**Метод createElement**

Метод **createElement** позволяет создать новый элемент, передав в параметре имя тега. После создания с элементом можно работать как с обычным элементом, а также его можно добавить на страницу (например методом appendChild).

Если записать результат работы createElement в переменную, то в этой переменной будет такой элемент, как будто бы мы получили его через getElementById.

Единственное отличие - наш элемент не будет размещен на странице. А так мы можем менять ему innerHTML, атрибуты, навешивать обработчики событий и в конце концов разместить его на странице.

См. также методы appendChild и insertBefore, с помощью которых можно добавлять созданные элементы на страницу.

**Синтаксис**

document.createElement('имя тега');

**Примеры**

**Пример**

Давайте создадим абзац, установим ему текст и поместим на страницу в блок #parent:

<div id="parent"></div>

var parent **=** document.getElementById('parent');

var p **=** document.createElement('p');

p.innerHTML **=** 'Содержимое абзаца';

parent.appendChild(p);

HTML код элемента станет выглядеть так:

<div id="parent">

<p>Содержимое абзаца</p>

</div>

**Пример**

Дан ul. Давайте разместим в нем 10 li, при этом их текстом сделаем порядковые номера:

<ul id="parent"></ul>

var parent **=** document.getElementById('parent');

for (var i **=** 1; i **<=** 10; i**++**) {

var li **=** document.createElement('li');

li.innerHTML **=** i;

parent.appendChild(li);

}

HTML код элемента станет выглядеть так:

<ul id="parent">

<li>1</li>

<li>2</li>

<li>3</li>

<li>4</li>

<li>5</li>

<li>6</li>

<li>7</li>

<li>8</li>

<li>9</li>

<li>10</li>

</ul>

**Пример**

Давайте заполним таблицу tr-ками и td-шками:

<table id="table"></table>

var table **=** document.getElementById('table');

for (var i **=** 1; i **<=** 3; i**++**) {

var tr **=** document.createElement('tr'); //создаем tr-ку

//Заполняем tr-ку td-шками:

for (var j **=** 1; j **<=** 3; j**++**) {

var td **=** document.createElement('td'); //создаем td-шку

td.innerHTML **=** j; //пишем в нее текст

tr.appendChild(td); //добавляем созданную td-шку в конец tr-ки

}

table.appendChild(tr); //добавляем созданную tr-ку в конец таблицы

}

HTML код элемента станет выглядеть так:

<table id="table">

<tr>

<li>1</li>

<li>2</li>

<li>3</li>

</tr>

<tr>

<li>1</li>

<li>2</li>

<li>3</li>

</tr>

<tr>

<li>1</li>

<li>2</li>

<li>3</li>

</tr>

</table>

**Метод appendChild**

Метод **appendChild** позволяет вставить в конец элемента переданный элемент. Чаще всего используется после создания элемента с помощью createElement.

**Синтаксис**

родитель.appendChild(элемент);

**Примеры**

**Пример**

Давайте создадим абзац, установим ему текст и поместим на страницу в блок #parent:

<div id="parent"></div>

var parent **=** document.getElementById('parent');

var p **=** document.createElement('p');

p.innerHTML **=** 'Содержимое абзаца';

parent.appendChild(p);

HTML код элемента станет выглядеть так:

<div id="parent">

<p>Содержимое абзаца</p>

</div>

**Пример**

Дан ul. Давайте разместим в нем 10 li, при этом их текстом сделаем порядковые номера:

<ul id="parent"></ul>

var parent **=** document.getElementById('parent');

for (var i **=** 1; i **<=** 10; i**++**) {

var li **=** document.createElement('li');

li.innerHTML **=** i;

parent.appendChild(li);

}

HTML код элемента станет выглядеть так:

<ul id="parent">

<li>1</li>

<li>2</li>

<li>3</li>

<li>4</li>

<li>5</li>

<li>6</li>

<li>7</li>

<li>8</li>

<li>9</li>

<li>10</li>

</ul>

**Пример**

Давайте заполним таблицу tr-ками и td-шками:

<table id="table"></table>

var table **=** document.getElementById('table');

for (var i **=** 1; i **<=** 3; i**++**) {

var tr **=** document.createElement('tr'); //создаем tr-ку

//Заполняем tr-ку td-шками:

for (var j **=** 1; j **<=** 3; j**++**) {

var td **=** document.createElement('td'); //создаем td-шку

td.innerHTML **=** j; //пишем в нее текст

tr.appendChild(td); //добавляем созданную td-шку в конец tr-ки

}

table.appendChild(tr); //добавляем созданную tr-ку в конец таблицы

}

HTML код элемента станет выглядеть так:

<table id="table">

<tr>

<li>1</li>

<li>2</li>

<li>3</li>

</tr>

<tr>

<li>1</li>

<li>2</li>

<li>3</li>

</tr>

<tr>

<li>1</li>

<li>2</li>

<li>3</li>

</tr>

</table>

**Метод insertBefore**

Метод **insertBefore** позволяет вставить элемент перед другим элементом. Чаще всего используется после создания элемента с помощью createElement.

**Синтаксис**

родитель.insertBefore(элемент, перед кем вставить);

В данном случае родитель - это родитель того элемента, перед которым произойдет вставка.

**Примеры**

**Пример**

Создадим абзац и поместим его перед вторым абзацем:

<div id="parent">

<p>Абзац 1</p>

<p id="before">Абзац 2</p>

<p>Абзац 3</p>

</div>

var parent **=** document.getElementById('parent');

var before **=** document.getElementById('before');

var p **=** document.createElement('p');

p.innerHTML **=** 'Новый абзац';

parent.insertBefore(p, before);

HTML код элемента станет выглядеть так:

<div id="parent">

<p>Абзац 1</p>

<p>Новый абзац</p>

<p>Абзац 2</p>

<p>Абзац 3</p>

</div>

**Пример**

Добавим абзац в начало элемента #parent. Для этого вставим наш абзац перед первым потомком #parent. Этого потомка можно найти с помощью firstElementChild:

<div id="parent">

<p>Абзац 1</p>

<p>Абзац 2</p>

</div>

var parent **=** document.getElementById('parent');

var p **=** document.createElement('p');

p.innerHTML **=** 'Новый абзац';

parent.insertBefore(p, parent.firstElementChild);

HTML код элемента станет выглядеть так:

<div id="parent">

<p>Новый абзац</p>

<p>Абзац 1</p>

<p>Абзац 2</p>

</div>

**Пример**

При передаче вторым параметром null метод **insertBefore** срабатывает как **appendChild**. В то же время, если в элементе нет дочерних элементов, **firstElementChild** возвращает null. Следовательно, добавлять в начало элемента можно даже тогда, когда в нем нет дочерних элементов:

<div id="parent"></div>

var parent **=** document.getElementById('parent');

var p **=** document.createElement('p');

p.innerHTML **=** 'Новый абзац';

parent.insertBefore(p, parent.firstChild);

HTML код элемента станет выглядеть так:

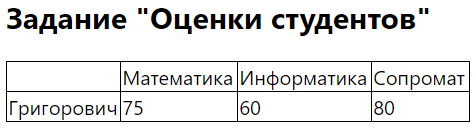
<div id="parent">

<p>Новый абзац</p>

</div>

**Задание**

Создайте HTML файл с таблицей.



В файле создайте массив с фамилиями и оценками по дисциплинам.

Сделайте так, чтобы при щелчке по строке таблицы под ней появлялась строка с информацией об очередном студенте. Данные о студенте берем их таблицы. Когда отобразятся все студенты должно появится всплывающее окне с надписью «Записи кончились».

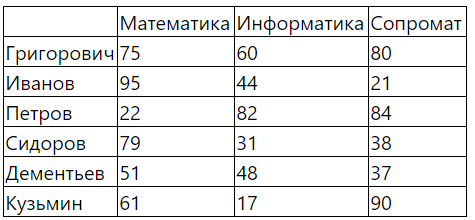


Таблица после работы скрипта.