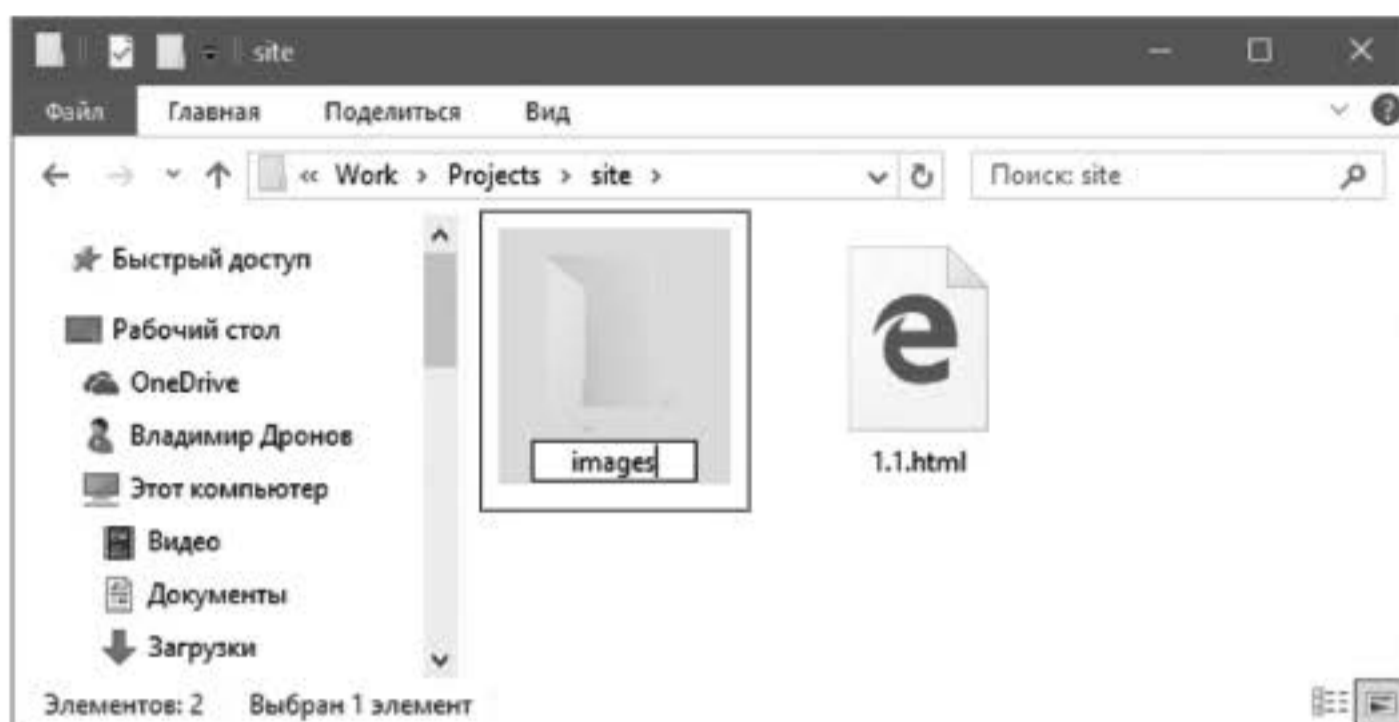
Имена тегов и атрибутов тегов набираются символами в нижнем регистре. Так рекомендует (хотя и не настаивает на этом) язык HTML 5.

1.4. Упражнение. Помещаем на веб-страницу изображения, знакомимся с одинарными тегами и атрибутами тегов

Что-то скучно выглядит наша страничка без картинок... Давайте поместим на нее парочку, применив для этого тег ``.

1. Графические изображения, помещаемые на страницы веб-сайта, рекомендуется сохранять в отдельной папке. Поэтому мы создадим для помещаемых на страницу изображений такую отдельную папку. Назовем ее `images`.

Годом рождения Всемирной паутины считается 1989-й. Именно тогда работавший в ЦЕРН британец Тим Бернерс-Ли разработал сетевой протокол HTTP, язык HTML, на котором пишутся веб-страницы, систему интернет-адресов, первый веб-обозреватель WorldWideWeb и веб-сервер httpd.



