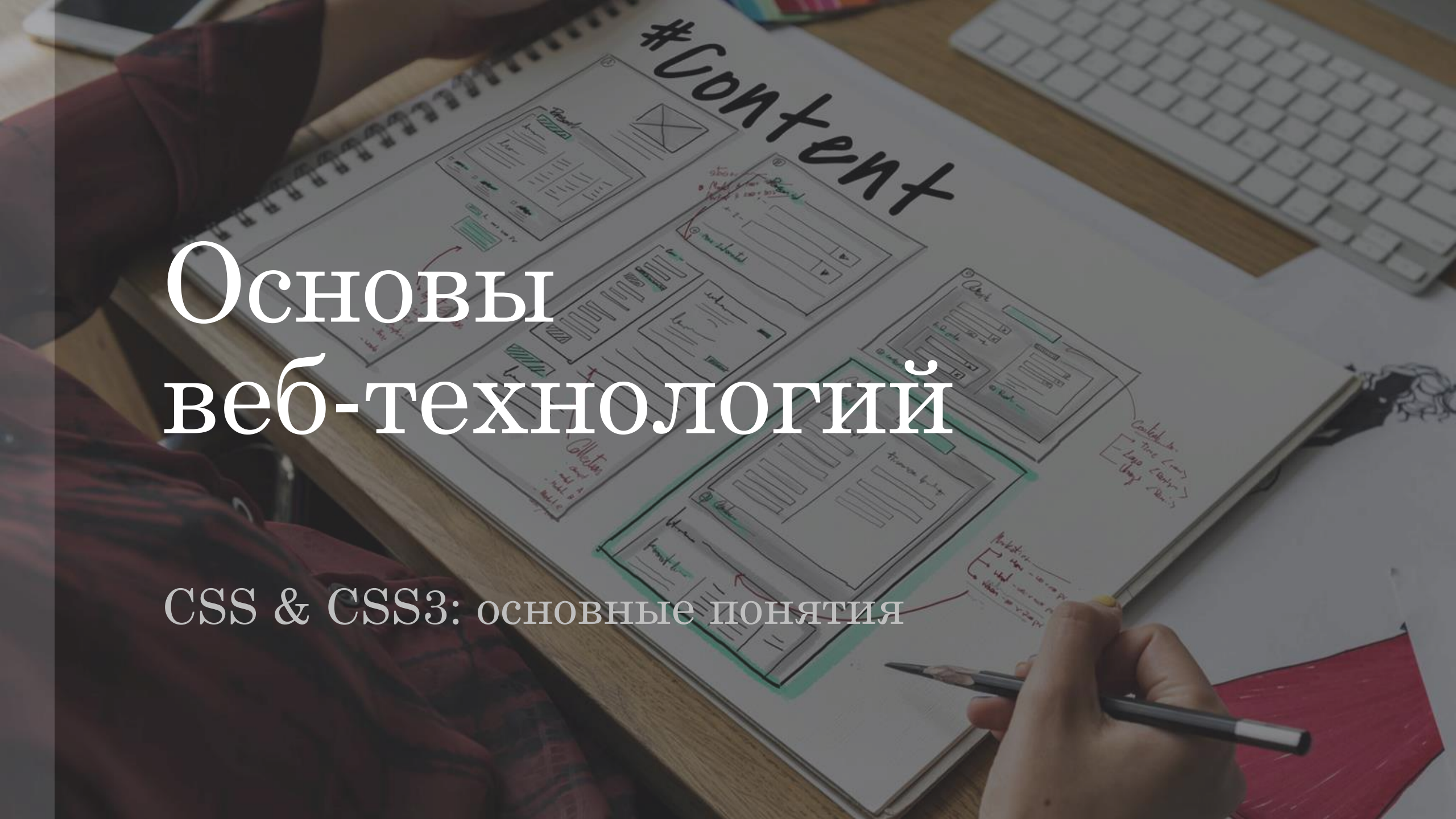


Основы веб-технологий

CSS & CSS3: основные понятия

CSS (Cascading Style Sheets)

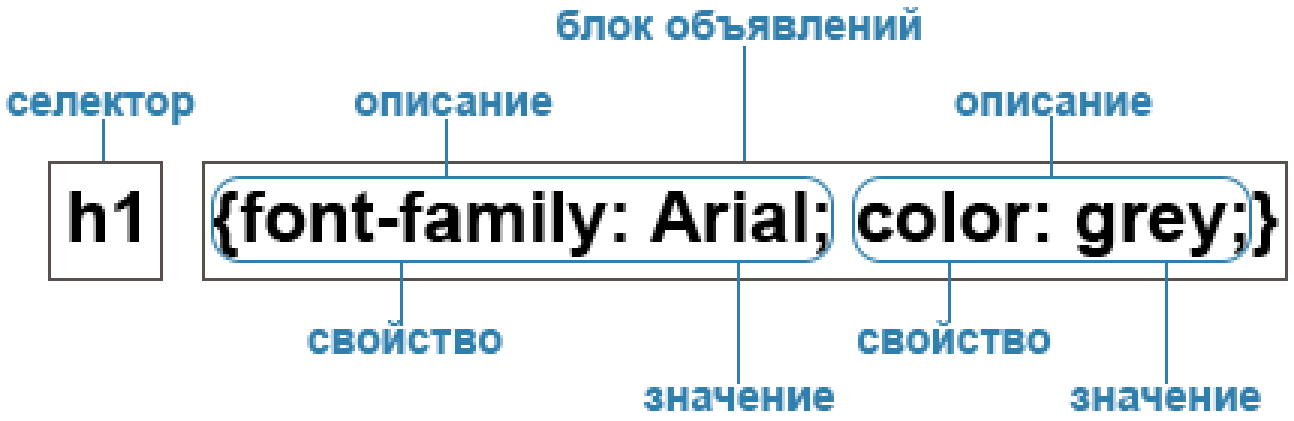
CSS (Cascading Style Sheets) — язык таблиц стилей, который позволяет прикреплять стиль к структурированным документам (например, документам HTML).

Отделяя стиль представления документов от содержимого документов, CSS упрощает создание веб-страниц и обслуживание сайтов.

Каскадные таблицы стилей описывают правила форматирования элементов с помощью свойств и допустимых значений этих свойств. Для каждого элемента можно использовать ограниченный набор свойств, остальные свойства не будут оказывать на него никакого влияния.

CSS (Cascading Style Sheets)

Объявление стиля состоит из двух частей: селектора и объявления.



