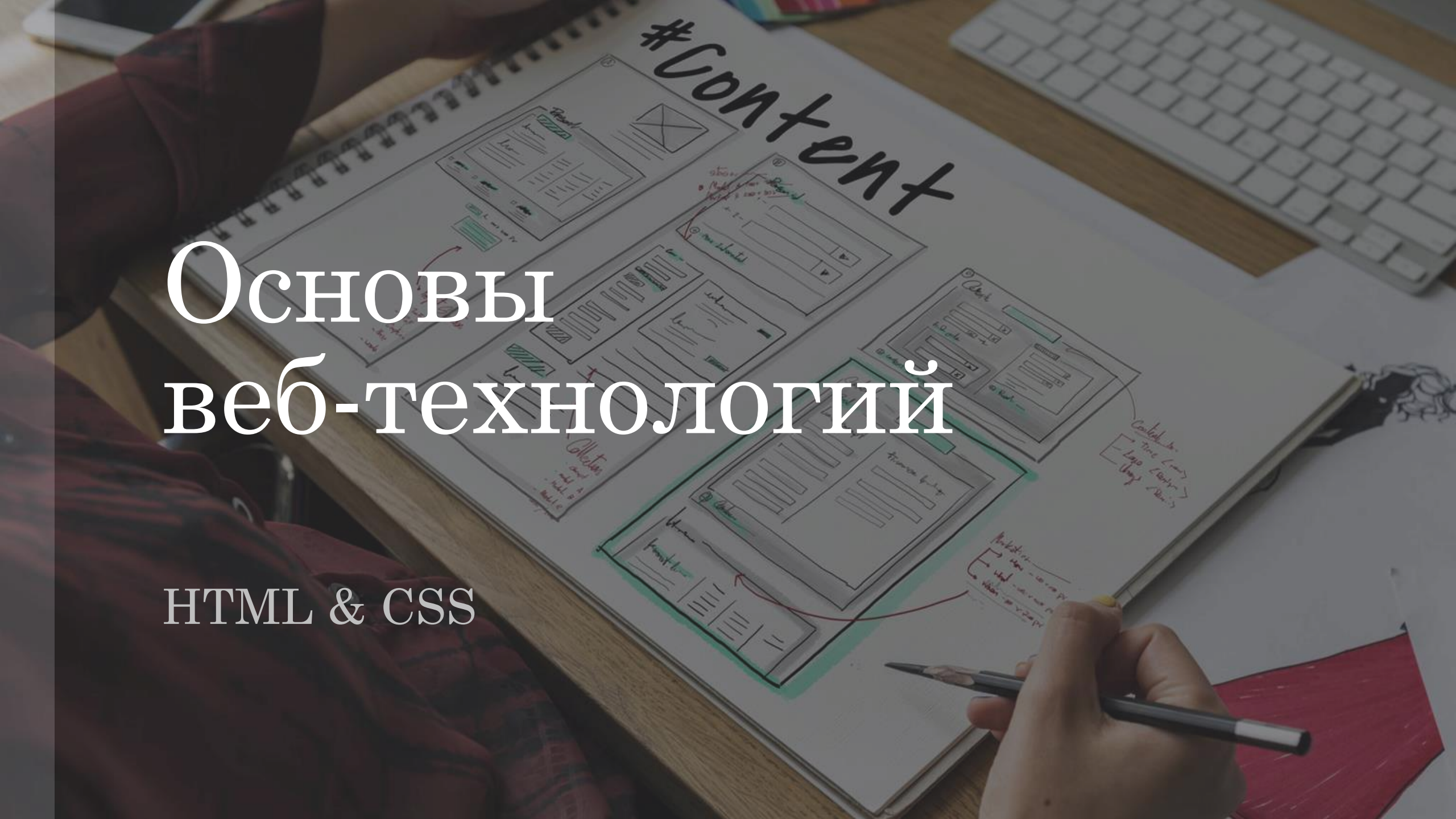


Основы веб-технологий

HTML & CSS

Виды селекторов (продолжение)

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**

Дочерний селектор

Дочерний элемент является прямым потомком содержащего его элемента. У одного элемента может быть несколько дочерних элементов, а родительский элемент у каждого элемента может быть только один. Дочерний селектор позволяет применить стили только если дочерний элемент идёт сразу за родительским элементом и между ними нет других элементов, то есть дочерний элемент больше ни во что не вложен.