2. Поместим в папку `images` файлы `sushi.jpg` и `drink.jpg` с изображениями подходящей тематики. Файлы эти можно взять из папки `1\!sources` сопровождающего книгу электронного архива (см. приложение 4).



Можно также найти подходящие изображения и в Интернете. Изображение `sushi.jpg` должно иметь размеры не менее 250×150 пикселей, а `drink.jpg` — не более 25×25 пикселей (как у символа текста).

3. Добавим в HTML-код нашей страницы необходимые теги (выделены здесь полужирным шрифтом):


```
<h1>Суши-бар <em>Йокогама</em></h1>
<p></p>
<p>У нас вы найдете: сашими, суши, роллы,
соусы, <strong>горячие блюда</strong>.</p>
<p>Также - безалкогольные напитки
.</p>
```

Тег ``, помещающий на страницу изображение, весьма примечателен. Прежде всего, он *одинарный*.

Одинарный тег

Не включает содержимого и закрывающей «пары».

А кроме того, он содержит *атрибут тега*.

Атрибут тега

Свойство элемента страницы, создаваемого тегом. Ставится внутри угловых скобок после имени тега и отделяется от него пробелом.



Значение атрибута тега

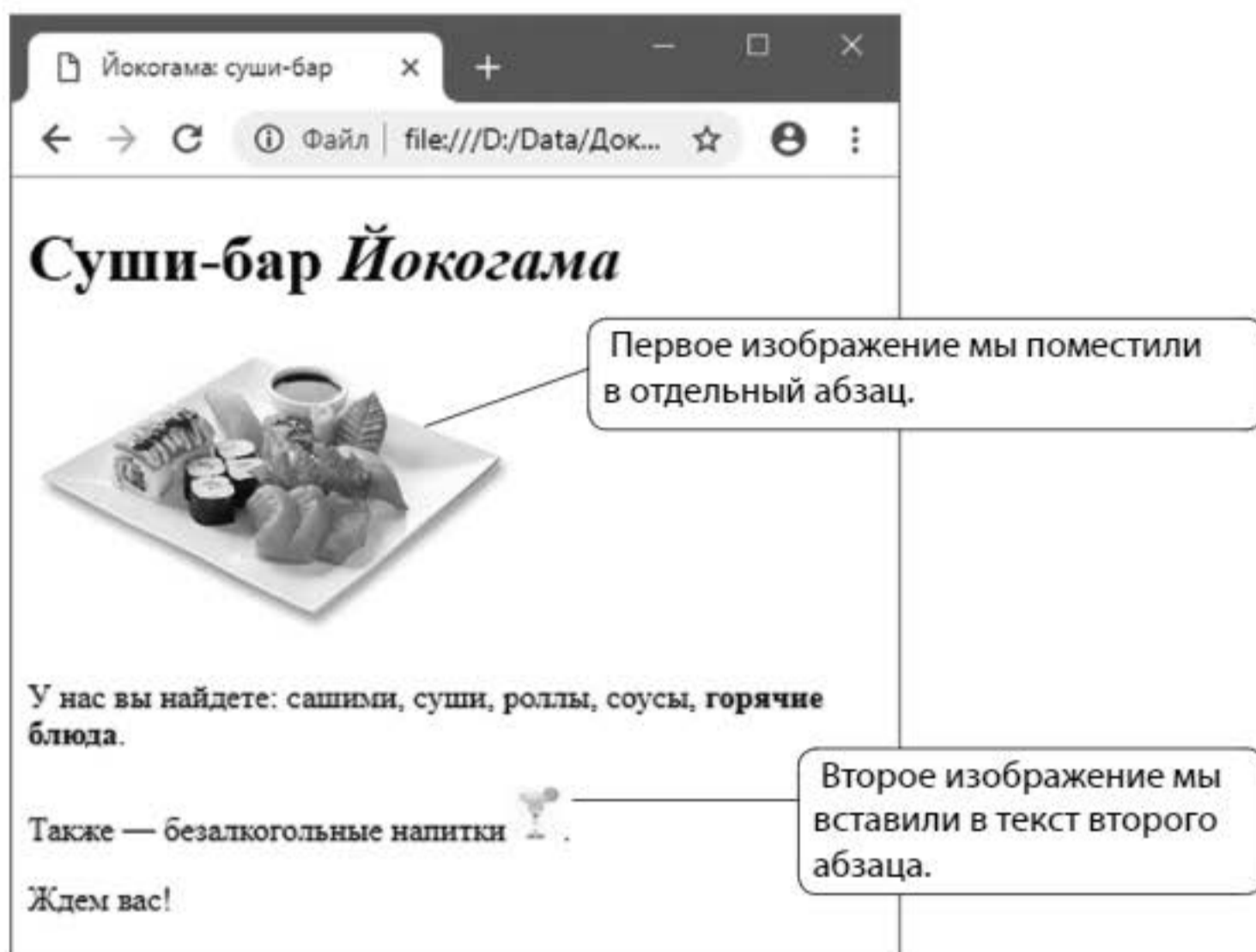
Ставится после имени атрибута тега, отделяется от него знаком равенства и берется в прямые кавычки.