Виды таблиц стилей

- Внешняя таблица стилей
- Внутренние стили
- Встроенные стили
- Правило @import

Встроенные стили

Когда мы пишем встроенные стили, мы пишем CSS-код в HTML-файл, непосредственно внутри элемента с помощью атрибута style

```
<p style="color: red;">
   Обратите внимание на этот текст.
</p>
```

Задание цвета

Существуют несколько основных способов представления цветов:

- В виде #123ABC. Представление в виде трёх пар шестнадцатеричных цифр, где каждая пара отвечает за свой цвет:
  - Две первые цифры — красный
  - Две в середине — зелёный
  - Две последние цифры — синий

  Возможно также краткое представление цвета в виде #ABC, что будет интерпретировано как #AABBCC.
- Представление ключевыми словами, например green, black. Во избежание случаев, когда указанное ключевое слово «не понимается» браузером, следует использовать лишь небольшой набор основных цветов, используемых во всех браузерах.
- В виде RGB(*,*,*), где «*» — числа от 0 до 255, обозначающих количество соответствующего цвета (красный, зелёный, синий) в получаемом цвете.
- Возможен и RGBA(*,*,*,*), где первые 3 «*» — компоненты цвета, задающиеся в диапазоне 0 до 255, а последняя «*» — уровень непрозрачности (альфа-канал), задающийся рациональными числами от 0 до 1.

Примеры цветов: https://www.w3schools.com/colors/colors_picker.asp

Внешняя таблица стилей

Внешняя таблица стилей представляет собой текстовый файл с расширением .css, в котором находится набор CSS-стилей элементов. Файл создаётся в редакторе кода, так же как и HTML-страница. Внутри файла могут содержатся только стили, без HTML-разметки. Внешняя таблица стилей подключается к веб-странице с помощью элемента `<link>`, расположенного внутри раздела `<head></head>`. Такие стили работают для всех страниц сайта.

```
<head>

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="css/assets.css" media="all">

</head>
```

Внутренние стили

Внутренние стили встраиваются в раздел `<head></head>` HTML-документа и определяются внутри элемента `<style></style>`. Внутренние стили имеют приоритет над внешними, но уступают встроенным стилям (заданным через атрибут `style`).

```
<head>
<style>
 h1,  h2 {
   color: red;
   font-family: "Times New Roman", Georgia, Serif;
   line-height: 1.3em;
  }
 </style>
</head>
<body>
...
</body>
```

Правило @import

Правило @import позволяет загружать внешние таблицы стилей. Чтобы директива @import работала, она должна располагаться в таблице стилей (внешней или внутренней) перед всеми остальными правилами:

```
<style>
 @import url(mobile.css);
 p {
  font-size: 0.9em;
  color: grey;
 }
</style>
```

Виды селекторов

Селекторы представляют структуру веб-страницы. С их помощью создаются правила для форматирования элементов веб-страницы. Селекторами могут быть элементы, их классы и идентификаторы, а также псевдоклассы и псевдоэлементы.

- **Универсальный селектор**
- **Селектор элемента**
- **Селектор класса**
- **Селектор идентификатора**
- **Селектор потомка**
- Дочерний селектор
- Сестринский селектор
- Селектор атрибута
- Селектор псевдокласса
- Селектор структурных псевдоклассов
- Селектор структурных псевдоклассов типа
- Селектор псевдоэлемента

Селектор элемента

Селекторы элементов позволяют форматировать все элементы данного типа на всех страницах сайта.

```
h1 {
  font-family: Lobster, cursive;
  color: red;
}
p {
 color: grey;
 border: solid;
}
```

Селектор класса

Селекторы класса позволяют задавать стили для одного и более элементов с одинаковым именем класса, размещенных в разных местах страницы или на разных страницах сайта.

```
<h1 class="headline">Тест</h1>
```

```
h1.headline {
  text-transform: uppercase;
  color: blue;
}
```

Универсальный селектор

Универсальный селектор обозначается звёздочкой (*). Соответствует любому тегу.

Убирая звёздочки с простых селекторов достигается тот же эффект. Например, * .warning и .warning считаются равными.

```
* {margin: 0; }

*.warning {color:red;}
*#maincontent {border: 1px solid blue;}
*[lang^=en]{color:green;}
```

Селектор класса и универсальный селектор

Если элемент имеет несколько атрибутов класса, их значения объединяются с пробелами.

```
<h1 class="headline rad">Тест</h1>
```

```
.headline {
  text-transform: uppercase;
  color: blue;
}
.rad {
  border: 1px solid blue;
}
```

Селектор идентификатора

Селектор идентификатора позволяет форматировать один конкретный элемент. Значение id должно быть уникальным, на одной странице может встречаться только один раз и должно содержать хотя бы один символ. Значение не должно содержать пробелов.

Нет никаких других ограничений на то, какую форму может принимать id, в частности, идентификаторы могут состоять только из цифр, начинаться с цифры, начинаться с подчеркивания, состоять только из знаков препинания и т. д.

Уникальный идентификатор элемента может использоваться для различных целей, в частности, как способ ссылки на конкретные части документа с использованием идентификаторов фрагментов, как способ нацеливания на элемент при создании сценариев и как способ стилизации конкретного элемента из CSS.

Селектор идентификатора

```
<div id="main">


</div>
```

```
#main {
border: 1px solid red;
}
```

Селектор потомка

Селекторы потомков применяют стили к элементам, расположенным внутри элемента-контейнера. Например, `ul li {color: green;}` — выберет все элементы `li`, являющиеся потомками всех элементов `ul`.

Если нужно отформатировать потомки определенного элемента, этому элементу нужно задать стилевой класс:

`p.grass a {color: green;}` — данный стиль применится ко всем ссылкам потомкам абзаца с классом grass;

`p .grass a {color: green;}` — если добавить пробел, то будут стилизованы ссылки, расположенные внутри любого элемента класса .grass, который является потомком элемента `<p>`;

`.grass a {color: green;}` — данный стиль применится к любой ссылке, расположенной внутри другого элемента, обозначенного классом .grass.

Группировка селекторов

Один и тот же стиль можно одновременно применить к нескольким элементам. Для этого необходимо в левой части объявления перечислить через запятую нужные селекторы:

```
h1, h2, p, span {
  color: blue;
  background-color: white;
}
```

Комбинация селекторов

Для более точного отбора элементов для форматирования можно использовать комбинации селекторов:

`#main-article p` — выберет все абзацы-потомки элемента с идентификатором p

`a[href][title]` — выберет все ссылки, для которых заданы атрибуты href и title;

`img[alt*="css"]:nth-of-type(even)` — выберет все четные картинки, альтернативный текст которых содержит слово css.

Наследование и каскадирование

- Наследование – это механизм, когда определенные в контейнере свойства, автоматически назначаются вложенным в этот контейнер элементам.
- Каскадность – это механизм CSS, который определяет какие стили в итоге будут применены к элементу и как конфликтующие правила переопределяют друг друга.

