

IT для начинающих
и опытных

CSS

для профессионалов

Руководство по разработке в примерах
от сообщества **Stack Overflow**

Издательство АСТ
Москва

УДК 004.774.6
ББК 32.973.202-018.2
Ц99

Последнюю версию книги «CSS Notes for Professionals» на английском языке можно загрузить по адресу: <https://goalkicker.com/CSSBook>.

Вы можете делиться этим PDF-документом с кем угодно бесплатно.

Книга создана на основе документации Stack Overflow (<https://archive.org/details/documentation-dump.7z>), содержание которой написано замечательными участниками сообщества Stack Overflow.

Текстовые материалы распространяются на условиях лицензии Creative Commons BY-SA.

Список авторов, участвовавших в создании различных глав, приведен в разделе «Благодарности» в конце книги.

Изображения могут являться объектами авторского права соответствующих владельцев, если не указано иное.

Ц99 **CSS** для профессионалов. Руководство по разработке в примерах от сообщества Stack Overflow. — Москва : Издательство АСТ, 2025. — 288 с. : ил. — (IT для начинающих и опытных).

ISBN 978-5-17-174752-7.

Это подробное руководство по каскадным таблицам стилей (CSS), основанное на лучших практиках и решениях, предложенных специалистами крупнейшего в мире сообщества разработчиков — Stack Overflow.

Книга охватывает весь спектр возможностей CSS: от базового синтаксиса и правил форматирования до сложных тем, таких как Flexbox, Grid, анимации, трансформации и управление слоями. В ней представлены разъяснения ключевых концепций, примеры кода и советы по оптимизации, которые помогут как начинающим, так и опытным разработчикам углубить свои знания и улучшить навыки веб-стилизации.

Вы узнаете, как правильно работать с селекторами, медиазапросами, типографикой, тенями, градиентами, масками, фильтрами и многими другими аспектами CSS. Особое внимание уделено кроссбраузерной совместимости, производительности и современным подходам к верстке.

Это идеальное руководство для тех, кто хочет освоить CSS с нуля или вывести свои знания на новый уровень, изучая проверенные временем решения, которыми делятся эксперты Stack Overflow.

УДК 004.774.6
ББК 32.973.202-018.2

ISBN 978-5-17-174752-7

Перевод на русский язык: ООО «Интеджер».
Издание на русском языке: ООО «Издательство АСТ».