Например, `p > strong` — выберет все элементы `strong`, являющиеся дочерними по отношению к элементу `p`.

Сестринский селектор

Сестринские отношения возникают между элементами, имеющими общего родителя. Селекторы сестринских элементов позволяют выбрать элементы из группы элементов одного уровня:

`h1 + p` — выберет все первые абзацы, идущие непосредственно за любым элементом `<h1>`, не затрагивая остальные абзацы;

`h1 ~ p` — выберет все абзацы, являющиеся сестринскими по отношению к любому заголовку `<h1>` и идущие сразу после него.

Селектор атрибута

Селекторы атрибутов выбирают элементы на основе имени атрибута или значения атрибута:

[атрибут] — все элементы, содержащие указанный атрибут, `[alt]` — все элементы, для которых задан атрибут alt;

селектор[атрибут] — элементы данного типа, содержащие указанный атрибут, `img[alt]` — только картинки, для которых задан атрибут alt;

селектор[атрибут="значение"] — элементы данного типа, содержащие указанный атрибут с конкретным значением, `img[title="flower"]` — все картинки, название которых содержит слово flower;

селектор[атрибут~="значение"] — элементы частично содержащие данное значение, например, если для элемента задано несколько классов через пробел, `p[class~="feature"]` — абзацы, имя класса которых содержит feature;

Селектор атрибута

селектор[атрибут^="значение"] — элементы, значение атрибута которых начинается с указанного значения, `a[href^="https://"]` — все ссылки, начинающиеся на https://;

селектор[атрибут$="значение"] — элементы, значение атрибута которых заканчивается указанным значением, `img[src$=".png"]` — все картинки в формате png;

селектор[атрибут*="значение"] — элементы, значение атрибута которых содержит в любом месте указанное слово, `a[href*="book"]` — все ссылки, название которых содержит book.

Селектор псевдокласса

Псевдоклассы — это классы, фактически не прикрепленные к HTML-элементам. Они позволяют применить CSS-правила к элементам при совершении события или подчиняющимся определенному правилу.

Псевдоклассы характеризуют элементы со следующими свойствами:

:link — не посещенная ссылка;

:visited — посещенная ссылка;

:hover — любой элемент, по которому проводят курсором мыши;

:focus — интерактивный элемент, к которому перешли с помощью клавиатуры или активировали посредством мыши;

:active — элемент, который был активизирован пользователем;

:lang() — элементы с текстом на указанном языке;

:target — элемент с символом #, на который ссылаются в документе;

Селектор псевдокласса (для форм)

:valid — поля формы, содержимое которых прошло проверку в браузере на соответствие указанному типу данных;

:invalid — поля формы, содержимое которых не соответствует указанному типу данных;

:enabled — все активные поля форм;

:disabled — заблокированные поля форм, т.е., находящиеся в неактивном состоянии;

:in-range — поля формы, значения которых находятся в заданном диапазоне;

:out-of-range — поля формы, значения которых не входят в установленный диапазон;

:not(селектор) — элементы, которые не содержат указанный селектор — класс, идентификатор, название или тип поля формы — :not([type="submit"]);

:checked — выделенные (выбранные пользователем) элементы формы.

Селектор структурных псевдоклассов

Структурные псевдоклассы отбирают дочерние элементы в соответствии с параметром, указанным в круглых скобках:

:nth-child(odd) — нечётные дочерние элементы;

:nth-child(even) — чётные дочерние элементы;

:nth-child(3n) — каждый третий элемент среди дочерних;

:nth-child(3n+2) — выбирает каждый третий элемент, начиная со второго дочернего элемента (+2);

:nth-child(n+2) — выбирает все элементы, начиная со второго;

:nth-child(3) — выбирает третий дочерний элемент;

Селектор структурных псевдоклассов

:nth-last-child() — в списке дочерних элементов выбирает элемент с указанным местоположением, аналогично с :nth-child(), но начиная с последнего, в обратную сторону;

:first-child — позволяет оформить только самый первый дочерний элемент;

:last-child — позволяет форматировать последний дочерний элемент;

:only-child — выбирает элемент, являющийся единственным дочерним элементом;

:empty — выбирает элементы, у которых нет дочерних элементов;

:root — выбирает элемент, являющийся корневым в документе — элемент html.