Наследование

Наследование является механизмом, с помощью которого определенные свойства передаются от предка к его потомкам. Спецификацией CSS предусмотрено наследование свойств, относящихся к **текстовому содержимому страницы**, таких как `color, font, text-alig и др.` Во многих случаях это удобно, так как не нужно задавать размер шрифта и семейство шрифтов для каждого элемента веб-страницы.

Свойства, относящиеся к **форматированию блоков**, не наследуются. Это `background, border, height и width, margin и др.`

Наследование

Принудительное наследование

- С помощью ключевого слова `inherit` можно принудить элемент наследовать любое значение свойства родительского элемента. Это работает даже для тех свойств, которые не наследуются по умолчанию.

Как задаются и работают CSS-стили

- Стили могут наследоваться от родительского элемента (наследуемые свойства или с помощью значения `inherit`).
- **Стили, расположенные в таблице стилей ниже, отменяют стили, расположенные в таблице выше.**

Каскадирование

Каскадирование — это механизм, который управляет конечным результатом в ситуации, когда к одному элементу применяются разные CSS-правила. Существует три критерия, которые определяют порядок применения свойств — **правило !important, специфичность и порядок**, в котором подключены таблицы стилей.

Правило `!important`

Вес правила можно задать с помощью ключевого слова `!important`, которое добавляется сразу после значения свойства, например,

```
span {font-weight: bold!important;}.
```

Правило необходимо размещать в конец объявления перед закрывающей скобкой, без пробела. Такое объявление будет иметь приоритет над всеми остальными правилами. Это правило позволяет отменить значение свойства и установить новое для элемента из группы элементов в случае, когда нет прямого доступа к файлу со стилями.

Порядок подключённых таблиц

Вы можете создать несколько внешних таблиц стилей и подключить их к одной веб-странице. Если в разных таблицах будут встречаться разные значения свойств одного элемента, то в результате к элементу применится правило, находящееся в таблице стилей, идущей в списке ниже.

Специфичность

Для каждого правила браузер вычисляет специфичность селектора, и если у элемента имеются конфликтующие объявления свойств, во внимание принимается правило, имеющее наибольшую специфичность. Значение специфичности состоит из четырех частей: 0, 0, 0, 0.

Специфичность селектора определяется следующим образом:

- для встроенного стиля, добавленного непосредственно к элементу — 1, 0, 0, 0;
- для id добавляется 0, 1, 0, 0;
- для class добавляется 0, 0, 1, 0;
- для каждого элемента и псевдоэлемента добавляется 0, 0, 0, 1;
- универсальный селектор не имеет специфичности.

В результате к элементу применятся те правила, специфичность которых больше.

Например, если на элемент действуют две специфичности со значениями 0, 0, 0, 2 и 0, 1, 0, 1, то выиграет второе правило.

Специфичность

```
h1 {color: lightblue;} /*специфичность 0, 0, 0, 1*/

em {color: silver;} /*специфичность 0, 0, 0, 1*/

h1 em {color: gold;} /*специфичность: 0, 0, 0, 1 + 0, 0, 0, 1 = 0, 0, 0, 2*/

div#main p.about {color: blue;} /*специфичность: 0, 0, 0, 1 + 0, 1, 0, 0 + 0, 0, 0, 1 + 0, 0, 1, 0 = 0, 1, 1, 2*/

.sidebar {color: grey;} /*специфичность 0, 0, 1, 0*/

#sidebar {color: orange;} /*специфичность 0, 1, 0, 0*/

li#sidebar  {color: aqua;} /*специфичность: 0, 0, 0, 1 + 0, 1, 0, 0 = 0, 1, 0, 1*/
```

Блочные и строчные элементы

Чтобы создать дерево блоков, CSS сначала использует каскадирование и наследование, позволяющие назначить вычисленное значение для каждого css-свойства каждому элементу и текстовому узлу в исходном дереве. Затем для каждого элемента CSS генерирует ноль или более блоков в соответствии со значением свойства `display` этого элемента. Как правило, элемент генерирует один основной блок, который представляет самого себя и содержит свое содержимое.

Положение блоков на странице определяется следующими факторами:

- размером элемента (с учётом того, заданы они явно или нет);
- типом элемента (строчный или блочный);
- схемой позиционирования (нормальный поток, позиционированные или плавающие элементы);
- отношениями между элементами в DOM (родительский — дочерний элемент);
- внутренними размерами содержащихся изображений;
- внешней информацией (например, размеры окна браузера).

Блочные элементы и блочные контейнеры

Блочные элементы — элементы высшего уровня, которые форматируются визуально как блоки, располагаясь на странице в окне браузера вертикально.

Значения свойства `display`, такие как `block`, `list-item` и `table` делают элементы блочными. Блочные элементы генерируют основной блок, который содержит только блок элемента.

Элементы со значением `display: list-item` генерируют дополнительные блоки для маркеров, которые позиционируются относительно основного блока.

Блочные элементы могут размещаться непосредственно внутри элемента `<body>`. Они создают разрыв строки перед элементом и после него, образуя прямоугольную область, по ширине занимающую всю ширину веб-страницы или блока-родителя.

Примеры: `<address>`, `<article>`, `<aside>`, `<blockquote>`, `<dd>`, `<div>`, `<dl>`, `<dt>`, `<details>`, `<fieldset>`, `<figcaption>`, `<figure>`, `<footer>`, `<form>`, `<h1>-<h6>`, `<header>`, `<li>`, `<nav>`, `<ol>`, `<output>`, `<p>`, `<section>`, `<summary>`, `<table>`, `<ul>` и др.

Строчные элементы и строчные контейнеры

Встроенные (строчные) элементы генерируют внутристрочные контейнеры. Они не формируют новые блоки контента. Значения свойства `display`, такие как `inline` и `inline-table` делают элементы строчными.

Примеры: `<a>`, `<cite>`, `<code>`, `<dfn>`, `<del>`, `<em>`, `<i>`, `<img>`, `<ins>`, `<kbd>`, `<label>`, `<map>`, `<mark>`, `<s>`, `<samp>`, `<small>`, `<span>`, `<strong>`, `<q>` и др.

Строчно-блочные элементы

Такие элементы являются встроенными, но для них можно задавать ширину и высоту.

```
{display: inline-block;}
```

Примеры: `<audio>`, `<button>`, `<canvas>`, `<input>`, `<meter>`, `<progress>`, `<select>`, `<textarea>`, `<video>` и др.

CSS блочная модель

Модуль CSS Box Model описывает свойства padding и margin, которые создают поля внутри и отступы снаружи CSS блока. Размеры блока также могут быть увеличены за счет рамки.

Каждый блок имеет прямоугольную область содержимого в центре, поля вокруг содержимого, рамку вокруг полей и отступ за пределами рамки. Размеры этих областей определяют свойства padding и его подсвойства — padding-left, padding-top и т.д., border и его подсвойства, margin и его подсвойства.

