



# Знакомство с HTML и CSS



### В этой главе

- Основы HTML.
- Основы CSS.
- Комбинирование HTML и CSS в целях создания веб-страниц.
- Знакомство с песочницей Web Design Playground.

Независимо от сложности произведения музыкант, играя, использует комбинации из 12 нот (от «до» до «си» и пять полутонов между ними). Художник, работая над шедевром, независимо от количества оттенков использует комбинации трех основных цветов (красного, желтого и синего). Поэты, создавая из слов самые проникновенные стихи, используют комбинации доступных букв алфавита. Эти примеры показывают, что для творчества и игр не нужны сложные ресурсы или исходные материалы. Воображение и любопытство в сочетании с несколькими строительными блоками — это все, что вам нужно, чтобы выразить себя практически в любом искусстве, в том числе в дизайне веб-страниц. Как вы узнаете из данной главы, да и, впрочем, из всей книги, HTML и CSS являются такими блоками. И хотя этих строительных блоков больше, чем музыкальных нот, основных цветов или даже букв алфавита, их количество ограничено, но его более чем достаточно, чтобы позволить вам выразить себя на захватывающем современном холсте — в Интернете.



## ЧТО ТАКОЕ HTML

Самое сложное в HTML — это его название. HTML расшифровывается как *Hypertext Markup Language* (*язык разметки гипертекста*), что звучит примерно так же пугающе, как налоговая проверка. Но вы успокойтесь, когда поймете значение всех терминов.

Начну с *гипертекста*. Как вы наверняка знаете, *ссылка* — это специальное слово либо фраза (или даже изображение) на веб-странице, которое «ссылается» на другую веб-страницу. Когда вы щелкаете кнопкой мыши на такой ссылке или касаетесь ее пальцем, ваш браузер сразу же открывает другую страницу. Люди, которые изобрели Интернет, использовали для обозначения такого специального текста длинный термин «*гипертекстовая ссылка*». (Приставка «*гипер*» означает «*за пределами*».) Гипертекстовые ссылки — отличительная черта Всемирной паутины, поэтому страницы часто называют *гипертекстовыми документами*. Итак, в *HTML* есть некий *гипертекст*, и вы используете его для создания гипертекстовых документов. (Хм, этот язык можно было бы назвать WPML, Web Page Markup Language — язык разметки веб-страниц.)

В словаре я нашел следующее определение *разметки*: «подробные стилистические инструкции по оформлению рукописи, которая подлежит верстке» (среди прочего). Исходя из целей текущей главы, я могу перефразировать это определение так: «подробные стилистические инструкции по оформлению текстового документа, который должен быть опубликован во Всемирной паутине». Это и есть краткое описание языка HTML. В нем используется несколько простых текстовых кодов — так называемых *тегов* — для подробного описания таких инструкций, как разделение текста на абзацы, создание маркированных списков, добавление изображений и (конечно же) оформление ссылок. Вы добавляете эти теги в определенных местах обычного текстового документа, а браузер выполняет всю остальную работу по преобразованию — или *отображению* — разметки в абзацы, списки, ссылки и т. д. Что происходит, когда вы загружаете текстовый документ в браузере? Он не показывает теги, а вместо этого оформляет содержимое страницы в соответствии с добавленными вами тегами.

Слово «*язык*» тоже может напугать, поскольку сразу возникает мысль, что HTML — это язык программирования. К счастью, HTML не имеет ничего общего с компьютерным программированием. Скорее, это «язык» в том смысле, что в нем есть небольшая коллекция «слов», то есть *тегов*, с помощью которых вы указываете, как должен выглядеть текст, — например, в виде заголовка или нумерованного списка.



Короче говоря, размечать содержимое с помощью HTML-тегов — значит вставлять их в стратегически важные места между фрагментами обычного текста таким образом, чтобы в итоге получилась самая настоящая веб-страница. Как бы неожиданно это ни звучало для вас сейчас, к концу работы над книгой вы создадите несколько интересных HTML-проектов.

