**Раздел 2. Создание стилей с применением CSS**

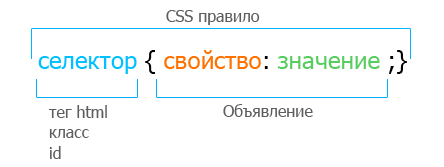
**2.1. Базовое представление о стиле**

Аббревиатура CSS расшифровывается как Cascading Style Sheets, что в переводе означает «каскадные таблицы стилей». Это язык разметки, используемый для визуального оформления веб-сайтов.

Объекты, расположенные на странице, размещаются с помощью HTML. А вот CSS отвечает за то, как эти объекты выглядят. Их размер, цвет, фоновое изображение, степень прозрачности, расположение относительно других элементов, поведение при наведении курсора, визуальное изменение кнопок при нажатии и т.п.

**2.2 Синтаксис CSS**

У языка CSS относительно простой синтаксис. Сначала прописывается селектор — он выбирает конкретный элемент на странице. Потом, после фигурных скобок, указываются свойства со значениями — между ними ставится двоеточие. Сами свойства отделяются друг от друга точкой с запятой.



Что такое селектор в CSS? Это конструкция, которая позволяет выбрать отдельные или однотипные элементы на странице, чтобы их стилизовать. С селектора начинается каждый блок объявлений в CSS:

.my-class {

**background-color**: #999;

}

В качестве селектора в примере используется my-class. Все элементы с этим классом получают единое оформление — серый фон цвета #999.

**2.3 Каскады в CSS**

И всё-таки почему CSS — это именно Cascading Style Sheets? Дело в том, что в единую схему стили организуются при помощи каскада. Вот простой пример, который также поможет понять, что такое CSS-код и как он выглядит:

p {

**color**: green;

**font-size**: 20px;

}

p {

**color**: red;

}

При помощи каскадов мы присвоили элементу p красный цвет. Зелёный цвет, указанный выше красного, учитываться не будет. Используется то значение параметра, которое указано ниже, и это помогает избежать конфликтов. При этом размер шрифта — 20 пикселей — не меняется. Если упростить написанное выше, то получится:

p {

**color**: red;

**font-size**: 20px;

}

Кроме того, один элемент можно сделать зависимым от нескольких селекторов разного типа. Пример:

p {

**color**: red;

}

p.important {

**font-size**: 20px;

}

#intro {

**font-style**: italic;

}

Чтобы использовать все параметры, в HTML указываем:

<**p** class="important" id="intro">

CSS упрощает форматирование документов.

</**p**>

Здесь мы присвоили элементу следующие свойства: он красный, размером 20 пикселей и написан курсивом. Важно, что в данном случае у разных селекторов — разный приоритет. Их порядок:

1. Селектор типа элемента (p).
2. Селектор класса (.important).
3. Селектор id (#intro).

1 — низкий приоритет, 3 — высокий.

**2.4 Приоритеты в CSS**

Обговорим подробнее вопрос приоритетов. Их иерархия работает следующим образом:

Самый высокий приоритет у свойств, в конце объявления которых указано !important.

Затем идут инлайновые стили, которые прописываются в теге через атрибут style.

Ещё ниже приоритет стилей, заданных в теге style в самом документе.

Далее следуют стили, подключённые к документу как внешний CSS-файл с использованием тега <link>

**2.6 Описание CSS-свойств**

CSS позволяет тегам HTML назначать любые визуальные стили, формируя таким образом уникальный внешний вид сайта. Делается это при помощи свойств CSS. В таблицах 4-13 приведены основные CSS-свойства, используемые для стилизации элементов страницы.