Физические свойства отступов: свойства margin-top, margin-right, margin-bottom, margin-left

Значения	Описание
длина	Размер отступа задается в единицах длины, например, px, in, em. Значение по умолчанию 0.
%	Вычисляется относительно ширины блока контейнера. Изменяются, если изменяется ширина родительского элемента.
auto	Для элементов уровня строки, плавающих (float) значения margin-left или margin-right вычисляются в 0. Если для элементов уровня блока задано margin-left: auto или margin-right: auto — соответствующее поле расширяется до края содержащего блока, если оба — их значения становятся равными, что горизонтально центрирует элемент относительно краев содержащего блока.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

```
margin-top: 20px;
margin-right: 1em;
margin-bottom: 5%;
margin-left: auto;
margin-top: inherit;
margin-right: initial;
```

Краткая запись отступов: свойство margin

Свойство margin является сокращенным свойством для установки `margin-top, margin-right, margin-bottom и margin-left` в одном объявлении.

Если существует только одно значение, оно применяется ко всем сторонам.

Если два — верхний и нижний отступы устанавливаются на первое значение, а правый и левый — устанавливаются на второе.

Если имеется три значения — верхний отступ устанавливается на первое значение, левый и правый — на второе, а нижний — на третье.

Если есть четыре значения — они применяются сверху, справа, снизу и слева соответственно.

```
p {margin: 20px auto; }
```

Поля элемента

Область полей представляет собой пространство между краем области содержимого и рамкой элемента. Свойства полей определяют толщину их области. Применяются ко всем элементам, кроме внутренних элементов таблицы (за исключением ячеек таблицы). Сокращенное свойство padding задает поля для всех четырех сторон, а подсвойства устанавливают только их соответствующие стороны. Краткая запись полей работает аналогично отступам.

длина	Поля элемента задаются при помощи единиц длины, например, px, pt, cm. Значение по умолчанию 0.
%	Вычисляются относительно ширины родительского элемента, могут меняться при изменении ширины элемента. Поля сверху и снизу равны полям слева и справа, т.е. верхние и нижние поля тоже вычисляются относительно ширины элемента.

```
padding-top: 0.5em;
padding-right: 0;
padding-bottom: 2cm;
padding-left: 10%;
padding-top: inherit;
padding-bottom: initial;
```

Ширина и высота

Свойство width определяет ширину содержимого блока. Это свойство не применяется к незамещаемым строчным элементам display: inline;. Ширина содержимого встроенных блоков определяется шириной отображаемого содержимого внутри них. Встроенные блоки сливаются в линейные блоки. Отрицательные значения не допускаются. Свойства min-width и max-width позволяют ограничивать ширину содержимого до определенного диапазона. Значения не могут быть отрицательными. Для min-width значение по умолчанию 0, для max-width — none.

Свойство height определяет высоту содержимого блока. Это свойство не применяется к незамещаемым строчным элементам. Значения длины не могут быть отрицательными.

Иногда полезно ограничить высоту элементов определенным диапазоном. Свойства min-height и max-height предлагают эту функциональность.

```
width: 100px;
width: 10em;
width: 50%;
width: auto;
width: inherit;
```

```
min-width: 100px;
min-width: 10em;
min-width: 50%;
min-width: inherit;
max-width: 500px;
max-width: 20em;
max-width: 80%;
max-width: none;
max-width: inherit;
```

```
height: 100px;
height: 10em;
height: 50%;
height: auto;
height: inherit;
```

```
min-height: 100px;
min-height: 2em;
min-height: 50%;
min-height: inherit;
max-height: 500px;
max-height: 20em;
max-height: 80%;
max-height: none;
max-height: inherit;
```

CSS-рамка

CSS-рамка элемента представляет собой одну или несколько линий, окружающих содержимое элемента и его поля padding. Рамка задаётся с помощью краткого свойства border. Стиль рамки задается с помощью трех свойств: стиль, цвет и ширина.

По умолчанию рамки всегда отрисовываются поверх фона элемента, фон распространяется до внешнего края элемента. Стиль рамки определяет ее отображение, без этого свойства рамки не будут видны вообще. Для элемента можно задавать рамку для всех сторон одновременно с помощью свойства `border-style` или для каждой стороны отдельно с помощью уточняющих свойств `border-top-style` и т.д. Не наследуется.

```
p {border-style: solid; }
p {border-top-style: solid;}
```

CSS-рамка

border-style: none;

border-style: hidden;

border-style: solid;

border-style: dotted;

border-style: dashed;

border-style: double;

border-style: groove;

border-style: ridge;

border-style: inset;

border-style: outset;

border-style: solid dotted dashed double;

CSS-рамка

border-color (border-top-color, border-right-color, border-bottom-color, border-left-color)	
transparent	Устанавливает прозрачный цвет для рамки. При этом ширина рамки остается. Можно использовать для смены цвета рамки при наведении курсора мыши на элемент, чтобы избежать смещение элемента. `p {border-color: transparent; }`
цвет	Цвет рамок задается при помощи значений свойства color. `p {border-color: #cacd58;}`

border-width (border-top-width, border-right-width, border-bottom-width, border-left-width)	
thin / medium / thick	Ключевые слова, устанавливают ширину рамки относительно друг друга. Первое значение уже, чем второе, второе — тоньше третьего. Значение по умолчанию — medium
Размер в px, em	{border-width: 5px;}

Свойство border позволяет объединить в себе следующие свойства: border-width, border-style, border-color, например:

```
p {
border: 2px solid grey;
}
```

CSS3-рамка

CSS3-рамка дополняет возможности форматирования границ элементов с помощью свойств, позволяющих закруглить углы элемента, а также использовать изображения для оформления границ элемента.

Закругление углов с помощью border-radius

Свойство позволяет закруглить углы строчных и блочных элементов. Кривая для каждого угла определяется с помощью одного или двух радиусов, определяющих его форму — круга или эллипса.