Селектор структурных псевдоклассов типа

Указывают на конкретный тип дочернего элемента:

:nth-of-type() — выбирает элементы по аналогии с :nth-child(), при этом берёт во внимание только тип элемента;

:first-of-type — выбирает первый дочерний элемент данного типа;

:last-of-type — выбирает последний элемент данного типа;

:nth-last-of-type() — выбирает элемент заданного типа в списке элементов в соответствии с указанным местоположением, начиная с конца;

:only-of-type — выбирает единственный элемент указанного типа среди дочерних элементов родительского элемента.

```
p:first-of-type {
  font-size: 1.25em;
}
div:nth-of-type(even) {
  background: red;
}
```

Селектор псевдоэлемента

Псевдоэлементы отличаются от псевдоклассов тем, что они не реагируют на состояние платформы, а действуют так, как если бы они вставляли новый элемент с помощью CSS. Кроме того, синтаксис псевдоэлементов отличается от синтаксиса псевдоклассов, потому что вместо одинарного двоеточия (:) в них используется двойное двоеточие (::).

Псевдоэлементы используются для добавления содержимого, которое генерируется с помощью свойства content:

::before — вставляет генерируемое содержимое перед элементом;

::after — добавляет генерируемое содержимое после элемента.

```
.my-element::before {
   content: 'Prefix - ';
}
```

Селектор псевдоэлемента

Перечень функций псевдоэлементов не ограничен вставкой содержимого. Их также можно использовать для нацеливания на определенные части элемента. Предположим, у вас есть список. С помощью псевдоэлемента `::marker` можно применить стиль к каждому маркеру (или номеру) в списке.

```
/* Теперь в списке будут либо красные точки, либо красные номера */
li::marker {
  color: red;
}
```

Кроме того, с помощью псевдоэлемента ::selection можно применять стили к содержимому, выделенному пользователем.

```
::selection {
  background: black;
  color: red;
}
```

CSS-ссылки

CSS-ссылки содержат свойства, которые отвечают за внешний вид гипертекстовых ссылок HTML-документа. Ссылки представляют собой основной способ навигации по сайту, поэтому применение CSS-стилей для оформления улучшит их визуальное восприятие.

Основной способ оформления ссылок заключается в стилизации подчеркивания ссылки и изменении цвета текста ссылки. Также можно изменить внешний вид курсора с помощью свойства `cursor`.

Псевдоклассы состояний

Большинство браузеров выделяют четыре основных состояния гиперссылок, каждому из которых соответствует свой псевдокласс селектора:

- Непосещенная — `a:link`
- Посещенная — по которой уже выполнялся переход — `a:visited`
- Сфокусированная — a:focus
- Не нажатая — над которой находится указатель мыши — `a:hover`
- Нажатая — которая удерживается мышью — `a:active`

Форматировать ссылки нужно в указанной последовательности, в противном случае состояние стилей перестанет работать (в силу механизма каскадности).

```
a:link {
  color: green;
}
a:visited {
  color: black;
}
a:focus, a:hover{
  color: pink;
}
a:active {
  color: red;
}
```

Ссылки-кнопки.

```
    <a href="#" class="link box-link-1">Ссылка</a>
    <a href="#" class="link box-link-2">Ссылка</a>
    <a href="#" class="link box-link-3">Ссылка</a>
    <a href="#" class="link box-link-4">Ссылка</a>
    <a href="#" class="link box-link-5">Ссылка</a>
```

```
a.link{
    text-decoration: none;
    display: inline-block;
    font-size: 1.5rem;
    margin: 5px;
}
a.box-link-1{
    border-radius: 10px;
    color: #EC4D3C;
    padding: 5px;
    background-color: transparent;
    transition: 0.5s;
}
a.box-link-1:hover{
    background-color: #ec4e3c21;
    box-shadow: 2px 2px 3px rgba(0, 0, 0, 0.3);
}
```

Свойство transition

CSS `transitions` предоставляют способ контролировать скорость анимации, при изменении CSS-свойств. Вместо того, чтобы свойство применилось сразу, вы можете сделать это действие происходящим в течение какого-то момента времени. Например, если вы смените цвет элемента с белого на чёрный, изменение произойдёт моментально, а вот с CSS `transitions`, изменения произойдут за временные интервалы, следующих кривой ускорения, все из которых могут быть настроены.