Таблица 4 - Шрифты

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Принимаемые значение** |
| font-family | Задает семейство шрифта, которое будет использоваться для оформления текста содержимого | *имя шрифта* | initial | inherit; |
| font-size | Определяет размер шрифта. | размер | % | medium | xx-small | x-small | small | large | x-large | xx-large | smaller | larger | initial | inherit; |
| font-style | Устанавливает стиль начертания шрифта - обычное, курсивное или наклонное | normal | italic | oblique | initial | inherit; |
| font-weight | Устанавливает насыщенность или "жирность" начертания шрифта | normal | bold | bolder | lighter | число | initial | inherit; |

Таблица 5 – Компоновка элементов на странице

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Принимаемые значение** |
| align-content | Выравнивает строки флекс-элементов по вертикали внутри флекс-контейнера | stretch | center | flex-start | flex-end | space-between | space-around | initial | inherit; |
| align-items | Выравнивает флекс-элементы внутри контейнера по высоте | stretch | center | flex-start | flex-end | baseline; |
| align-self | Устанавливает выравнивание по высоте отдельно взятого флекс-элемента | auto | stretch | center | flex-start | flex-end | baseline; |
| flex | Универсальное свойство, которое позволяет указать параметры элемента, чтобы он эффективно заполнял доступное пространство | flex-grow flex-shrink flex-basis | auto | none | initial | inherit; |
| flex-basis | Устанавливает базовую ширину флекс-элемента | *ширина* | auto | initial | inherit; |
| flex-direction | Определяет направление главной оси во флекс-контейнере | row | row-reverse | column | column-reverse | initial | inherit; |
| flex-flow | Позволяет за одну декларацию задать направление главной оси и возможность переноса флекс-элементов на новую строку | *flex-direction flex-wrap* | initial | inherit; |
| flex-grow | Определяет коэффициент увеличения ширины флекс-элемента относительно других флекс-элементов | *число* | initial | inherit; |
| flex-shrink | Определяет коэффициент уменьшения ширины флекс-элемента относительно других флекс-элементов | *число* | initial | inherit; |
| flex-wrap | Устанавливает, следует ли флекс-элементам располагаться в одну строку или можно занять несколько строк | nowrap | wrap | wrap-reverse | initial | inherit; |
| justify-content | Выравнивает флекс-элементы по ширине флекс-контейнера | center | flex-start | flex-end | space-between | space-around | initial | inherit; |
| order | Определяет порядок вывода флекс-элементов внутри флекс-контейнера | *число* | initial | inherit; |

Таблица 6 – Цвет и фон

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Принимаемые значение** |
| background | Короткое свойство, устанавливающее все свойства фона элемента за одну декларацию | *bg-color bg-image bg-position/bg-size bg-repeat bg-origin bg-clip bg-attachment* | initial | inherit; |
| background-attachment | Определяет, будет ли фоновое изображение зафиксировано, или оно будет прокручиваться вместе с остальным содержимым страницы | scroll | fixed | local | initial | inherit; |
| background-clip | Устанавливает область прорисовки фонового изображения | padding-box | border-box | content-box | initial | inherit; |
| background-color | Определяет фоновый цвет элемента | *цвет* | transparent | initial | inherit; |
| background-image | Определяет фоновое изображение элемента | url('*URL*') | none | initial | inherit; |
| background-repeat | Определяет, как будет повторяться фоновое изображение, установленное с помощью параметра background-image | repeat | repeat-x | repeat-y | no-repeat | initial | inherit; |
| background-size | Устанавливает размер фонового изображения | auto | *размеры* | % | cover | contain | initial | inherit; |
| color | Это свойство определяет цвет текста элемента | *цвет* | initial | inherit; |

Таблица 7 – Рамка и контур

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Принимаемые значение** |
| border | Это свойство позволяет определить все атрибуты рамки HTML элемента за одну декларацию | *border-width border-style border-color* | initial | inherit; |
| border-bottom | Это свойство позволяет определить все атрибуты нижней рамки HTML элемента за одну декларацию | *border-width border-style border-color* | initial | inherit; |
| border-bottom-color | Это свойство определяет цвет нижней стороны рамки HTML элемента | *цвет* | transparent | initial | inherit; |
| border-color | Это свойство определяет цвет всех четырех сторон рамки HTML элемента | *цвет* | transparent | initial | inherit; |
| border-radius | Устанавливает радиус скругления уголков рамки | *радиус* |% [/ *радиус*|%] | initial | inherit; |
| outline | Универсальное свойство, которое устанавливает атрибуты контура элемента | *outline-color outline-style outline-width* | initial | inherit; |
| outline-color | Определяет цвет контура | *цвет* | invert | initial | inherit; |
| outline-offset | Задает величину отступа контура от элемента | *расстояние* | initial | inherit; |
| outline-width | Определяет толщину контура | medium | thin | thick | *толщина* | initial | inherit; |