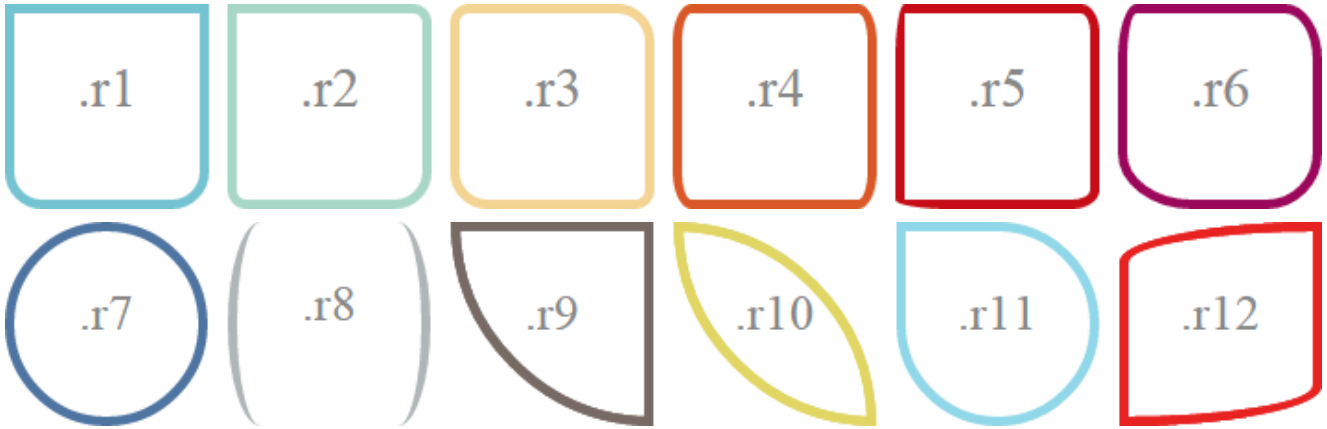
Свойство `border-radius` позволяет закруглить все `border-top-left-radius`, `border-top-right-radius`, `border-bottom-right-radius`, `border-bottom-left-radius` углы одновременно, а с помощью этих свойств можно закруглить каждый угол отдельно.

Если задать два значения для свойства `border-radius`, то первое значение закруглит верхний левый и нижний правый угол, а второе — верхний правый и нижний левый.

Значения, заданные через /, определяют горизонтальные и вертикальные радиусы.

Закругление углов с помощью border-radius

```
div {width: 100px; height: 100px; border: 5px red solid;}
.r1 {border-radius: 0 0 20px 20px;}
.r2 {border-radius: 0 10px 20px;}
.r3 {border-radius: 10px 20px;}
.r4 {border-radius: 10px/20px;}
.r5 {border-radius: 5px 10px 15px 30px/30px 15px 10px 5px;}
.r6 {border-radius: 10px 20px 30px 40px/30px;}
.r7 {border-radius: 50%;}
.r8 {border-top: none; border-bottom: none; border-radius: 30px/90px;}
.r9 {border-bottom-left-radius: 100px;}
.r10 {border-radius: 0 100%;}
.r11 {border-radius: 0 50% 50% 50%;}
.r12 {border-top-left-radius: 100% 20px; border-bottom-right-radius: 100% 20px;}
```



Свойство box-sizing

По умолчанию в блочной модели CSS ширина и высота, которая задаётся элементу, применяется только для контента элемента. Если у элемента есть рамка или поле, то они добавляются к ширине и высоте, чтобы получить отображаемый на экране размер. Например, если у вас есть четыре блока с width: 25%; , и у какого-нибудь из них есть рамка или поле слева или справа, то по умолчанию они не поместятся на одной строке.

Свойство box-sizing может изменять это поведение:

- `content-box` даёт стандартное поведение свойства box-sizing. Если вы выставите элементу ширину 100 пикселей, то ширина его контента будет 100 пикселей, а ширина рамки и внутренних рллей при рендере будет добавлена к финальной ширине, делая элемент шире ста пикселей.
- `border-box` говорит браузеру учитывать любые границы и внутренние отступы в значениях, которые вы указываете в ширине и высоте элемента. Если вы выставите элементу ширину 100 пикселей, то эти 100 пикселей будут включать в себя рамку и внутренние поля, а контент сожмётся, чтобы выделить для них место. Обычно это упрощает работу с размерами элементов.

CSS-фон

Свойство `background` позволяет установить фоновое изображение, цвет фона и другие свойства стиля для заданного элемента.

`background` также объединяет в себе несколько других свойств:

- `background-color` — задаёт цвет фона элемента;
- `background-image` — устанавливает фоновое изображение для элемента;
- `background-position` — определяет начальную позицию фонового изображения;
- `background-repeat` — указывает, должно ли фоновое изображение повторяться;
- `background-size` — устанавливает размер фонового изображения;
- `background-attachment` — определяет, будет ли фоновое изображение прокручиваться вместе с содержимым элемента.

Цвет фона: свойство background-color

Каждый блок html-элемента имеет фоновый слой, который может быть полностью прозрачным (по умолчанию) или заполнен цветом и/или одним или несколькими изображениями.

Свойства фона не наследуются, но фон родительского блока будет просвечивать по умолчанию из-за начального значения в `background-color: transparent.`

Фон не отображается у пустых элементов с нулевой высотой. Отрицательные значения свойства `margin` не влияют на фон элемента.

```
background-color: salmon;
background-color: #00ff00;
background-color: rgba(255, 128, 128, 0.5);
background-color: transparent;
```

Источник изображения: свойство background-image

Свойство `background-image` устанавливает фоновое изображение (одно или несколько) элемента. Значение none считается слоем изображения, но ничего не рисует. Изображение, которое является пустым (нулевой ширины или нулевой высоты), которое не загружается или не может быть отображено (например, потому что оно не в поддерживаемом формате изображения) также считается слоем, но ничего не рисует.

```
background-image: none;
background-image: url(http://site.ru/rose.png);
background-image: url(tl.png), url(tr.png);
background-image: linear-gradient(white, gray);
background-image: repeating-radial-gradient(circle closest-side at 20px 30px,
red, yellow, green 100%, yellow 150%, red 200%);
background-image: image("sprites.svg#xywh=40,0,20,20");
background-image: inherit;
```

Укладка изображений: свойство background-position

У свойства background-position два значения, положение по горизонтали (может быть — left, center, right) и вертикали (может быть — top, center, bottom). Кроме использования ключевых слов положение также можно задавать в процентах, пикселях или других единицах. Если применяются ключевые слова, то порядок их следования не имеет значения, при процентной записи вначале задаётся положение рисунка по горизонтали, а затем, через пробел, положение по вертикали.

```
background-position: center bottom;
background-position: center;
background-position: 25% 75%; // 100% 100% и bottom right равнозначны
```

Укладка изображений: свойство background-repeat

Свойство `background-repeat` определяет, как фоновые изображения укладываются в области фона после того, как для них установлены размеры и позиционирование. Если значение свойства имеет два ключевых слова, первое используется для горизонтального направления, второе — для вертикального.

- no-repeat - устанавливает одно фоновое изображение в элементе без его повторений, положение которого определяется свойством background-position (по умолчанию в левом верхнем углу). Аналогично no-repeat no-repeat.
- repeat - фоновое изображение повторяется по горизонтали и вертикали. Аналогично repeat repeat.
- repeat-x - фоновый рисунок повторяется только по горизонтали. Аналогично repeat no-repeat.
- repeat-y - фоновый рисунок повторяется только по вертикали. Аналогично no-repeat repeat.
- space - изображение повторяется столько раз, чтобы полностью заполнить область; если это не удаётся, между картинками добавляется пустое пространство.
- round - изображение повторяется так, чтобы в области поместилось целое число рисунков; если это не удаётся сделать, то фоновые рисунки масштабируются.