`Transition позволяет` задать значения `transition-property`, `transition-duration`, `transition-timing-function` и `transition-delay`.

Свойство transition

transition-property

Указывает имя или имена свойств, чьи переходы должны анимироваться. Только свойства, указанные здесь, анимируются в переходах; изменение других свойств будет происходить обычным образом.

```
transition-property: all;
transition-property: font-size;
transition-property: width, height;
```

transition-duration

Свойство `transition-duration` определяет продолжительность выполнения анимации. Значение по умолчанию равняется 0s, т.е. отсутствие анимации.

transition-delay

Определяет как много должно пройти времени, перед тем как начнётся переход.

Свойство transition

`transition-timing-function`
Устанавливает, насколько быстро должно изменяться значение стилевого свойство для которого применяется эффект перехода.

```
transition-timing-function:  ease|ease-in|ease-out|ease-in-out|linear|step-start|step-end|steps|cubic-Bezier
```

- ease - анимация начинается медленно, затем ускоряется и к концу движения опять замедляется.
- ease-in - анимация медленно начинается, к концу ускоряется.
- ease-out - анимация начинается быстро, к концу замедляется.
- ease-in-out - анимация начинается и заканчивается медленно.
- linear - одинаковая скорость от начала и до конца.
- step-start - как таковой анимации нет. Стилевые свойства сразу же принимают конечное значение.
- step-end - как таковой анимации нет. Стилевые свойства находятся в начальном значении заданное время, затем сразу же принимают конечное значение.
- steps - ступенчатая функция, имеющая заданное число шагов.
- cubic-bezier - задаёт функцию движения в виде кривой Безье.

Ссылки-кнопки. Примеры

```css
a.box-link-2{
    background-color: black;
    color:aquamarine;
    padding: 5px 15px;
    border: 1px solid aquamarine;
    border-radius: 0px  15px  15px  0px;
    transition:  1s  ;
}
a.box-link-2:hover{
    border-color: aqua;
    color: white;
    background-image: linear-gradient(to top, aqua 3px, black 3px);
}
a.box-link-3{
    padding: 5px;
    color: grey;
    width: 120px;
    text-align: center;
    border: 1px solid lightgrey;
}
a.box-link-3:hover{
    font-weight: bold;
    color: rgb(86, 86, 86);
}
```

Ссылки-кнопки. Примеры

```css
a.box-link-4 {
    color: white;
    font-weight: bold;
    letter-spacing: 2px;
    padding: 15px 30px;
    border-radius: 10px;
    box-shadow: inset 0 0 40px 40px #F137A6 ;
    transition: 200ms ;
}
a.box-link-4:hover {
    box-shadow: inset 0 0 10px 0 #F137A6 ,
                      0 0 10px 4px #F137A6;
    color: #F137A6;
}

a.box-link-5 {
    padding: 5px;
    width: 100px ;
    text-align: center;
    border-radius: 45px;
    color: #524f4e;
    background: white;
    box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.1);
    transition: 0.3s;
}
a.box-link-5:hover {
    box-shadow: 0px 15px 20px rgba(0,0,0,0.3);
    transform: translateY(-3px);
}
```

Свойство cursor

Вид курсора можно задать с помощью значения – названия, например курсор со знаком вопроса:

```
.help { cursor: help;}
```

Примеры свойства cursor: https://html5css.ru/cssref/pr_class_cursor.php

Все браузеры дают возможность установить элементу свой курсор из файла по URL, например:

```
body {  cursor: url('/pointer.cur') 4 1;}
```

Вторым параметром задаются координаты активной точки, в которой регистрируется нажатие мыши. По умолчанию это левый верхний угол (0,0).

У свойства cursor имеется возможность указать несколько курсоров, в приоритете будет первый в списке поддерживаемый формат.

CSS-списки

CSS-списки — набор свойств, отвечающих за оформление списков. Использование HTML-списков очень распространено при создании панелей навигации по сайту. Элементы списка представляют набор блочных элементов.