Таблица 8 – Свойства таблиц

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Принимаемые значение** |
| border-collapse | Это свойство определяет, какую модель рамки вокруг ячеек таблицы следует использовать | separate | collapse | initial | inherit; |
| border-spacing | Это свойство определяет расстояние между рамками соседних ячеек таблицы | *расстояние* | initial | inherit; |
| caption-side | Это сойство определяет, где будет отображаться заголовок таблицы | top | bottom | initial | inherit; |
| empty-cells | Это свойство задает отображение границ и фона в ячейке таблицы, если она пустая | show | hide | initial | inherit; |
| table-layout | Определяет, как браузер должен вычислять высоту и ширину ячеек таблицы, основываясь на ее содержимом | auto | fixed | initial | inherit; |

Таблица 9 - Позиционирование и отображение

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Принимаемые значение** |
| bottom | Это свойство определяет расстояние от нижнего края HTML элемента до нижнего края его родительского элемента | auto | *расстояние* | initial | inherit; |
| clear | Это свойство определяет, с какой стороны элемента запрещено его обтекание другими элементами | none | left | right | both | initial | inherit; |
| clip | Это свойство определяет область позиционированного элемента, в которой будет показано его содержимое | auto | rect(*Y1, X1, Y2, X2*) | initial | inherit; |
| display | Определяет, как элемент должен быть показан в документе | Inline | block | flex | inline-block |
| float | Определяет, по какой стороне будет выравниваться элемент, при этом остальные элементы будут обтекать его с другой стороны | none | left | right | initial | inherit; |
| left | Это свойство определяет расстояние от левого края HTML элемента до левого края его родительского элемента, т.е. расстояние его смещения вправо | auto | *расстояние* | initial | inherit; |
| overflow | Определяет, каким образом будет отображаться содержимое элемента, если оно не помещается в область вывода данного элемента | visible | hidden | scroll | auto | initial | inherit; |
| overflow-wrap | Определяет, должен ли браузер переносить длинные слова на новую строку при переполнении контейнерного элемента | normal | anywhere | break-word | initial | inherit; |
| overflow-x | Определяет, как будет отображаться содержимое элемента, если оно не помещается слева/справа в область вывода данного элемента | visible | hidden | scroll | auto | initial | inherit; |
| overflow-y | Определяет, как будет отображаться содержимое элемента, если оно не помещается вверху/внизу в область вывода данного элемента | visible | hidden | scroll | auto | initial | inherit; |
| position | Устанавливает способ позиционирования элемента | static | absolute | fixed | relative | initial | inherit; |
| right | Это свойство определяет расстояние от правого края HTML элемента до правого края его родительского элемента | auto | *расстояние* | initial | inherit; |
| top | Это свойство определяет расстояние от верхнего края HTML элемента до верхнего края его родительского элемента | auto | *расстояние* | initial | inherit; |
| visibility | Определяет видимость элемента | visible | hidden | collapse | initial | inherit; |
| z-index | Определяет порядок позиционированных элементов по z-оси | auto | *число* | initial | inherit; |

Таблица 10 – Тени и прозрачность

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Принимаемые значение** |
| box-shadow | Добавляет тень к элементу | none | *x-сдвиг y-сдвиг размытие растяжение цвет* | inset | initial | inherit; |
| opacity | Определяет уровень прозрачности элемента | *число* | initial | inherit; |
| text-shadow | Добавляет тень к тексту | none | *x-сдвиг y-сдвиг размытие цвет* | initial | inherit; |

