

Лекция 4

Знакомство с jQuery

Что такое jQuery?

jQuery — это легковесная JavaScript библиотека типа «пиши меньше, делай больше».

Цель jQuery — упростить использование JavaScript на вашем веб-сайте.

jQuery берет множество общих задач, для выполнения которых требуется много JavaScript кода, и оборачивает их в методы, которые можно вызывать с помощью одной команды.

jQuery также упрощает многие сложные вещи из JavaScript, такие как вызовы AJAX и манипулирование DOM.

Библиотека jQuery содержит следующие функции:

- для манипуляции с HTML/DOM
- для манипуляции с CSS
- для работы с HTML событиями
- эффекты и анимация
- AJAX
- Утилиты

Кроме того, у jQuery есть плагины практически для любой задачи.

Почему используют jQuery?

Существует множество других библиотек JavaScript, но jQuery, вероятно, является самой популярной, а также самой расширяемой.

Многие из крупнейших компаний в Интернете используют jQuery, например:

- Google
- Microsoft
- IBM
- Netflix

jQuery будет работать одинаково во всех основных браузерах.

Добавление jQuery на web-страницу

Есть несколько способов начать использовать jQuery на своем веб-сайте. Вы можете:

- Загрузите библиотеку jQuery с сайта [jQuery.com](http://jquery.com)
- Включите jQuery из CDN, например из Google

Для загрузки доступны две версии jQuery:

- рабочая (продакшн) версия - это для вашего работающего веб-сайта, потому что он был минимизирован и сжат.
- версия для разработчиков - для тестирования и разработки (несжатый и читаемый код)

Библиотека jQuery представляет собой отдельный файл JavaScript, и вы ссылаетесь на него с помощью HTML тега **<script>**

```
3 <head>
4 |   <script src="jquery-3.6.0.min.js"></script>
5 </head>
```

jQuery CDN

Если вы не хотите загружать и размещать jQuery самостоятельно, вы можете включить его из CDN (сети доставки контента).

Google является примером того, кто размещает jQuery:

```
3 <head>
4   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5 </head>
```

<https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js>

Синтаксис jQuery

Синтаксис jQuery создан специально для выбора HTML элементов и выполнения некоторых действий над ними.

`$(selector).action()`

- знак **\$** для определения/доступа к jQuery
- **selector** для «запроса (или поиска)» HTML элементов
- **action** для выполнения действий/операций над элементами

`$(this).hide()` - скрывает текущий элемент.

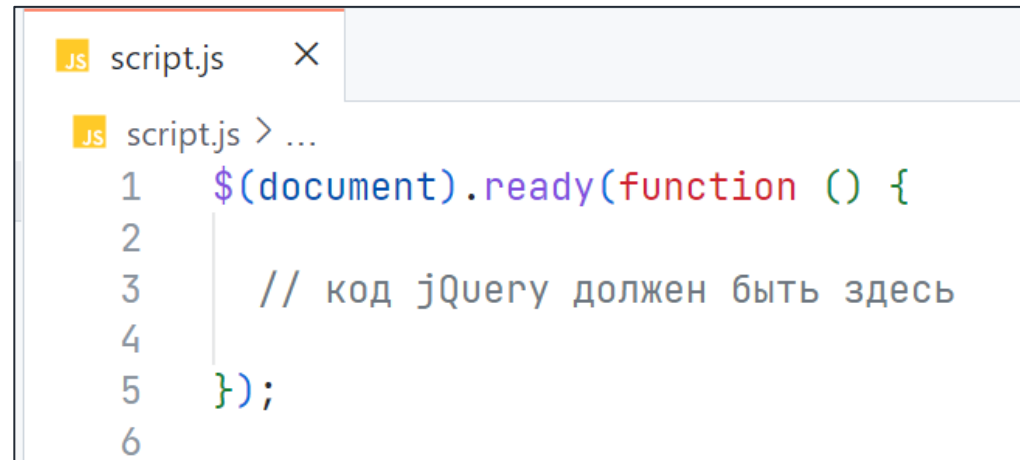
`$("p").hide()` - скрывает все элементы `<p>`.

`$(".test").hide()` - скрывает все элементы с `class="test"`.

`$("#test").hide()` - скрывает элемент с `id="test"`.

Событие, что документ готов

Все методы jQuery должны находится внутри события готовности документа.



```
script.js  X
script.js > ...
1  $(document).ready(function () {
2
3      // код jQuery должен быть здесь
4
5  });
6
```

Это делается для предотвращения запуска любого кода jQuery до того, как документ будет загружен (будет готов).

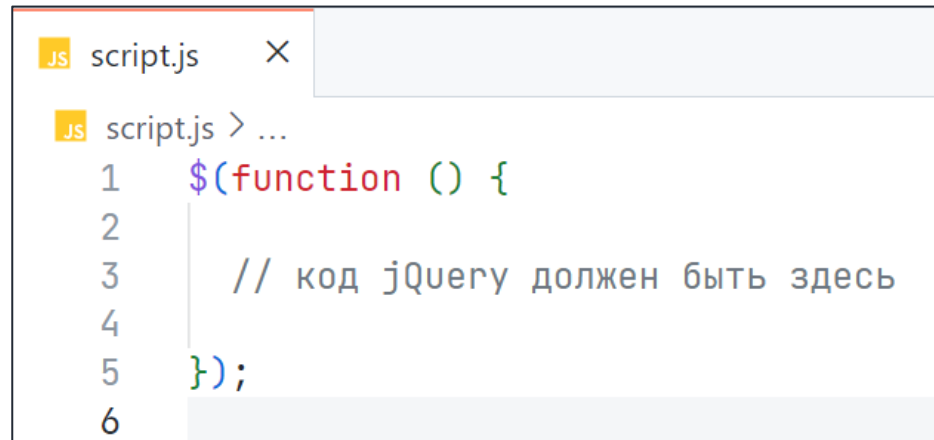
Хорошей практикой является дождаться полной загрузки и готовности документа, прежде чем работать с ним. Это также позволяет размещать код JavaScript перед телом документа, в разделе заголовка.

Событие, что документ готов

Вот несколько примеров действий, которые могут завершиться ошибкой, если методы запускаются до полной загрузки документа:

- Попытка скрыть элемент, который еще не создан
- Попытка получить размер изображения, которое еще не загружено

Команда jQuery также создала еще более короткий метод для события готовности документа:

A screenshot of a code editor window titled 'script.js'. The editor shows a jQuery document ready event handler. The code is as follows:

```
1 $(function () {  
2  
3     // код jQuery должен быть здесь  
4  
5 });  
6
```


jQuery селекторы

jQuery селекторы позволяют выбирать элементы HTML и управлять ими.

Селекторы jQuery используются для «нахождения» (или выбора) HTML элементов на основе их имени, идентификатора, классов, типов, атрибутов, значений атрибутов и многого другого. Он основан на существующих селекторах CSS и, кроме того, имеет несколько собственных селекторов.

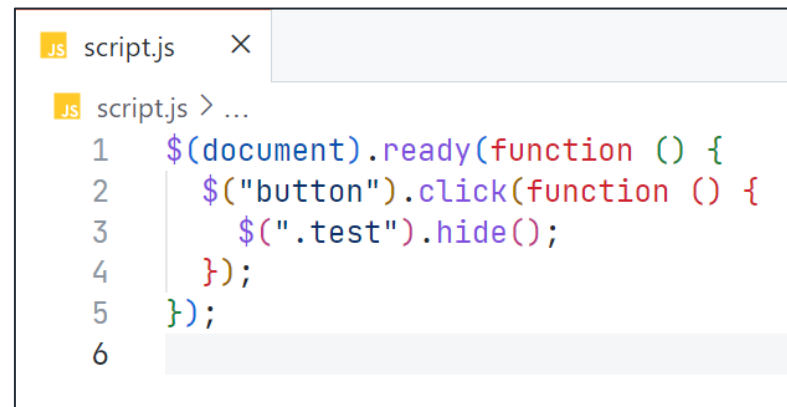
Все селекторы в jQuery начинаются со знака доллара и круглых скобок: **\$()**.

`$(this).hide()` - скрывает текущий элемент.

`$("p").hide()` - скрывает все элементы `<p>`.

`$(".test").hide()` - скрывает все элементы с `class="test"`.

`$("#test").hide()` - скрывает элемент с `id="test"`.

A screenshot of a code editor window titled 'script.js'. The editor shows a jQuery script that uses the 'ready' method to attach a click event to a button. When the button is clicked, it calls the 'hide' method on all elements with the class 'test'. The code is as follows:

```
1 $(document).ready(function () {  
2   $("button").click(function () {  
3     $(".test").hide();  
4   });  
5 });  
6
```

Описание функций в отдельном файле

Если ваш веб-сайт содержит много страниц и вы хотите, чтобы ваши функции jQuery было легко поддерживать, вы можете поместить свои функции jQuery в отдельный файл .js.

```
3 <head>
4   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5   <script src="script.js"></script>
6 </head>
```

Получить контент и атрибуты

Одна очень важная часть jQuery — возможность манипулировать DOM.

jQuery поставляется с набором методов, связанных с DOM, которые упрощают доступ к элементам и атрибутам и управление ими.

Три простых, но полезных метода jQuery для манипулирования DOM:

text() - задает или возвращает текстовое содержимое выбранных элементов

html() - задает или возвращает содержимое выбранных элементов (включая HTML-разметку)

val() - задает или возвращает значение полей формы

```
script.js  ×
script.js > ...
1  $("#btn1").click(function () {
2    alert("Text: " + $("#test").text());
3  });
4  $("#btn2").click(function () {
5    alert("HTML: " + $("#test").html());
6  });
7  |
```

```
script.js  ×
script.js > ...
1  $("#btn1").click(function () {
2    alert("Value: " + $("#test").val());
3  });
4  |
```

Получить контент и атрибуты

Метод jQuery **attr()** используется для получения значений атрибутов.

A screenshot of a code editor window titled 'script.js'. The editor shows a jQuery script with four lines of code. Line 1: `1 $("button").click(function () {`. Line 2: `2 | alert($("#w3s").attr("href"));`. Line 3: `3 | });`. Line 4: `4 |`. The code is color-coded: `$` is blue, `function` is red, `click` is blue, `attr` is blue, and `href` is red. The cursor is at the end of line 4.

```
JS script.js  X
JS script.js > ...
1  $("button").click(function () {
2  |  alert($("#w3s").attr("href"));
```

Изменить контент и атрибуты

Мы будем использовать те же три метода, что и на предыдущих слайдах, для установки контента:

text() - задает или возвращает текстовое содержимое выбранных элементов

html() - задает или возвращает содержимое выбранных элементов (включая HTML-разметку)

val() - задает или возвращает значение полей формы



```
script.js  ×
script.js > ...
1  $("#btn1").click(function () {
2    |   $("#test1").text("Hello world!");
3  |   });
4  $("#btn2").click(function () {
5    |   $("#test2").html("<b>Hello world!</b>");
6  |   });
7  $("#btn3").click(function () {
8    |   $("#test3").val("Dolly Duck");
9  |   });
10 |
```

Функция обратного вызова

Все три вышеуказанных метода jQuery: **text()**, **html()** и **val()** также имеют функцию обратного вызова. Функция обратного вызова имеет два параметра:

- индекс текущего элемента в списке выбранных элементов;
- исходное (старое) значение.

Затем вы возвращаете строку, которую хотите использовать в качестве нового значения функции.

```
script.js  ×
script.js > ...
1  $("#btn1").click(function(){
2      $("#test1").text(function(i, origText){
3          return "Old text: " + origText + " New text: Hello world! index: " + i + " ";
4      });
5  });
6
7  $("#btn2").click(function(){
8      $("#test2").html(function(i, origText){
9          return "Old html: " + origText + " New html: Hello <b>world!</b> (index: " + i + " ";
10     });
11 });
12
```

Изменить атрибут

Метод jQuery **attr()** также используется для установки/изменения значений атрибутов.

```
JS script.js ×
JS script.js > ...
1  $("button").click(function () {
2    |  $("#w3s").attr("href", "https://www.w3schools.com/jquery/");
3  });
4  |
```

Метод **attr()** также позволяет одновременно устанавливать несколько атрибутов.

```
JS script.js ×
JS script.js > ...
1  $("button").click(function () {
2    |  $("#w3s").attr({
3    |    href: "https://ssau.ru",
4    |    title: "Самарский университет",
5    |  });
6  });
7  |
```

Функция обратного вызова для attr()

Метод jQuery **attr()** также имеет функцию обратного вызова. Функция обратного вызова имеет два параметра: индекс текущего элемента в списке выбранных элементов и исходное (старое) значение атрибута. Затем вы возвращаете строку, которую хотите использовать в качестве нового значения атрибута из функции.



```
script.js  X
script.js > ...
1  $("button").click(function () {
2      $("#w3s").attr("href", function (i, origValue) {
3          return origValue + "/jquery/";
4      });
5  });
6  |
```


Добавление элемента

jQuery позволяет легко добавлять новые элементы/контент на страницу.