Содержание

Раздел 1. Начало работы с CSS	7
Глава 1.1. Внешняя таблица стилей	7
Глава 1.2. Внутренние стили	9
Глава 1.3. Правило CSS @import (одно из правил CSS at-rule)	9
Глава 1.4. Встроенные стили (Inline Styles)	10
Глава 1.5. Изменение CSS с помощью JavaScript.....	11
Глава 1.6. Стилизация списков с помощью CSS	12
Раздел 2. Структура и форматирование CSS-правил	13
Глава 2.1. Списки свойств	13
Глава 2.2. Множественные селекторы.....	13
Глава 2.3. Правила, селекторы и блоки объявлений.....	14
Раздел 3. Комментарии	15
Глава 3.1. Однострочные комментарии	15
Глава 3.2. Многострочные комментарии	15
Раздел 4. Селекторы	16
Глава 4.1. Базовые селекторы.....	16
Глава 4.2. Селекторы атрибутов.....	17
Глава 4.3. Комбинаторы.....	20
Глава 4.4. Псевдоклассы	22
Глава 4.5. Псевдокласс :nth-child	26
Глава 4.6. Селекторы по имени класса	27
Глава 4.7. Выбор элемента по его ID без высокой специфичности селектора ID	28
Глава 4.8. Селектор :last-of-type	28
Глава 4.9. Пример селектора :in-range в CSS3	29
Глава 4.10. А. Пример псевдокласса :not () В. Псевдокласс :focus-within в CSS.....	29
Глава 4.11. Глобальный булевый флажок с использованием checkbox: checked и ~ (общий комбинатор соседних элементов)	30
Глава 4.12. Селекторы по ID	31
Глава 4.13. Как стилизовать элемент Range input	32
Глава 4.14. Пример псевдокласса :only-child.....	32
Раздел 5. Фоны	33
Глава 5.1. Цвет фона	33
Глава 5.2. Градиенты фона	36
Глава 5.3. Фоновое изображение.....	38
Глава 5.4. Сокращенная запись при работе с фоном (Background Shorthand).....	39
Глава 5.5. Размер фона (Background Size)	41
Глава 5.6. Позиция фона (Background Position)	46
Глава 5.7. Свойство background-origin	47
Глава 5.8. Множественные фоновые изображения.....	50
Глава 5.9. Свойство background-attachment.....	50
Глава 5.10. Свойство background-clip.....	52
Глава 5.11. Свойство background-repeat.....	53
Глава 5.12. Свойство background-blend-mode	54
Глава 5.13. Цвет фона с прозрачностью	54
Раздел 6. Центрирование	55
Глава 6.1. Использование Flexbox	55
Глава 6.2. Использование CSS transform	56
Глава 6.3. Использование margin: 0 auto;.....	58
Глава 6.4. Использование text-align.....	59
Глава 6.5. Использование position: absolute.....	60
Глава 6.6. Использование calc()	61
Глава 6.7. Использование line-height.....	61
Глава 6.8. Вертикальное выравнивание чего угодно с помощью трех строк кода	62
Глава 6.9. Центрирование относительно другого элемента	62
Глава 6.10. Метод «призрачного элемента»	63
Глава 6.11. Центрирование по вертикали и горизонтали без учета высоты и ширины.....	65
Глава 6.12. Вертикальное выравнивание изображения внутри div.....	66
Глава 6.13. Центрирование с фиксированным размером	67
Глава 6.14. Вертикальное выравнивание элементов с динамической высотой	68
Глава 6.15. Горизонтальное и вертикальное центрирование с использованием табличной верстки.....	69
Раздел 7. Блочная модель (Box Model)	70
Глава 7.1. Что такое блочная модель?.....	70
Глава 7.2. box-sizing	73
Раздел 8. Внешние отступы (Margins)	75
Глава 8.1. Схлопывание отступов (Margin Collapsing).....	75
Глава 8.2. Применение внешнего отступа (margin) на определенной стороне.....	78
Глава 8.3. Упрощение свойства margin	79
Глава 8.4. Горизонтальное центрирование элементов на странице с использованием margin	80
Глава 8.5. Пример 1.....	80
Глава 8.6. Отрицательные отступы (margin)	81

