# Глава 1. Основы HTML и CSS

## Теги HTML

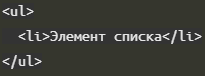
### 1.1.1 Парные и одиночные теги

Теги бывают парными и одиночными. Пример парного: *<h1>Текст заголовка</h1>.*

Пример одиночного: <img src="keks.png" width="200" height="100">

Комментарии: Код, заключённый между символами <!-- и --> - не работает

В парные теги можно вкладывать другие теги. Например, как в списках:

**

У вложенных тегов всегда нужно следить за правильным порядком закрытия. Вложенный тег не может закрываться позже родительского:



И ещё, не все теги можно вкладывать в другие теги, например уже знакомый вам тег <h1> нельзя вкладывать в <p>. Эти правила вложенности для каждого тега вы узнаете постепенно

## CSS

### 1.2.1 Описание стиля и его применение

Описание некого стиля в .css файле в классе feature-kitten:



Применение класса к абзацу:



Все параметры отображения тегов, на которые влияет CSS, задаются с помощью свойств в следующем формате:



Например:



**Комментарии:** /\* и \*/

### 1.2.2 Селекторы и классы

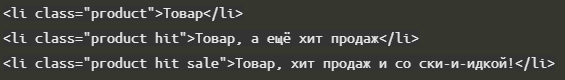
Селектор указывает, к каким тегам применятся свойства из CSS-правила. Проще всего работают селекторы по тегам: они выбирают все теги с подходящим именем. Например:



В примере селектором является p, и он выбирает все теги с именем p (то есть теги <p>), а теги с другим именем, например h1, не выбирает.

Селекторы могут быть классами:

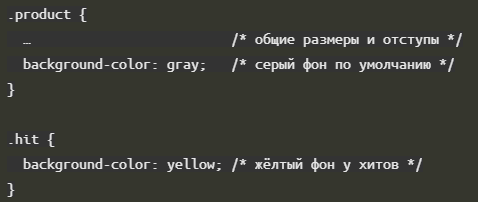




### 1.2.3 Использование нескольких классов

Это типовой приём, который ещё называется миксованием классов. Обычно его используют так: в один класс выносят общее оформление, а в дополнительных классах описывают его модификации. В примере выше размеры карточек товаров можно описать в CSS-правиле .product, а особый фон для хита продаж — в правиле .hit. В общем, удобный приём, помогает сократить дублирование кода.

**Миксование**:



Оба CSS-правила подействуют на второй товар, ведь у него есть два подходящих класса. Но чтобы цвет фона стал жёлтым, **правило .hit должно находиться в коде ниже** правила .product. В этом случае свойство background-color из второго правила переопределит значение свойства background-color из первого правила.

# Глава 2. Структура HTML документа

## 2.1 Структурные и смысловые теги на html странице

### 2.1.1 <!DOCTYPE html>

**<!DOCTYPE html>** - это декларация типа документа



Простейшая HTML-страница состоит как минимум из трёх тегов: <html>, <head> и <body>. Тег <html> располагается в документе сразу после доктайпа и содержит все остальные теги, включая <head> и <body>. Тегу <html> обычно добавляют важный атрибут lang в котором задаётся язык документа.



### 2.1.1 Тег <head>

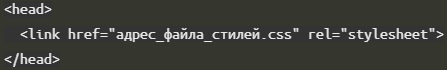
Тег <head> хранит важную служебную информацию о странице.

*\*Атрибут lang можно задавать любым тегам, если нужно уточнить, на каком языке написан текст внутри тега. Но если задать его тегу <html>, то действие атрибута распространится на весь документ целиком.*

Внутри <head> указывается:

* Заголовок
* ключевые слова
* описание страницы
* подключаются стили и другие внешние ресурсы
* и другое

Пример подключения стиля:

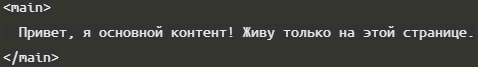


### 2.1.2 Тег <title> (внутри head)

В нём задаётся заголовок страницы, который отображается во вкладках браузера



### 2.1.3 Тег main (внутри body)

И первый такой тег — это тег <main>. Он выделяет основное содержание страницы, которое не повторяется на других страницах. 

### 2.1.4 Теги header и footer, шапка и подвал (внутри body)

Вводную часть страницы, которую чаще называют «шапкой», описывает тег <header>. Заключительную часть страницы, или «подвал», описывает тег <footer>

Обычно на странице появляется по одному тегу <header> и <footer>, но их может быть и больше.