Мы рассмотрим четыре метода, которые используются для добавления нового контента:

append() - добавляет контент внутрь выбранных элементов в конец, после имеющихся.

prepend() - добавляет контент внутрь выбранных элементов перед существующим контентом

after() - вставляет контент после каждого выбранного элемента

before() - вставляет контент перед каждым выбранным элементом

Добавление элемента

Метод **append()** добавляет контент внутрь выбранных элементов в конец, после имеющихся.

```
index.html > ...
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5   <script src="script.js"></script>
6 </head>
7 <body>
8   <p>Это параграф</p>
9   <p>Это другой параграф</p>
10  <button id="btn">Добавить</button>
11 </body>
12 </html>
13
```

Это параграф

Это другой параграф

Добавить

```
script.js > ...
1 $(document).ready(function () {
2   $("#btn").click(function () {
3     $("p").append("<b> Добавленный текст</b>.");
4   });
5 });
6
```

Это параграф **Добавленный текст.**

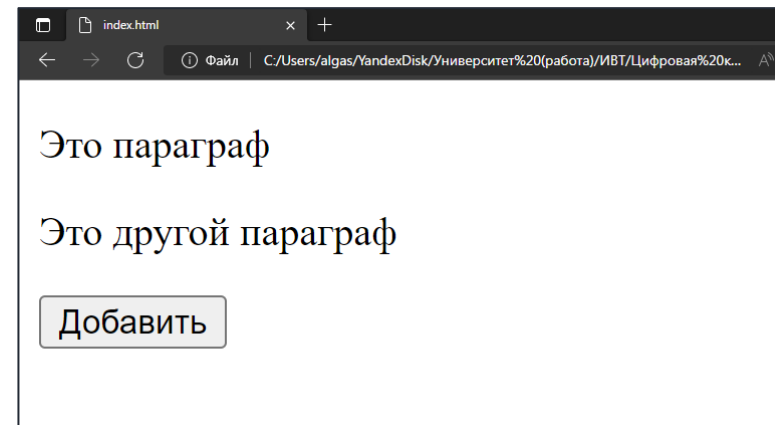
Это другой параграф **Добавленный текст.**

Добавить

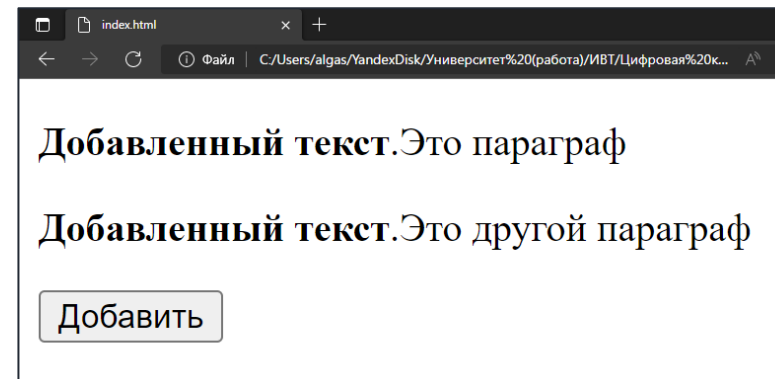
Добавление элемента

Метод **prepend()** добавляет контент внутри выбранных элементов перед существующим контентом

```
index.html > ...
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5   <script src="script.js"></script>
6 </head>
7 <body>
8   <p>Это параграф</p>
9   <p>Это другой параграф</p>
10  <button id="btn">Добавить</button>
11 </body>
12 </html>
13
```



```
script.js > ...
1 $(document).ready(function () {
2   $("#btn").click(function () {
3     $("p").prepend("<b> Добавленный текст</b>.");
4   });
5 });
6
```



Добавление нескольких новых элементов

В обоих приведенных ранее примерах мы добавляли некоторые элементы только в начало/конец выбранных элементов HTML.

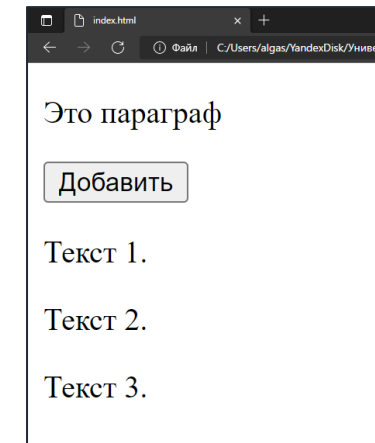
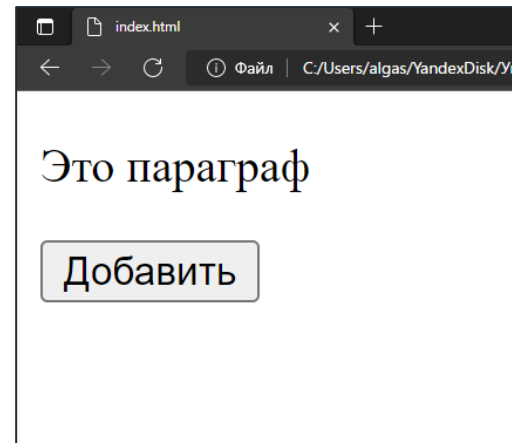
Однако оба метода **append()** и **prepend()** могут принимать в качестве параметров бесконечное количество новых элементов. Новые элементы могут быть сгенерированы с помощью HTML (как мы сделали в приведенных выше примерах), с помощью jQuery или с помощью кода JavaScript и элементов DOM.

В следующем примере мы создадим несколько новых элементов. Элементы будут созданы с помощью текста/HTML, jQuery и JavaScript/DOM. Затем мы добавим новые элементы в текст с помощью метода **append()**.

Добавление нескольких новых элементов

```
index.html ×
index.html > ...
1  <!DOCTYPE html>
2  <html lang="ru">
3    <head>
4      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5      <script src="script.js"></script>
6    </head>
7    <body>
8      <p>Это параграф</p>
9      <button onclick="appendText()">Добавить</button>
10   </body>
11 </html>
12
```

```
script.js ×
script.js > ...
1  function appendText() {
2    var txt1 = "<p>Текст 1.</p>";
3    var txt2 = $("<p></p>").text("Текст 2.");
4    var txt3 = document.createElement("p");
5    txt3.innerHTML = "Текст 3.";
6    $("body").append(txt1, txt2, txt3);
7  }
8
```



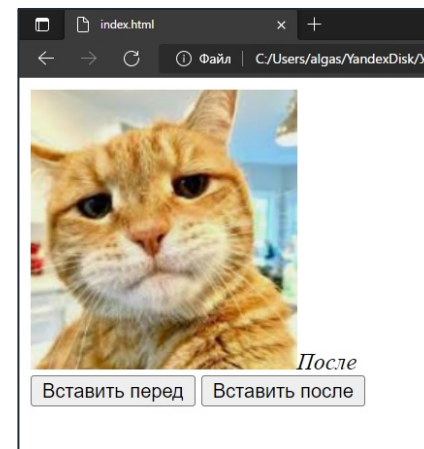
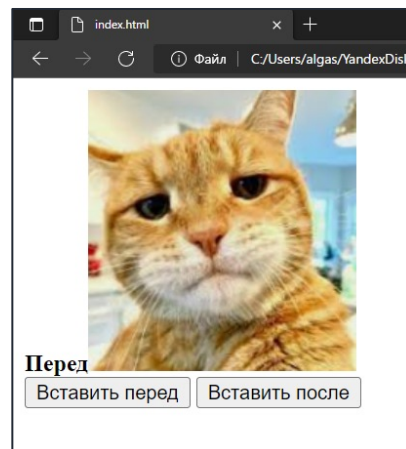
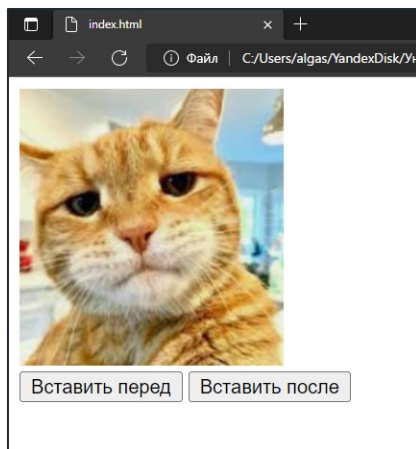
Методы after() и before()

Метод **after()** вставляет контент после каждого выбранного элемента.

Метод **before()** вставляет контент перед каждым выбранным элементом.

```
index.html > ...
1 <!DOCTYPE html>
2 <html lang="ru">
3   <head>
4     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5     <script src="script.js"></script>
6   </head>
7   <body>
8     
9     <div>
10      <button id="btn1">Вставить перед</button>
11      <button id="btn2">Вставить после</button>
12    </div>
13  </body>
14 </html>
15
```

```
script.js > ...
1 $(document).ready(function () {
2   $("#btn1").click(function () {
3     $("#img").before("<b>Перед</b>");
4   });
5
6   $("#btn2").click(function () {
7     $("#img").after("<i>После</i>");
8   });
9 });
10
```



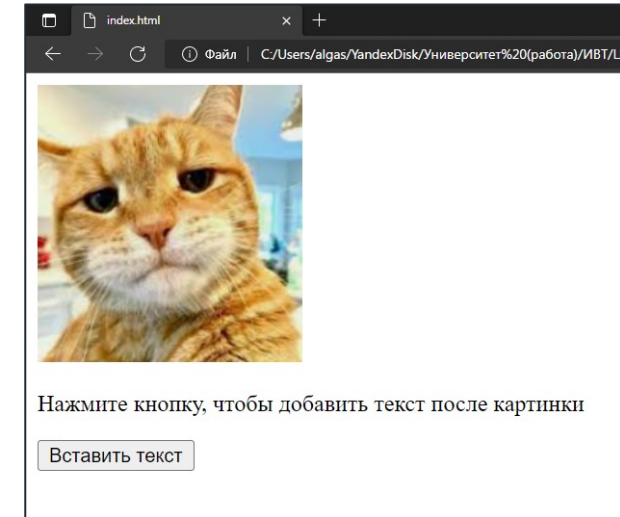
Добавление нескольких новых элементов

Кроме того, методы **after()** и **before()** могут принимать в качестве параметров бесконечное количество новых элементов. Новые элементы могут быть сгенерированы с помощью HTML, с помощью jQuery или с помощью кода JavaScript и элементов DOM.

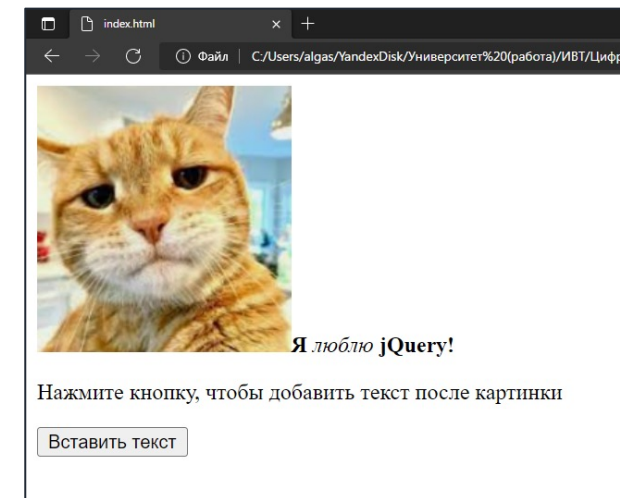
В следующем примере мы создаем несколько новых элементов. Элементы создаются с помощью HTML, jQuery и JavaScript/DOM. Затем мы вставляем новые элементы в текст с помощью метода `after()`.

Добавление нескольких новых элементов

```
index.html > ...
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5   <script src="script.js"></script>
6 </head>
7 <body>
8   
9   <p>Нажмите кнопку, чтобы добавить текст после картинки</p>
10  <button onclick="afterText()">Вставить текст</button>
11 </body>
12 </html>
13
```



```
script.js > ...
1 function afterText() {
2   var txt1 = "<b>Я </b>";
3   var txt2 = $("<i></i>").text("люблю ");
4   var txt3 = document.createElement("b");
5   txt3.innerHTML = "jQuery!";
6   $("img").after(txt1, txt2, txt3);
7 }
8
```



Удаление элементов

С помощью jQuery можно легко удалить существующие элементы HTML.

Для удаления элементов и контента есть два метода в jQuery:

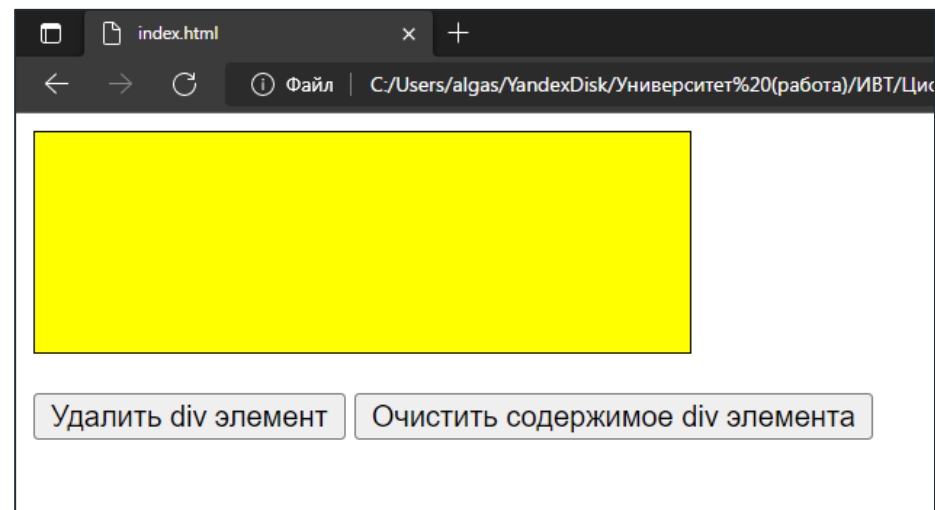
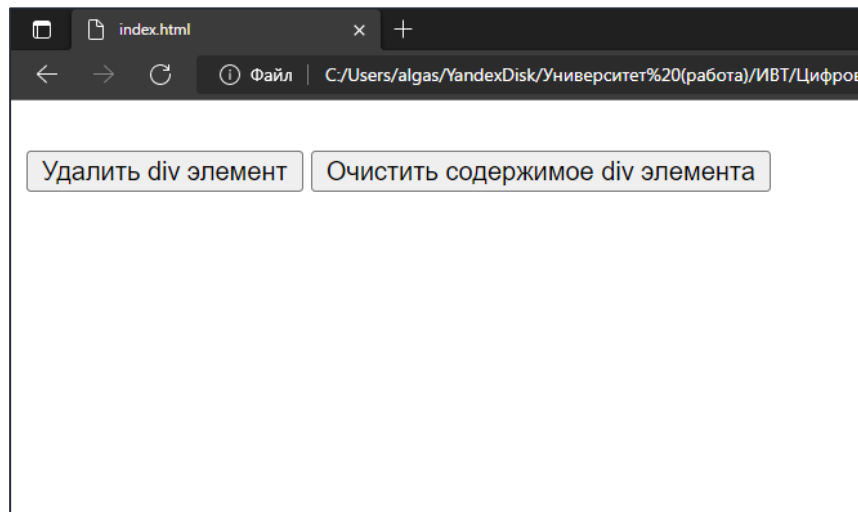
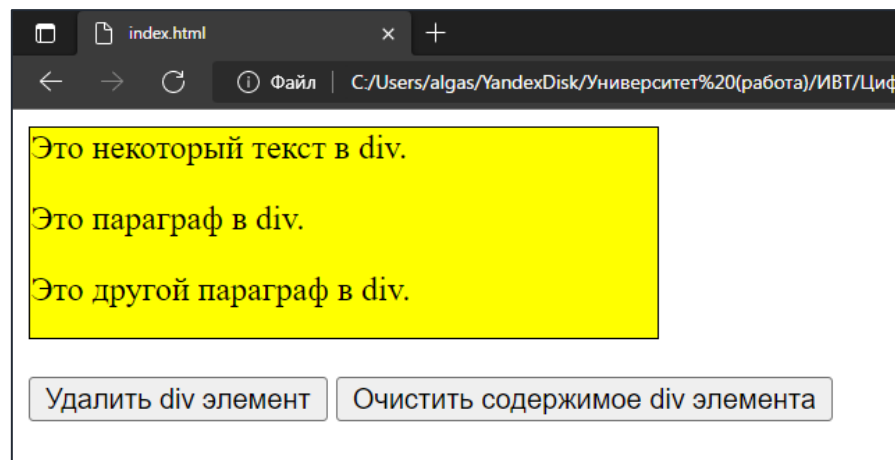
remove() - удаляет выбранный элемент (и его дочерние элементы)