С помощью стандартных CSS-свойств можно изменить внешний вид маркера списка, добавить изображение для маркера, а также изменить местоположение маркера.

Псевдоэлемент ::marker применяет стили к маркеру элемента списка, которые обычно содержит значок или номер. Работает с любым элементом или псевдоэлементом, к которого установлен display: list-item, например, у такого как <li>:

```
li::marker{
    font-size: 1.5em;
    color: red;
}
```

Тип маркера списка list-style-type

Свойство изменяет типа маркера или удаляет маркер для маркированного и нумерованного списков. Свойство наследуется.

```
list-style-type: disc; /* Значение по умолчанию. В качестве маркера элементов списка выступает закрашенный кружок. */
list-style-type: circle;
list-style-type: square;
list-style-type: decimal;
list-style-type: decimal-leading-zero;
list-style-type: lower-roman; /*i, ii, iii, iv, v, …*/
list-style-type: upper-roman;
list-style-type: lower-greek;
list-style-type: lower-latin;
list-style-type: upper-latin;
list-style-type: lower-alpha;
list-style-type: upper-alpha;
list-style-type: none;
```

Изображения для элементов списка list-style-image

В качестве маркера элементов списка можно использовать изображения.

```
list-style-image: url("images/romb.png");
list-style-image: none;
```

Местоположение маркера списка list-style-position

Данное свойство предоставляет возможность располагать маркер вне или внутри содержимого элемента списка.

```
list-style-position: inside;
list-style-position: outside;
```

```
/* Пример короткого оформления списка */
ul {
  list-style: url("images/romb.png") inside;
}
```

Пример оформления списка

```html
<ul class="list-2">
    <li><a href="#">Ссылка 1</a></li>
    <li><a href="#">Ссылка 2</a></li>
    <li><a href="#">Ссылка 3</a></li>
    <li><a href="#">Ссылка 4</a></li>
    <li><a href="#">Ссылка 5</a></li>
</ul>
```

```css
.list-2{
    list-style: none;
    padding: 0;
}
.list-2 a{
    color: black;
    font: 1.5em bold;
    text-decoration: none;
    display: inline-block;
    width: 100%;
}
.list-2 li{
    background-color: orange;
    border: 1px solid black;
    //display: inline-block; /* Попробуйте изменить
свойство отображения и посмотреть, что получится */
    padding: 5px;
}
.list-2 li:hover{
    background-color: lime;
}
```

Пример оформления списка

```html
<ul class="list-border">
    <li>Элемент списка</li>
    <li>Элемент списка</li>
    <li>Элемент списка</li>
    <li>Элемент списка</li>
    <li>Элемент списка</li>
</ul>
```

```css
.list-border {
    list-style: none;
    padding: 0;
}
.list-border li {
    padding: 7px 20px;
    margin-bottom: 10px;
    border-radius: 5px;
    border-left: 10px solid #f05d22;
    box-shadow: 2px -2px 5px 0 rgba(0, 0, 0, .1),
        -2px -2px 5px 0 rgba(0, 0, 0, .1),
        2px 2px 5px 0 rgba(0, 0, 0, .1),
        -2px 2px 5px 0 rgba(0, 0, 0, .1);
    font-size: 20px;
    letter-spacing: 2px;
    transition: 0.3s all linear;
}

.list-border li:hover {
    border-left: 10px solid transparent;
    border-right: 10px solid #f05d22;
}
```

HTML-таблицы

Таблица – это структурированный набор данных, состоящий из строк и столбцов (табличных данных). Таблицы позволяют быстро и легко посмотреть значения, показывающие некоторую взаимосвязь между различными типами данных.

HTML-таблицы упорядочивают и выводят на экран данные с помощью строк или столбцов. Таблицы состоят из ячеек, образующихся при пересечении строк и столбцов.

Ячейки таблиц могут содержать любые HTML-элементы, такие как заголовки, списки, текст, изображения, элементы форм, а также другие таблицы. Каждой таблице можно добавить связанный с ней заголовок, расположив его перед таблицей или после неё.

Таблицы больше не используются для вёрстки веб-страниц и компоновки отдельных элементов, потому что такой приём не обеспечивает гибкость структуры и адаптивность сайта, существенно увеличивая HTML-разметку.