Атрибут `src` тега `` задает ссылку на файл, в котором хранится выводимое на страницу изображение. Такой ссылкой может быть путь (как в нашем случае) или интернет-адрес (о тех и других мы поговорим позже).

4. Сохраним исправленный код и обновим открытую в веб-обозревателе страницу.

В дальнейшем не забываем выполнять эти действия после каждого изменения кода. Автор более не будет напоминать об этом!

Результат



В теге можно записать несколько атрибутов, также разделив их пробелами. Пример:

```

```

Атрибут `title`, поддерживаемый почти всеми тегами HTML, задает текст *всплывающей подсказки*, которая выводится на экран при наведении на элемент страницы курсора мыши.



Не забывайте ставить закрывающие теги!

Пропуск закрывающего тега в парном теге — например, здесь пропущен закрывающий тег `</h1>`:

```
<h1>Суши-бар Йокогама
<p>У нас вы найдете: сашими, суши, роллы,
соусы, горячие блюда.</p>
```

может привести к тому, что веб-страница будет выведена неправильно.

Поэтому никогда не забывайте ставить закрывающие теги.

Неопытные веб-верстальщики часто забывают поставить в закрывающем теге символ слеша, в результате чего у них получается еще один открывающий тег. Такую ошибку выявить довольно трудно, поэтому лучше проверять HTML-код в процессе его набора.

1.5. Структурирование HTML-кода и метаданные

Ранее говорилось, что HTML-код *структурируется* с применением особых тегов — иначе веб-обозреватель его не «поймет». В коде страницы также записываются и теги, хранящие данные о самой странице. Сейчас мы познакомимся с этими тегами (здесь они выделены полужирным шрифтом>):

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Йокогама: суши-бар</title>
  </head>
  <body>
    <h1>Суши-бар <em>Йокогама</em></h1>
    . . .
    <p>Ждем вас!</p>
  </body>
</html>
```

- ◆ `<!doctype>` — одинарный тег, создающий *пролог*.

Пролог

Пометка, обозначающая язык, на котором написана веб-страница, и версию этого языка.

Атрибут без значения `html`, помещенный в тег `<!doctype>`, обозначает язык HTML 5.

Атрибут тега без значения

Атрибут тега, для которого не указывается значение.

- ◆ `<html>` — парный тег, в который помещается весь остальной код страницы.
- ◆ `<head>` — парный тег, создающий *секцию заголовка* страницы.

Секция заголовка

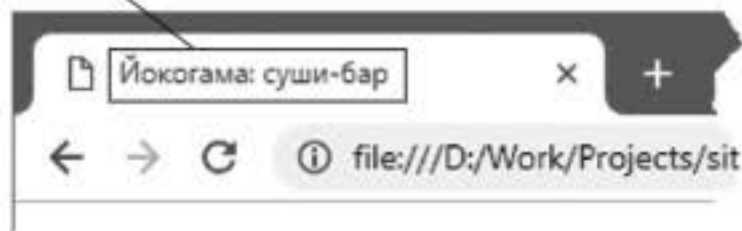
Служит для записи *метаданных* веб-страницы.

Метаданные

Данные, описывающие саму страницу: кодировку, название и др. Они не выводятся на экран, но используются веб-обозревателем при выводе страницы.

- ◆ `<meta>` — одинарный тег, хранящий метаданные с техническими сведениями о странице.
Атрибут `charset` этого тега задает кодировку, в которой сохранена страница (в нашем случае — UTF-8).
Для указания прочих технических сведений применяются другие атрибуты тега `<meta>`. Мы узнаем о них на *уроке 9*.
- ◆ `<title>` — парный тег (не путать с одноименным атрибутом тега!), указывающий *название* страницы.

Название веб-страницы



Название веб-страницы

Выводится на корешке вкладки веб-обозревателя, на которой открыта страница.

- ◆ `<body>` — парный тег, создающий *секцию тела* страницы.

Секция тела

В ней помещается HTML-код, который создает «видимое», непосредственно выводимое на экран содержание страницы.

1.6. Комментарии HTML

Мы можем помещать в HTML-код страниц *комментарии* — замечания и пояснения для себя или своих коллег.

Комментарий HTML

Фрагмент HTML-кода, не обрабатываемый веб-обозревателем и не выводимый на экран. Применяется для внесения в код всевозможных заметок и примечаний.

Пример комментария (выделен полужирным шрифтом):