*\*Тег <header> — это не только шапка сайта с логотипом и меню, он может использоваться, например, и как «шапка» какой-нибудь статьи или раздела сайта. Конечно, в случае со статьёй <header> называют не «шапкой», а вводной частью, в которой могут содержаться заголовки, оглавление и так далее.*

*С <footer> ситуация аналогичная. В привычном понимании это подвал сайта, с копирайтами, контактной информацией и так далее. Но <footer> может использоваться и в других разделах сайта. Например, внутри статьи в «подвале» можно разместить дополнительную информацию: данные об авторе, дополнительные ссылки и так далее.*

### 2.1.5 Тег section, смысловой раздел (обычно, внутри main)

*\*Небольшая подсказка как отличать смысловые или логические разделы от структурных разделов. Логический раздел вы можете осмысленно назвать одним словом или словосочетанием: «программа обучения», «каталог», «наши преимущества». Если же в голову приходят названия, которые указывают на его положение на странице («шапка», «подвал», «левая колонка»), либо перечисления («новости и галерея», «фильтры и товары»), в таком случае раздел структурный и тег <section> для него не подходит.*

### 2.1.6 Тег nav, основная навигация

Обычно в <nav> включают ссылки на другие страницы или навигацию по текущей странице.

### 2.1.7 Тег article

тег <article>обозначает цельный, законченный и самостоятельный фрагмент информации.

### 2.1.8 Тег aside

Тег <aside> включает в себя дополнительное содержание, не связанное напрямую с основным. Такие блоки ещё часто называют «сайдбарами» или боковыми панелями.

### 2.1.9 h1

Заголовок <h1> — самый важный на странице. В него нужно включать текст, который в целом описывает содержание страницы. Очень важно, чтобы заголовок первого уровня на странице был только один. На главных страницах заголовок верхнего уровня часто добавляют в шапку сайта. На внутренних страницах заголовок первого уровня чаще всего находится в основном контенте, то есть внутри тега <main>.

### 2.1.10 Тег p, параграф

Тег <p> позволяет лишь явно выделять параграфы, группируя элементы с фразовым типом содержимого. И это могут быть не только блоки текста, а, например, изображения, ссылки или поля ввода. Не все теги могут быть включены в <p>. Например, внутри абзаца не могут располагаться крупные структурные теги.

### 2.1.11 Тег meta, кодировка страницы (в head)

Сообщает кодировку текста, описание контента и так далее.

Пример:



**Ключевые слова**

Перечень ключевых слов задаётся тегом <meta>, у которого атрибут name имеет значение keywords. Ключевые слова (самые важные слова из содержания страницы) перечисляются в атрибуте content через запятую:



Краткое описание страницы задаётся похожим образом, только значение атрибута name меняется на description:

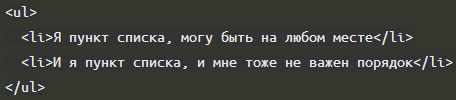


# Глава 3: Разметка текста

## 3.1 Списки и пр.

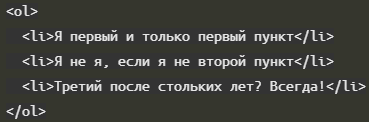
### 3.1.1 Тег ul, неупорядоченный список

Его используют, когда порядок элементов не важен. Например, для разметки перечня ссылок в меню. В теге <ul> могут находиться только теги <li> которые обозначают элементы списка. Пунктов может быть неограниченное количество, но должно быть не менее одного.



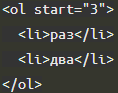
По умолчанию элементы <ul> отмечаются маркерами такого же цвета, как цвет текста.

### 3.1.2 Тег ol, упорядоченный список



### 3.1.3 Атрибуты start и reversed в ol

С помощью start можно поменять стартовое число списка.

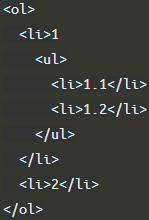


Атрибут reversed меняет направление нумерации на противоположный.

С помощью ещё одного атрибута, type, можно задавать различные типы маркеров.

### 3.1.4 Вложенные списки

Сначала нужно создать список первого уровня, а затем между тегами <li> и </li> этого списка добавить ещё один список.



