**II Введение в язык стилей CSS**

**4 Работа с каскадными таблицами стилей**

CSS – это формальный язык, служащий для описания оформления внешнего вида документа, созданного с использованием языка разметки (HTML, XHTML, XML). Название происходит от английского Cascading Style Sheets, что означает “каскадные таблицы стилей”.

Любой HTML-документ без использования стилей будет выглядеть лаконично. Каскадные таблицы стилей (CSS) определяют представление документа и его внешний вид. Назначение CSS – отделять то, что задает внешний вид страницы, от ее содержания. Если документ создан только с использованием HTML, то в нем определяется не только каждый элемент, но и способ его отображения (цвет, шрифт, положение блока и т. д.). Если же подключены каскадные таблицы стилей, то HTML описывает только очередность объектов. А за все их свойства отвечает CSS [5].

Стиль в CSS - это правило, которое сообщает веб-браузеру, как форматировать элемент. Форматирование может включать установку цвета фона элемента, установку цвета и типа шрифта и т. д. Определение стиля состоит из двух частей: селектора, указывающего на элемент, и блока объявления стиля, т.е. набора команд, устанавливающих правила форматирования. Например:

**div {background-color:blue; width: 100px;  height: 60px;}**

В данном случае селектором является div. Этот селектор указывает, что этот стиль будет применен ко всем элементам div. После селектора находится блок объявления стиля, заключенный в фигурные скобки. Команды определяются между открывающими и закрывающими скобками. Каждая команда состоит из свойства и значения. Таким образом, в следующем выражении:

**background-color:blue;**

background-color представляет свойство, а blue - значение. Свойство определяет определенный стиль. Существует множество свойств CSS. Например, background-color определяет цвет фона. После двоеточия указывается значение этого свойства. Например, приведенная выше команда устанавливает для свойства background-color значение красного цвета. Другими словами, цвет фона элемента устанавливается “blue”, то есть синий.

После каждой команды ставится точка с запятой, отделяющая данную команду от остальных. Наборы таких стилей часто называют таблицами стилей или CSS (Cascading Style Sheets или каскадные таблицы стилей). Существуют различные способы определения стилей.

**4.1 Виды каскадных таблиц стилей**

**4.1.1 Внедренные таблицы стилей. Атрибут style**

Первый способ заключается во встраивании стилей непосредственно в элемент с помощью атрибута style.

**<body>**

**<h2 style="color: purple;"> Использование стилей </h2>**

**<div style="width: 200px; height: 200px; background-color: blue;">**

**</div>**

**</body>**

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.1 - Inline-стили элементов HTML5

Здесь определены два элемента: заголовок h2 и div. Заголовок имеет синий цвет текста, определенный с помощью свойства color. Блок div имеет свойства, определенные для ширины, высоты и цвета фона.

**4.1.2 Внутренние таблицы стилей. Тег style**

Второй способ - использовать элемент стиля в html-документе. Этот элемент сообщает браузеру, что данные внутри представляют собой код CSS, а не HTML:

**<head>  <title>Использование стилей</title>**

**<style>    h2{color: purple; }**

**div{width: 200px; height: 200px;   background-color: blue; }**

**</style>    </head>**

**<body>  <h2>Использование стилей</h2>**

**<div></div></body>**

Результат в будет абсолютно такой же, как в предыдущем примере.

**4.1.3 Внешние таблицы стилей. Файл \*.css**

Третий способ - перенесение стилей во внешний файл. Создается текстовый документ, переименовывается в style.css и определяется следующее содержимое:

**h2 {color: purple;}**

**div {width: 200px;   height: 200px; background-color: blue;}**

Это те же стили, что были внутри элемента style. Исходный код html-страницы будет выглядеть следующим образом:

**<head> <title>Использование стилей</title>**

**<link rel="stylesheet" type="text/css" href="styles.css"/>**

**</head>  <body> <h2>Использование стилей</h2> <div></div> </body>**

Здесь больше нет элемента style, но есть элемент ссылки link, который связывает созданный выше файл style.css:

**<link rel="stylesheet" type="text/css" href="styles.css"/>**

Поэтому, определяя стили во внешнем файле, HTML-код становится чище, структура страницы отделяется от ее стилей. Стили, определенные таким образом, гораздо легче изменить, если бы они были определены внутри элементов или в элементе стиля. Возможно объединить все эти подходы, и для одного элемента некоторые свойства CSS определяются внутри самого элемента, другие свойства CSS определяются в элементе стиля, а другие находятся во внешнем файле.

Например:

**<head>**

**<link rel="stylesheet" type="text/css" href="styles.css"/>**

**<style>**

**div {width:100px;}**

**</style>  </head>**

**<body> <div style="width:200px;"></div></body>**

А в файле styles.css определены стили:

**div{ width:10px; height:300px;  background-color:blue;}**

В этом случае элемент div имеет свойство ширины, определенное в трех местах с разными значениями. К элементу будет применена следующая система приоритетов: если у элемента определены встроенные стили (inline-стили), то они имеют наивысший приоритет, т.е. в примере выше конечная ширина будет равна 200 пикселей. Следующими по приоритету являются стили, определенные в элементе style. Стили с самым низким приоритетом определены во внешнем файле.

**4.2 Задание стилей с помощью селекторов**

**4.2.1 Селекторы типа**

Определение стиля начинается с селектора. В CSS3 определено три вида селекторов – селектор типа, селектор класса и ID-селектор. Селектор типа определяет HTML-элемент, к которому будет применен стиль. Селекторы типа легко определить в стилях, так как они имеют одноимённые наименования с форматируемыми элементами, например: p - сообщает браузеру, что необходимо отформатировать все HTML теги <p> (параграф).

**P {font-family: 'Times New Roman', color: green;}**

Задаются селекторы типа в заголовке html документа, в то время как переопределение стиля происходит в теле документа.