```
<!--  
    Вывести этот заголовок  
    по центру страницы  
-->  
<h1>Суши-бар Йокогама</h1>
```

- ◆ `<!--` — открывающий тег комментария.
Содержание комментария может иметь любой объем и быть каким угодно. Оно даже может включать в себя фрагмент HTML-кода, который в таком случае не будет обработан веб-обозревателем.
- ◆ `-->` — закрывающий тег комментария.

Как еще можно использовать комментарии?

Мы можем поместить в комментарий часть HTML-кода страницы (*закомментировать* его), чтобы выяснить, как страница отображается без этого фрагмента. Это может помочь при выявлении ошибок в HTML-коде.

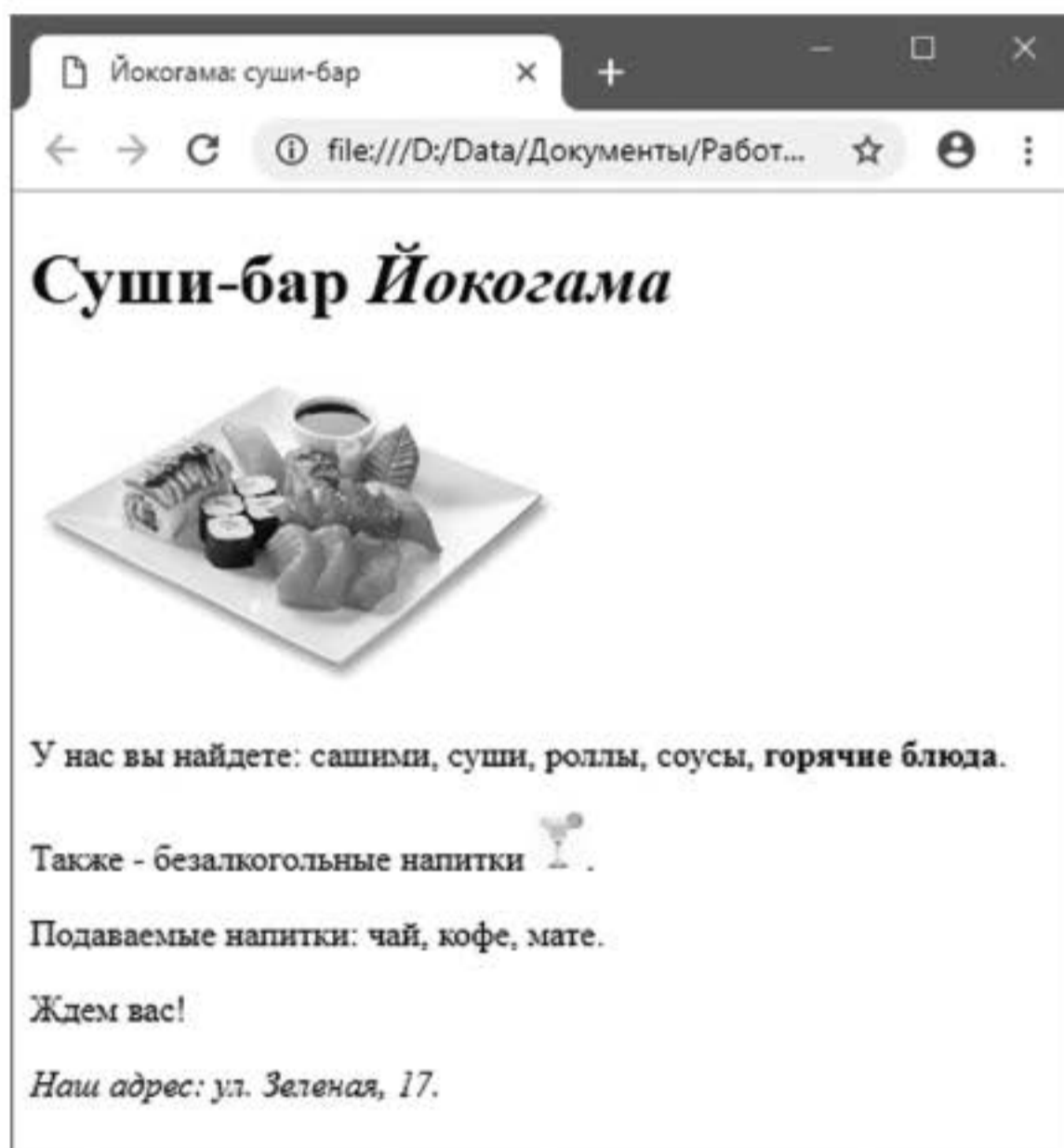
Выявив ошибку, мы уберем теги комментария, в который заключили фрагмент (*раскомментируем* его).

1.7. Самостоятельные упражнения

Добавьте на созданную ранее веб-страницу 1.1.html:

- ◆ список подаваемых напитков (отдельным абзацем);
- ◆ физический адрес суши-бара (используйте парный тег `<address>`).

У вас должно получиться так:



Урок 2. CSS

Оформление веб-страниц

Таблицы стилей

Стили

Атрибуты стилей

Наследование

Правила каскадности