Размер фонового изображения: background-size

Свойство background-size позволяет изменять размер фонового изображения. Если фоновая картинка не совпадает по размеру с размерами блока, то при помощи этого свойства можно сделать так, чтобы она, картинка занимала всю площадь блока или, наоборот, была определённого размера.

Значением свойства служат любые единицы для размеров, либо ключевые слова auto, cover или contain.

- auto — фон будет иметь натуральный размер, такой, как реальный размер картинки фона. Если же auto задано только для одной стороны, то по этой стороне фон будет масштабироваться так, чтобы иметь неискаженные пропорции.
- cover — масштабирует картинку так, чтобы она накрыла собой весь блок с сохранением пропорций. Картинка будет стараться поместиться целиком, но это не всегда будет получаться, поэтому какая-то ее часть будет обрезаться. Блок всегда будет покрыт картинкой целиком.
- contain — масштабирует картинку так, чтобы она целиком влезла в блок с сохранением пропорций. При этом она может занять или всю ширину, или всю высоту (зависит от пропорций картинки и от размеров элемента). Блок в общем случае будет покрыт картинкой не целиком (зато картинка всегда будет видна вся, хоть и в уменьшенном варианте).

Прокрутка фона: background-attachment

Свойство background-attachment задает каким образом прокручивать фоновую картинку элемента: вместе с текстом или текст будет скользить по картинке.

- fixed – картинка фона будет неподвижной, а текст будет скользить по ней.
- scroll – картинка фона будет прокручиваться вместе с текстом.
- local – фон фиксируется с учетом поведения элемента. Если элемент имеет прокрутку, то фон будет прокручиваться вместе с содержимым, но фон выходящий за рамки элемента остается на месте.

Свойство background

Свойство background является сокращением, которое устанавливает следующие свойства в одном объявлении: background-clip, background-color, background-image, background-origin, background-position, background-repeat, background-size, и background-attachment.

```
background: green;

background: url("test.jpg") repeat-y;

/* Одно изображение, центрированное и масштабированное */
background: no-repeat center/80% url("../img/image.png");
```

CSS-градиент

CSS-градиент представляет собой переходы от одного цвета к другому.

Градиенты создаются с помощью функций linear-gradient() и radial-gradient().

Градиентный фон можно устанавливать в свойствах background, background-image, border-image и list-style-image.

Линейный градиент linear-gradient()

Линейный градиент создается с помощью двух и более цветов, для которых задано направление, или **линия градиента**. Если направление не указано, используется значение по умолчанию — **сверху-вниз**.
Цвета градиента по умолчанию распределяются равномерно в направлении, перпендикулярном линии градиента.

Направление градиента может быть задано двумя способами:

- с помощью угла наклона в градусах deg, значение которого определяет угол наклона линии внутри элемента.
- с помощью ключевых слов to top, to right, to bottom, to left, которые соответствуют углу градиента, равному 0deg, 90deg, 180deg и 270deg соответственно.

```
{background: linear-gradient(угол/сторона или угол наклона с помощью ключевого слова (пары ключевых слов), первый цвет, второй цвет и т.д.);}
```

Точки остановки цвета

Через пробел после значения цвета можно задавать начальную и конечную точки цвета. Можно использовать любую единицу длины либо проценты. В начальной точке заканчивается плавный переход из предыдущего цвета. Конечная точка — это точка, с которой начинается плавный переход в следующий цвет.

Линейный градиент linear-gradient()

```
background: linear-gradient(45deg, #EECFBA, #C5DDE8);

background: linear-gradient(to right, #F6EFD2, #CEAD78);

background: linear-gradient(to right, #FFDDD6 20%, #FFF9ED 20% 80%, #DBDBDB
80%);
```

Радиальный градиент radial-gradient()

Радиальный градиент отличается от линейного тем, что цвета выходят из одной точки (центра градиента) и равномерно распределяются наружу, рисуя форму круга или эллипса.

Форма градиента определяется ключевыми словами `circle` или `ellipse`. Если форма не задана, по умолчанию радиальный градиент принимает форму эллипса.

Позиция центра задаётся с помощью ключевых слов, используемых в `свойстве background-position`, с добавлением приставки `at`. Если позиция центра не задана, используется значение по умолчанию `at center`.

С помощью пары значений, указанных в единицах длины %, em или px, можно управлять размером эллипсообразного градиента. Первое значение задает ширину эллипса, второе — высоту.

Размер градиента задаётся с помощью ключевых слов. Значение по умолчанию `farthest-corner` (к дальнему углу).

```
background: radial-gradient(форма градиента/размер/позиция центра, первый цвет, второй цвет и т.д.);
```

closest-side	Размер градиента рассчитывается из расстояния до любой ближней стороны блока для circle или до ближних сторон по X и по У для ellipse.
farthest-side	Размер рассчитывается из расстояния до дальних сторон.
closest-corner	Размер рассчитывается из расстояния до ближних углов.
farthest-corner	Размер рассчитывается из расстояния до дальних углов.

Радиальный градиент radial-gradient()

```
div.c4 {
height: 200px;
background: radial-gradient(white, #FFA9A1);
}
div.c5 {
height: 200px;
background: radial-gradient(at top, #FEFFFF, #A7CECC);
}
div.c6 {
height: 200px;
background: radial-gradient(40% 50%, #FAECD5, #CAE4D8);
}
div.c7 {
height: 200px;
background: radial-gradient(circle farthest-corner at 100px 50px, #FBF2EB,
#352A3B);
}
```

CSS3-тень блока

Свойство box-shadow добавляет элементу одну или более теней. Тень представляет собой копию элемента, смещенную на указанное расстояние. Тени бывают внешние или внутренние, размытые или плоские, они могут следовать контурам блоков со скругленными углами. С помощью ключевого слова inset создаются тени внутри элемента, делая элемент визуально объёмным или вдавленным.