empty() - удаляет дочерние элементы из выбранного элемента

```
index.html X
index.html > ...
1 <!DOCTYPE html>
2 <html lang="ru">
3   <head>
4     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5     <script src="script.js"></script>
6   </head>
7   <body>
8     <div
9       id="div1"
10      style="
11        height: 100px;
12        width: 300px;
13        border: 1px solid black;
14        background-color: yellow;
15      "
16    >
17      Это некоторый текст в div.
18      <p>Это параграф в div.</p>
19      <p>Это другой параграф в div.</p>
20    </div>
21    <br />
22
23    <button id="btn1">Удалить div элемент</button>
24    <button id="btn2">Очистить содержимое div элемента</button>
25  </body>
26 </html>
27
```

```
JS script.js X
JS script.js > ...
1 $(document).ready(function () {
2   $("#btn1").click(function () {
3     $("#div1").remove();
4   });
5
6   $(document).ready(function () {
7     $("#btn2").click(function () {
8       $("#div1").empty();
9     });
10  });
11
12
```

Удаление элементов



Отфильтровать элементы, которые нужно удалить

Метод **remove()** также принимает один параметр, который позволяет фильтровать удаляемые элементы.

Параметр может быть любым селектором jQuery.

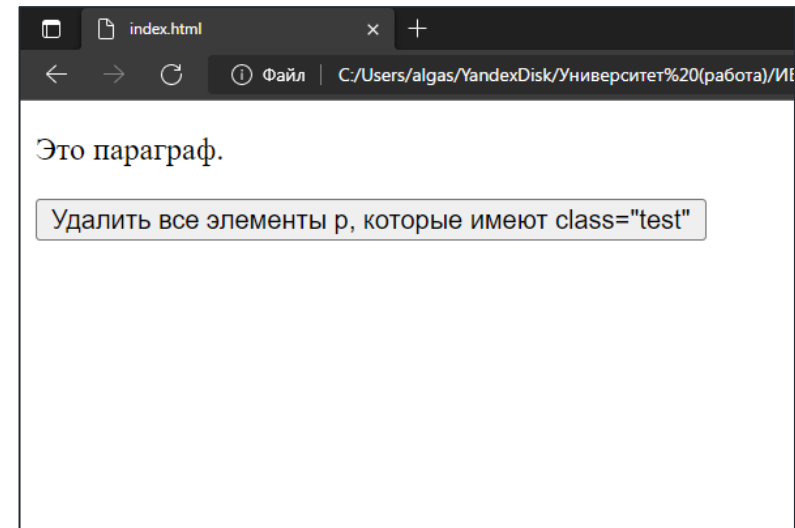
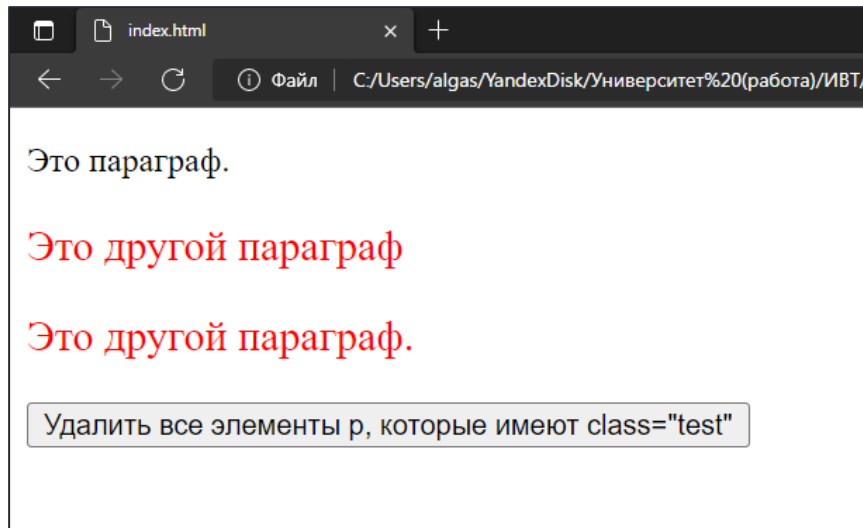
В следующем примере удаляются все элементы `<p>` с `class="test"`:

```
index.html X
index.html > ...
1 <!DOCTYPE html>
2 <html lang="ru">
3   <head>
4     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5     <script src="script.js"></script>
6     <link rel="stylesheet" href="style.css" />
7   </head>
8   <body>
9     <p>Это параграф.</p>
10    <p class="test">Это другой параграф</p>
11    <p class="test">Это другой параграф.</p>
12
13    <button>Удалить все элементы p, которые имеют class="test"</button>
14  </body>
15 </html>
16
```

```
style.css X
style.css > ...
1 .test {
2   color: red;
3   font-size: 20px;
4 }
5
```

Отфильтровать элементы, которые нужно удалить

```
script.js
script.js > ...
1  $(document).ready(function () {
2    $("button").click(function () {
3      $("p").remove(".test");
4    });
5  });
6
```



Получить и изменить CSS class

С jQuery легко манипулировать стилями элементов.

jQuery имеет несколько методов для работы с CSS. Мы рассмотрим следующие методы:

addClass() - добавляет один или несколько классов к выбранным элементам

removeClass() - удаляет один или несколько классов из выбранных элементов

toggleClass() - переключение между добавлением/удалением классов из выбранных элементов

css() - устанавливает или возвращает атрибут стиля

Метод addClass()

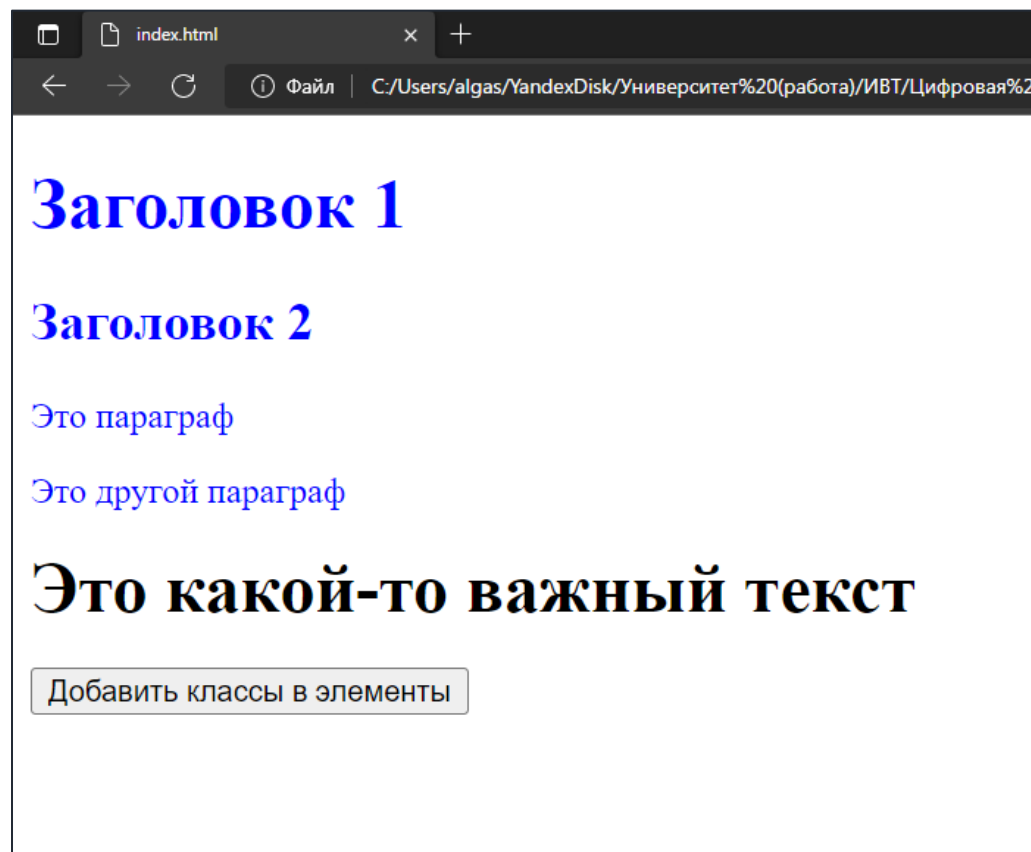
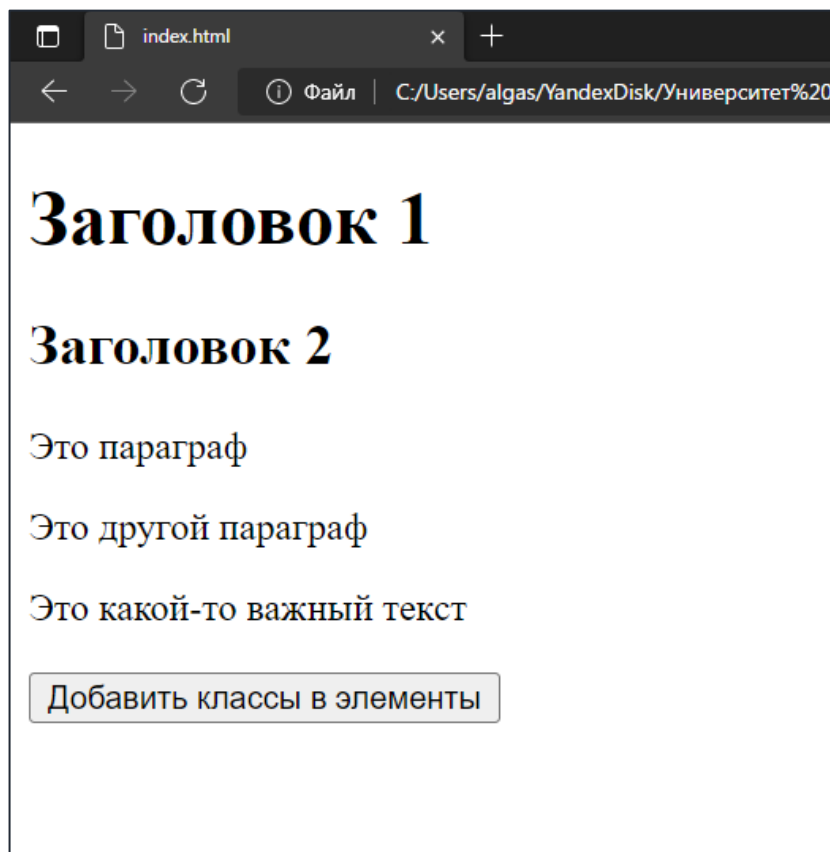
В следующем примере показано, как добавить атрибуты класса к различным элементам. Конечно, вы можете выбрать несколько элементов при добавлении классов:

```
index.html X
index.html > ...
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5   <script src="script.js"></script>
6   <link rel="stylesheet" href="style.css" />
7 </head>
8 <body>
9   <h1>Заголовок 1</h1>
10  <h2>Заголовок 2</h2>
11
12  <p>Это параграф</p>
13  <p>Это другой параграф</p>
14
15  <div>Это какой-то важный текст</div>
16  <br />
17
18  <button>Добавить классы в элементы</button>
19 </body>
20 </html>
21
```

```
script.js X
script.js > ...
1 $(document).ready(function () {
2   $("button").click(function () {
3     $("h1, h2, p").addClass("blue");
4     $("div").addClass("important");
5   });
6 };
7
```

```
style.css X
style.css > ...
1 .important {
2   font-weight: bold;
3   font-size: xx-large;
4 }
5
6 .blue {
7   color: blue;
8 }
9
```

Метод addClass()



Метод removeClass()

