*Узлы nodes*

***Свойство childNodes***

Свойство childNodes хранит в себе псевдомассив непосредственных потомков элементов (включая комментарии и текстовые узлы).

См. также похожее свойство children, которое содержит только элементы

***элемент.childNodes***

Узнаем сколько потомков у элемента с id, равным 'parent' (включая комментарии, текстовые и прочие узлы):

<div id="parent"> <p>Абзац</p> <!-- Коментарий --> </div>

var parent = document.getElementById('parent');  
document.write(parent.childNodes.length);

Результат выполнения кода: 5

Свойство childNodes содержит всех потомков, в том числе и комментарии и текстовые узлы (текст, и отступы между тегами). Поэтому в предыдущем примере у элемента получилось 5 потомков.

Вот что произойдет, если убрать все отступы:

<div id="parent"><p>Абзац</p><!-- Коментарий --></div>

var parent = document.getElementById('parent');  
console.log(parent.childNodes.length);

Результат выполнения кода: 2

***Свойство firstChild***

Свойство firstChild хранит первого потомка данного элемента. Потомками являются не только элементы, но и просто текст, и комментарий HTML.

Если у элемента нет дочерних элементов - возвращается null.

См. похожее свойство firstElementChild, которое также содержит первого потомка, но комментарии и текст не учитываются.

***элемент.firstChild***

Давайте попробуем получить содержимое первого потомка элемента с id, равным 'parent'. Нас будет ждать подвох.

Суть подвоха: первый элемент из #parent - это абзац. Однако firstChild содержит не только элементы, но и текст (в том числе и пробельные символы, которые находится между тегами). Поэтому, если попробовать прочитать свойство innerHTML для первого потомка - мы увидим undefined, так как был получен текстовый узел, у которого нет свойства innerHTML:

<div id="parent"> <p>Первый элемент</p> <div>Последний элемент</div> </div>

var parent = document.getElementById('parent');  
document.write(parent.firstChild.innerHTML);

Результат выполнения кода: undefined

Давайте уберем все пробелы и отступы - результат будет совсем другим:

<div id="parent"><p>Первый элемент</p><div>Второй элемент</div></div>

var parent = document.getElementById('parent');  
document.write(parent.firstChild.innerHTML);

Результат выполнения кода: Первый элемент

***Свойство lastChild***

Свойство lastChild хранит последнего потомка данного элемента. Потомками являются не только элементы, но и просто текст, и комментарий HTML.

Если у элемента нет дочерних элементов - возвращается null.

См. также похожее свойство lastElementChild, которое также содержит последнего потомка, но комментарии и текст не учитываются.

элемент.lastChild

Свойство parentNode содержит родительский элемент.

Существует также почти идентичное свойство parentElement. Отличия: для тега htmlсвойство parentNode возвращает document, а parentElement возвращает null.

***элемент.parentNode***

Давайте получим родителя элемента #elem и выведем на экран его id:

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

var elem = document.getElementById('elem');  
var parent = elem.parentNode;  
document.write(parent.id);

Результат выполнения кода: parent

Для тега html свойство parentElement также возвращает родителя - это document:

var html = document.documentElement;  
document.write(html.parentNode);

Результат выполнения кода: document