Таблица 11 – Размеры

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Принимаемые значение** |
| box-sizing | Устанавливает алгоритм вычисления ширины и высоты элемента | content-box | border-box | initial | inherit; |
| height | Устанавливает высоту блочных или заменяемых элементов | auto | *значение* | % | initial | inherit; |
| max-height | Устанавливает максимальную высоту элемента. | *значение* | % | none | initial | inherit; |
| max-width | Устанавливает максимальную ширину элемента | *значение* | % | none | initial | inherit; |
| min-height | Устанавливает минимальную высоту элемента | *значение* | % | none | initial | inherit; |
| min-width | Устанавливает минимальную ширину элемента | *значение* | % | none | initial | inherit; |
| resize | Определяет, может ли пользователь изменять размер элемента | none | both | horizontal | vertical | initial | inherit; |
| width | Устанавливает ширину блочных или заменяемых элементов | auto | *значение* | % | initial | inherit; |

Таблица 12 - Внешние и внутренние отступы

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Принимаемые значение** |
| margin | Универсальное свойство, которое позволяет устанавливать размер всех отступов элемента за одну декларацию | *размер* | auto | initial | inherit; |
| margin-bottom | Это свойство устанавливает размер нижнего отступа элемента | *размер* | auto | initial | inherit; |
| margin-left | Это свойство устанавливает размер левого отступа элемента | *размер* | auto | initial | inherit; |
| margin-right | Это свойство устанавливает размер правого отступа элемента | *размер* | auto | initial | inherit; |
| margin-top | Это свойство устанавливает размер верхнего отступа элемента | *размер* | auto | initial | inherit; |
| padding | Универсальное свойство, которое позволяет устанавливать размер всех полей элемента за одну декларацию | *размер* | auto | initial | inherit; |
| padding-bottom | Это свойство устанавливает размер нижнего поля элемента | *размер* | auto | initial | inherit; |
| padding-left | Это свойство устанавливает размер левого поля элемента | *размер* | auto | initial | inherit; |
| padding-right | Это свойство устанавливает размер правого поля элемента | *размер* | auto | initial | inherit; |
| padding-top | Это свойство устанавливает размер верхнего поля элемента | *размер* | auto | initial | inherit; |

Таблица 13 - Свойства текста

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Принимаемые значение** |
| letter-spacing | Устанавливает расстояние между символами в пределах элемента | normal | *значение* | initial | inherit; |
| line-height | Устанавливает межстрочный интервал текста | normal | *множитель* | *значение* | % | initial | inherit; |
| text-align | Устанавливает горизонтальное выравнивание текста элемента | left | right | center | justify | initial | inherit; |
| text-decoration | Добавляет оформление текста в виде подчеркивания, перечеркивания, линии над текстом | none | underline | overline | line-through | initial | inherit; |
| text-decoration-color | Устанавливает цвет линии, которая добавляется через свойство text-decoration | *цвет* | initial | inherit; |
| text-decoration-line | Устанавливает тип оформления текста - подчеркивание, перечеркивание, линия над текстом | none | underline | overline | line-through | initial | inherit; |
| text-decoration-style | Устанавливает стиль декоративной линии оформления текста | solid | double | dotted | dashed | wavy | initial | inherit; |
| text-overflow | Определяет параметры видимости текста в блоке, если текст целиком не помещается в заданную область | clip | ellipsis | *строка* | initial | inherit; |
| text-transform | Управляет преобразованием текста элемента в заглавные или прописные символы | none | capitalize | uppercase | lowercase | initial | inherit; |
| white-space | Устанавливает, как отображать пробелы между словами в пределах элемента | normal | nowrap | pre | pre-line | pre-wrap | initial | inherit; |
| word-break | Определяет, как делать перенос на новую строку внутри слов | normal | break-all | keep-all | initial | inherit; |
| word-spacing | Устанавливает расстояние между словами в пределах элемента | normal | *значение* | initial | inherit; |
| word-wrap | Устанавливает, можно или нельзя переносить на новую строку длинные слова | normal | break-word | initial | inherit; |