В HTML для создания таблиц существует набор семантических тегов:

```
<table>
<caption> <thead> <tbody> <tfoot>
<tr>
<th> <td>
```

Структурные теги

`<table>`

Встречая данный тег в разметке, браузер понимает, что дальше будет таблица.

`<tr>`

Любая таблица в первую очередь состоит из строк. Чтобы в таблице появились строки, используйте парный тег `<tr>`. Сколько нужно строк — столько раз нужно написать `<tr>` внутри `<table>`.

`<td>`

Чтобы создать ячейку под данные, нужен парный тег `<td>`. Пишем столько `<td>` внутри `<tr>`, сколько нужно ячеек таблицы.

Ячейки формируют из себя столбцы. В HTML нет специального тега для столбцов.

```
<table>
    <tr>
    <td>Модель</td>
    <td>Цена</td>
    </tr>
    <tr>
    <td>iPhone 12 Pro</td>
    <td>$999</td>
    </tr>
    <tr>
    <td>iPhone 12</td>
    <td>$799</td>
    </tr>
    <tr>
    <td>iPhone 12 mini</td>
    <td>$699</td>
    </tr>
</table>
```

Оформление таблиц

По умолчанию таблица и ячейки не имеют видимых границ.

Границы задаются с помощью свойства `border` и `border-collapse`.

```css
/* внешние границы таблицы серого цвета толщиной 1px */
table {
    border: 1px solid grey;
    border-collapse: collapse;
    width: 600px;
}

/* границы ячеек тела таблицы */
td {
    border: 1px solid grey;
}
```

Структурные теги

Специально для заголовков ячеек или строк есть тег `<th>`.

К ячейкам, обёрнутым тегом `<th>`, применяются дефолтные стили: текст становится жирным и выравнивается по центру ячейки. Это помогает внешне отделить заголовки от остальных данных таблицы

```
/* границы ячеек первого ряда таблицы */
th {
    border: 1px solid grey;
}
```

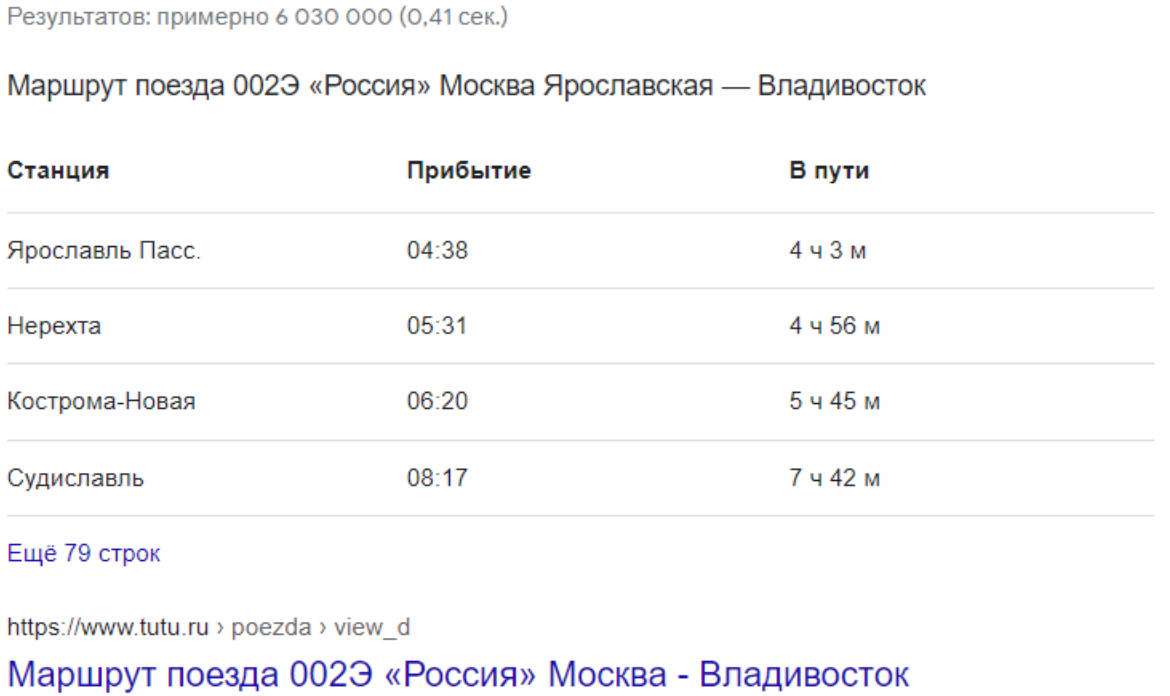
```
<tr>
    <th>Модель</th>
    <th>Цена</th>
</tr>
<tr>
    <td>iPhone 12 Pro</td>
    <td>$999</td>
</tr>
<tr>
    <td>iPhone 12</td>
    <td>$799</td>
</tr>
<tr>
    <td>iPhone 12 mini</td>
    <td>$699</td>
</tr>
</table>
```