### 3.1.5 Теги dl, dt и dd, список описаний

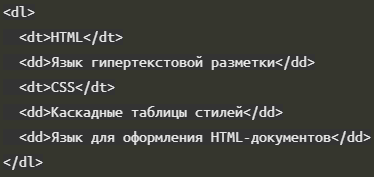
<dl> (сокращение от «description list») обозначает сам список описаний;

<dt> (сокращение от «description term») обозначает термин;

<dd> (сокращение от «description definition») обозначает описание или определение.

Теги <dt> и <dd> пишутся внутри <dl>.

Каждый список <dl> может содержать один или несколько терминов и одно или несколько описаний для каждого термина. Пример кода:



## 3.2 Цитирование и форматирование текста

### 3.2.1 Тег pre, преформатированный текст

Браузер сохраняет и отображает все пробелы и переносы, которые есть внутри тега <pre>

### 3.2.2 Символы-мнемоники в HTML

Мнемоники — это особые строки, которые начинаются с амперсанда (&) и заканчиваются точкой с запятой (;). Например, знак меньше на страницу можно вставить мнемоникой &lt; (less than), а знак больше мнемоникой &gt; (greater than)

### 3.2.3 Тег code, фрагмент кода

Для обозначения фрагментов кода есть специальный тег, который так и называется <code>

### 3.2.4 Тег q, встроенная цитата

Предназначен для выделения цитат внутри предложения.

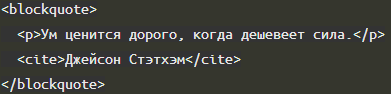
Если известен интернет-адрес цитаты, то его можно указать в атрибуте cite:



### 3.2.5 Тег blockquote, отдельная цитата

Для выделения длинных цитат, которые могут состоять из нескольких абзацев, предназначен тег <blockquote>.

Может быть атрибут cite, содержащий ссылку на источник цитаты. И также в паре с <blockquote> можно использовать **тег** <cite>, который обычно размещается внутри цитаты.



### 3.2.6 Тег br, перенос строки

Иногда этот тег применяют неправильно, для разбиения текста на «как бы абзацы». Никогда так не делайте. Для разметки абзацев нужно использовать тег <p>.

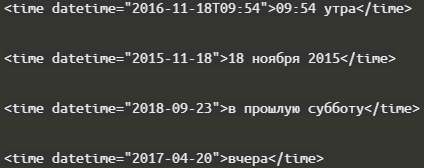
### 3.2.7 Теги sub и sup, нижний и верхний индексы

Тег <sup> (сокращение от «superscript») отображает текст в виде верхнего индекса, а тег <sub> (сокращение от «subscript») в виде нижнего индекса.

Для создания более сложных формул <sup> и <sub> можно использовать внутри друг друга.

### 3.2.8 Тег time, дата и время

С помощью <time> можно описывать даты одновременно и для человека, и для машины. Дата для человека описывается внутри тега, а дата для машины — внутри атрибута datetime в формате ISO 8601:



### 3.2.9 Теги em и i, акцентирование внимания (курсив)

Тег <em> определяет текст, на который сделан особый акцент, меняющий смысл предложения. (интонация ридера)

Тег <i> применяется для обозначения текста, который отличается от окружающего текста, но не является более важным (названия, термины?)

### 3.2.10 Теги strong и b, важность и выделение (жирный)

Тег <strong> указывает на важность (интонация ридера). Тег <b> предназначен для выделения текста с целью привлечения к нему внимания, но без придания ему особой важности.

### 3.2.11 Теги del и ins, выделение изменений

Тег <del> выделяет текст, который был удалён в новой версии документа. В браузере этот текст перечёркивается.

Тег <ins> выделяет текст, который был добавлен в новой версии документа. В браузере этот текст подчёркивается.

Оба тега имеют атрибут datetime, в котором можно указать дату и время, когда была внесена та или иная правка.

### 3.2.11 Теги div и span, контейнеры для стилизации

Это «чистые» элементы, которые отлично подходят для визуальной группировки. Это не смысловые теги, а стиливые, они не имеют никакого оформления по умолчанию.

# Глава 4:Ссылки и изображения

## 4.1Ссылки

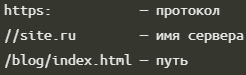
### 4.1.1 Что такое ссылка, тег a

Ссылки создаются с помощью очень короткого тега <a> (сокращение от «anchor»). Например, вот так:



### 4.1.2 Абсолютные адреса

Абсолютные адреса содержат минимум три части: протокол, имя сервера и путь.



### 4.1.3 Ссылка на файл

Для того чтобы предотвратить открытие файлов прямо в браузере, у тега <a> существует атрибут download, который поможет именно скачать файл.