**<head> <title>Использование селекторов типа</title>**

**<style> h2 { font-family: 'Courier New', Courier, monospace; } </style>**

**</head> <body>**

**<h1>Селекторы в CSS (без стилей)</h1>**

**<h2>Селекторы в CSS (с использованием стилей)</h2>**

**</body>**

Здесь на странице показаны заголовки, которые имеют стили и нет:

Изображение выглядит как текст, снимок экрана, Шрифт, электроника

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.2 – Селекторы типа CSS

**4.2.2 Селекторы класса**

Иногда одни и те же элементы требуют разного оформления. В этом случае можно использовать классы. Чтобы определить селектор класса в CSS, следует поставить точку перед именем класса:

**.greenBlock{background-color:green;}**

Имя класса может быть любым. Например, в данном случае имя класса – “greenBlock”. Однако в имени класса могут использоваться буквы, цифры, дефисы и подчеркивания, и имя класса должно начинаться с буквы. Также стоит учитывать регистр зависимость имен: имена «article» и «ARTICLE» представляют разные классы.

После определения класса его можно применить к элементу с помощью атрибута class. Например:

**<div class="greenBlock"></div>**

Далее приведен пример использования селекторов класса.

**<head> <style>**

**.blocks {width: 100vw; display: flex; justify-content: start;}**

**div {width: 250px;margin: 10px; padding: 10px;}**

**span {font-size: 20px;}**

**.greenBlock {background-color: green; color: white;}**

**.yellowBlock {background-color: yellow;}**

**.blueBlock {background-color: blue; color: white;}**

**</style> </head><body> <h2>Использование классов</h2>**

**<div class="blocks"><div class="greenBlock"> <h2>Зеленый цвет</h2> <span> <b>Значение.</b> Зелёный цвет способен гасить эмоции, успокаивать, создавать ощущение умиротворения и безопасности и потому воспринимается как один из самых комфортных.</span>**

**</div><div class="yellowBlock"> <h2>Жёлтый цвет</h2> <span> <b>Значение.</b> Это один из самых тёплых цветов видимого спектра. В первую очередь жёлтый ассоциируется с солнцем и в основном вызывает положительные эмоции. </span> </div>**

**<div class="blueBlock"> <h2>Синий цвет</h2> <span> <b>Значение.</b> Синий также воспринимается как успокаивающий, снижающий тревогу. Он создаёт приятные ассоциации с морем и небом. </span> </div> </div> </body>**

Реализация данного кода представлена на рисунке 4.3.

Изображение выглядит как текст, электроника, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.3 – Селекторы классов в CSS3

Другой пример использования селектора класса. Допустим, необходимо создать Web-страницу, на которой будет два вида абзацев <p>, причем оба вида будут постоянно чередоваться и часто повторяться.

Типичный пример такой страницы - интервью, в котором чередуются вопросы журналиста и ответы интервьюируемого. При создании такой страницы необходимо визуально отделить вопросы и ответы друг от друга. Это будет выглядеть следующим образом:

**<head> <style> body { background-color: rgba(247, 255, 171, 0.449); }**

**.ask { font-family: Arial, sans-serif;**

**font-style: italic; font-weight: bold;**

**font-size: 14px; margin-left: 15px; }**

**.answer, ul {font-family: Courier, serif;**

**font-size: 14px; margin-left: 10px;}**

**</style> </head> <body>**

**<h3>Экзамен по дисциплине “Основы веб-разработки”</h3>**

**<p class="ask">1. Что такое CSS?</p>**

**<p class="answer"> CSS - это язык стилизации, который делает веб-страницы HTML более презентабельными. Он позволяет, среди прочего, добавлять на веб-сайт цвет,**

**дизайн и кнопки.</p>**

**<p class="ask">2. Каковы преимущества использования CSS?</p>**

**<p class="answer"> Использование CSS имеет множество преимуществ, в том числе:**

**<ul> <li>Специальные возможности</li> <li>Переформатирование страницы</li>**

**<li>Согласованность для всего сайта</li> <li>Пропускная способность</li>**

**<li>Разделение полосы пропускания презентации</li>**

**</ul> <p class="ask">3. Каковы недостатки CSS?</p>**

**<p class="answer">Недостатками использования CSS являются:**

**<ul> <li>Тексты, оформление и правила, ориентированные на конкретную аудито-рию, не допускаются</li>**

**<li>Нет выражений</li>**

**<li>Отсутствует динамическое поведение для управления псевдоклассом</li>**

**<li>Невозможно подняться по селекторам</li>**

**<li>Вертикальный контроль ограничен</li>**

**<li>Объявление столбца отсутствует</li> </ul> </body>**

Реализация данного кода представлена на рисунке 4.4.

Изображение выглядит как текст, электроника, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.4 - Использование селектора класса

**4.2.3 Id-селектор**

Для идентификации уникальных на веб-станице элементов используются идентификаторы, которые определяются с помощью атрибутов id. Определение стилей для идентификаторов аналогично определению классов, только вместо точки ставится символ решетки #.

Например,

**<head> <style>**

**#mystyle {text-color: aqua;}**

**</style></head>**

Вызов стилей реализуется в теле документа в теге <body>.

**<h1 id="mystyle"></div>**

Далее представлен пример использования id-селекторов при задании стилей для тега <div>.