Группировка разделов таблицы

Существуют также теги логической группировки `<thead>`, `<tbody>`, `<tfoot>`.

Эти теги помогают лучше читать разметку сложных таблиц и отделять зёрна от плевел: структурные части таблицы друг от друга. Например: сложную шапку от тела с данными, и всё это — от результатов подсчёта в подвале.

К тому же, правильно свёрстанная таблица может отобразиться в поисковике в виде сниппета.



Группировка разделов таблицы

`<thead>`

Тег `<thead>` отвечает за шапку таблицы. Внутри этого тега могут располагаться одна или более строк с заголовками таблицы. `<thead>` должен располагаться в разметке сразу за открывающим `<table>` или после `<caption>`, но строго до `<tbody>` и `<tfoot>`.

```
<table>
    <thead>
    <tr>
        <th>Модель</th>
        <th>Цена</th>
    </tr>
    </thead>
    <tr>
        <td>iPhone 12 Pro</td>
        <td>$999</td>
    </tr>
    <tr>
        <td>iPhone 12</td>
        <td>$799</td>
    </tr>
    <tr>
        <td>iPhone 12 mini</td>
        <td>$699</td>
    </tr>
</table>
```

Группировка разделов таблицы

`<tbody>`

Этот тег предназначен для основной части таблицы. Внутрь него помещаются строки с данными.

Можно использовать несколько `<tbody>` внутри таблицы, разделяя тем самым данные на отдельные блоки.

```
<table>
    <thead>
        <tr>
            <th>Модель</th>
            <th>Цена</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>iPhone 12 Pro</td>
            <td>$999</td>
        </tr>
        <tr>
            <td>iPhone 12</td>
            <td>$799</td>
        </tr>
        <tr>
            <td>iPhone 12 mini</td>
            <td>$699</td>
        </tr>
    </tbody>
    <tbody>
        <tr>
            <td>Xiaomi Mi 10</td>
            <td>$768</td>
        </tr>
    </tbody>
</table>
```

Группировка разделов таблицы

`<tfoot>`

Тег `<tfoot>` нужен для строки «Итого» — некой строки с итогом данных таблицы. В таблице может быть только один блок `<tfoot>`.

Браузер всегда отрисовывает `<tfoot>` внизу таблицы, даже если этот блок идёт в разметке не последним (хоть это и не очень логично).

Если по какой-то причине вы не использовали в таблице `<thead>` или `<tbody>`, то футер будет отрисован после всех `<tr>`.

```
<table>
    <thead>
        <tr>
            <th>Модель</th>
            <th>Цена</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>iPhone 12 Pro</td>
            <td>$999</td>
        </tr>
        <tr>
            <td>iPhone 12</td>
            <td>$799</td>
        </tr>
        <tr>
            <td>iPhone 12 mini</td>
            <td>$699</td>
        </tr>
    </tbody>
    <tbody>
        <tr>
            <td>Xiaomi Mi 10</td>
            <td>$768</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td>Средняя цена:</td>
            <td>$816.25</td>
        </tr>
    </tfoot>
</table>
```

Заголовок таблицы

`<caption>`

Если нужно подписать таблицу, дать ей определение, то можно использовать парный тег `<caption>`. В него помещается общая информация о таблице.