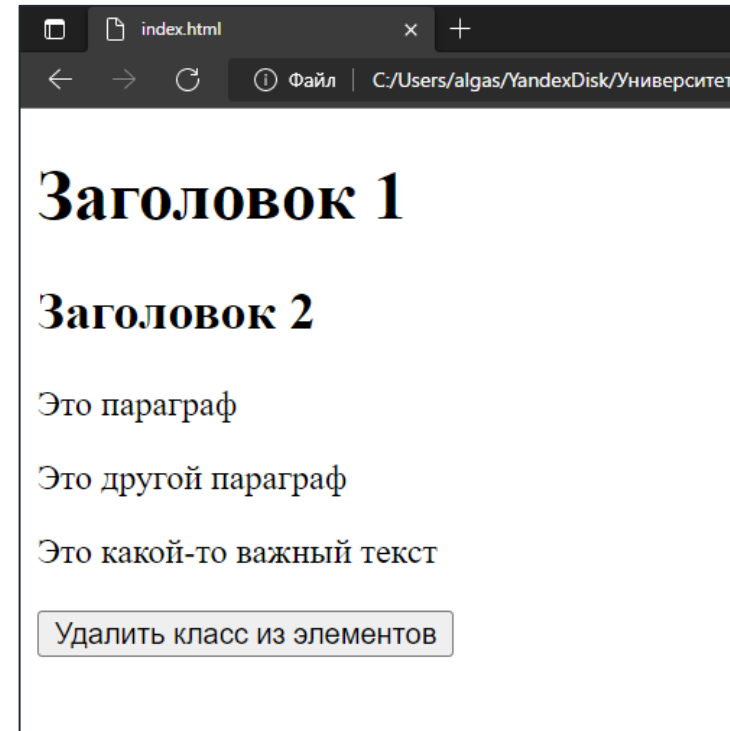
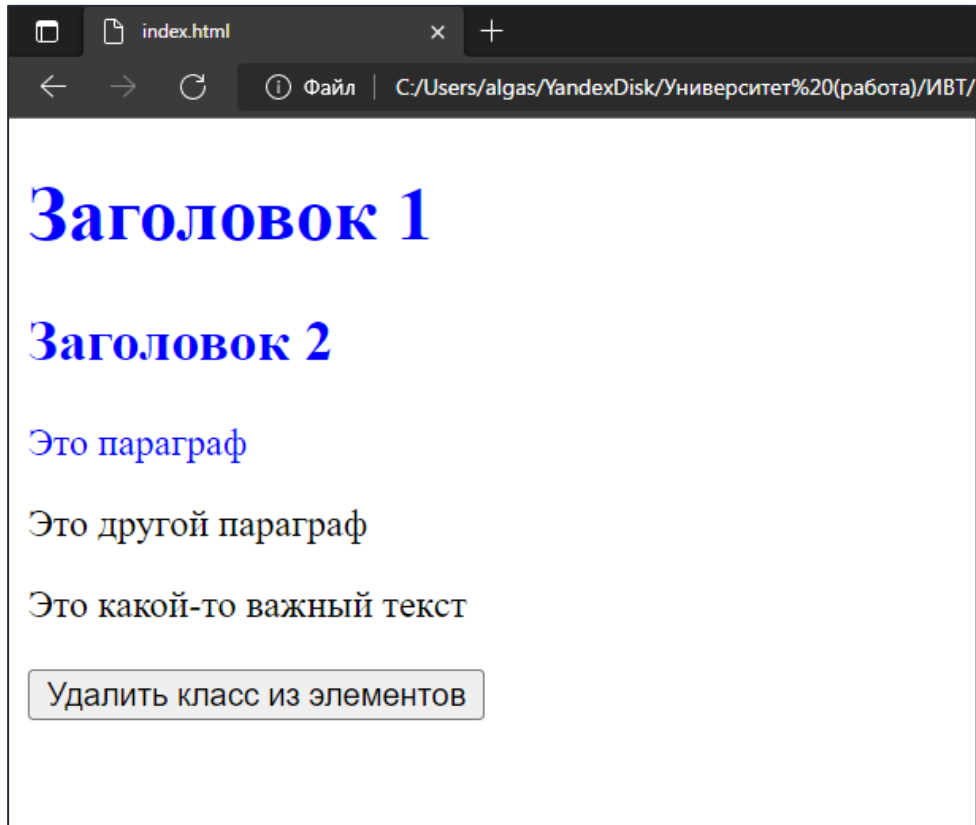
В следующем примере показано, как удалить определенный атрибут класса из разных элементов

```
index.html X
index.html > ...
1 <!DOCTYPE html>
2 <html lang="ru">
3   <head>
4     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
5     <script src="script.js"></script>
6     <link rel="stylesheet" href="style.css" />
7   </head>
8   <body>
9     <h1 class="blue">Заголовок 1</h1>
10    <h2 class="blue">Заголовок 2</h2>
11
12    <p class="blue">Это параграф</p>
13    <p>Это другой параграф</p>
14
15    <div>Это какой-то важный текст</div>
16    <br />
17
18    <button>Удалить класс из элементов</button>
19  </body>
20 </html>
21
```

```
JS script.js X
script.js > ...
1 $(document).ready(function () {
2   $("button").click(function () {
3     $("h1, h2, p").removeClass("blue");
4   });
5 });
6
```

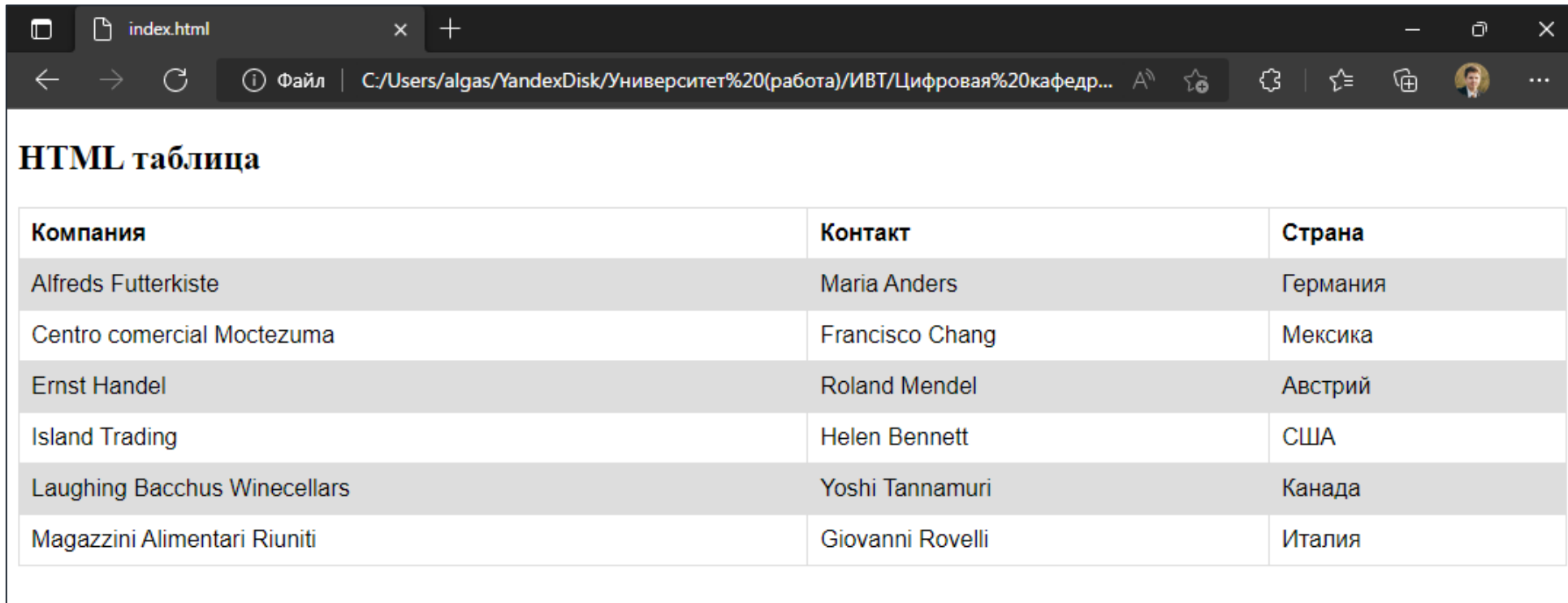
```
style.css X
style.css > ...
1 .blue {
2   color: blue;
3 }
4
```


Метод removeClass()



HTML таблицы

HTML таблицы позволяют веб-разработчикам упорядочивать данные в строки и столбцы.



The screenshot shows a web browser window with a single tab titled 'index.html'. The address bar shows the file path 'C:/Users/algas/YandexDisk/Университет%20(работа)/ИВТ/Цифровая%20кафедр...'. The page content features a heading 'HTML таблица' followed by a table with three columns: 'Компания', 'Контакт', and 'Страна'. The table contains seven rows of data, alternating between light gray and white background colors for each row.

Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика
Ernst Handel	Roland Mendel	Австрий
Island Trading	Helen Bennett	США
Laughing Bacchus Winecellars	Yoshi Tannamuri	Канада
Magazzini Alimentari Riuniti	Giovanni Rovelli	Италия

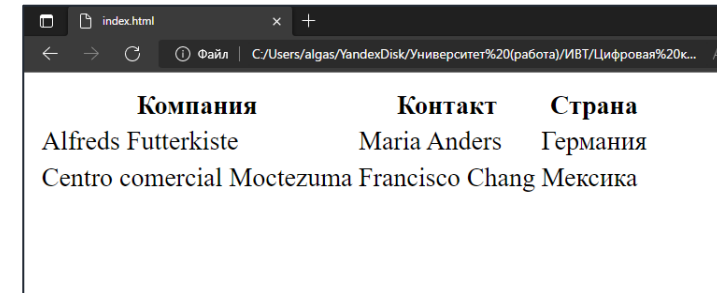
Ячейки таблицы

HTML таблица состоит из ячеек, находящихся внутри строк и столбцов.

Каждая ячейка таблицы определяется тегами **<td>** и **</td>**.

Все, что находится между **<td>** и **</td>**, является содержимым ячейки таблицы.

Ячейка таблицы может содержать всевозможные HTML элементы: текст, изображения, списки, ссылки, другие таблицы и т. д.



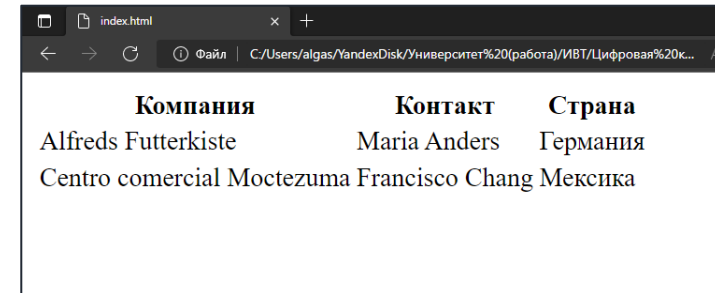
Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

```
8 <body>
9   <table>
10    <tr>
11      <th>Компания</th>
12      <th>Контакт</th>
13      <th>Страна</th>
14    </tr>
15    <tr>
16      <td>Alfreds Futterkiste</td>
17      <td>Maria Anders</td>
18      <td>Германия</td>
19    </tr>
20    <tr>
21      <td>Centro comercial Moctezuma</td>
22      <td>Francisco Chang</td>
23      <td>Мексика</td>
24    </tr>
25  </table>
26 </body>
```

Строки таблицы

Каждая строка таблицы начинается с тега `<tr>` и заканчивается тегом `</tr>`.

В таблице может быть сколько угодно строк; просто убедитесь, что количество ячеек одинаково в каждой строке.



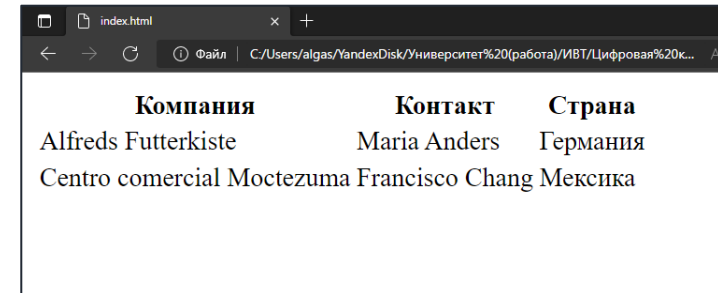
Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

```
8 <body>
9 <table>
10 <tr>
11   <th>Компания</th>
12   <th>Контакт</th>
13   <th>Страна</th>
14 </tr>
15 <tr>
16   <td>Alfreds Futterkiste</td>
17   <td>Maria Anders</td>
18   <td>Германия</td>
19 </tr>
20 <tr>
21   <td>Centro comercial Moctezuma</td>
22   <td>Francisco Chang</td>
23   <td>Мексика</td>
24 </tr>
25 </table>
26 </body>
```

Заголовки таблиц

Иногда вы хотите, чтобы ваши ячейки были ячейками заголовка таблицы. В этих случаях используйте тег **<th>** вместо тега **<td>**:

По умолчанию текст в элементах **<th>** выделяется жирным шрифтом и располагается по центру, но вы можете изменить это с помощью CSS.



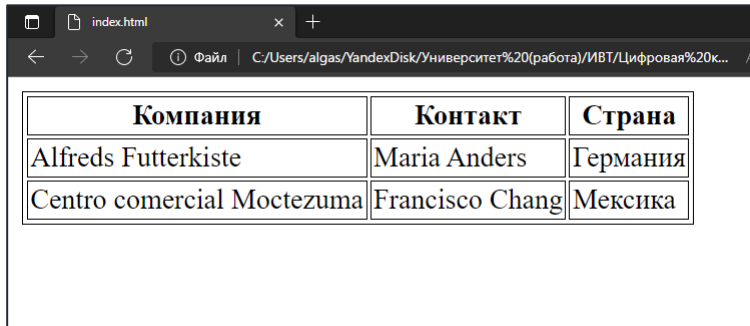
Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

```
8 <body>
9 <table>
10 <tr>
11   <th>Компания</th>
12   <th>Контакт</th>
13   <th>Страна</th>
14 </tr>
15 <tr>
16   <td>Alfreds Futterkiste</td>
17   <td>Maria Anders</td>
18   <td>Германия</td>
19 </tr>
20 <tr>
21   <td>Centro comercial Moctezuma</td>
22   <td>Francisco Chang</td>
23   <td>Мексика</td>
24 </tr>
25 </table>
26 </body>
```

Границы HTML таблиц

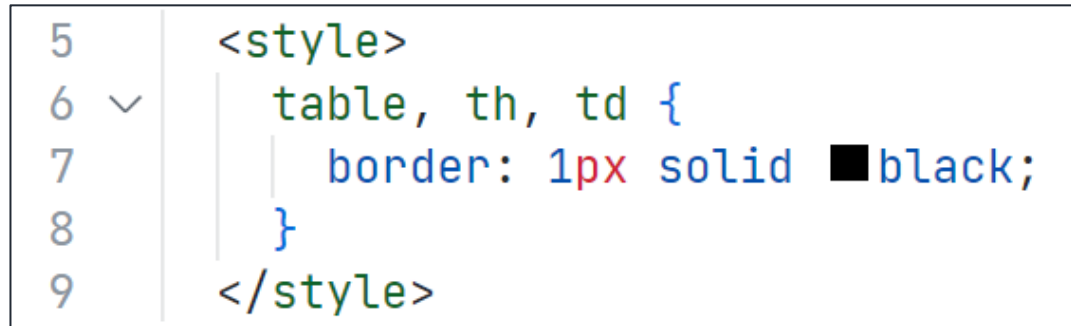
HTML таблицы могут иметь границы разных стилей и форм.