**<head> <style> div{ margin: 10px;**

**border: 2px solid #000000;}**

**#header{ height: 65px; background-color: #ffE87C;}**

**#content{ height: 95px; background-color: gold;}**

**#footer{ height: 25px; background-color: #FFF380;}**

**</style></head>**

**<body> <div id="header"> <h2>Золотая рыбка</h2> </div>**

**<div id="content">** **Золотая рыбка (Carassius auratus) это по праву самый узнаваемый вид декоративных рыб, который занимает почетное первое место в списке наиболее популярных аквариумных жителей. Свое название золотая рыбка получила, прежде всего, за цвет чешуи: стандартный цвет тела – красновато-золотистый, зачастую с металлическим отливом.**  **</div>**

**<div id="footer">Copyright &copy; Golgfish 2024 </div>**

**</body></html>**

Реализация данного кода представлена на рисунке 4.5.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.5 – Иcпользование id-селекторов в CSS3

Однако стоит заметить, что идентификаторы в большей степени относятся к структуре веб-странице и в меньшей степени к стилизации. Для стилизации преимущественно используются классы, нежели идентификаторы.

**4.2.4 Универсальный селектор**

Помимо селекторов тегов, классов и идентификаторов, CSS также имеет универсальный селектор, который представляет собой знак звездочки (\*). Он применяет стили ко всем элементам HTML-страницы:

**\*{ background-color: #00FA9A;}**

Далее представлен пример использования универсальных селекторов.

**<head> <style>**

**\* { background-color: #FF69B4; }**

**div { margin: 15px;border: 2px solid #000000; }**

**</style> </head> <body>**

**<div>Оренбургский государственный университет - ОГУ</div>**

**<div>История Оренбургского государственного университета началась 14 сентября 1955 года, когда в городе Чкалове (ныне Оренбурге) было открыто вечернее отделение Куйбышевского индустриального института им. В. В. Куйбышева</div>**

**<div> Copyright &copy; 2024</div> </body>**

Реализация данного кода представлена на рисунке 4.6.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.6 – Использование универсальных селекторов

**4.2.5 Стилизация группы селекторов**

Иногда к набору селекторов применяются определенные стили. Например, необходимо применить подчеркивание ко всем заголовкам. В этом случае можно перечислить селекторы всех элементов через запятую [1, 16]:

**<head> <style> h1, h2, h3 {color: crimson;} </style> </head>**

**<body> <h1>Актеры<h1> <h2>Никита Кологривый</h2> <h3>Биография</h3>**

**<p>Никита Сергеевич Кологривый - российский актёр театра и кино.**

**Родился 16 октября 1994 года в Новосибирске. В юношестве занимался боксом и планировал поступать в педагогический университет на тренера, но поступил в Новосибирский театральный институт. После окончания второго курса отправился в Москву, где поступил в ГИТИС на курс театральных режиссёров Сергея Голомазова и Павла Хомского. С отличием окончил вуз в 2018 году.</p> </body>**

Реализация данного кода представлена на рисунке 4.7.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.7 - Группа селекторов в CSS3

Группа селекторов может содержать как селекторы тегов, так и селекторы классов и идентификаторов, например:

**h1, #header, .blueBlock {color: blue;}**

**4.3 Селекторы потомков**

Веб-страница может иметь сложную организацию, одни элементы внутри себя могут определять другие элементы. Вложенные элементы иначе можно назвать потомками. А контейнер этих элементов - родителем.

Например, пусть элемент body на веб-странице имеет следующее содержимое:

**<body> <h2>Название</h2>**

**<div> <p>Контент</p> </div> </body>**

Внутри элемента body определено три вложенных элемента: h2, div, p. Все эти элементы являются потомками элемента body. А внутри элемента div определен только один вложенный элемент - p, поэтому элемент div имеет только одного потомка. Используя специальные селекторы, можно стилизовать вложенные элементы или потомков внутри строго определенных элементов.

Например, на странице могут быть параграфы внутри блока с основным содержимым и внутри блока футера. Но для параграфов внутри блока основного содержимого можно установить один шрифт, а для параграфов футера другой.

**<head> <style>**

**#main p { font-size: 17px; background-color: mediumspringgreen;}**

**#footer p { background-color: mediumseagreen; color: white;}**

**</style> </head> <body> <div id="main"><p>Интересные факты о цветах</p>**

**<p> По факту, если мы видим бутон белого цвета, значит, его лепестки совершенно бесцветные и свободно отражают белый солнечный свет. Своего оттенка у них нет. Цветов чистого черного цвета тоже не бывает: скорее всего, они темно-лиловые, фиолетовые, темно-бордовые. Больше у цветов нет никаких ограничений по оттенкам: они могут быть абсолютно любыми, кроме черного и белого. </p> </div>**

**<div id="footer"><p> Интересно, что пчелы никогда не ошибаются в выборе цветка: садятся только на те, у которых есть нектар. Энтомологи выяснили: это происходит потому, что на лепестках цветов есть невидимые линии и пятна, видимые только в ультрафиолете. По этим светящимся узорам насекомые ориентируются по цветам. </p> </div> </body>**

Реализация данного кода представлена на рисунке 4.8.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.8 - Селекторы вложенных элементов в CSS3

Далее представлен еще один пример работы с дочерними селекторами.

**<head> <style> li .blueLink {color: blue;}</style></head>**

**<body> <h2>Жанры книг</h2> <ul>**

**<li>Жанр книги “Призрак Оперы”: <a class="blueLink">Мистика</a></li>**

**<li>Жанр книги “Безмолвный пациент”: <a>Ужасы</a></li>**

**<li>Жанр книги “Судьба человека”: <a>Драма</a></li>**

**<li>Жанр книги “Укрощение строптивой”: <a>Комедия</a></li>**

**</ul></body>**

Реализация данного кода представлена на рисунке 4.9.

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.9 - Стилизация дочерних элементов в CSS3

Здесь стиль применяется к элементам с классом "blueLink", которые находятся внутри элемента <li>. И соответственно браузер окрасит эти элементы в синий цвет:

Пробел: li .blueLink играет большое значение и указывает как раз, что элементы с классом blueLink должны быть вложенными по отношению к элементу <li>.