## ***Возможности HTML***

Добавляя HTML-разметку в документ, вы, по сути, даете браузеру инструкции, как страница должна выглядеть в его окне. С помощью HTML-разметки формируется общая структура страницы, и браузер понимает, какой должна быть каждая ее часть. В HTML вы задаете, к примеру, такие инструкции:

- использовать указанную строку текста в качестве основного заголовка страницы;
- оформить указанные строки текста как подзаголовки;
- выделить указанный фрагмент текста в отдельный абзац;
- превратить указанные пять последовательных пунктов в маркированный список;
- преобразовать указанные шесть последовательных шагов в нумерованный список;
- превратить указанную фразу в ссылку.

Эти инструкции могут казаться немного абстрактными, поэтому я покажу конкретный пример HTML-разметки.

## ***От обычного текста к HTML: пример***

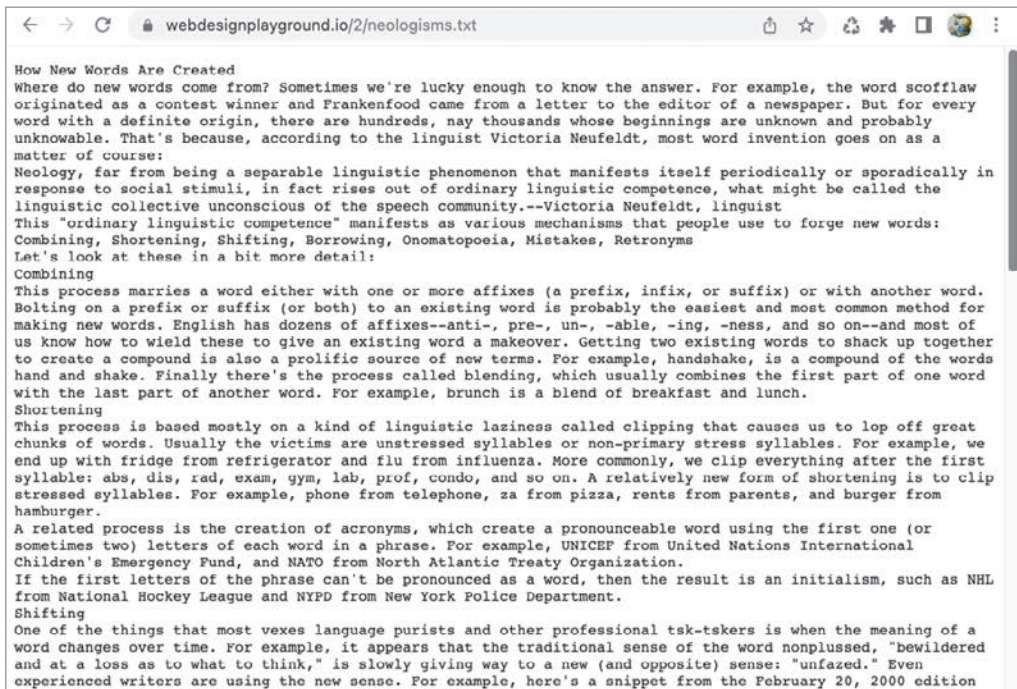
На рис. 1.1 показан обычный текстовый документ, отображаемый в браузере. Как видите, за исключением нескольких разрывов строк, браузер выводит массив неотформатированного, неразмеченного текста. Его трудно читать и из него очень сложно извлечь смысл, поскольку он никак не дифференцирован.

Язык HTML предоставляет средства, позволяющие сделать обычный сплошной текст более читабельным и отобразить его так, чтобы его было легче понять. На рис. 1.2 показан текст с рис. 1.1, но уже с HTML-разметкой.