Когда вы добавляете границу к таблице, вы также добавляете границы вокруг каждой ячейки таблицы.



A screenshot of a web browser window displaying an HTML table. The browser's address bar shows the file path: C:/Users/algas/YandexDisk/Университет%20(работа)/ИВТ/Цифровая%20к... The table has three columns: 'Компания', 'Контакт', and 'Страна'. It contains two data rows. The table and its cells are outlined with a solid black border.

Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

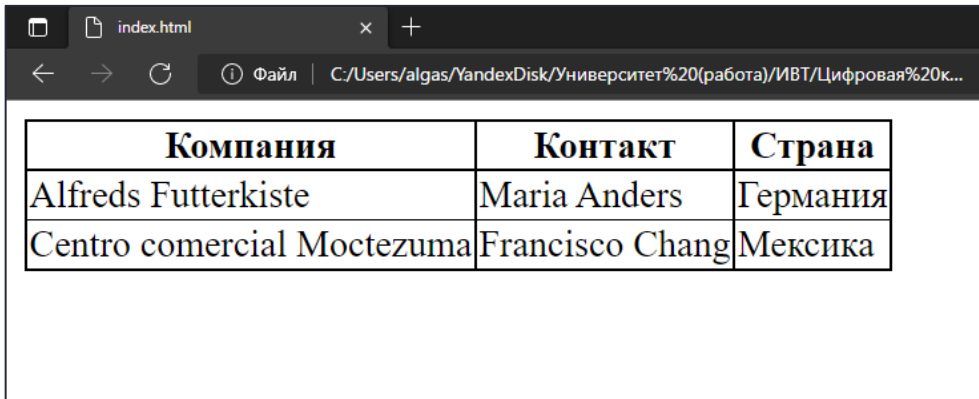


A code editor snippet showing CSS code to style a table. The code is as follows:

```
5 <style>
6   table, th, td {
7     border: 1px solid black;
8   }
9 </style>
```

Границы HTML таблиц

Чтобы избежать двойных границ, как в приведенном выше примере, установите для CSS свойства **border-collapse** значение «**collapse**».

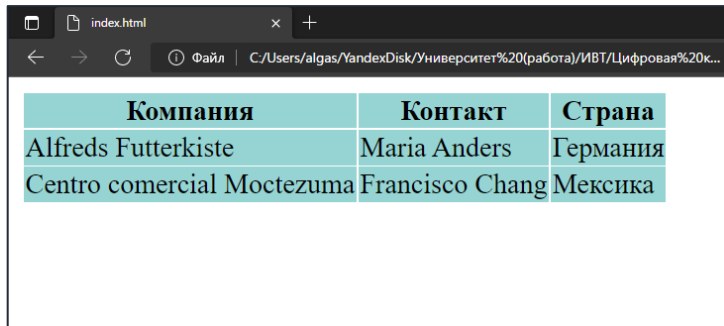


Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

```
5 <style>
6   table, th, td {
7     border: 1px solid black;
8     border-collapse: collapse;
9   }
10 </style>
```

Границы HTML таблиц

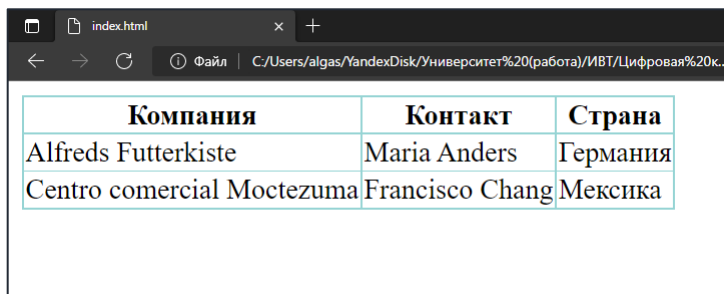
Если вы установите цвет фона для каждой ячейки и зададите границы белый цвет (такой же, как фон документа), вы получите эффект невидимой границы:



Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

```
5 <style>
6   table, th, td {
7     border: 1px solid white;
8     border-collapse: collapse;
9   }
10
11   th, td {
12     background-color: #96D4D4;
13   }
14 </style>
```

С помощью свойства **border-color** вы можете установить цвет границы.



Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

```
5 <style>
6   table, th, td {
7     border: 1px solid white;
8     border-collapse: collapse;
9   }
10
11   th, td {
12     border-color: #96D4D4;
13   }
14 </style>
```

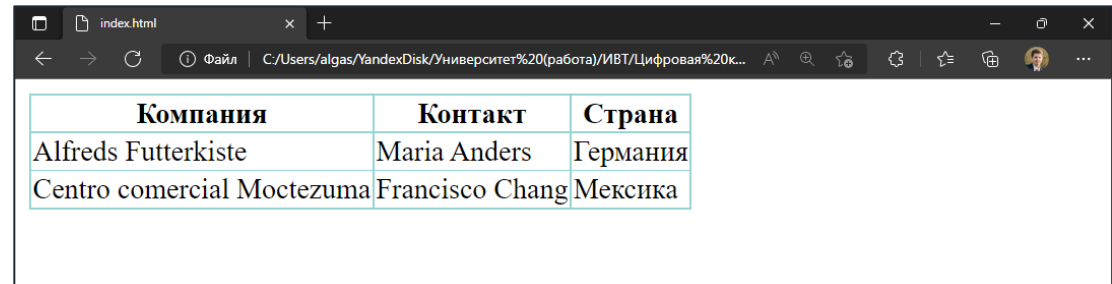

Размеры HTML-таблиц

HTML-таблицы могут иметь разные размеры для каждого столбца, строки или всей таблицы.

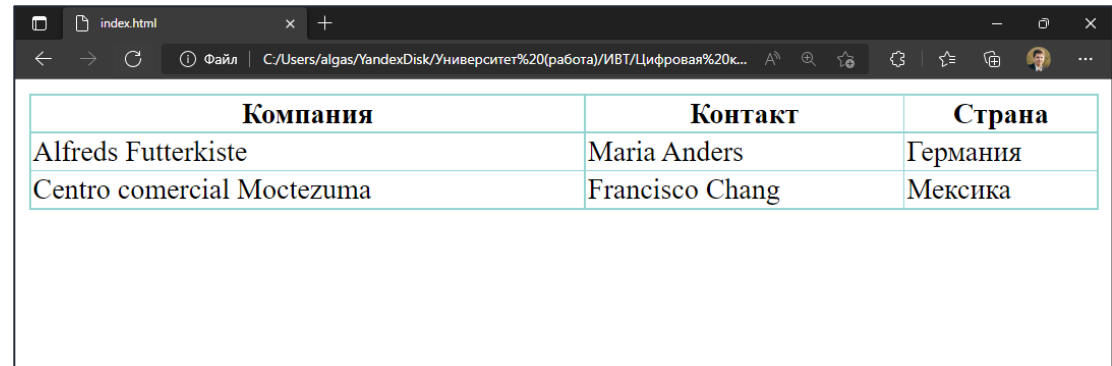
Размеры HTML-таблиц

Чтобы установить ширину таблицы, добавьте атрибут **style** к элементу **<table>**

```
20 <body>
21   <table style="width:100%">
22     <tr>
23       <th>Компания</th>
24       <th>Контакт</th>
25       <th>Страна</th>
26     </tr>
27     <tr>
28       <td>Alfreds Futterkiste</td>
29       <td>Maria Anders</td>
30       <td>Германия</td>
31     </tr>
32     <tr>
33       <td>Centro comercial Moctezuma</td>
34       <td>Francisco Chang</td>
35       <td>Мексика</td>
36     </tr>
37   </table>
38 </body>
```



Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

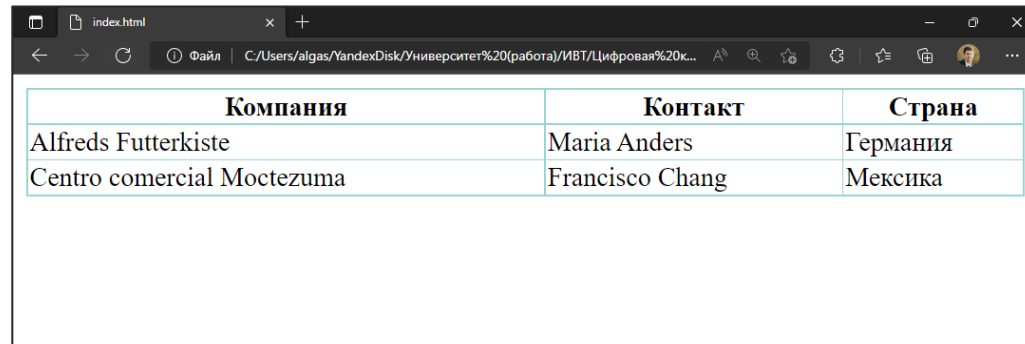


Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

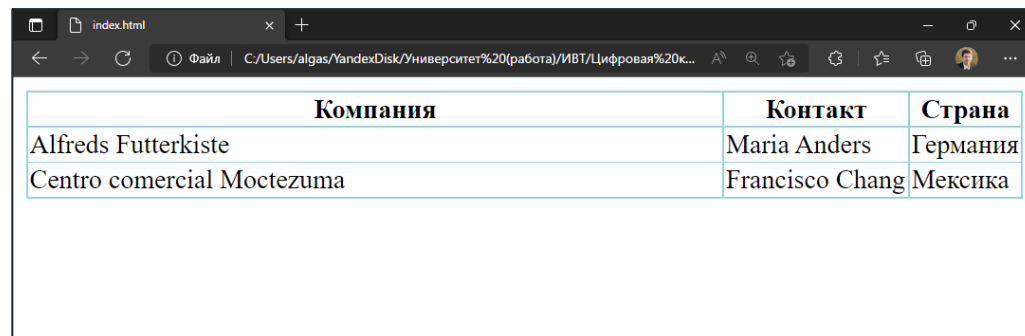
Размеры HTML-таблиц

Чтобы установить размер определенного столбца, добавьте атрибут **style** к элементу **<th>** или **<td>**

```
20 <body>
21   <table style="width:100%">
22     <tr>
23       <th style="width:70%">Компания</th>
24       <th>Контакт</th>
25       <th>Страна</th>
26     </tr>
27     <tr>
28       <td>Alfreds Futterkiste</td>
29       <td>Maria Anders</td>
30       <td>Германия</td>
31     </tr>
32     <tr>
33       <td>Centro comercial Moctezuma</td>
34       <td>Francisco Chang</td>
35       <td>Мексика</td>
36     </tr>
37   </table>
38 </body>
```



Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

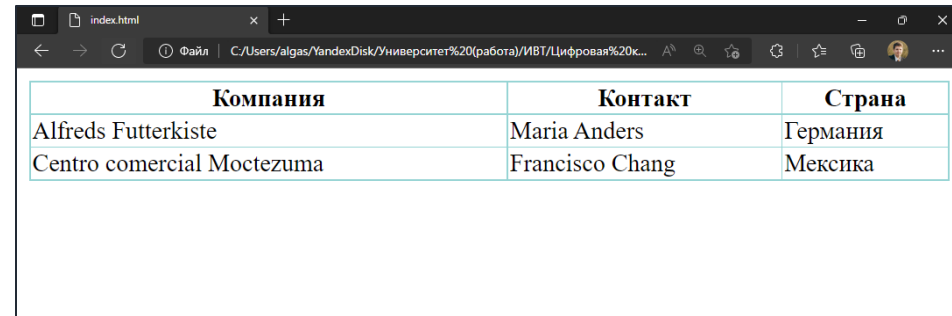


Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

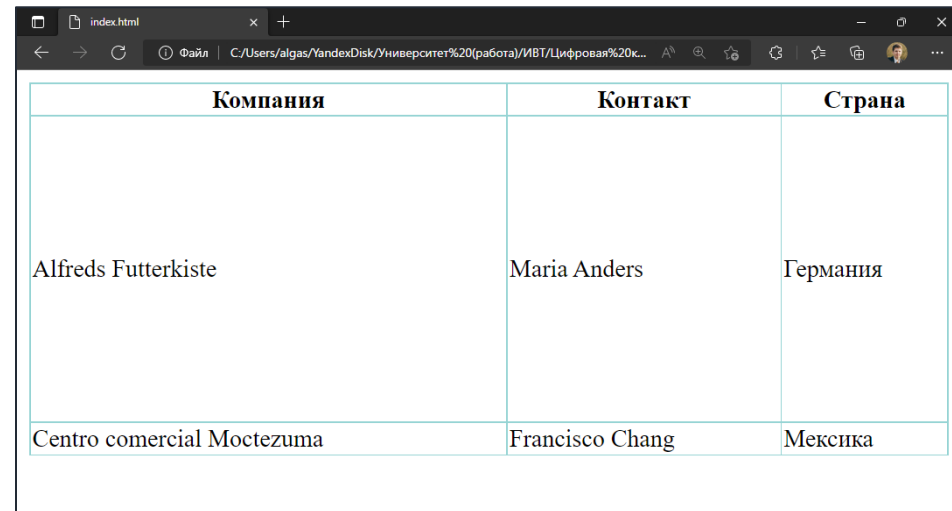
Размеры HTML-таблиц

Чтобы установить высоту определенной строки, добавьте атрибут **style** к элементу строки таблицы

```
20 <body>
21   <table style="width:100%">
22     <tr>
23       <th>Компания</th>
24       <th>Контакт</th>
25       <th>Страна</th>
26     </tr>
27     <tr style="height:200px">
28       <td>Alfreds Futterkiste</td>
29       <td>Maria Anders</td>
30       <td>Германия</td>
31     </tr>
32     <tr>
33       <td>Centro comercial Moctezuma</td>
34       <td>Francisco Chang</td>
35       <td>Мексика</td>
36     </tr>
37   </table>
38 </body>
```



Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика



Компания	Контакт	Страна
Alfreds Futterkiste	Maria Anders	Германия
Centro comercial Moctezuma	Francisco Chang	Мексика

Заголовки HTML таблиц

Таблицы HTML могут иметь заголовки для каждого столбца или строки или для нескольких столбцов/строк.