Свойство box-shadow прикрепляет одну или несколько теней к блоку. Свойство принимает либо значение none, которое указывает на отсутствие теней, либо список теней через запятую, упорядоченный от начала к концу.

```
box-shadow: none | <тень> [,<тень>]*

где <тень>: inset <сдвиг по x> <сдвиг по y> <размытие> <растяжение> <цвет>
```

- `inset` — тень выводится внутри элемента. Необязательный параметр.
- `сдвиг по x` — смещение тени по горизонтали относительно элемента. Положительное значение этого параметра задает сдвиг тени вправо, отрицательное — влево. **Обязательный параметр**.
- `сдвиг по y` — смещение тени по вертикали относительно элемента. Положительное значение задает сдвиг тени вниз, отрицательное — вверх. **Обязательный параметр.**
- `размытие` — задает радиус размытия тени. Чем больше это значение, тем сильнее тень сглаживается, становится шире и светлее. Если этот параметр не задан, по умолчанию устанавливается равным 0, тень при этом будет четкой, а не размытой.
- `растяжение` — положительное значение растягивает тень, отрицательное, наоборот, ее сжимает. Если этот параметр не задан, по умолчанию устанавливается 0, при этом тень будет того же размера, что и элемент.
- `цвет` — цвет тени в любом доступном CSS формате, по умолчанию тень черная. Необязательный параметр.

CSS3-тень блока. Пример

```
<p class="example-shadow-2"><span></span></p>
```

```
.example-shadow-2 {
  background: #e8e8e8;
  text-align: center;
}
.example-shadow-2 span {
  background: white;
  display: inline-block;
  width: 200px;
  height: 100px;
  margin: 50px;
  box-shadow: 0 14px 28px rgba(0,0,0,0.25), 0 10px 10px rgba(0,0,0,0.22);
}
```

CSS-текст

Модуль **CSS-текст** описывает функции CSS, которые управляют переводом исходного текста в форматированный и переносом строк. Различные свойства CSS обеспечивают контроль над преобразованием регистра, обработкой пробелов, правилами переноса и переносом текста и строк, выравниванием, интервалами и отступами.

Преобразование текста: свойство text-transform

Свойство text-transform стилизует текст. Оно не влияет на базовое содержимое и не должно влиять на содержимое операции копирования и вставки простого текста.

```
text-transform: capitalize | lowercase | uppercase | none | inherit
```

```
h1 {text-transform: uppercase; /* Заглавные буквы */
}
p {text-transform: capitalize; /* Каждое слово начинается с заглавной буквы */
}
```

Обработка пробелов и переносы строк: свойство white-space

Свойство `white-space` устанавливает, как отображать пробелы между словами. В обычных условиях любое количество пробелов в коде HTML показывается на веб-странице как один. Исключением является тег `<pre>`, помещенный в этот контейнер текст выводится со всеми пробелами, как он был отформатирован пользователем. Таким образом, `white-space` имитирует работу тега `<pre>`, но в отличие от него не меняет шрифт на моноширинный.

```
white-space: normal | nowrap | pre | pre-line | pre-wrap | inherit
```

Значение	Перенос текста	Пробелы
normal	Переносится	Не учитываются
nowrap	Не переносится	Не учитываются
pre	Не переносится	Учитываются
pre-line	Переносится	Не учитываются
pre-wrap	Переносится	Учитываются

Выравнивание: свойство text-align

Блок текста представляет собой набор линейных блоков. Свойство text-align задает свойства text-align-all и text-align-last и описывает, как блоки на уровне строки в каждом линейном блоке выравниваются относительно начальной и конечной сторон линейного блока.

start	Содержимое на уровне строки выравнивается по начальному краю линейного блока. Значение по умолчанию.
end	Содержимое на уровне строки выравнивается по конечному краю линейного блока.
left	Содержимое на уровне строки выравнивается по левому краю строки линейного блока. В вертикальных системах письменности это будет физический верх или низ, в зависимости от ориентации текста.
right	Содержимое на уровне строки выравнивается по правому краю строки линейного блока. В вертикальных системах письменности это будет физический верх или низ, в зависимости от ориентации текста.
center	Содержимое на уровне строки центрируется внутри линейного блока.
justify	Текст выравнивается по ширине линейного блока, чтобы точно заполнить поле строки, прижимаюсь к левому и правому краям родительского элемента. Если иное не указано в text-align-last, последняя строка перед принудительным разрывом или конец блока выравнивается по началу. Пробелы между словами и буквами распределяются таким образом, чтобы длина всех строк была равна. Разные браузеры могут увеличить как отступы между словами, так и интервалы между буквами.
justify-all	Устанавливает text-align-all и text-align-last в justify, также выравнивая последнюю строку.
match-parent	Значение ведет себя так же, как inherit за исключением того, что унаследованное значение start или end интерпретируется относительно значения direction (или исходного содержащего блока, если нет родителя) и приводит к вычисленному значению left или right.

Отступ первой строки: свойство text-indent

Свойство text-indent задает отступ, применяемый к строкам встроенного содержимого в блоке. Отступ обрабатывается как поле, примененное к начальному краю линейного блока.

Если в первой строке блочного элемента присутствует изображение, то оно сдвинется вместе с остальным текстом.

```
p.c1 { text-indent: 1.5em;
       text-align: justify;}
p.c2 {
       text-indent: 2em;}
```

CSS-шрифты

Шрифт в CSS — это ресурс, содержащий визуальное представление символов. На самом простом уровне он содержит информацию, которая сопоставляет коды символов с фигурами (называемые глифами), представляющие эти символы.

Шрифты, использующие общий стиль дизайна, обычно группируются в семейства шрифтов, классифицируемые набором стандартных свойств шрифта. Внутри семейства форма, отображаемая для данного символа, может варьироваться в зависимости от толщины обводки, наклона или относительной ширины.

Ресурсы шрифтов могут быть установлены локально на устройстве, в котором работает браузер. Для локальных ресурсов шрифта описательная информация может быть получена непосредственно из ресурса шрифта. Для загружаемых ресурсов шрифтов, также называемых веб-шрифтами, описательная информация включена со ссылкой на ресурс шрифта.

Используя различные шрифты для заголовков, абзацев и других элементов, можно задавать определенный стиль письменных сообщений, передавая желаемые эмоции и настроение.

Не рекомендуется использовать более двух шрифтов на странице, а желаемого контраста можно достигнуть за счет комбинирования шрифтов разной толщины, размера, начертания или же при помощи цвета.

Семейство шрифтов: свойство font-family

Свойство font-family используется для выбора начертания шрифта. Поскольку невозможно предсказать, установлен тот или иной шрифт на компьютере посетителя вашего сайта, рекомендуется прописывать все возможные варианты однотипных шрифтов. В таком случае браузер будет проверять их наличие, последовательно перебирая предложенные варианты.

```
font-family: "Times New Roman", Georgia, Serif;
font-family: serif;
font-family: sans-serif;
font-family: monospace;
font-family: cursive;
```

Шрифты Google Fonts (https://fonts.google.com/)

Насыщенность шрифта: свойство font-weight

normal	Значение по умолчанию, устанавливает нормальную насыщенность шрифта. Эквивалентно значению насыщенности, равной 400.
bold	Делает шрифт текста полужирным. Эквивалентно значению насыщенности, равной 700.
bolder	Насыщенность шрифта будет больше, чем у предка.
lighter	Насыщенность шрифта будет меньше, чем у предка.
100, 200, 300, 400, 500, 600, 700, 800, 900	Значение 100 соответствует самому легкому варианту начертания шрифта, а 900 — самому плотному. При этом, эти числа не определяют конкретной плотности, т.е. 100, 200, 300 и 400 могут соответствовать одному и тому же варианту слабой насыщенности начертания шрифта; 500 и 600 — средней насыщенности, а 700, 800 и 900 могут выводить одинаковое очень насыщенное начертание. Распределение плотности так же зависит от количества уровней насыщенности, определенных в конкретном семействе шрифтов.