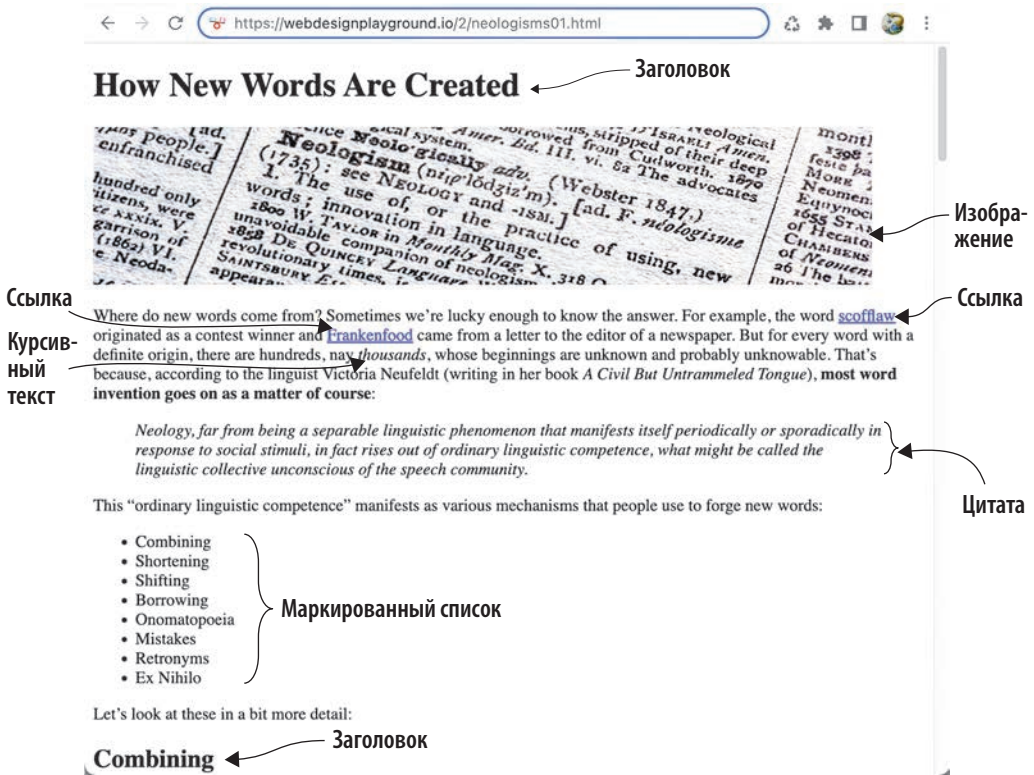
Здесь я использовал заголовки для оформления названия статьи в верхней части и раздела в нижней части. Обратите внимание, что



заголовок раздела отображается шрифтом чуть меньшего размера, чем название, благодаря чему понятна структура текста. Кроме того, с помощью HTML я добавил изображение, помогающее привлечь внимание. Чтобы вы поняли, как устроен гипертекст, скрывающийся под буквой *H* в названии *HTML*, я оформил два слова в виде ссылок на другие сайты. В этой главе вы узнаете, что обычно текст форматируется с помощью CSS, но для простого форматирования можно использовать и HTML, например выделяя текст курсивом, как показано в примере. Еще я оформил цитату, которую браузер отображает с отступом от обычного текста, и выделил ее курсивом. Наконец, с помощью HTML-разметки я создал маркированный список. Теперь, когда вы увидели возможности HTML, пришло время поговорить об инструкциях для браузера, содержащих указание о том, как должна выглядеть ваша страница.



► **Рис. 1.1.** Браузер может отображать обычные текстовые файлы, но их очень трудно читать



► **Рис. 1.2.** Благодаря определенным HTML-элементам текст с рис. 1.1 легче читать и понимать и в нем проще ориентироваться

## ЧТО ТАКОЕ CSS

Одним из первых этапов работ при строительстве дома является возведение фундамента — базовой структуры для пола, стен и крыши. Именно такие основные элементы вы создаете, добавляя HTML-разметку на свою страницу: вы указываете, что должно появиться на ней, какими должны быть ее различные элементы (например, заголовок, абзац или список) и в каком порядке их следует отображать.