Раздел 9. Внутренние отступы (Padding)	82
Глава 9.1. Сокращенная запись внутренних отступов (padding)	82
Глава 9.2. Внутренние отступы для определенной стороны	84
Раздел 10. Граница	85
Глава 10.1. border-radius	85
Глава 10.2. border-style	87
Глава 10.3. Множественные границы	87
Глава 10.4. border (сокращенные записи)	88
Глава 10.5. border-collapse	89
Глава 10.6. border-image	89
Глава 10.7. Создание многоцветной границы с использованием border-image	90
Глава 10.8. border-[left right top bottom]	91
Раздел 11. Контуры (Outlines)	92
Глава 11.1. Обзор	92
Глава 11.2. outline-style	93
Раздел 12. Переполнение (Overflow)	95
Глава 12.1. overflow-wrap	95
Глава 12.2. overflow-x и overflow-y	97
Глава 12.3. overflow: scroll	97
Глава 12.4. overflow: visible	98
Глава 12.5. Создание блочного контекста форматирования с помощью overflow	99
Раздел 13. Медиазапросы	101
Глава 13.1. Терминология и структура	102
Глава 13.2. Базовый пример	103
Глава 13.3. Тип медиа (mediatype)	104
Глава 13.4. Медиазапросы для Retina и не-Retina экранов	105
Глава 13.5. Ширина (width) против области просмотра (viewport)	106
Глава 13.6. Использование медиазапросов для адаптации к разным размерам экрана	107
Глава 13.7. Использование в теге <link>	107
Глава 13.8. Медиазапросы и IE8	108
Раздел 14. Плавающие элементы (Floats)	109
Глава 14.1. Обтекание изображения текстом	109
Глава 14.2. Свойство clear	110
Глава 14.3. Clearfix	111
Глава 14.4. Строчные DIV с использованием float	113
Глава 14.5. Использование свойства overflow для очистки плавающих элементов	114
Глава 14.6. Простой макет из двух колонок фиксированной ширины	114
Глава 14.7. Простой макет из трех колонок фиксированной ширины	116
Глава 14.8. Двухколоночный макет «Ленивый/Жадный»	117
Раздел 15. Типографика	119
Глава 15.1. Сокращенная запись для шрифта (Font Shorthand)	119
Глава 15.2. quotes	120
Глава 15.3. Размер шрифта	121
Глава 15.4. Направление текста	121
Глава 15.5. Стек шрифтов (Font Stacks).....	122
Глава 15.6. Переполнение текста (Text Overflow)	122
Глава 15.7. Тень текста (Text Shadow).....	123
Глава 15.8. Трансформация текста (Text Transform).....	124
Глава 15.9. Межбуквенный интервал (Letter Spacing).....	124
Глава 15.10. Отступ текста (Text Indent).....	125
Глава 15.11. Оформление текста (Text Decoration).....	125
Глава 15.12. Межсловный интервал (Word Spacing).....	126
Глава 15.13. Вариант шрифта (Font Variant)	127
Раздел 16. Flexbox	128
Глава 16.1. Динамическое вертикальное и горизонтальное выравнивание (align-items, justify-content)	128
Глава 16.2. Липкий футер с переменной высотой	132
Глава 16.3. Оптимальное размещение элементов внутри контейнера	133
Глава 16.4. «Святой Грааль» (Holy Grail Layout) с использованием Flexbox	134
Глава 16.5. Идеальное выравнивание кнопок внутри карточек с использованием Flexbox	136
Глава 16.6. Одинаковая высота вложенных контейнеров	139
Раздел 17. Каскадирование и специфичность	140
Глава 17.1. Расчет специфичности селекторов	140
Глава 17.2. Объявление !important	143
Глава 17.3. Каскадирование (Cascading)	144
Глава 17.4. Более сложный пример специфичности	146
Раздел 18. Цвета	148
Глава 18.1. currentColor	148
Глава 18.2. Ключевые слова цветов	149
Глава 18.3. Шестнадцатеричные значения (Hexadecimal Value)	157
Глава 18.4. Нотация rgb().....	158
Глава 18.5. Нотация rgba().....	158
Глава 18.6. Нотация hsl().....	159
Глава 18.7. Нотация hsla()	160
Раздел 19. Прозрачность (Opacity)	161
Глава 19.1. Свойство opacity	161
Глава 19.2. Совместимость с IE для свойства opacity	161
Раздел 20. Единицы измерения длины	162
Глава 20.1. Создание масштабируемых элементов с использованием rem и em	163
Глава 20.2. Размер шрифта с rem	164