По умолчанию подпись визуально располагается сразу перед таблицей. Но её положением можно управлять при помощи свойства `caption-side`. Вне зависимости от визуального расположения подписи скринридер прочитает её перед таблицей.

```
<table>
    <caption>
        Цены модели iPhone и
Xiaomi
    </caption>
    <thead>
        <tr>
            <th>Модель</th>
            <th>Цена</th>
        </tr>
    </thead>
</table>
```

Таблицы для пользователей с ограниченными возможностями

Люди с ослабленным зрением часто используют скринридер, который читает им информацию с веб-страницы. Это не проблема когда вы читаете простой текст, но интерпретация таблицы может быть сложной проблемой для слепых людей. Тем не менее, вместе с правильной разметкой мы можем заменить визуальные ассоциации программными.

Скринридеры будут определять все заголовки и использовать их создавая программные ассоциации между этими заголовками и ячейками к которым они относятся. Сочетание заголовков столбцов и строк будет определять и интерпретировать данные в каждой ячейке так, что пользователи скринридеров могут интерпретировать таблицу также как это делают зрячие пользователи.

Атрибут **scope** определяет, является ли заголовочная ячейка заголовком для колонки, строки или группы колонок или строк.

Атрибут **scope** никак не отображается обычными браузерами и предназначен для использования речевыми браузерами.

```
<table>
    <tr>
        <th></th>
        <th scope="col">Месяц</th>
        <th scope="col">Заработано</th>
    </tr>
    <tr>
        <td scope="row">1</td>
        <td>Январь</td>
        <td>152</td>
    </tr>
    <tr>
        <td scope="row">2</td>
        <td>Февраль</td>
        <td>176</td>
    </tr>
    <tr>
        <td scope="row">3</td>
        <td>Март</td>
        <td>244</td>
    </tr>
</table>
```

Пример оформления для таблицы

Цены модели iPhone и Xiaomi

Модель	Цена
iPhone 12 Pro	$999
iPhone 12	$799
iPhone 12 mini	$699
Xiaomi Mi 10	$768
Средняя цена:	*$758.8*

```
<table>
    <caption> Цены модели iPhone и Xiaomi </caption>
    <thead>
        <tr>
            <th>Модель</th>
            <th>Цена</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>iPhone 12 Pro</td>
            <td>$999</td>
        </tr>
        <tr>
            <td>iPhone 12</td>
            <td>$799</td>
        </tr>
        <tr>
            <td>iPhone 12 mini</td>
            <td>$699</td>
        </tr>
        <tr>
            <td>Xiaomi Mi 10</td>
            <td>$768</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td>Средняя цена:</td>
            <td>$758.8</td>
        </tr>
    </tfoot>
</table>
```

Объединение ячеек

Атрибуты `colspan` и `rowspan` объединяют ячейки таблицы. Атрибут `colspan` задает количество ячеек, объединенных по горизонтали, а `rowspan` — по вертикали.

Студент	2023						
	Сентябрь		Октябрь		Ноябрь		
	6	20	4	18	1	15	29
Иванов П.Т.	н		+			5	
Прокопьев Ф.А.	н	н	н			0	
Якимов П.Т.	н		+			2	
Минкина П.Р.		н	+			3	

```
<table>
    <thead>
        <tr>
        <th rowspan="3">  Студент  </th>  <th colspan="7">  2023 </th>
        </tr>
        <tr>
        <th colspan="2">  Сентябрь  </th> <th colspan="2">  Октябрь  </th>  <th colspan="3">  Ноябрь  </th>
        </tr>
        <tr>
        <th>6</th> <th>20</th> <th>4</th> <th>18</th> <th>1</th> <th>15</th> <th>29</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Иванов П.Т.</td>  <td>н</td> <td></td> <td>+</td><td></td> <td></td> <td>5</td> <td></td>
        </tr>
        <tr>
            <td>Прокопьев Ф.А.</td> <td>н</td> <td>н</td> <td>н</td> <td></td> <td></td> <td>0</td>   <td></td>
        </tr>
        <tr>
            <td>Якимов П.Т.</td>  <td>н</td> <td></td> <td>+</td><td></td> <td></td> <td>2</td> <td></td>
        </tr>
        <tr>
            <td>Минкина П.Р.</td>  <td></td> <td>н</td> <td>+</td><td></td> <td></td> <td>3</td> <td></td>
        </tr>
    </tbody>
</table>
```