Ширина шрифта: свойство font-stretch

Свойство font-stretch позволяет выбрать нормальное, сжатое или расширенное начертание символа из семейства шрифтов. Свойство не работает на любом шрифте, а только на шрифтах, для которых разработаны различными начертания, соответствующими определенным размерам.

ultra-condensed	Указывает на наиболее сжатый шрифт.
extra-condensed	Указывает на второй по сжатости шрифт.
condensed	Указывает на сжатый шрифт.
semi-condensed	Указывает на немного сжатый шрифт.
normal	Значение по умолчанию.
semi-expanded	Слегка расширенный шрифт.
expanded	Расширенный шрифт.
extra-expanded	Второй по расширенности шрифт.
ultra-expanded	Максимально расширенный шрифт.

Начертание шрифта: свойство font-style

Свойство font-style позволяет выбрать стиль начертания для шрифта. При этом разница между курсивом и наклонным начертанием заключается в том, что курсив вносит небольшие изменения в структуру каждого символа, в то время как наклонное начертание представляет собой наклонную версию прямого шрифта.

normal	Значение по умолчанию, устанавливает для текста обычное начертание шрифта.
italic	Выделяет текст курсивом.
oblique	Устанавливает наклонное начертание шрифта.

Размер шрифта: свойство font-size

Свойство font-size указывает желаемую высоту глифов из шрифта.

absolute-size	xx-small, x-small, small, medium, large, x-large, xx-large. В качестве стандартного размера принимается medium.
relative-size	smaller, larger. Относительные размеры обусловливают изменение размера шрифта элемента относительно родителя. При этом размер шрифта может выйти за рамки размеров, предполагаемых для xx-small и xx-large.
длина	Размер шрифта устанавливается с помощью положительных значений единиц длины, например, px, em, как целых, так и дробных.
%	Относительное значение, вычисляется на основании любого размера, унаследованного от родительского элемента. Обеспечивает более точную настройку вычисляемого размера шрифта. Задание размеров шрифта с помощью em эквивалентно процентному значению.

Сокращенная запись свойств шрифта: свойство font

Свойство font является сокращенным свойством для установки font-style, font-variant, font-weight, font-stretch, font-size/line-height, font-family.

```
font: 12pt/14pt sans-serif;
font: 80% sans-serif;
font: x-large/110% "new century schoolbook", serif;
font: bold italic large Palatino, serif;
font: normal small-caps 120%/120% fantasy;
font: condensed oblique 12pt "Helvetica Neue", serif;
```

CSS Text Decoration

CSS Text Decoration – модуль CSS, который определяет возможности, относящиеся к оформлению текста, такие как подчёркивание, тень текста и метки выделения.

Свойство **`text-decoration`** является сокращением и позволяет указать значения для свойств:

- `text-decoration-line` — положение декоративной линии;
- `text-decoration-style` — стиль декоративной линии;
- `text-decoration-color` — цвет декоративной линии.

Но, как правило, его используют только для указания положения линии **`text-decoration-line`**:

- `underline` — подчёркнутый текст.
- `line-through` — перечёркнутый текст.
- `overline` — надчёркнутый текст, линия находится над словами.
- `none` — отменяет все эффекты.

```
text-decoration-line: overline;
```

CSS Text Decoration

Свойство **text-decoration-style** определяет стиль линий, нарисованных для оформления текста, указанного в элементе. Значения имеют то же значение, что и для свойства `border-style`.

`solid` — сплошная линия. Значение по умолчанию.

`double` — двойная линия.

`dotted` — точечная линия.

`dashed` — пунктирная линия.

`wavy` — волнистая линия.

Цвет линии оформления: свойство **text-decoration-color**

```
text-decoration-color: #00ff00;
```

Свойство **text-decoration** является краткой формой записи свойств `text-decoration-line` `text-decoration-style` `text-decoration-color` в одном объявлении. Пропущенные значения устанавливаются на их начальные значения. Значение по умолчанию

```
text-decoration: none solid currentColor;
```

```
text-decoration: underline double #ff0000;
```

Тень текста: свойство text-shadow

Свойство `text-shadow` используется для добавления тени к тексту. Тень текста — интересный инструмент, который позволяет создавать удивительные эффекты. Тени могут быть однослойными или многослойными, размытыми, цветными или полупрозрачными. Задавая тень для элемента, можно указывать только одно значение длины и цвет, таким образом создавая цветную копию отдельного символа или слова. Также, с помощью тени можно сделать текст более читаемым, если контраст между цветом текста и фоном невелик.

Одновременно можно задавать несколько теней, указывая их через запятую. Тени накладываются друг на друга, но не перекрывают сам текст. Первая тень всегда расположена сверху над остальными тенями. Свойство наследуется.

```
text-shadow: x-offset y-offset blur цвет;
```

Тень текста: свойство text-shadow

```
.text1 {
text-shadow: 4px 3px 0px #fff, 7px 6px 0px rgba(0,0,0,0.15);    /* дает две
тени */
}
.text2{
color: #3CF;
text-shadow: -1px 0 1px white, 0 -1px 1px white, 0 1px 1px white, 1px 0 1px
white, 0 0 8px white, 0 0 8px white, 0 0 8px white, 2px 2px 3px black;
}
.text3{
text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;
}
.text4{
background: #91877b;
text-shadow: 0 1px 0 rgba(255, 255, 255, 0.4);
}
```

Прозрачность элементов: свойство opacity

Свойство `opacity` определяет уровень прозрачности элемента. `Opacity` - это противоположный эффект прозрачности. Это свойство дает возможность сделать элемент полностью или частично прозрачным.

Свойство принимает значения в диапазоне от 0 до 1. Значение по умолчанию равно 1, при котором элемент становится полностью непрозрачным. Значение 0 делает элемент полностью прозрачным.

Значения между 0 и 1 делают элемент полупрозрачным.

```
opacity: 0.5;
```

Справочники

- https://developer.mozilla.org/ru/docs/Web/CSS/Reference
- https://webref.ru/css
- https://html5css.ru/css/default.php
- https://www.w3schools.com/css/default.asp