### 4.1.4 Ссылка-якорь

Вот так выглядит адрес, состоящий из одного якоря:



При щелчке по такой ссылке браузер найдёт на странице элемент с соответствующим атрибутом id и прокрутит окно страницы к нему.



### 4.1.5 Ссылки с пустым href, атрибут title

Тег <a> можно использовать вообще без адреса, то есть без атрибута href. Такой тег обозначает «ссылку-заглушку». Часто ссылки-заглушки используют, чтобы показать, что мы находимся на текущей странице.

Если удаляем href у ссылки, то лучше оставить подсказку о том, почему мы это сделали. Подсказку можно добавить с помощью атрибута title

## 4.2 Изображения

### 4.2.1 Тег img, изображение

Он одиночный, то есть, как мы уже знаем, не требует закрывающего тега. Сам по себе тег <img> бесполезен без своего атрибута src, который указывает путь к картинке.



### 4.2.2 Форматы изображений

.svg – векторный, очень легкий, простые логотипы

.jpeg – сжатые фотографии

.png – без сжатия, если важна четкость. Поддерживает прозрачность

.gif – простейшая анимация – поддерживает прозрачность, но не полупрозрачность

### 4.2.3 Размеры изображения

Чтобы управлять шириной и высотой изображения, нужно использовать атрибуты width и height. Размеры задаются без единиц измерения px. Например:



Если задать только один из размеров, то вторую размерность браузер вычислит самостоятельно исходя из пропорций изображения.

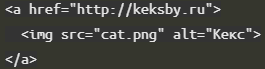
### 4.2.4 Атрибут alt

Альтернативный текст изображения задаётся с помощью атрибута alt. Например:

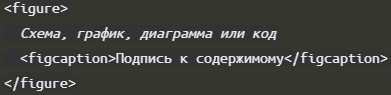


### 4.2.5 Изображение-ссылка

Ссылки можно делать не только с помощью текста, но и с помощью изображений. Для этого нужно обернуть тег <img> в тег <a>. Например:



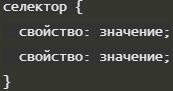
### 4.2.6 Теги figure и figcaption, демонстративный материал



# Глава 5:Основы CSS

### CSS-правила, вводная информация

CSS-правило состоит из селектора и перечня свойств и их значений:

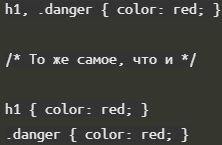


### Селекторы

Определяет к каким HTML-элементам применятся свойства и значения из правила. Простейшие (и самые популярные) селекторы — это селекторы по тегам и по классам. Селекторы по тегам содержат имя тега без символов < и > и применяются ко всем подходящим тегам. Селекторы по классам начинаются с точки, за которой идёт имя класса, и применяются ко всем тегам с подходящим атрибутом class

### Группировка селекторов

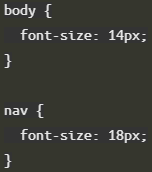
Если у CSS-правил отличаются *только* селекторы, а свойства и значения одинаковые, то их можно сгруппировать через запятую. Например:



### Наследование

Это механизм, с помощью которого значения свойств элемента-родителя передаются его элементам-потомкам.

Стили, присвоенные одному элементу, наследуются всеми потомками (вложенными элементами), но только в том случае, если они где-то явно не переопределены. Например, размер шрифта и его цвет достаточно применить к body, чтобы большинство элементов внутри имели те же свойства:



Размер шрифта у всего текста на странице, кроме текста внутри навигации, станет равен 14px. У nav есть своё объявленное значение размера шрифта (18px), и оно будет использоваться вместо наследуемого от body значения (14px). А ещё 18px станет новым наследуемым значением для потомков nav.

### Наследуемые свойства

Не все свойства в CSS наследуются. К наследуемым относятся в основном свойства, определяющие параметры отображения текста. Параметры текста зачастую не меняются в пределах крупных блоков страницы: меню, основного содержания, информационных панелей. Поэтому общие параметры текста (цвет, размер, гарнитура) обычно указывают в стилях этих крупных блоков.

### Ненаследуемые свойства

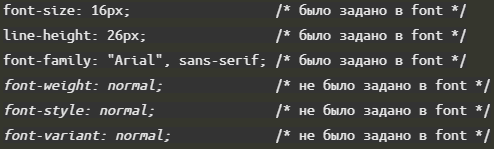
Но не все свойства наследуются. Основные ненаследуемые свойства — это параметры позиционирования, размеров, отступов, фона, рамок.