Но если убрать пробел:

**li.blueLink {color: blue;}**

то смысл селектора изменится. Теперь будет подразумеваться, что стиль применяется к элементам <li>, которые имеют класс blueLink. Например, к следующему элементу:

**<li class="blueLink"> Жанр книги “Безмолвный пациент”: <a> Ужасы </a></li>**

Но никак не к элементу:

**<li>Жанр книги “Призрак Оперы”: <a class="blueLink">Мистика</a></li>**

**4.4 Селекторы дочерних элементов**

Селекторы дочерних элементов отличаются от селекторов потомков тем, что позволяют выбрать элементы только первого уровня вложенности. Например:

**<body>**

**<h2>Название</h2><div> <p>Контент</p> </div>**

**</body>**

Хотя вложенными в элемент body элементами являются целых три - h2, div, p, но дочерними из них являются только два - div и h2, так как они находятся в первом уровне вложенности. А элемент p находится на втором уровне вложенности, так как вложен внутрь элемента div, а не просто элемента body.

Для обращения к дочерним элементам используется знак угловой скобки. Далее представлен пример работы с дочерними элементами.

**<head> <style> .article > p {font-weight: bold; color: steelblue; boder: 2 px;} </style>**

**</head> <body> <div class="article">**

**<p>Хасеки Хюрем Султан</p>**

**<div class="content">**

**<p> Хюрем -**[**наложница**](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%BB%D0%BE%D0%B6%D0%BD%D0%B8%D1%86%D0%B0)**, а затем супруга**[**османского султана**](https://ru.wikipedia.org/wiki/%D0%9E%D1%81%D0%BC%D0%B0%D0%BD%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%BC%D0%BF%D0%B5%D1%80%D0%B8%D1%8F)[**Сулеймана I Великолепного**](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%BB%D0%B5%D0%B9%D0%BC%D0%B0%D0%BD_I)**, мать султана**[**Селима II**](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BB%D0%B8%D0%BC_II)**. Являлась одной из самых влиятельных женщин в истории**[**Османской империи**](https://ru.wikipedia.org/wiki/%D0%9E%D1%81%D0%BC%D0%B0%D0%BD%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%BC%D0%BF%D0%B5%D1%80%D0%B8%D1%8F)**. </p> <p> Во время одного из**[**набегов крымских татар**](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D1%8B%D0%BC%D1%81%D0%BA%D0%BE-%D0%BD%D0%BE%D0%B3%D0%B0%D0%B9%D1%81%D0%BA%D0%B8%D0%B5_%D0%BD%D0%B0%D0%B1%D0%B5%D0%B3%D0%B8_%D0%BD%D0%B0_%D0%A0%D1%83%D1%81%D1%8C)**девушка попала в плен и после нескольких перепродаж, была подарена**[**Сулейману**](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%BB%D0%B5%D0%B9%D0%BC%D0%B0%D0%BD_I) **по случаю восшествия на престол**[**.**](https://ru.wikipedia.org/wiki/%D0%A0%D0%BE%D0%BA%D1%81%D0%BE%D0%BB%D0%B0%D0%BD%D0%B0#cite_note-autogenerated2-26)**</p> </div> </div> </body>**

Реализация данного кода представлена на рисунке 4.10.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

AI-generated content may be incorrect.

Рисунок 4.10 - Селекторы дочерних элементов в CSS3

В блоке с классом article есть два параграфа. Селектор .article > p выбирает только те параграфы, который находятся непосредственно в блоке article.

Если бы использовался другой селектор без символа >

**.article p {font-weight: bold; color: steelblue; border: 2 px;}**

Тогда стиль бы применялся ко всем параграфам на всех уровнях вложенности (рисунок 4.11).

Изображение выглядит как текст, электроника, снимок экрана, мультимедиа

AI-generated content may be incorrect.

Рисунок 4.11 - Селекторы вложенных элементов в CSS

**4.5 Селекторы элементов одного уровня**

Селекторы элементов одного уровня или смежных элементов позволяют выбрать элементы, которые находятся на одном уровне вложенности. Иногда такие элементы еще называют сиблинги (siblings) или сестринскими элементами.

**<body>**

**<h2>Заголовок</h2>**

**<div> <p>Текст первого блока</p></div>**

**<div> <p>Текст второго блока</p></div>**

**</body>**

Здесь элементы h2 и оба блока div являются смежными, так как находятся на одном уровне. А элементы параграфов и заголовок h2 не являются смежными, так как параграфы вложены в блоки div.

Чтобы стилизовать первый смежный элемент по отношению к определенному элементу, используется знак плюса +.

Далее представлен пример работы с селекторами одного уровня.

**<head><style> h2+div {color: LimeGreen; font-style: italic;} </style> </head>**

**<body> <h2>Гарик Харламов</h2>**

**<div> <p> Родился 28 февраля 1980 года в Москве. Отец Гарика после развода переехал в Чикаго. Сын по окончании школы отправился за ним и получал образование в знаменитой актерской школе. Через 5 лет молодой человек вернулся в Москву. Окончил Государственный университет управления по специальности “Управление персоналом”. </p></div>**

**<div> <p> 7 лет играл в КВН за команды “Сборная Москвы” и “Незолотая молодёжь”. Участник юмористического шоу “Comedy Club” (ТНТ), выступает в дуэте с Тимуром Батрутдиновым. В отдельных выпусках Гарик выходит на сцену не под настоящим именем, а под псевдонимом**[**Эдуард Суровый**](https://24smi.org/person/9707-eduard-surovyi.html)**.</p> </div> </body>**

Реализация данного кода представлен на рисунке 4.12.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

AI-generated content may be incorrect.

