Creating & Appending Nodes

* document.createElement(element) Makes a new element
* After creating an element it doesn’t exists in the page but it’s in memory
* To add the element with the document use appendChild()
* node.appendChild(element) Adds element inside a node

Controlling Node insertions

* appendChild() is great way to add element with the document but it has limitations
* appendChild() lacks precision
* Need to insert a node anywhere in the node list
* Use insertBefore() for surgical insertions
* The insertBefore() method inserts a node as a child, right before an existing child, which you specify - node.insertBefore(newnode,existingnode)

Cloning & removing nodes

* Make a copy of a node is called cloning
* cloneNode() makes a copy
* We can then reposition the node anywhere in the any existing dom
* removeChild(node) removes the node
* removeChild() has to be called from a parent node
* Cloning is right something big
* The Node.cloneNode() method returns a duplicate of the node on which this method was called.
* The cloneNode() method creates an exact copy of a specified node.
* We should be copy the element to create a new copy of the element & use it in another without copying the original element repositioned and lost its actual position

Remove Child

* The removeChild() method removes a specified child node of the specified element.
* Use the appendChild() or insertBefore() method to insert the removed node into the same document.
* To insert it to another document, use the document.adoptNode() or document.importNode() method
* node.removeChild(node)
* Remove all child nodes of a list:

// Get the <ul> element with id="myList"

var list = document.getElementById("myList");

// As long as <ul> has a child node, remove it

while (list.hasChildNodes()) {

list.removeChild(list.firstChild);

}

* Remove a <li> element with id="myLI" from its parent element (without specifying its parent node):

var item = document.getElementById("myLI");

item.parentNode.removeChild(item);

* Remove a <li> element from its parent, and insert it again:

var item = document.getElementById("myLI");

function removeLi() {

item.parentNode.removeChild(item);

}

function appendLi() {

var list = document.getElementById("myList");

list.appendChild(item);

}

* Remove a <span> element from its parent and insert it to an <h1> element in another document:

var child = document.getElementById("mySpan");

function removeLi() {

child.parentNode.removeChild(child);

}

function myFunction() {

var frame = document.getElementsByTagName("IFRAME")[0]

var h = frame.contentWindow.document.getElementsByTagName("H1")[0];

var x = document.adoptNode(child);

h.appendChild(x);

}

* The Node.removeChild() method removes a child node from the DOM. Returns removed node.

Replace Child

* replaceChild() – takes a node & replaces a node with another node
* We must call it from the parent node
* Saves us the step of having to delete the original