Глава 20.3. <code>min</code> и <code>max</code>	165
Глава 20.4. <code>vh</code> и <code>vw</code>	165
Глава 20.5. Использование процентов %	165
Раздел 21. Псевдоэлементы	167
Глава 21.1. Псевдоэлементы	168
Глава 21.2. Псевдоэлементы в списках	168
Раздел 22. Позиционирование	170
Глава 22.1. Перекрытие элементов с помощью <code>z-index</code>	170
Глава 22.2. Абсолютное позиционирование	172
Глава 22.3. Фиксированное позиционирование	173
Глава 22.4. Относительное позиционирование	173
Глава 22.5. Статическое позиционирование	174
Раздел 23. Управление макетом	175
Глава 23.1. Свойство <code>display</code>	176
Глава 23.2. Создание структуры таблицы с использованием <code>div</code>	179
Раздел 24. Grid	180
Глава 24.1. Базовый пример	180
Раздел 25. Таблицы	182
Глава 25.1. <code>table-layout</code>	182
Глава 25.2. <code>empty-cells</code>	182
Глава 25.3. <code>border-collapse</code>	183
Глава 25.4. <code>border-spacing</code>	184
Глава 25.5. <code>caption-side</code>	184
Раздел 26. Переходы (Transitions)	185
Глава 26.1. Сокращенная запись перехода (Transition shorthand)	185
Глава 26.2. Функция <code>cubic-bezier</code>	186
Глава 26.3. Переход (длинная запись)	187
Раздел 27. Анимации	188
Глава 27.1. Анимации с использованием ключевых кадров (keyframes)	188
Глава 27.2. Анимации с использованием свойства <code>transition</code>	190
Глава 27.3. Примеры синтаксиса	191
Глава 27.4. Увеличение производительности анимаций с использованием атрибута <code>will-change</code>	192
Раздел 28. 2D-трансформации	193
Глава 28.1. Поворот (Rotate)	194
Глава 28.2. Масштабирование (Scale)	194
Глава 28.3. Скос (Skew)	195
Глава 28.4. Множественные трансформации	195
Глава 28.5. Перемещение (Translate)	196
Глава 28.6. Центр трансформации (Transform Origin)	197
Раздел 29. 3D-трансформации	198
Глава 29.1. Форма стрелки компаса или иглы с использованием 3D-трансформаций	198
Глава 29.2. 3D-эффект текста с тенью	199
Глава 29.3. <code>backface-visibility</code>	200
Глава 29.4. 3D-куб	201
Раздел 30. Свойство Filter	203
Глава 30.1. Размытие (Blur)	203
Глава 30.2. Тень (Drop Shadow) (используйте <code>box-shadow</code> , если возможно)	204
Глава 30.3. Поворот оттенка (Hue Rotate)	204
Глава 30.4. Множественные значения фильтров	205
Глава 30.5. Инверсия цвета (Invert Color)	206
Раздел 31. Стилизация курсора	207
Глава 31.1. Изменение типа курсора	207
Глава 31.2. <code>pointer-events</code>	208
Глава 31.3. <code>caret-color</code>	208
Раздел 32. <code>box-shadow</code>	210
Глава 32.1. Тень только снизу с использованием псевдоэлемента	210
Глава 32.2. Внешняя тень	211
Глава 32.3. Внутренняя тень	211
Глава 32.4. Множественные тени	212
Раздел 33. Формы для плавающих элементов (Floats)	213
Глава 33.1. Свойство <code>Shape Outside</code> с базовой формой — <code>circle()</code>	213
Глава 33.2. Свойство <code>Shape Margin</code>	215
Раздел 34. Стили списков	217
Глава 34.1. Позиция маркера	217
Глава 34.2. Удаление маркеров / номеров	217
Глава 34.3. Тип маркера или нумерации	218
Раздел 35. Счетчики (Counters)	219
Глава 35.1. Применение стиля римских цифр к выводу счетчика	219
Глава 35.2. Нумерация каждого элемента с использованием CSS-счетчика	220
Глава 35.3. Реализация многоуровневой нумерации с использованием CSS-счетчиков	221
Раздел 36. Функции	223
Глава 36.1. Функция <code>calc()</code>	223
Глава 36.2. Функция <code>attr()</code>	223
Глава 36.3. Функция <code>var()</code>	224
Глава 36.4. Функция <code>radial-gradient()</code>	224
Глава 36.5. Функция <code>linear-gradient()</code>	224
Раздел 37. Пользовательские свойства (переменные)	225
Глава 37.1. Переменные для цвета	225