Рисунок 4.12 - Селекторы смежных элементов в CSS3

Селектор h2+div позволяет определить стиль (в данном случае красный цвет текста) для блока div, который идет непосредственно после заголовка h2. Причем этот селектор будет стилизовать блок div, если он будет идти непосредственно после заголовка. Если же между заголовком и блоков div будет находиться еще какой-либо элемент, то к нему не будет применяться стиль, например:

**<h2>Заголовок</h2>**

**<p>Элемент между заголовком и блоком div</p>**

**<div><p>Текст первого блока</p></div>**

Если надо стилизовать вообще все смежные элементы одного уровня, неважно непосредственно идут они после определенного элемента или нет, то в этом случае вместо знака плюса необходимо использовать знак тильды “~”.

**<head> <title>Селекторы в CSS3</title> <style>**

**h2 ~ div {color: BlueViolet; font-weight: bold;} </style> </head>**

**<body> <h2>Иван Филиппович Янковский</h2>**

**<div> <p> Иван Филиппович Янковский - российский актёр театра и кино. Пятикратный лауреат премии “Золотой орёл”: дважды за “Лучшую мужскую роль в кино», дважды за “Лучшую мужскую роль второго плана” и один раз за “Лучшего актёра онлайн-сериала”. </p></div>**

**<div> <p> Янковский женат, его спутница - коллега по цеху Диана Пожарская. В 2021 году стало известно, что у них родился сын Олег. До супружества Янковский встречался с дочерью известного бизнесмена Камиллой Байсаровой, а также был замечен в отношениях с дочерью лидера группы “Алиса” Верой Панфиловой. </p></div> </body>**

Реализация данного кода представлено на рисунке 4.13.

Изображение выглядит как текст, снимок экрана, мультимедиа, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.13 - Стилизация элементов одного уровня в CSS

**4.6 Псевдоклассы**

В дополнение к селекторам тегов, классов и идентификаторов доступны селекторы псевдоклассов, которые несут дополнительные возможности по выбору нужных элементов. Список доступных псевдоклассов:

* :root: позволяет выбрать корневой элемент веб-страницы;
* :link: применяется к ссылкам и представляет ссылку в обычном состоянии, по которой еще не совершен переход;
* :visited: применяется к ссылкам и представляет ссылку, по которой совершен переход;
* :active: применяется к ссылкам и представляет ссылку в тот момент, когда пользователь осуществляет по ней переход;
* :hover: представляет элемент, на который пользователь навел указатель мыши;
* :focus: представляет элемент, который получает фокус;
* :not: позволяет исключить элементы из списка элементов, к которым применяется стиль;
* :lang: стилизует элементы на основании значения атрибута lang;
* :empty: выбирает элементы, которые не имеют вложенных элементов, то есть являются пустыми.

При применении псевдоклассов перед ними всегда ставится двоеточие. Далее представлен пример, в котором стилизуются ссылки, используя псевдоклассы.

<**head> <title>Использование превдоклассов</title> <style>**

**a:link {font-size: 20px; color: ForestGreen; font-style: italic; }**

**a:visited {font-size: 12 px; color: FireBrick;}**

**a:hover {font-size: 24px; color: DarkViolet; text-decoration: underline;}**

**a:active {font-size: 18px; color: DodgerBlue; свойство font-weight: bold;}**

**</style></head>**

**<body> <h2>Моя домашняя страница</h2>**

**<h3>Добро пожаловать на мою домашнюю страницу. Здесь я поделюсь самым сокровенным. Расскажу про свои увлечения, питомцев, друзей и многое другое. Переходи по ссылкам, будет интересно. </h3>**

**<a href="https://family.html/">Моя семья</a><br />**

**<a href="https://friends.html/">Мои друзья </a><br />**

**<a href="https://pets.html/">Мои животные</a><br />**

**<a href="https://hobbi.html/">Мое хобби</a><br /> </body>**

Реализация данного кода представлена на рисунке 4.14.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

AI-generated content may be incorrect.

Рисунок 4.14 – Пример стилизации ссылок

Селектор :lang выбирает элементы на основании атрибута lang. Далее приведен пример работы с псевдоклассом lang.

**<head> <style>**

**:lang(ru) { color: blue;}**

**</style> </head> <body> <form>**

**<p lang="ru">Моя любимая актриса Марина Александрова </p>**

**<p lang="en">My favorite actress is Marina Alexandrova</p>**

**<p lang="de">Meine Lieblingsschauspielerin ist Marina Alexandrova</p>**

**</form></body>**

Реализация данного кода представлена на рисунке 4.15.

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

AI-generated content may be incorrect.

Рисунок 4.15 – Использование псевдокласса :lang

**4.7 Псевдоклассы дочерних элементов**

Особую группу псевдоклассов образуют псевдоклассы, которые позволяют выбрать определенные дочерние элементы [17]:

* :first-child: первый дочерний элемент;
* :last-child: последний дочерний элемент;
* :only-child: единственный дочерний элементом в некотором контейнере;
* :only-of-type: единственный элемент определенного типа (тега) в некотором контейнере;
* :nth-child(n): дочерний элемент, который имеет определенный номер n, например, второй дочерний элемент;
* :nth-last-child(n): дочерний элемент, который имеет определенный номер n, начиная с конца;
* :nth-of-type(n): дочерний элемент определенного типа, который имеет определенный номер;
* :nth-last-of-type(n): дочерний элемент определенного типа, который имеет определенный номер, начиная с конца.

**4.7.1 Псевдокласс first-child**

Далее приведен пример использования псевдокласса first-child для выбора первых ссылок в блоках.

**<head> <style> a:first-child {font-weight:bold; color: BlueViolet;}**

**</style> </head>**

**<body> <h3>Воздушный транспорт</h3>**

**<div> <a>самолёт</a><br/>**

**<a>геликоптер (вертолёт)</a><br/>**

**<a>планер</a><br/>**

**<a>аэростат и дирижабль</a><br/> </div>**

**<h3>Наземный транспорт</h3>**

**<div> <p>городской транспорт:</p>**

**<a>автомобиль</a><br/>**

**<a>трамвай</a><br/>**

**<a>автобус</a><br/>**

**<a>троллейбус</a> </div> </body>**

Реализация данного кода представлена на рисунке 4.16.