Дом без отделки непригоден для жилья. Точно так же и ваш документ не будет современной веб-страницей, если не использовать CSS в качестве такой финишной отделки. CSS расшифровывается как *Cascading Style Sheets* — *каскадные таблицы стилей*, и, как и в случае



с HTML, его название сложнее, чем предназначение. Проясню для вас значение каждого слова, хотя в данном случае в целях упрощения буду обращаться к ним не по порядку.

Начнем с того, что *стиль* — это инструкция, выдающая браузеру указание изменить способ отображения какого-либо элемента на странице, например слова, абзаца или всех экземпляров определенного HTML-элемента. Обычно такие изменения касаются форматирования, например изменения шрифта или цвета текста. Но с помощью стилей можно управлять и макетом страницы и даже добавлять анимированные эффекты. Если вы когда-нибудь использовали стили в текстовых редакторах, то уже имеете представление о том, каковы возможности стилей веб-страниц.

А что такое *таблица*? На заре издательского бизнеса компании выпускали руководства, в которых определяли предпочтительное форматирование шрифтов, заголовков, цитат и т. д. Это форматирование было известно как *фирменный стиль*, а руководство называлось *таблицей стилей*. В веб-дизайне таблица стилей выполняет, по сути, те же функции — это набор стилей, которые применяются к конкретной веб-странице.

Теперь перейдем к слову «каскадные» в названии CSS. Существуют разные источники стилей. Одни стили поступают от браузера, другие — от пользователя (например, если пользователь настроил свой браузер на использование другого размера шрифта по умолчанию), а третьи — из ваших таблиц стилей. Когда эти стили накладываются друг на друга, браузер использует сложный алгоритм, чтобы решить, какой стиль будет применен. Этот алгоритм называется *каскадом*.

Таким образом, CSS используется для определения внешнего вида страницы. Вам кажется, что CSS предназначен только для того, чтобы страница выглядела привлекательно? Это действительно так: CSS предлагает инструменты для внесения только стандартных и несущественных изменений. Однако то, как выглядит ваша страница, не менее важно, чем то, что она содержит, поскольку мало кто станет читать текст, который отформатирован плохо или бессистемно.

## ***Разделение структуры и представления***

Пока вы пытаетесь разобраться в различиях между HTML и CSS, обращу ваше внимание на ключевую разницу. Я немного обобщил, но основная разница в том, что:

- HTML определяет общую структуру веб-страницы;
- CSS определяет ее визуальное представление.

### **ВНИМАНИЕ**

*Принцип каскадности, безусловно, самый сложный и запутанный аспект CSS. Я расскажу о нем позже (в главе 19).*



Частично они перекрываются (например, HTML может влиять на вид страницы, а CSS — на структуру макета), но в целом отвечают каждый за свое: HTML — за структуру, CSS — за представление. Это различие важно учитывать, поскольку, разделяя эти два момента при верстке веб-страницы, легче ее создавать, поддерживать и настраивать.

## ***Возможности CSS***

Добавляя CSS-стили в документ, вы сообщаете браузеру, как должны выглядеть определенные элементы. Каждый стиль — своего рода инструкция по форматированию. Такие инструкции можно использовать самыми разными способами, как показано в следующих примерах:

- отобразить все ссылки красным текстом;
- использовать определенный шрифт для всех заголовков;
- добавить дополнительное пространство вокруг абзаца;
- добавить тень к фотографии;
- использовать строчные римские цифры для всех нумерованных списков;
- закрепить указанный раздел текста в правой крайней части окна;
- повернуть указанный рисунок на 45 градусов.

Приведу наглядный пример, чтобы конкретизировать этот список.

## ***От структуры к представлению: пример CSS***

Ранее в этой главе я взял обычный текстовый документ (см. рис. 1.1) и добавил немного HTML-тегов, чтобы структурировать его и облегчить восприятие информации (см. рис. 1.2). На рис. 1.3 я применил несколько стилей, чтобы страница выглядела лучше.