EMIL	TOBIAS	LINUS

8:00		
9:00		
10:00		
11:00		
12:00		
13:00		

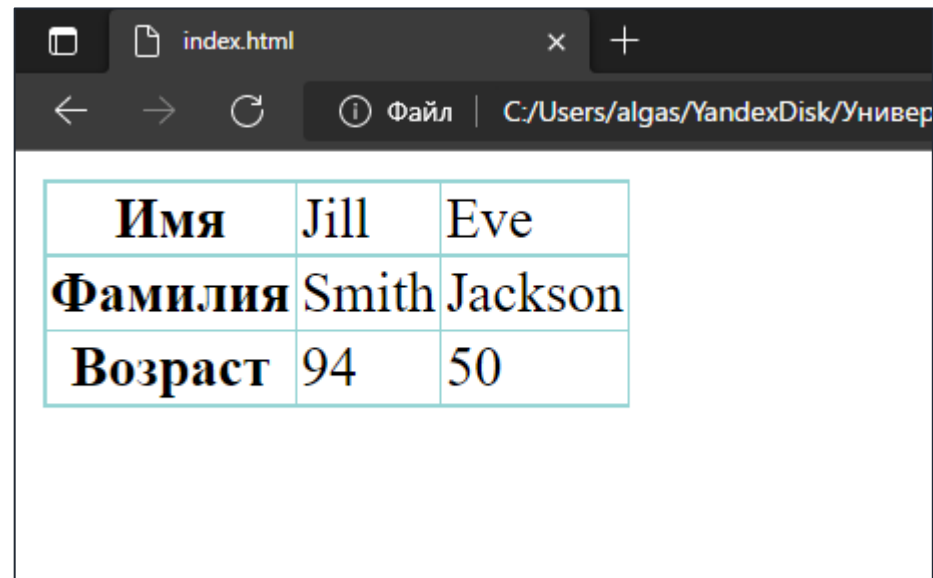
	MON	TUE	WED	THU	FRI
8:00					
9:00					
10:00					
11:00					
12:00					

DECEMBER		

Заголовки HTML таблиц

Чтобы использовать первый столбец в качестве заголовков таблицы, определите первую ячейку в каждой строке как элемент **<th>**

```
21 <table>
22   <tr>
23     <th>Имя</th>
24     <td>Jill</td>
25     <td>Eve</td>
26   </tr>
27   <tr>
28     <th>Фамилия</th>
29     <td>Smith</td>
30     <td>Jackson</td>
31   </tr>
32   <tr>
33     <th>Возраст</th>
34     <td>94</td>
35     <td>50</td>
36   </tr>
37 </table>
```



The screenshot shows a web browser window with the title 'index.html'. The address bar displays the file path 'C:/Users/algas/YandexDisk/Универ'. The rendered table is displayed below the browser interface.

Имя	Jill	Eve
Фамилия	Smith	Jackson
Возраст	94	50

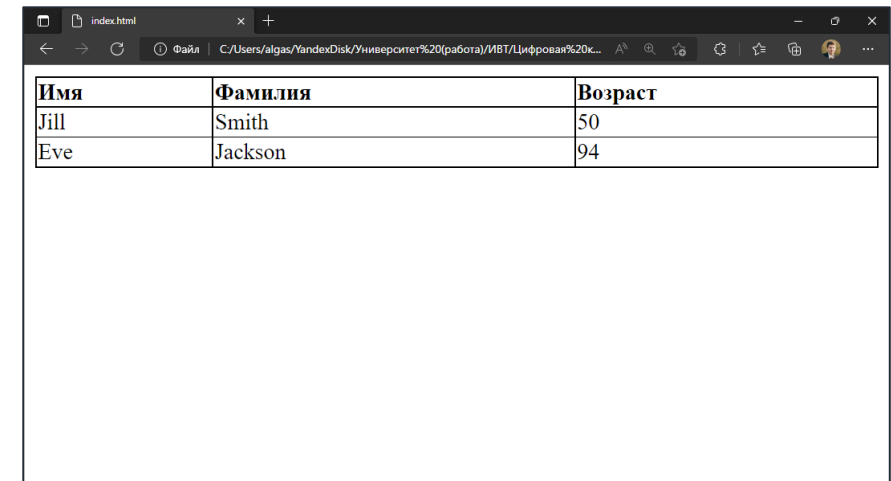
Заголовки HTML таблиц

По умолчанию заголовки таблиц выделены жирным шрифтом и расположены по центру

Чтобы выровнять заголовки таблицы по левому краю, используйте CSS свойство **text-align**

```
20 <table style="width:100%">
21   <tr>
22     <th>Имя</th>
23     <th>Фамилия</th>
24     <th>Возраст</th>
25   </tr>
26   <tr>
27     <td>Jill</td>
28     <td>Smith</td>
29     <td>50</td>
30   </tr>
31   <tr>
32     <td>Eve</td>
33     <td>Jackson</td>
34     <td>94</td>
35   </tr>
36 </table>
```

```
5 <style>
6   table, th, td {
7     border: 1px solid black;
8     border-collapse: collapse;
9   }
10
11   th {
12     text-align: left;
13   }
14 </style>
```



The screenshot shows a web browser window with a single tab titled 'index.html'. The address bar shows the file path 'C:/Users/algas/OneDrive/Рабочий стол/ИИТ/Цифровая%20...'. The rendered table is displayed with three columns: 'Имя', 'Фамилия', and 'Возраст'. The first row contains 'Jill', 'Smith', and '50'. The second row contains 'Eve', 'Jackson', and '94'. The table has a black border and collapsed borders between cells. The header row is bolded, and the text is left-aligned.

Имя	Фамилия	Возраст
Jill	Smith	50
Eve	Jackson	94

Заголовки HTML таблиц

У вас может быть заголовок, который охватывает два или более столбца.

Для этого используйте атрибут **colspan** элемента **<th>**

```
20 <table style="width:100%">
21   <tr>
22     <th colspan="2">Имя</th>
23     <th>Возраст</th>
24   </tr>
25   <tr>
26     <td>Jill</td>
27     <td>Smith</td>
28     <td>50</td>
29   </tr>
30   <tr>
31     <td>Eve</td>
32     <td>Jackson</td>
33     <td>94</td>
34   </tr>
35 </table>
```

index.html

⌵

+

⏪

⏩

🔄

🛂

📄

🔍

🔊

🔍

🌟

⚙️

🔖

🗂️

👤

⋮

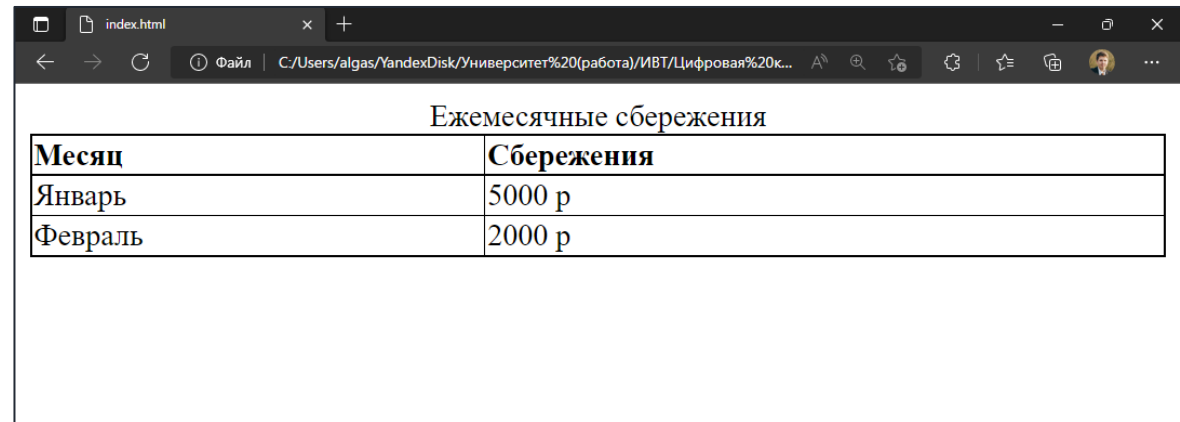
Имя		Возраст
Jill	Smith	50
Eve	Jackson	94

Заголовки HTML таблиц

Вы можете добавить заголовок, который будет служить заголовком для всей таблицы.

Чтобы добавить заголовок к таблице, используйте тег **<caption>**.

```
20 <table style="width:100%">
21   <caption>Ежемесячные сбережения</caption>
22   <tr>
23     <th>Месяц</th>
24     <th>Сбережения</th>
25   </tr>
26   <tr>
27     <td>Январь</td>
28     <td>5000 р</td>
29   </tr>
30   <tr>
31     <td>Февраль</td>
32     <td>2000 р</td>
33   </tr>
34 </table>
```



The screenshot shows a web browser window with the title 'index.html'. The address bar shows the file path 'C:/Users/algas/YandexDisk/Университет%20(работа)/ИИТ/Цифровая%20к...'. The browser displays the rendered HTML table, which has a caption 'Ежемесячные сбережения' centered above the table. The table has two columns: 'Месяц' and 'Сбережения'. The first row contains 'Январь' and '5000 р', and the second row contains 'Февраль' and '2000 р'.

Месяц	Сбережения
Январь	5000 р
Февраль	2000 р

Отступы в HTML таблицах

В HTML таблицах можно регулировать отступы внутри ячеек, а также пространство между ячейками.

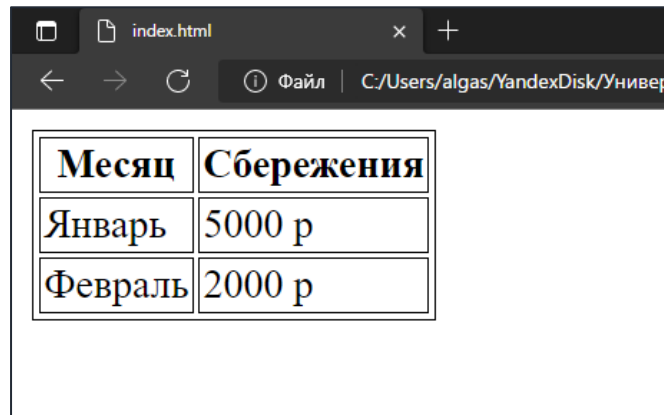
hello	hello	hello
hello	hello	hello
hello	hello	hello

hello	hello	hello
hello	hello	hello
hello	hello	hello

Отступы в HTML таблицах

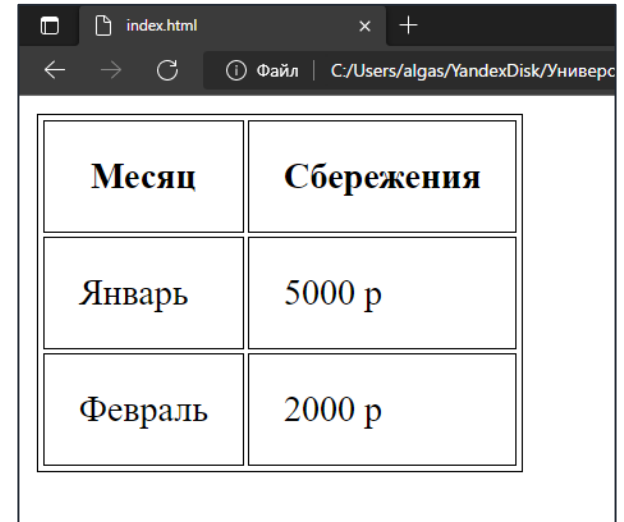
Чтобы добавить отступы к ячейкам таблицы, используйте CSS свойство **padding**

```
5  <style>
6    table, th, td {
7      border: 1px solid black;
8    }
9
10   th, td {
11     padding: 15px;
12   }
13 </style>
```



A screenshot of a web browser window displaying a table. The table has two columns: 'Месяц' (Month) and 'Сбережения' (Savings). The first row contains the headers. The second row shows 'Январь' (January) and '5000 р' (5000 rubles). The third row shows 'Февраль' (February) and '2000 р' (2000 rubles). The table cells have a black border and padding, making the text clearly visible.

Месяц	Сбережения
Январь	5000 р
Февраль	2000 р



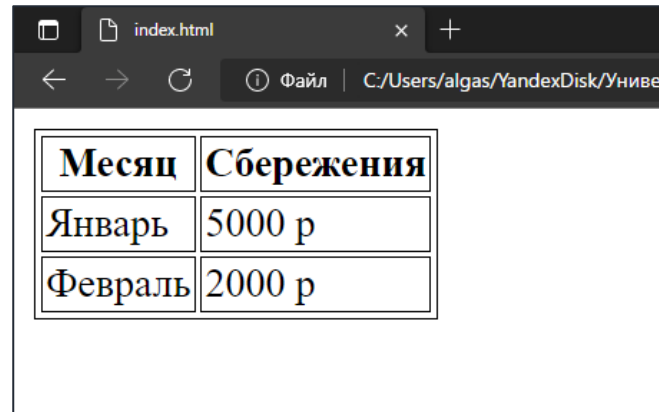
A screenshot of a web browser window displaying a table. The table has two columns: 'Месяц' (Month) and 'Сбережения' (Savings). The first row contains the headers. The second row shows 'Январь' (January) and '5000 р' (5000 rubles). The third row shows 'Февраль' (February) and '2000 р' (2000 rubles). The table cells have a black border and padding, making the text clearly visible.