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.16 - Селектор first-child в CSS 3

Стиль по селектору a:first-child применяется к ссылке, если она является первым дочерним элементом любого элемента. В первом блоке элемент ссылки является первым дочерним элементом, поэтому к нему применяется заданный стиль. Во втором блоке первым элементом является параграф <p>, к нему не применяется стиль.

**4.7.2 Псевдокласс last-child**

Далее представлено использование псевдокласса last-child.

**<head> <style>**

**a:last-child{font-weight:bold; color:seagreen;} </style> </head>**

**<body> <h3>Воздушный транспорт</h3>**

**<div> <a>самолёт</a><br/>**

**<a>геликоптер (вертолёт)</a>**

**<br/> <a>планер</a><br/>**

**<a>аэростат и дирижабль</a> </div>**

**<h3>Наземный транспорт</h3>**

**<div> <a>автомобиль</a><br/>**

**<a>трамвай</a><br/>**

**<a>автобус</a><br/>**

**<a>троллейбус</a>**

**<p> - городской транспорт</p> </div> </body>**

Реализация данного кода представлена на рисунок 4.17.

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.17 - Селектор last-child в CSS 3

Селектор a:last-child определяет стиль для ссылок, которые являются последними дочерними элементами.

В первом блоке как раз последним дочерним элементом является ссылка. А вот во втором последним дочерним элементом является параграф, поэтому во втором блоке стиль не применяется ни к одной из ссылок.

**4.7.3 Селектор only-child**

Селектор :only-child выбирает элементы, которые являются единственными дочерними элементами в контейнерах.

**<head> <style> p:only-child{color: #8A2BE2; }**

**</style> </head>**

**<body> <h2>Александр Петров</h2>**

**<div><p>Александр Петров, несмотря на молодой возраст, давно вошел в число самых востребованных российских актеров.</p></div>**

**<div><p>Он активно снимается в кино, играет в театре, пишет стихи и пробует свои силы в режиссуре. </p> <p>Александр родился 25 января 1989 г. Родиной артиста является городок Переславль-Залесский в Ярославской области.</p> </div>**

**<div> <p>Особый успех Александру Петрову принес сериал «Полицейский с Рублёвки», который с 2016 года выходил на канале ТНТ.  </p> </div> </body>**

Реализация данного кода представлена на рисунке 4.18.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.18 - Селектор only-child в CSS3

В данном случае первый и последний параграфы с текстами являются единственными дочерними элементами в своих внешних контейнерах, поэтому к ним применяется стиль - фиолетовый цвет шрифта.

**4.8 Псевдоклассы форм**

Ряд псевдоклассов используется для работы с элементами форм [18]:

* :enabled: выбирает элемент, если он доступен для выбора (то есть у него не установлен атрибут disabled);
* :disabled: выбирает элемент, если он не доступен для выбора (то есть у него установлен атрибут disabled);
* :checked: выбирает элемент, если у него установлен атрибут checked;
* :default: выбирает элементы по умолчанию;
* :valid: выбирает элемент, если его значение проходит валидацию HTML5;
* :invalid: выбирает элемент, если его значение не проходит валидацию;
* :in-range: выбирает элемент, если его значение находится в определенном диапазоне (для элементов типа ползунка);
* :out-of-range: выбирает элемент, если его значение не находится в определенном диапазоне;
* :required: выбирает элемент, если у него установлен атрибут required;
* :optional: выбирает элемент, если у него не установлен атрибут required.

**4.8.1 Псевдоклассы enabled и disabled**

Псевдоклассы enabled и disabled выбирают элементы форм в зависимости от того, установлен ли у них атрибут disabled.

Далее приведен приме использования псевдоклассов enabled и disabled.

**<head>**

**<style>**

**:enabled {border: 2px black solid; color: rgb(11, 184, 103); }**

**</style>**

**</head>**

**<body>**

**<h2>Регистрация пассажиров:</h2>**

**<h3>Введите свои данные</h3>**

**<p> Фамилия <input type="text" value="Обязательное поле для заполнения" /></p>**

**<p> Имя <input type="text" value="Обязательное для заполнения поле" /></p>**

**<p>Отчество<input type="text" disabled value="Необязательное поле" /></p>**

**</body>**

Реализация данного кода представлена на рисунке 4.19.

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.19 – Использование псевдоклассов enabled и disabled в CSS3

**4.8.2 Псевдокласс checked**

Псевдокласс checked стилизует элементы формы, у которых установлен атрибут checked. Далее представлен пример работы с этим псевдоклассом. В данном случае селектор :checked + span позволяет выбрать элемент, соседний с отмеченным элементом формы.

**<head>**

**<style>**

**:checked + span {color: SteelBlue; font-weight: bold; } </style>**

**</head>**

**<body>**

**<h2>Туристическая фирма</h2>**

**<h3>Выберите курорт для отдыха:</h3**

**<p> <input type="checkbox" checked/><span>Турция</span> </p>**

**<p> <input type="checkbox"/><span>Греция</span></p>**

**<p> <input type="checkbox"/><span>Кипр</span>**

**</p> <h3>Выберите класс отеля:</h3>**

**<p> <input type="radio" checked /><span>Пять звезд</span>**

**</p><p><input type="radio" /><span>Четыре звезды</span> </p>**

**</body>**

Реализация данного кода представлена на рисунке 4.20.

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

AI-generated content may be incorrect.

Рисунок 4.20 - Псевдокласс checked в CSS 3

**4.8.3 Псевдокласс default**

Псевдокласс :default выбирает стандартный элемент на форме. Как правило, в роли такого элемента выступает кнопка отправки.