Глава 37.2. Переменные для размеров.....	225
Глава 37.3. Каскадирование переменных	226
Глава 37.4. Допустимые и недопустимые имена	226
Глава 37.5. Использование с медиазапросами	228
Раздел 38. Фигуры из одного элемента	229
Глава 38.1. Трапеция.....	229
Глава 38.2. Треугольники.....	229
Глава 38.3. Круги и эллипсы	232
Глава 38.4. «Взрывы».....	233
Глава 38.5. Квадрат	235
Глава 38.6. Куб	235
Глава 38.7. Пирамида.....	236
Раздел 39. Колонки	238
Глава 39.1. Простой пример (column-count)	238
Глава 39.2. Ширина колонки	239
Раздел 40. Многоколоночная верстка	240
Глава 40.1. Создание нескольких колонок	240
Глава 40.2. Базовый пример.....	240
Раздел 41. Верстка с использованием Inline-Block	242
Глава 41.1. Выровненная по ширине навигационная панель	242
Раздел 42. Наследование	244
Глава 42.1. Автоматическое наследование	244
Глава 42.2. Принудительное наследование.....	244
Раздел 43. CSS-спрайты для изображений	246
Глава 43.1. Базовая реализация.....	246
Раздел 44. Обрезка и маскирование	248
Глава 44.1. Обрезка и маскирование: обзор и различия.....	249
Глава 44.2. Простая маска, которая делает изображение плавно прозрачным.....	250
Глава 44.3. Обрезка (круг).....	251
Глава 44.4. Обрезка (многоугольник).....	252
Глава 44.5. Использование масок для вырезания отверстия в центре изображения.....	253
Глава 44.6. Использование масок для создания изображений с неправильными формами	253
Раздел 45. Фрагментация	255
Глава 45.1. Разрыв страницы для печати (Media print page-break)	255
Раздел 46. CSS Object Model (CSSOM)	256
Глава 46.1. Добавление правила background-image через CSSOM	256
Глава 46.2. Введение	256
Раздел 47. Feature Queries	257
Глава 47.1. Основное использование @supports	257
Глава 47.2. Цепочка проверок возможностей.....	257
Раздел 48. Контекст наложения (Stacking Context)	259
Глава 48.1. Контекст наложения	259
Раздел 49. Контексты форматирования блоков	263
Глава 49.1. Использование свойства overflow со значением, отличным от visible	263
Раздел 50. Вертикальное центрирование	265
Глава 50.1. Центрирование с помощью display: table	265
Глава 50.2. Центрирование с помощью Flexbox.....	265
Глава 50.3. Центрирование с помощью Transform	266
Глава 50.4. Центрирование текста с помощью line-height	266
Глава 50.5. Центрирование с помощью position: absolute	267
Глава 50.6. Центрирование с помощью псевдоэлемента	268
Раздел 51. Масштабирование и размещение объектов	269
Глава 51.1. object-fit.....	269
Раздел 52. Шаблоны проектирования CSS	271
Глава 52.1. BEM.....	271
Раздел 53. Поддержка браузеров и префиксы	273
Глава 53.1. Переходы (Transitions)	273
Глава 53.2. Трансформации (Transform).....	273
Раздел 54. Нормализация стилей браузера	274
Глава 54.1. normalize.css.....	274
Глава 54.2. Подходы и примеры	275
Раздел 55. Хаки для Internet Explorer	277
Глава 55.1. Добавление поддержки Inline Block для IE6 и IE7	277
Глава 55.2. Режим высокой контрастности в Internet Explorer 10 и выше.....	277
Глава 55.3. Только для Internet Explorer 6 и Internet Explorer 7.....	278
Глава 55.4. Только для Internet Explorer 8.....	278
Раздел 56. Производительность	279
Глава 56.1. Используйте transform и opacity, чтобы избежать срабатывания пересчета макета (layout).....	279
Благодарности	281

Раздел 1. Начало работы с CSS

Версия	Дата выпуска
1	17-12-1996
2	12-05-1998
3	13-10-2015

Глава 1.1. Внешняя таблица стилей

Внешняя таблица стилей CSS может быть применена к любому количеству HTML-документов путем добавления элемента `<link>` в каждый HTML-документ.

Атрибут `rel` тега `<link>` должен быть установлен в значение `stylesheet`, а атрибут `href` — в относительный или абсолютный путь к таблице стилей. Хотя использование относительных путей считается хорошей практикой, абсолютные пути также могут быть задействованы. В HTML5 атрибут `type` может быть опущен.

Рекомендуется размещать тег `<link>` внутри тега `<head>` HTML-документа, чтобы стили загружались до того, как будут применены к элементам. В противном случае пользователи могут увидеть кратковременное отображение нестилизованного контента.

Пример

hello-world.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <h1>Hello world!</h1>
    <p>I ♥ CSS</p>
```

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

style.css

```
h1 {
    color: green;
    text-decoration: underline;
}
p {
    font-size: 25px;
    font-family: 'Trebuchet MS', sans-serif;
}
```

Убедитесь, что вы указали правильный путь к вашему CSS-файлу в атрибуте href. Если CSS-файл находится в той же папке, что и HTML-файл, то путь указывать не требуется (как в примере выше). Однако если файл сохранен в папке, укажите путь следующим образом:

```
<link rel="stylesheet" type="text/css" href="foldername/style.css">
```

Внешние таблицы стилей считаются лучшим способом управления CSS. Причина проста: если вы управляете сайтом из, скажем, 100 страниц, контролируемых одной таблицей стилей, и хотите изменить цвет ссылок с синего на зеленый, гораздо проще внести это изменение в CSS-файл и позволить изменениям «каскадироваться» на все 100 страниц, чем вручную вносить изменения в каждую из 100 страниц. Кроме того, если вы захотите полностью изменить внешний вид вашего сайта, вам нужно будет обновить только этот один файл.

Вы можете загружать любое количество CSS-файлов на вашу HTML-страницу:

```
<link rel="stylesheet" type="text/css" href="main.css">
<link rel="stylesheet" type="text/css" href="override.css">
```

Правила CSS применяются с учетом определенных базовых правил, и порядок имеет значение. Например, если у вас есть файл main.css с таким кодом:

```
p.green { color: #00FF00; }
```

...все ваши параграфы с классом green будут отображаться светло-зеленым цветом. Однако вы можете переопределить это, подключив другой CSS-файл после main.css. Например, в файле override.css можно указать:

```
p.green { color: #006600; }
```

Теперь все параграфы с классом `green` будут отображаться темно-зеленым цветом вместо светло-зеленого.

Также применяются другие принципы, такие как правило `!important`, специфичность и наследование.

Когда кто-то впервые посещает ваш сайт, его браузер загружает HTML текущей страницы и связанный CSS-файл. При переходе на другую страницу браузеру нужно загрузить только HTML этой страницы; CSS-файл кешируется, поэтому его не нужно загружать снова. Поскольку браузеры кешируют внешние таблицы стилей, ваши страницы загружаются быстрее.

Глава 1.2. Внутренние стили

CSS, заключенный в теги `<style></style>` внутри HTML-документа, функционирует аналогично внешней таблице стилей, за исключением того, что он находится в самом HTML-документе, который стилизует, и поэтому может быть применен только к этому документу. Обратите внимание, что этот элемент должен находиться внутри `<head>` для валидации HTML (хотя он будет работать во всех современных браузерах, даже если размещен в `<body>`):

```
<head>
  <style>
    h1 {
      color: green;
      text-decoration: underline;
    }
    p {
      font-size: 25px;
      font-family: 'Trebuchet MS', sans-serif;
    }
  </style>
</head>
<body>
  <h1>Hello world!</h1>
  <p>I ♥ CSS</p>
</body>
```

Глава 1.3. Правило CSS `@import` (одно из правил CSS `at-rule`)

Правило CSS `@import` используется для импорта стилей из других таблиц стилей. Эти правила должны предшествовать всем другим типам правил, за исключением правил `@charset`; так как оно не является вложенным

оператором, `@import` не может использоваться внутри условных групповых правил.

Почитать про `@import` можно, перейдя по ссылке <https://developer.mozilla.org/en-US/docs/Web/CSS/@import>.



Как использовать `@import`

Вы можете использовать правило `@import` следующими способами:

А. С внутренним тегом стиля

```
<style>
  @import url('/css/styles.css');
</style>
```

В. С внешней таблицей стилей

Следующая строка импортирует CSS-файл с именем `additional-styles.css`, находящийся в корневом каталоге, в CSS-файл, в котором она указана:

```
@import '/additional-styles.css';
```

Импорт внешнего CSS также возможен. Частым примером использования являются файлы шрифтов:

```
@import 'https://fonts.googleapis.com/css?family=Lato';
```

Необязательный второй аргумент правила `@import` — это список медиа-запросов:

```
@import '/print-styles.css' print;
@import url('landscape.css') screen and (orientation:landscape);
```

Глава 1.4. Встроенные стили (Inline Styles)

Используйте встроенные стили для применения стилей к конкретному элементу. Обратите внимание, что это не самый оптимальный способ. Размещение правил стилей в теге `<style>` или во внешнем CSS-файле рекомендуется для поддержания разделения между содержимым и представлением.

Встроенные стили переопределяют любые CSS-правила, заданные в теге `<style>` или во внешней таблице стилей. Хотя это может быть полезно в некоторых случаях, чаще всего это снижает удобство поддержки проекта.

Стили в следующем примере применяются непосредственно к элементам, к которым они прикреплены:

```
<h1 style="color: green; text-decoration: underline;">Hello world!</h1>
<p style="font-size: 25px; font-family: 'Trebuchet MS' ;">I ♥ CSS</p>
```

Встроенные стили, как правило, являются самым безопасным способом обеспечения совместимости отображения в различных почтовых клиентах, программах и устройствах, но их написание может быть трудоемким, а управление ими — несколько сложным.

Глава 1.5. Изменение CSS с помощью JavaScript

Чистый JavaScript

Можно добавлять, удалять или изменять значения CSS-свойств с помощью JavaScript через свойство `style` элемента:

```
var el = document.getElementById("element");
el.style.opacity = 0.5;
el.style.fontFamily = 'sans-serif';
```

Обратите внимание, что свойства стилей именуются в нижнем регистре (lower camel case). В примере видно, что CSS-свойство `font-family` становится `fontFamily` в JavaScript.

В качестве альтернативы работе непосредственно с элементами вы можете создать элемент `<style>` или `<link>` в JavaScript и добавить его в `<body>` или `<head>` HTML-документа.

jQuery

Изменение CSS-свойств с помощью jQuery еще проще:

```
$('#element').css('margin', '5px');
```

Если нужно изменить более одного правила стиля:

```
$('#element').css({
  margin: "5px",
  padding: "10px",
  color: "black"
});
```

jQuery включает два способа изменения CSS-правил, содержащих дефисы (например `font-size`). Вы можете заключить их в кавычки или использовать `camel-case` для имени правила:

```
$('.example-class').css({
  "background-color": "blue",
  fontSize: "10px"
});
```

Глава 1.6. Стилизация списков с помощью CSS

Существует три различных свойства для стилизации элементов списка: `list-style-type`, `list-style-image` и `list-style-position`, которые должны объявляться в таком порядке. Значения по умолчанию: `disc`, `outside` и `none` соответственно. Каждое свойство может быть объявлено отдельно или с использованием сокращенного свойства `list-style`.

`list-style-type` определяет форму или тип маркера, используемого для каждого элемента списка.

Некоторые допустимые значения для `list-style-type`:

- `disc`
- `circle`
- `square`
- `decimal`
- `lower-roman`
- `upper-roman`
- `none`



Полный список см. в спецификации W3C (<https://developer.mozilla.org/en-US/docs/Web/CSS/list-style-type>).

Чтобы использовать квадратные маркеры для каждого элемента списка, например, вы бы использовали следующее свойство-значение:

```
li {  
    list-style-type: square;  
}
```

Свойство `list-style-image` определяет, задается ли значок элемента списка с помощью изображения, и принимает значение `none` или URL, указывающего на изображение:

```
li {  
    list-style-image: url(images/bullet.png);  
}
```

Свойство `list-style-position` определяет, где позиционировать маркер элемента списка, и принимает одно из двух значений — `inside` или `outside`:

```
li {  
    list-style-position: inside;  
}
```

Раздел 2. Структура и форматирование CSS-правила

Глава 2.1. Списки свойств

Некоторые свойства могут принимать несколько значений, которые вместе называются **списком свойств** (property list):

```
/* Два значения в этом списке свойств */
span {
    text-shadow: yellow 0 0 3px, green 4px 4px 10px;
}
/* Альтернативное форматирование */
span {
    text-shadow:
        yellow 0 0 3px,
        green 4px 4px 10px;
}
```

Глава 2.2. Множественные селекторы

Когда вы группируете CSS-селекторы, вы применяете одинаковые стили к нескольким различным элементам без повторения стилей в вашей таблице стилей. Используйте запятую для разделения нескольких сгруппированных селекторов:

```
div, p { color: blue; }
```

Таким образом, синий цвет применяется ко всем элементам `<div>` и всем элементам `<p>`. Без запятой только дочерний элемент `<div>` был бы красным.

Это также применимо ко всем типам селекторов:

```
p, .blue, #first, div span { color: blue; }
```

Это правило применяется к:

- Элементам `<p>`
- Элементам с классом `blue`

- Элементу с идентификатором `first`
- Каждому элементу `` внутри `<div>`

Глава 2.3. Правила, селекторы и блоки объявлений

CSS-правило состоит из селектора (например `h1`) и блока объявлений (`{}`):

```
h1 {}
```

Раздел 3. Комментарии

Глава 3.1. Однострочные комментарии

```
/* Это однострочный комментарий в CSS */  
div {  
    color: red; /* Это однострочный комментарий в CSS */  
}
```

Глава 3.2. Многострочные комментарии

```
/*  
    Это  
    многострочный  
    комментарий  
    в CSS  
*/  
div {  
    color: red;  
}
```