Месяц	Сбережения
Январь	5000 р
Февраль	2000 р

Отступы в HTML таблицах

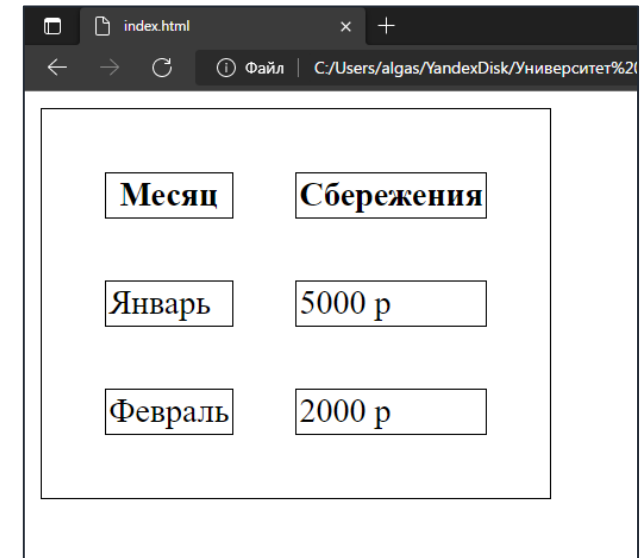
Чтобы изменить расстояние между ячейками таблицы, используйте CSS свойство **border-spacing** для элемента таблицы

```
5 <style>
6   table, th, td {
7     border: 1px solid black;
8   }
9
10  table {
11    border-spacing: 30px;
12  }
13 </style>
```



A screenshot of a web browser window displaying a table. The table has two columns: 'Месяц' (Month) and 'Сбережения' (Savings). The rows are: 'Январь' (January) with '5000 р' (5000 rubles), and 'Февраль' (February) with '2000 р' (2000 rubles). The cells are separated by a 30px gap, as defined by the CSS border-spacing property.

Месяц	Сбережения
Январь	5000 р
Февраль	2000 р



A screenshot of a web browser window displaying a table. The table has two columns: 'Месяц' (Month) and 'Сбережения' (Savings). The rows are: 'Январь' (January) with '5000 р' (5000 rubles), and 'Февраль' (February) with '2000 р' (2000 rubles). The cells are separated by a small default gap, as defined by the default border-spacing property.

Месяц	Сбережения
Январь	5000 р
Февраль	2000 р

Colspan и Rowspan

HTML таблицы могут иметь ячейки, которые охватывают несколько строк и/или столбцов.

NAME		

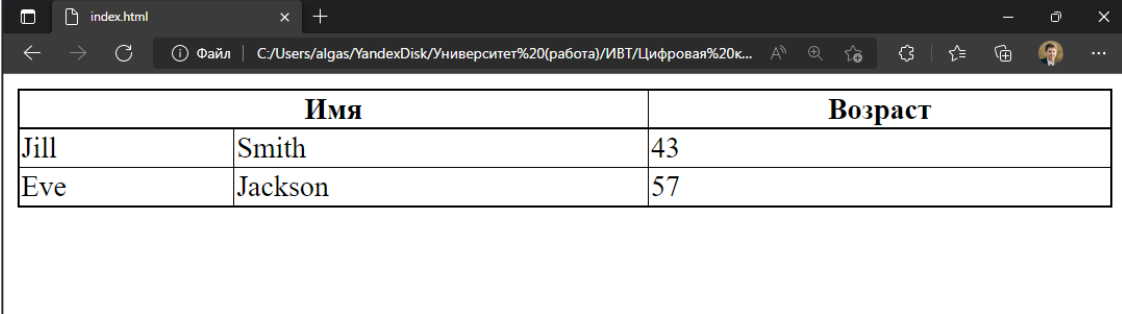
APRIL		

2022		
FIESTA		

Colspan и Rowspan

Чтобы создать диапазон ячеек для нескольких столбцов, используйте атрибут **colspan**

```
16 <table style="width: 100%">
17   <tr>
18     <th colspan="2">Имя</th>
19     <th>Возраст</th>
20   </tr>
21   <tr>
22     <td>Jill</td>
23     <td>Smith</td>
24     <td>43</td>
25   </tr>
26   <tr>
27     <td>Eve</td>
28     <td>Jackson</td>
29     <td>57</td>
30   </tr>
31 </table>
```

A screenshot of a web browser window showing the rendered HTML table. The browser's address bar shows the file path. The table has three columns: 'Имя' (Name), 'Имя' (Name), and 'Возраст' (Age). The first row contains 'Jill', 'Smith', and '43'. The second row contains 'Eve', 'Jackson', and '57'.

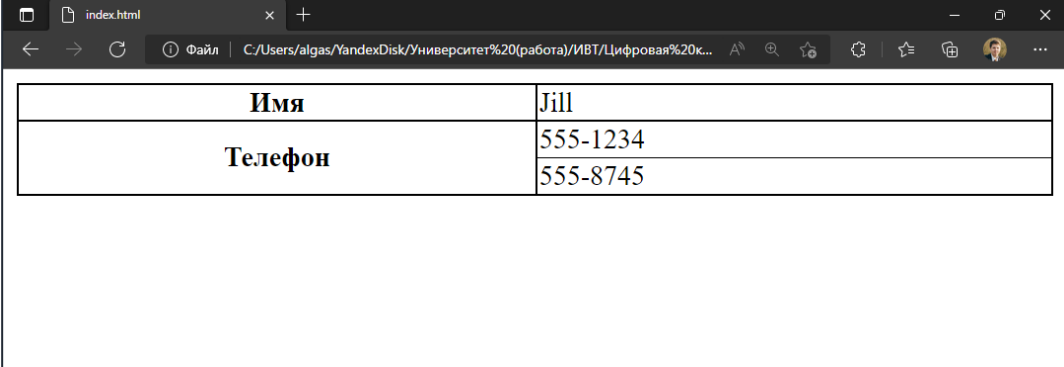
Имя		Возраст
Jill	Smith	43
Eve	Jackson	57

Значение атрибута **colspan** представляет количество столбцов, которые нужно охватить.

Colspan и Rowspan

Чтобы сделать ячейку, охватывающую несколько строк, используйте атрибут **rowspan**

```
16 <table style="width: 100%;">
17   <tr>
18     <th>Имя</th>
19     <td>Jill</td>
20   </tr>
21   <tr>
22     <th rowspan="2">Телефон</th>
23     <td>555-1234</td>
24   </tr>
25   <tr>
26     <td>555-8745</td>
27   </tr>
28 </table>
```



The screenshot shows a web browser window with a single tab titled 'index.html'. The address bar shows the file path 'C:/Users/algas/YandexDisk/Университет%20(работа)/ИВТ/Цифровая%20к...'. The browser displays a table with two columns. The first column has two rows: the first row has the header 'Имя' and the second row has the header 'Телефон'. The second column has three rows: the first row has the value 'Jill', the second row has the value '555-1234', and the third row has the value '555-8745'. The 'Телефон' header spans the two rows below it.

Имя	Jill
Телефон	555-1234
	555-8745

Значение атрибута **rowspan** представляет количество строк, которые нужно охватить.

Стилизация HTML таблиц

Используйте CSS, чтобы ваши таблицы выглядели лучше.

Если вы добавите цвет фона в каждую вторую строку таблицы, вы получите приятный эффект полос зебры.

Чтобы стилизовать любой другой элемент строки таблицы, используйте селектор **:nth-child(even)** следующим образом

```
5 <style>
6   table {
7     border-collapse: collapse;
8     width: 100%;
9   }
10
11   th, td {
12     text-align: left;
13     padding: 8px;
14   }
15
16   tr:nth-child(even) {
17     background-color: #D6EEEE;
18   }
19 </style>
```



Имя	Фамилия	Доход
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

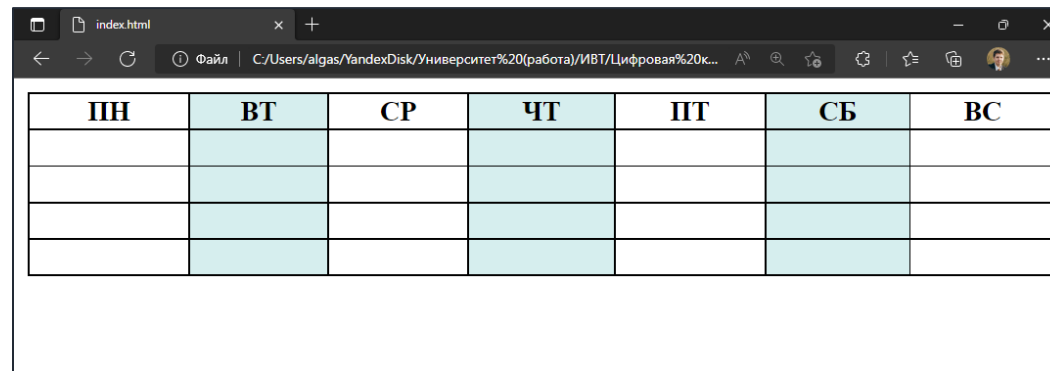
Если вы используете **(odd)** вместо **(even)**, стилизация будет происходить в строках 1,3,5 и т. д. вместо 2,4,6 и т. д.

Стилизация HTML таблиц

Чтобы сделать вертикальные полосы зебры, стилизуйте каждую вторую колонку, а не каждую вторую строку.

Установите **:nth-child(even)** для элементов данных таблицы следующим образом:

```
5  <style>
6    table, th, td {
7      border: 1px solid black;
8      border-collapse: collapse;
9    }
10
11    th:nth-child(even),
12    td:nth-child(even) {
13      background-color: #D6EEEE;
14    }
15  </style>
```



ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС

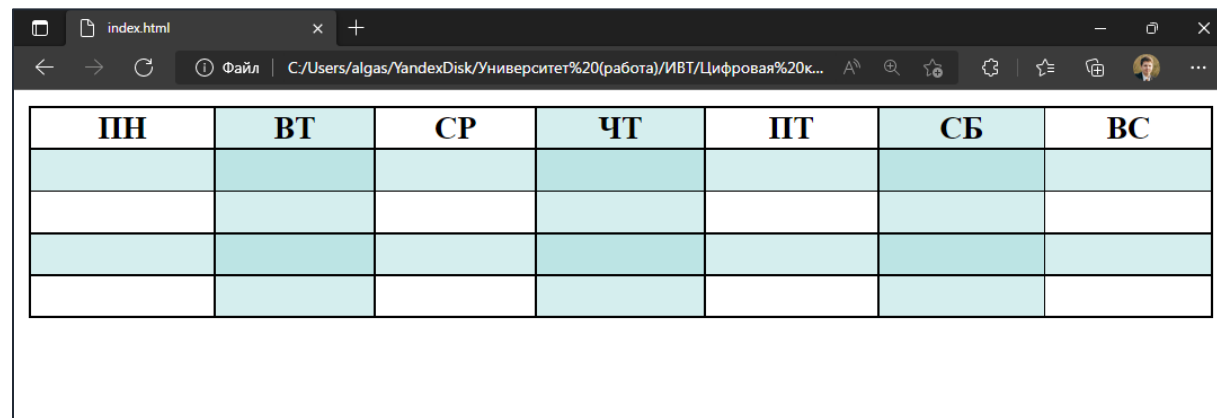
Поместите селектор **:nth-child()** на элементы **th** и **td**, если вы хотите, чтобы стили применялись как к заголовкам, так и к обычным ячейкам таблицы.

Стилизация HTML таблиц

Вы можете комбинировать стиль из двух приведенных выше примеров, и у вас будут полосы в каждой второй строке и каждом втором столбце.

Если вы используете прозрачный цвет, вы получите эффект перекрытия.

```
5 <style>
6   table, th, td {
7     border: 1px solid black;
8     border-collapse: collapse;
9   }
10
11   tr:nth-child(even) {
12     background-color: rgba(150, 212, 212, 0.4);
13   }
14
15   th:nth-child(even),
16   td:nth-child(even) {
17     background-color: rgba(150, 212, 212, 0.4);
18   }
19 </style>
```



The screenshot shows a web browser window with a single tab titled 'index.html'. The address bar shows the file path 'C:/Users/algas/YandexDisk/Университет%20(работа)/ИВТ/Цифровая%20к...'. The browser displays a table with 7 columns and 5 rows. The first row contains the headers 'ПН', 'ВТ', 'СР', 'ЧТ', 'ПТ', 'СБ', and 'ВС'. The table is styled with a light blue background for even-numbered rows and columns, and a white background for odd-numbered rows and columns. The borders are thin and black, and the text is centered in each cell.

ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС

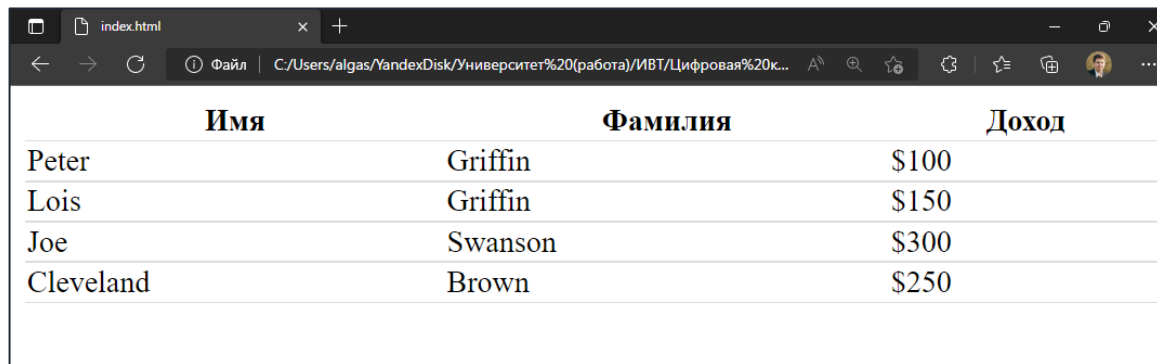
Используйте цвет **rgba()**, чтобы указать прозрачность цвета

Стилизация HTML таблиц

Если вы укажете границы только внизу каждой строки таблицы, у вас будет таблица с горизонтальными разделителями.