**<head> <style>**

**:default { border: 2px solid rgb(76, 0, 255);}</style> </head>**

**<body>**

**<form> <input name="login"/>**

**<input type="submit" value="Авторизироваться" />**

**</form> </body>**

Реализация данного кода представлена на рисунке. 4.21.

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Рисунок 4.21 - Псевдокласс default в CSS 3

**4.8.4 Псевдоклассы valid и invalid**

Псевдоклассы :valid и :invalid стилизуют элементы формы в зависимости от того, проходят они валидацию или нет. Далее представлен пример работы с этими псевдоклассами.

**<head> <style>**

**input:invalid {border: 3px solid FireBrick;}**

**input:valid {border: 3px solid ForestGreen;} </style></head>**

**<body><form> <h4>Почтовый адрес:</h4>**

**<p> Город:<input type="text" name="gorod" placeholder="Наименование города" required /></p>**

**<p>Улица:<input type="text" name="ulica" placeholder="Наименование улицы" required /></p> <input type="submit" value="Войти" /> </form> </body>**

Реализация данного кода представлена на рисунке 4.22.

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.22 - Использование псевдоклассов valid и invalid в CSS3

**4.8.5 Псевдоклассы in-range и out-of-range**

Псевдоклассы :in-range и :out-of-range стилизуют элементы формы в зависимости от того, попадает ли их значение в определенный диапазон. Это в первую очередь относится к элементу <input type="number" >.

**<head> <style>**

**:in-range {border: 4px solid rgb(35, 193, 35);}**

**:out-of-range { border: 4px solid rgb(247, 65, 65);} </style> </head>**

**<body> <form> <p> <label for="age">Введите возраст вашей собаки:</label>**

**<p> <input type="number" min="1" max="20" value="1" id="age" name="age"/>**

**</p> <input type="submit" value="Готово" /> </form> </body>**

**</html>**

Реализация данного кода представлена на рисунках 4.23-4.24.

**Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Рисунок 4.23 – Использование псевдоклассов out-of-range (корректный ввод)

**Изображение выглядит как текст, снимок экрана, мультимедиа, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Рисунок 4.24 - Псевдоклассы out-of-range (некорректный ввод)

Здесь атрибуты min и max задают диапазон, в которое должно попадать вводимое в поле значение:

**4.9 Селекторы атрибутов**

Кроме селекторов элементов также можно использовать селекторы их атрибутов. Например, есть на веб-странице несколько полей input, и необходимо окрасить в красный цвет только текстовые поля. В этом случае как раз можно проверять значение атрибута type: если оно имеет значение text, то это текстовое поле, и соответственно его надо окрасить в синий цвет. Определение стиля в этом случае выглядело бы так:

**input[type="text"]{border: 3px solid blue;}**

После элемента в квадратных скобках идет атрибут и его значение. То есть в данном случае для текстового поля надо установить границу синего цвета три пикселя толщиной, сплошной линией.

**<head> <style>input[type="text"]{border: 3px solid blue;}**

**</style> </head>**

**<body> <p><input type="text" id="email" /></p>**

**<p><input type="password" id="password" /></p>**

**<input type="submit" value="Отправить" /> </body>**

Реализация данного кода представлена на рисунке 4.25.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.25 - Селекторы атрибутов в CSS3

Селекторы атрибутов можно применять не только к элементам, но и классам и идентификаторам. Например, в ниже представленном коде представлен пример использования селекторов атрибутов при работе с классом.

**<head> <style>.link[href="https://ru.wikipedia.org/wiki/Музыкант"]{color: aqua;}**

**</style> </head> <body>**

**<h2>Интересное про профессии:</h2>**

**<a class="link" href="https://ru.wikipedia.org/wiki/Музыкант">Музыкант</a>**

**| <a class="link" href="https:// ru.wikipedia.org/ wiki/Пекарь"> Пекарь</a>**

**| <a class="link" href="https://ru.wikipedia.org/wiki/Стилист">Стилист</a>**

**| <a class="link" href="https://ru.wikipedia.org/wiki/Водитель">Водитель</a>**

**| <a class="link" href="https://ru.wikipedia.org/wiki/Психолог">Психолог</a>**

**</body>**

Реализация данного кода представлена на рисунке 4.26.

Изображение выглядит как текст, снимок экрана, мультимедиа, программное обеспечение

AI-generated content may be incorrect.

Рисунок 4.26 - Селекторы атрибутов в CSS

Специальные символы позволяют конкретизировать значение атрибутов. Например, символ ^ позволяет выбрать все атрибуты, которые начинаются на определенный текст. Например, надо выбрать все ссылки, которые используют протокол https, то есть ссылка должна начинаться на "https://". В этом случае можно применить следующий селектор:

**a[href^="https://"]{ color: aqua; }**

Если значение атрибута должно иметь в конце определенный текст, то для проверки используется символ $. Например, надо выбрать все изображения в формате jpg. В этом случае можно проверить, оканчивается ли значение атрибута src на текст ".jpg":

**img[src$=".jpg"]{width: 100px;}**

**4.10 Наследование и каскадность стилей**

**4.10.1 Наследование в CSS**

Для упрощения определения стилей в CSS применяется механизм наследования стилей. Этот механизм предполагает, что вложенные элементы могут наследовать стили своих элементов-контейнеров. Веб-страница имеет следующую структуру:

Изображение выглядит как диаграмма, линия, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.27 - Структура HTML и наследование стилей в CSS

Для элемента div переопределяется цвет текста. И так как элемент h2 и один из параграфов находятся в элементе div, то они наследуют стиль именно элемента div. Второй параграф находится непосредственно в элементе body и поэтому наследует стиль элемента body.

Однако не ко всем свойствам CSS применяется наследование стилей. Например, свойства, которые представляют отступы (margin, padding) и границы (border) элементов, не наследуются.

