***Вставка элементов***

***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.

document.createElement('имя тега');

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>

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>

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.

См. также метод insertBefore, который вставляет элементы в перед заданным элементом.  
См. также метод insertAdjacentHTML, который вставляет элементы в заданное место.  
См. также метод removeChild, который удаляет элементы.

родитель.appendChild(элемент);

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>

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>

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.

См. также метод appendChild, который вставляет элементы в конец родителя.  
См. также метод insertAdjacentHTML, который вставляет элементы в заданное место.  
См. также метод removeChild, который удаляет элементы.

родитель.insertBefore(элемент, перед кем вставить);

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

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>

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>

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>

***Метод cloneNode***

Метод cloneNode позволяет клонировать элемент и получить его точную копию. Эту копию затем можно вставить на страницу с помощью appendChild или других подобных методов.

В параметре метод получает true либо false. Если передан true, то элемент клонируется полностью, вместе со всем атрибутами и дочерними элементами, а если false - только сам элемент (без дочерних элементов).

См. также метод createElement, с помощью которого можно создать новый элемент.

элемент.cloneNode(true или false);

Сделаем копию блока с классом elem и вставим его в конец блока #parent:

<div id="parent">  
 <div class="elem">  
 <p>Первый абзац</p> <p>Второй абзац</p>  
 </div>  
</div>

var parent = document.getElementById('parent');  
var elem = parent.querySelector('.elem');  
var clone = elem.cloneNode(true);  
parent.appendChild(clone);

HTML код элемента станет выглядеть так:

<div id="parent">  
 <div class="elem">  
 <p>Первый абзац</p> <p>Второй абзац</p>  
 </div>  
 <div class="elem">  
 <p>Первый абзац</p> <p>Второй абзац</p>  
 </div>  
</div>

С полученным клоном можно работать как с обычным элементом:

<div id="parent">  
 <div class="elem"> <p>Первый абзац</p> <p>Второй абзац</p>  
 </div>  
</div>

var parent = document.getElementById('parent');  
var elem = parent.querySelector('.elem');  
var clone = elem.cloneNode(true);  
clone.children[0].innerHTML = 'Новое содержимое первого абзаца';  
clone.children[1].innerHTML = 'Новое содержимое второго абзаца';

***parent.appendChild(clone);***

HTML код элемента станет выглядеть так:

<div id="parent">  
 <div class="elem">  
 <p>Первый абзац</p>  
 <p>Второй абзац</p>  
 </div>  
 <div class="elem">  
 <p>Новое содержимое первого абзаца</p>  
 <p>Новое содержимое второго абзаца</p>  
 </div>  
</div>

***Метод removeChild***

Метод removeChild позволяет удалить элемент. Применяется к родительскому элементу с указанием элемента, который нужно удалить. См. также методы insertBefore, appendChild, которые выполняют вставку элементов.

***родитель.removeChild(элемент);***

Давайте удалим абзац #elem. Для этого получим его родителя #parent и метод removeChild применим к полученному родителю, а параметром метода передадим элемент, который хотим удалить:

<div id="parent">   
 <p>Первый абзац</p>  
 <p id="elem">Второй абзац</p>  
</div>

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

***parent.removeChild(elem);***

HTML код элемента станет выглядеть так:

<div id="parent"> <p>Первый абзац</p> </div>

Если нужно удалить элемент, но ссылки на его родителя нет, - родителя можно получить через parentNode:

<div> <p>Первый абзац</p> <p id="elem">Второй абзац</p> </div>

var elem = document.getElementById('elem');  
elem.parentNode.removeChild(elem);

HTML код элемента станет выглядеть так:

<div id="parent"> <p>Первый абзац</p> </div>

***Метод insertAdjacentHTML***

Метод insertAdjacentHTML позволяет вставить строку HTML кода в любое место страницы. Код вставляется относительно опорного элемента.

Можно сделать вставку перед опорным элементом (способ вставки beforeBegin), после него (способ вставки afterEnd), а также в начало (способ вставки afterBegin) или в конец (способ вставки beforeEnd) опорного элемента.

См. методы insertBefore и appendChild, которые также позволяют вставлять новые элементы.

опорный элемент.insertAdjacentHTML(способ вставки, код для вставки);

Пример . Способ beforeBegin Пусть опорный элемент - это элемент #target. Вставим перед ним новый абзац:

<div id="target"> <p>Абзац</p> </div>

var target = document.getElementById('target');  
target.insertAdjacentHTML('beforeBegin', '<p>Новый абзац</p>');

HTML код элемента станет выглядеть так:

<p>Новый абзац</p> <div id="target"> <p>Абзац</p> </div>

Пример . Способ afterEnd А теперь вставим новый абзац после опорного элемента:

<div id="target"> <p>Абзац</p> </div>

var target = document.getElementById('target');  
target.insertAdjacentHTML('afterEnd', '<p>Новый абзац</p>');

HTML код элемента станет выглядеть так:

<div id="target"> <p>Абзац</p> </div> <p>Новый абзац</p>

Пример . Способ afterBegin Вставим новый абзац в начало опорного элемента:

<div id="target"> <p>Абзац</p> </div>

var target = document.getElementById('target');  
target.insertAdjacentHTML('afterBegin', '<p>Новый абзац</p>');

HTML код элемента станет выглядеть так:

<div id="target"> <p>Новый абзац</p> <p>Абзац</p> </div>

Пример . Способ beforeEnd Вставим новый абзац в конец опорного элемента:

<div id="target"> <p>Абзац</p> </div>

var target = document.getElementById('target');  
target.insertAdjacentHTML('beforeEnd', '<p>Новый абзац</p>');

HTML код элемента станет выглядеть так:

<div id="target"> <p>Абзац</p> <p>Новый абзац</p> </div>