Добавьте свойство **border-bottom** ко всем элементам **tr**, чтобы получить горизонтальные разделители:

```
5 <style>
6   table {
7     border-collapse: collapse;
8     width: 100%;
9   }
10
11   tr {
12     border-bottom: 1px solid #ddd;
13   }
14 </style>
```



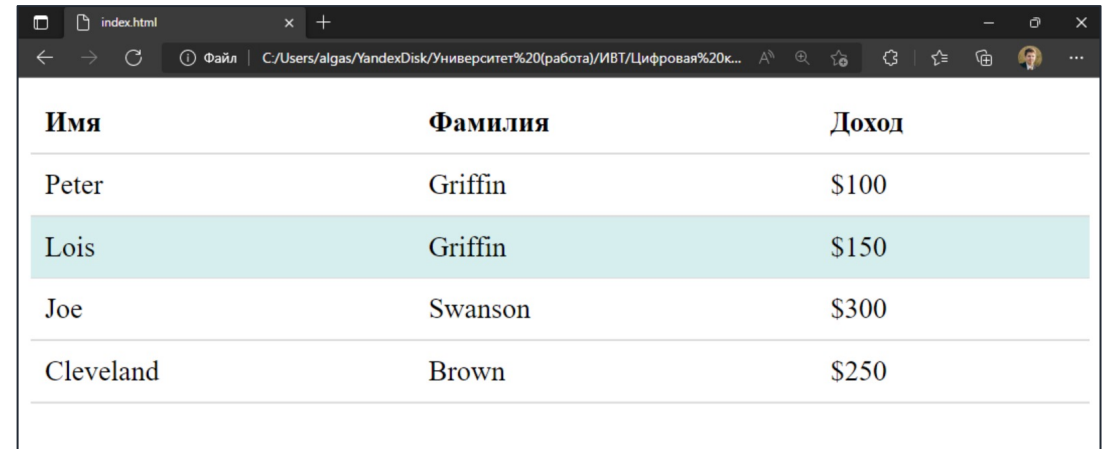
The screenshot shows a web browser window with a single tab titled 'index.html'. The address bar shows a local file path. The browser displays a table with three columns: 'Имя' (Name), 'Фамилия' (Surname), and 'Доход' (Income). The table has four data rows and one empty row at the bottom. Each row is separated by a horizontal border, which is the result of the CSS rule 'border-bottom: 1px solid #ddd;' applied to the 'tr' elements.

Имя	Фамилия	Доход
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

Стилизация HTML таблиц

Используйте селектор **:hover** на **tr**, чтобы выделить строки таблицы при наведении курсора мыши

```
5 <style>
6   table {
7     border-collapse: collapse;
8     width: 100%;
9   }
10
11   th, td {
12     padding: 8px;
13     text-align: left;
14     border-bottom: 1px solid #DDD;
15   }
16
17   tr:hover {
18     background-color: #D6EEEE;
19   }
20 </style>
```



Имя	Фамилия	Доход
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

Стилизация HTML таблиц

Элемент **<colgroup>** используется для стилизации определенных столбцов таблицы.

Если вы хотите стилизовать два первых столбца таблицы, используйте элементы **<colgroup>** и **<col>**.

Элемент **<colgroup>** следует использовать в качестве контейнера для спецификаций столбцов.

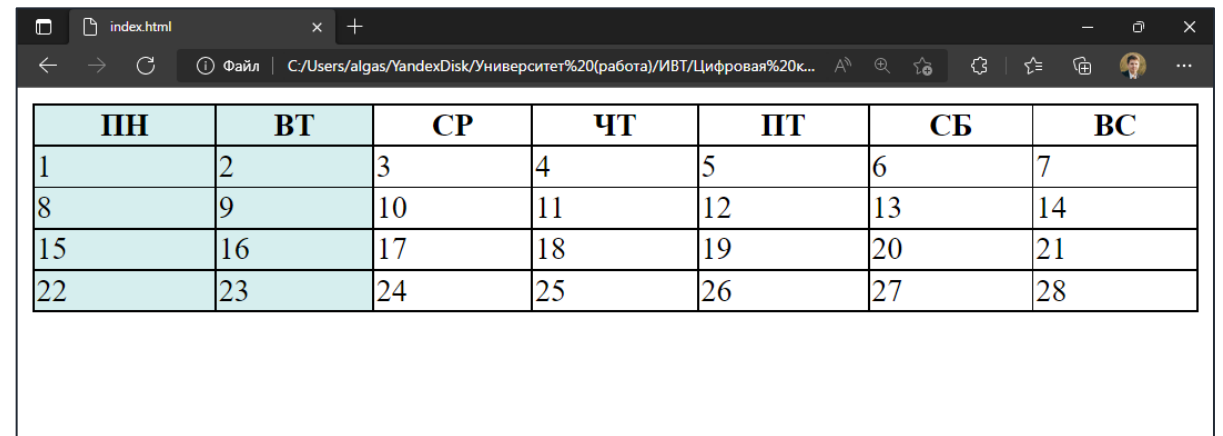
Каждая группа указывается с помощью элемента **<col>**.

Атрибут **span** указывает, сколько столбцов получает стиль.

Атрибут **style** указывает стиль, который будет задан для столбцов.

Стилизация HTML таблиц

```
16 <table style="width: 100%;">
17   <colgroup>
18     <col span="2" style="background-color: #D6EEEE">
19   </colgroup>
20   <tr>
21     <th>ПН</th>
22     <th>ВТ</th>
23     <th>СР</th>
24     <th>ЧТ</th>
25     <th>ПТ</th>
26     <th>СБ</th>
27     <th>ВС</th>
28   </tr>
29 >   <tr>...
37 </tr>
38 >   <tr>...
46 </tr>
47 >   <tr>...
55 </tr>
56 >   <tr>...
64 </tr>
65 </table>
```



ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Тег **<colgroup>** должен быть дочерним элементом элемента **<table>** и должен располагаться перед любыми другими элементами таблицы, такими как **<thead>**, **<tr>**, **<td>** и т. д., но после элемента **<caption>**, если он присутствует.

Стилизация HTML таблиц

Существует лишь очень ограниченный набор свойств CSS, которые разрешено использовать в **colgroup**.

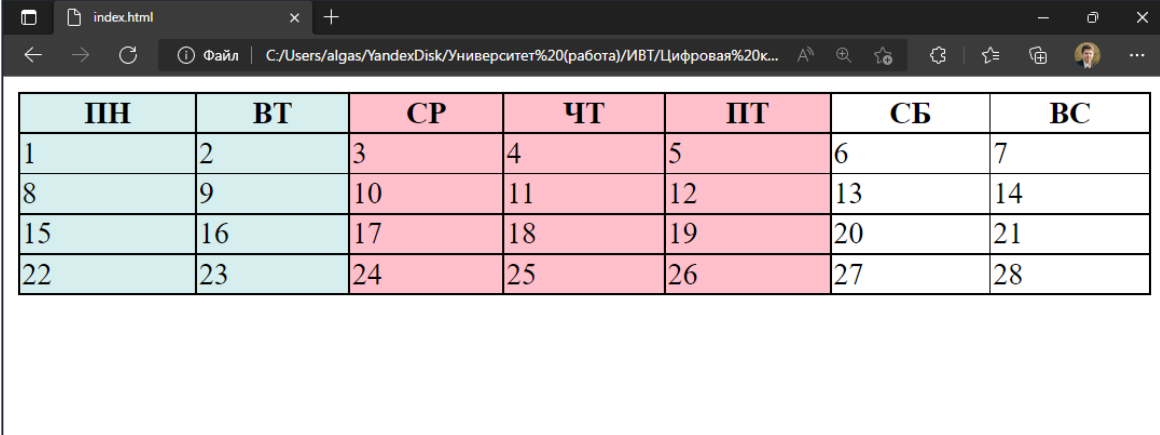
- width
- visibility
- background
- border

Все остальные свойства CSS не будут влиять на ваши таблицы.

Стилизация HTML таблиц

Если вы хотите оформить больше столбцов с разными стилями, используйте больше элементов **<col>** внутри **<colgroup>**.

```
16 <table style="width: 100%;">
17   <colgroup>
18     <col span="2" style="background-color: #D6EEEE">
19     <col span="3" style="background-color: pink">
20   </colgroup>
21   <tr>
22     <th>ПН</th>
23     <th>ВТ</th>
24     <th>СР</th>
25     <th>ЧТ</th>
26     <th>ПТ</th>
27     <th>СБ</th>
28     <th>ВС</th>
29   </tr>
30   <tr>...
38   </tr>
39   <tr>...
47   </tr>
48   <tr>...
56   </tr>
57   <tr>...
65   </tr>
66 </table>
```

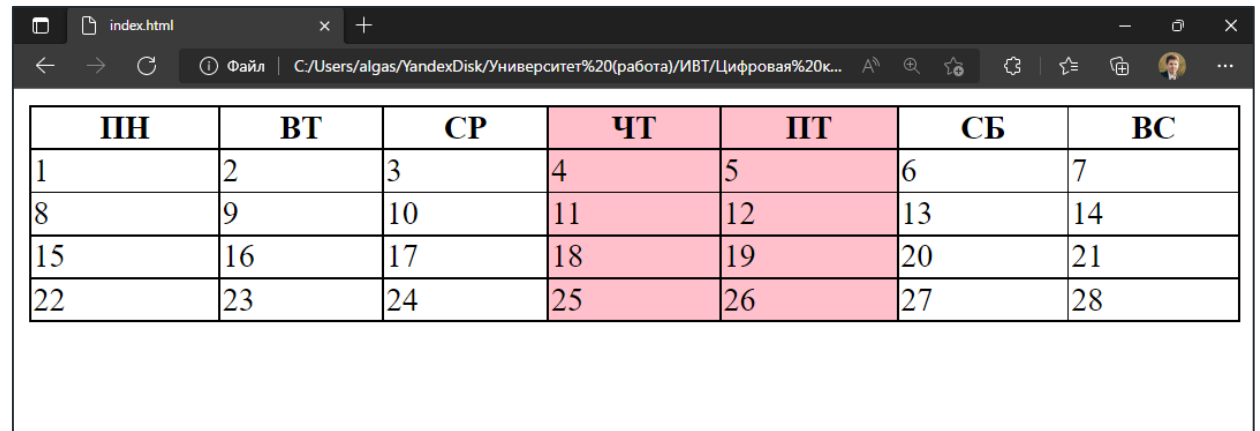


ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Стилизация HTML таблиц

Если вы хотите стилизовать столбцы в середине таблицы, вставьте «пустой» элемент <col> (без стилей) для столбцов в начале.

```
16 <table style="width: 100%;">
17   <colgroup>
18     <col span="3">
19     <col span="2" style="background-color: pink">
20   </colgroup>
21   <tr>
22     <th>ПН</th>
23     <th>ВТ</th>
24     <th>СР</th>
25     <th>ЧТ</th>
26     <th>ПТ</th>
27     <th>СБ</th>
28     <th>ВС</th>
29   </tr>
30   <tr>...
38   </tr>
39   <tr>...
47   </tr>
48   <tr>...
56   </tr>
57   <tr>...
65   </tr>
66 </table>
```

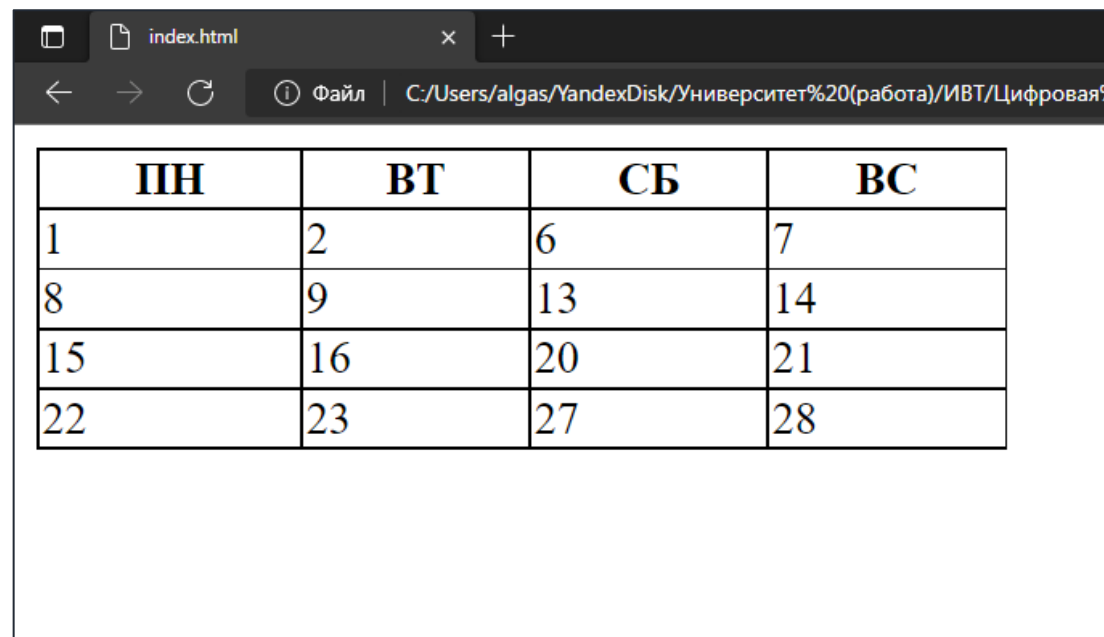


ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Стилизация HTML таблиц

Вы можете скрыть столбцы с помощью свойства **visible: collapse**.

```
16 <table style="width: 100%;">
17   <colgroup>
18     <col span="2">
19     <col span="3" style="visibility: collapse">
20   </colgroup>
21   <tr>
22     <th>ПН</th>
23     <th>ВТ</th>
24     <th>СР</th>
25     <th>ЧТ</th>
26     <th>ПТ</th>
27     <th>СБ</th>
28     <th>ВС</th>
29   </tr>
30 >   <tr>...
38   </tr>
39 >   <tr>...
47   </tr>
48 >   <tr>...
56   </tr>
57 >   <tr>...
65   </tr>
66 </table>
```



ПН	ВТ	СБ	ВС
1	2	6	7
8	9	13	14
15	16	20	21
22	23	27	28

Спасибо за внимание

Ссылка на демо с лекции: <https://github.com/Algashev1/3-4>