**4.10.2 Каскадность стилей**

Когда к определенному элементу применяется один стиль, то все относительно просто. Однако если же к одному и тому же элементу применяется сразу несколько различных стилей, то возникает вопрос, какой же из этих стилей будет в реальности применяться. В CSS действует механизм каскадности, которую можно определить как набор правил, определяющих последовательность применения множества стилей к одному и тому же элементу.

К примеру, определена следующая веб-страница, в которой применяются несколько стилей.

**<head> <style>**

**.BVLink {color: blueviolet;}**

**.main a {font-weight: bold;}**

**a {text-decoration: underline;}**

**</style> </head><body>**

**<p class="main"> Каскадность - это главный принцип CSS - приоритет одних правил/стилей над другими, когда одни стили перебивают другие.  Это означает, что CSS имеет внутреннюю иерархию, и стили с более высоким приоритетом перезаписывают правила с более низким приоритетом.**

**<a class=" BVLink" href="index.js">Подробнее узнать о каскадировании можно узнать по ссылке</a></p></body>**

Реализация данного кода представлена на рисунке 4.28.

Изображение выглядит как текст, снимок экрана, мультимедиа, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.28 - Каскадность в CSS 3

В данном примере определено три стиля и все они применяются к ссылке. Если к элементу веб-страницы применяется несколько стилей, которые не конфликтуют между собой, то браузер объединяет их в один стиль. Так, в данном случае, все три стиля не конфликтуют между собой, поэтому все эти стили будут применяться к ссылке.

Если же стили конфликтуют между собой, например, определяют разный цвет текста, то в этом случае применяется сложная система правил для вычисления значимости каждого стиля. Для определения стиля к элементу могут применяться различные селекторы, и важность каждого селектора оценивается в баллах. Чем больше у селектора пунктов, тем он важнее, и тем больший приоритет его стили имеют над стилями других селекторов.

* Селекторы тегов имеют важность, оцениваемую в 1 балл.
* Селекторы классов, атрибутов и псевдоклассов оцениваются в 10 баллов.
* Селекторы идентификаторов оцениваются в 100 баллов.
* Встроенные inline-стили (задаваемые через атрибут style) оцениваются в 1000 баллов.

Далее представлен пример применения каскадирования.

**<head> <style> #index {color: navy;}**

**.redLink {color: ; font-size: 20px;}**

**a {color: black; font-weight: bold;}**

**</style> </head> <body>**

**<a id="index" class="redLink" href="index.js">Основы CSS 3</a> </body>**

В данном случае к ссылке применяется сразу три стиля. Эти стили содержат два не конфликтующих правила:

**font-size: 20px;**

**font-weight: bold;**

которые устанавливают высоту шрифта 20 пикселей и выделение ссылки жирным. Так как каждое из этих правил определено только в одном стиле, то в итоге они будут суммироваться и применяться к ссылке без проблем.

Кроме того, все три стиля содержат определение цвета текста, но каждый стиль определяет свой цвет текста. Так как селекторы идентификаторов имеют больший удельный вес, то в конечном счете будет применяться темно-синий цвет, задаваемый селектором:

**#index {color: navy;}**

Если селектор является составным, то происходит сложение баллов всех входящих в селектор подселекторов. Так, рассматривается следующий пример.

**<head> <style> a {font-size: 18px;}**

**.nav li a {color: red;}**

**#menu a {color: navy;}**

**.nav .menuItem {color: green;}**

**a.menuItem:not(.newsLink) {color: orange;}**

**div ul li a {color: gray;}**

**</style> </head><body> <h3><Меню ресторана</h3>**

**<div id="menu"> <ul class="nav">**

**<li><a class="menuItem"> Филе миньон, приготовленный на углях с картофельным гратеном и шалотом конфи. </a></li>**

**<li><a class="menuItem">Говяжий язык с имбирной грушей и нежным спайси соусом. </a></li>**

**<li><a class="menuItem"> Тар-Тар из говяжьей вырезки с трюфельным айоли, булочкой бриошь и пармезаном. </a></li>**

**<li><a class="menuItem"> Карпаччо из говядины с трюфельной заправкой и сыром пармезан </a></li>**

**<li><a class="menuItem"> Креветки васаби с авокадо, специями шичимии и соусом понзу </a></li> </ul>**

**</div>**

**</body>**

Реализация данного кода представлена на рисунке 4.29.

Изображение выглядит как текст, снимок экрана, мультимедиа, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.29 - Составной селектор

В данном примере определено пять различных селекторов, которые устанавливают цвет ссылок. В итоге браузер выберет селектор #menu a и окрасит ссылки в темно-синий цвет. Этот выбор основан на суммировании баллов по каждому из пяти селекторов, представленный в таблице 4.1.

Таблица 4.1 – Присвоение баллов селекторам

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Селектор | Идентификаторы | Классы | Теги | Сумма |
| .nav li a | 0 | 1 | 2 | 12 |
| #menu a | 1 | 0 | 1 | 101 |
| .nav .menuItem | 0 | 2 | 0 | 20 |
| a.menuItem:not(.newsLink) | 0 | 2 | 1 | 21 |
| div ul li a | 0 | 0 | 4 | 4 |

Для селектора #menu a в колонке сумма оказалось больше всего баллов - 101. То есть в нем 1 идентификатор (100 баллов) и один тег (1 балл), которые в сумме дают 101 балл. В селекторе .nav .menuItem два селектора класса, каждый из которых дает 10 баллов, то есть в сумме 20 баллов. При этом псевдокласс :not в отличие от других псевдокдассов не учитывается, однако учитывается тот селектор, который передается в псевдокласс not. CSS предос тавляет возможность полностью отменить значимость стилей. Для этого в конце стиля указывается значение !important:

**a {font-size: 18px; color: red !important;}**

**#menu a {color: navy;}**

В этом случае вне зависимости от наличия других селекторов с большим количеством баллов к ссылкам будет применяться красный цвет, определяемый первым стилем.