### Составные свойства

Есть обычные свойства, управляющие одним параметром отображения, и есть составные свойства, управляющие одновременно несколькими параметрами. Пример:



Будет преобразовано в:



### Типы значений: абсолютные и относительные

Абсолютные единицы измерения привязаны к настоящим физическим размерам.

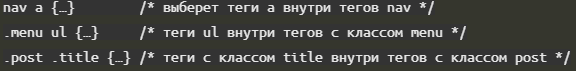
Относительные единицы измерения описывают значения, которые зависят от других значений. Например, ширина элемента в процентах зависит от ширины родительского элемента, а ширина элемента в em зависит от размера шрифта самого элемента. К относительным единицам относятся em, rem, vh, vw и некоторые другие. Например, проценты используют для «резиновой» вёрстки, а em применяют в вёрстке государственных сайтов с особыми дополнительными требованиями к масштабированию текста.

### Селекторы по тегам и по классам

Бывают селекторы по тегам и по классам

### Вложенные селекторы

Вы можете комбинировать любые типы селекторов через пробел. Такие селекторы называются вложенными или контекстными и читаются справа налево. Например:



### CSS классы через пробел и без него

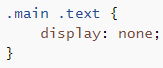
Имеем следующую структуру:



Для того чтобы обратиться к блоку, стили в котором идут друг за другом, то пробел не нужно делать. Так, мы обратимся к блоку div:



А таким образом к элементу span:



### Стили по умолчанию

Некоторым элементам можно не задавать никаких стилей, но у них всё равно будет какое-то оформление.

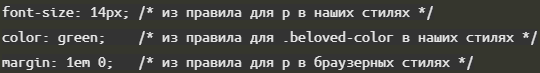
У списка есть отступы и маркеры, но откуда они берутся? Эти стили хранятся внутри браузера, это браузерные стили по умолчанию. Размер шрифта body не будет наследоваться заголовками. Это происходит из-за того, что размер шрифта заголовков явно задан внутри браузерных стилей по умолчанию.

### Каскадирование

Если в наших стилях есть такой код:



То у нашего абзаца про цвет будет такой итоговый набор свойств и значений:



Этот механизм комбинирования стилей из разных источников в итоговый набор свойств и значений для каждого тега и называется каскадностью.

### Конфликт свойств

Если в css правилах есть одинаковые свойства с разными значениями, то возникает конфликт. Например:



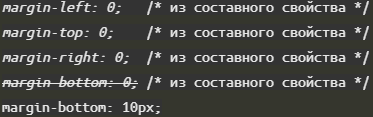
Конфликт разрешается максимум за три шага:

* Сравниваются приоритеты стилевых файлов, в которых находятся конфликтующие свойства. Например, авторские (то есть наши) стили приоритетнее браузерных.
* Сравнивается специфичность селекторов у CSS-правил с конфликтующими свойствами. Например, селектор по классу более специфичен, чем селектор по тегу.
* Побеждает то свойство, которое находится ниже в коде.

Каскад работает и внутри CSS-правил:

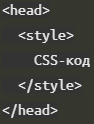


Превратится в



### Встроенные стили, атрибут style

Вариант подключения стилей в теге <style> используется чаще всего для оптимизации загрузки страницы:



Предположим, что нужно иметь возможность управлять в разметке шириной с помощью свойства width в диапазоне от 0% до 100%. Чтобы сделать это через CSS, пришлось бы создать 100 классов, и применять их. Проще сделать так:



# Глава 6: Оформление текста из CSS

### Свойство font-size, размер шрифта

Значение свойства задаёт желаемую высоту символа шрифта. Причём единицы измерения могут быть абсолютными или относительными. Самая часто используемая единица измерения размера шрифта — пиксели px.

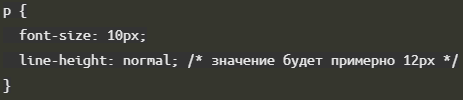
### Относительный размер шрифта

Хочется сделать так, чтобы при изменении основного размера шрифта для родителя (например, для body), его дочерние элементы пропорционально меняли свои размеры шрифта. Для этого есть единица измерения — em.

Величина 1em — это такой же размер, что и у родителя. 2em – в два раза больше родителя.