Приведу краткое описание основных изменений в стиле, которые я внес:

- основной заголовок отображен увеличенным шрифтом, дан малыми прописными буквами и выровнен по центру;
- к фотографии добавлена тень;
- Обычный текст немного уменьшился;
- убрано подчеркивание из ссылок;
- цитата в тексте оформлена более светлым цветом;
- маркированный список разделен на два столбца;
- увеличены боковые поля.



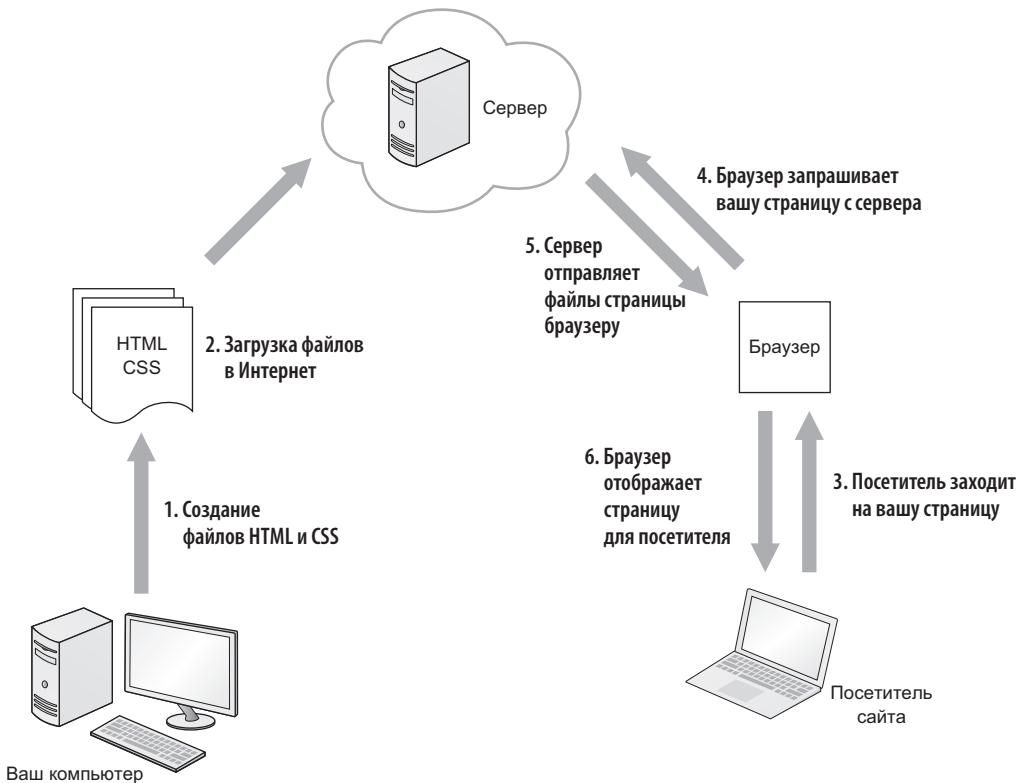


Для выполнения подобных задач требуется язык программирования, такой как JavaScript или PHP, которые не рассматриваются в этой книге.

## КАК HTML И CSS СОЗДАЮТ ВСЕМИРНУЮ ПАУТИНУ

С Интернетом связано множество удивительных фактов. Приведу один из них: за исключением дополнительных элементов, таких как изображения, видео и звуки, страницы Интернета состоят только из текста. Именно так: почти все, что вы видите в Сети, было создано из совокупностей букв, цифр и символов, набранных на клавиатуре.

Этот факт вызывает много вопросов, поэтому я немного углублюсь в детали. Процесс состоит из следующих этапов (рис. 1.4).



► **Рис. 1.4.** Чтобы превратить HTML- и CSS-код в веб-страницу, вы отправляете его на веб-сервер, а посетители используют свои браузеры для получения и отображения вашего кода на странице