

javascript node cheatsheet

node relationships

Property	Description
<code>node.children</code>	A collection of all of the element nodes contained within <code>node</code>
<code>node.firstChild</code>	The first element within <code>node</code>
<code>node.lastElementChild</code>	The last element within <code>node</code>
<code>node.nextElementSibling</code>	The next sibling element to <code>node</code>
<code>node.previousElementSibling</code>	The previous sibling element to <code>node</code>
<code>node.childElementCount</code>	The number of child elements within <code>node</code>
<code>node.children.length</code>	The number of child elements within <code>node</code>

element node relationships

Expression	Description
<code>node.firstChild</code>	The first child of <code>node</code>
<code>node.lastChild</code>	The last child of <code>node</code>
<code>node.childNodes</code>	A collection of all of the nodes that are direct children of <code>node</code>
<code>node.previousSibling</code>	The sibling prior to <code>node</code>
<code>node.nextSibling</code>	The sibling after <code>node</code>
<code>node.ownerDocument</code>	The root node of the document
<code>node.parentNode</code>	The parent of <code>node</code>

methods to create nodes

Method	Description
<code>document.createAttribute(<i>att</i>)</code>	Creates an attribute node with the name <i>att</i>
<code>document.createComment(<i>text</i>)</code>	Creates a comment node containing the comment <i>text</i>
<code>document.createElement(<i>elem</i>)</code>	Creates an element node with the name <i>elem</i>
<code>document.createTextNode(<i>text</i>)</code>	Creates a text node containing the text string <i>text</i>
<code>node.cloneNode(<i>deep</i>)</code>	Creates a copy of <i>node</i> where <i>deep</i> is a Boolean value that indicates whether to copy all descendants of <i>node</i> (true) or only <i>node</i> itself (false)

methods to append and replace nodes

Method	Description
<code>node.appendChild(<i>new</i>)</code>	Appends <i>new</i> node as a child of <i>node</i>
<code>node.insertBefore(<i>new</i>, <i>child</i>)</code>	Inserts <i>new</i> node directly before <i>child</i> node (if no <i>child</i> node is specified then <i>new</i> node is added as the last child node)
<code>node.normalize()</code>	Traverses all of the child nodes of <i>node</i> ; any adjacent text nodes are merged into a single text node
<code>node.removeChild(<i>old</i>)</code>	Removes <i>old</i> node from <i>node</i>
<code>node.replaceChild(<i>new</i>, <i>old</i>)</code>	Replaces <i>old</i> node with <i>new</i> node.

creating elements and text nodes

creates element nodes for the h1 and ol elements

```
function makeOutline() {
  // Location of the document outline
  var outline = document.getElementById("outline");

  // Source document for the outline
  var source = document.getElementById("doc");

  var mainHeading = document.createElement("h1");
  var outlineList = document.createElement("ol");
  var headingText = document.createTextNode("Outline");
}
```

creates a text node containing the text string "Outline"

attaching element and text nodes

```
var mainHeading = document.createElement("h1");
var outlineList = document.createElement("ol");
var headingText = document.createTextNode("Outline");

mainHeading.appendChild(headingText);
outline.appendChild(mainHeading);
outline.appendChild(outlineList);
}
```

appends the text node to the h1 element

appends an ordered list to the outline

appends the h1 heading to the outline