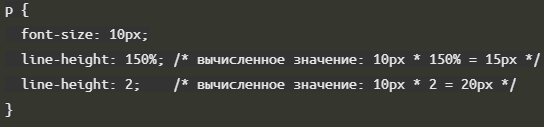
### Свойство line-height, высота строки (межстрочный интервал)

По умолчанию это свойство имеет значение normal. Оно указывает браузеру, что межстрочный интервал нужно подобрать автоматически, исходя из размера шрифта. Спецификация рекомендует устанавливать его в пределах 100-120% от размера шрифта. То есть:



### Относительная высота строки (межстрочный интервал)

в зависимости от font-size:



### Свойство font-family, семейство шрифта

В значении этого свойства указывается список названий шрифтов, перечисленных через запятую. В начале списка располагают самый желаемый шрифт, затем менее желаемый, а в самом конце списка — общий тип шрифта (системный).

### Свойство font-weight, толщина шрифта

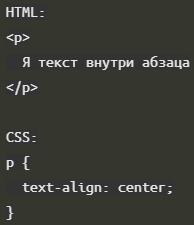
В качестве значения можно использовать ключевое слово или число. Самые часто встречающиеся значения:

400 или normal — обычный шрифт, значение по умолчанию;

700 или bold — жирный шрифт.

### Свойство text-align, горизонтальное выравнивание текста

Описывает, как выравнивается текст внутри блока по горизонтали (на самом деле не только текст). Может принимать следующие значения: left, right, center, justify. Важно помнить, что свойство text-align применяется именно к самому блоку-контейнеру, внутри которого находится текстовый контент:



### Выравнивание text-align не только для текста

Свойство text-align способно влиять не только на тексты, но и другие элементы. Эти элементы должны быть инлайновыми, то есть это изображения, инлайн-блоки, инлайн-таблицы и другие.

### Свойство vertical-align, вертикальное выравнивание текста

Этим свойством можно выравнивать инлайновые элементы относительно содержащей его строки.

У свойства vertical-align много значений, но самые часто используемые:

top — выравнивание по верхнему краю строки;

middle — по середине строки;

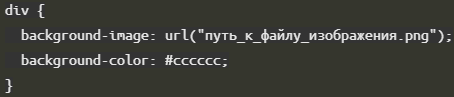
bottom — по нижнему краю строки;

baseline — по базовой линии строки (значение по умолчанию).

В отличие от text-align свойство vertical-align задаётся самому элементу, а не содержащему его контейнеру.

### Использование фона для оформления текста

Фоновое изображение для блока задаётся свойством background-image, а фоновый цвет — свойством background-color:



### Свойство color, цвет текста

Если не хочется иметь дело с шестнадцатеричными значениями, можно воспользоваться специальной функцией rgb:



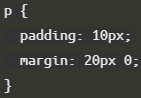
У функции rgb есть расширенная версия — rgba. В этом случае помимо указания цвета последним значением указывается степень непрозрачности цвета — alpha. Значение может быть от 0 (полностью прозрачный) до 1 (полностью непрозрачный):



### Использование отступов для оформления текста (padding)

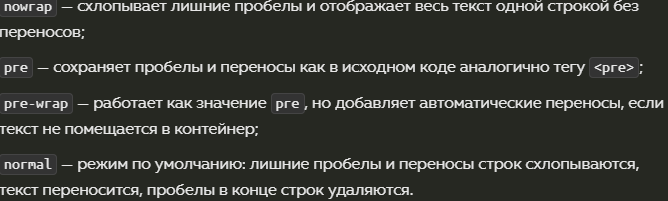
Вокруг текста должно быть достаточно «воздуха», он не должен «прилипать» к краям, ему не должно быть «тесно».

За отступы в CSS отвечают два свойства: padding задаёт внутренние отступы в блоке, а margin задаёт внешние отступы. Пример, как можно их использовать:

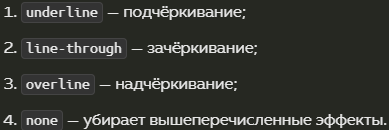
 сверху/снизу – 20, справа/слева – 0;

### Свойство white-space, управление пробелами

C помощью CSS управлять пробелами и переносами можно более гибко с помощью white-space:

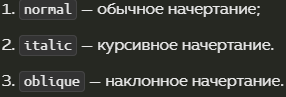


### Свойство text-decoration, подчёркивание и другие эффекты



Свойство text-decoration — составное, а это значит, что его можно разложить

### Свойство font-style, курсив



### Свойство text-transform, регистр символов