2.1. Оформление по умолчанию

Посмотрим на нашу веб-страничку:

Сейчас она оформлена по умолчанию:

- ◆ заголовок выводится полужирным шрифтом большого размера (который установлен веб-обозревателем);
- ◆ абзацы и адрес выводятся шрифтом обычной насыщенности и небольшого размера (размер также установлен веб-обозревателем);
- ◆ гарнитура шрифта для вывода текста выбирается веб-обозревателем;
- ◆ текст и изображение выравниваются по левому краю;
- ◆ между элементами делаются отступы сверху и снизу.



Оформление веб-страницы по умолчанию

Оформление, которое веб-обозреватель самостоятельно задает для страницы в том случае, если веб-верстальщик не указал для нее свое оформление.

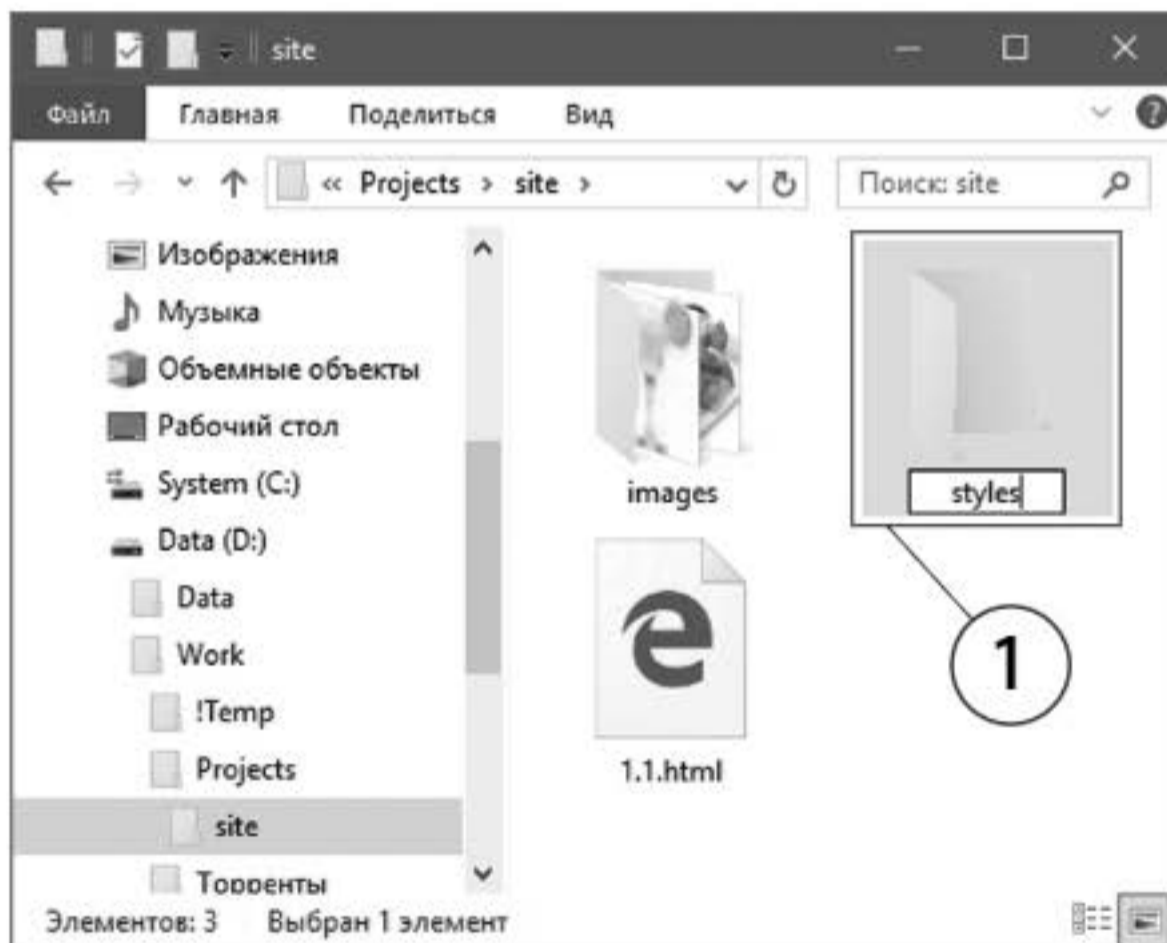
2.2. Упражнение. Задаем оформление для веб-страницы

Давайте же зададим свое собственное оформление для нашей странички, и в частности:

- ◆ для вывода текста определим шрифт Arial;
- ◆ для заголовка установим размер (*кегель*) шрифта, равным 32 пункта, и выравнивание по центру;
- ◆ а для абзаца — 14 пунктов и выравнивание по ширине.

Чтобы осуществить намеченное:

1. Создадим для файлов с представлением отдельную папку (файлы, в которых хранится представление веб-страниц, рекомендуется сохранять в отдельной папке). Назовем ее *styles*.



2. Запустим Блокнот и наберем в нем следующий код, стараясь не допускать ошибок:

```
body {font-family: Arial;}
h1 {
    font-size: 32pt;
    text-align: center;
}
```

```
p {  
    font-size: 14pt;  
    text-align: justify;  
}
```

3. Сохраним набранный код в файле в созданной ранее папке `styles`. Дадим файлу имя `2.1.css`. В качестве кодировки укажем: **UTF-8**.

Не забываем в раскрывающемся списке **Тип файла** указывать **Все файлы (*.*)**, чтобы Блокнот при сохранении не добавил к имени сохраняемого файла расширение `txt`.

4. В HTML-код веб-страницы добавим следующий фрагмент (выделен полужирным шрифтом):

```
<head>  
    <meta charset="UTF-8">  
    <title>Йокогама: суши-бар</title>  
    <link href="styles/2.1.css" rel="stylesheet"  
        type="text/css">  
</head>
```

5. Откроем страницу `1.1.html` в веб-обозревателе (или обновим, если она уже открыта).

Результат

