

JS DOM Manipulation Cheat Sheet

By Web Dev Simplified <https://courses.webdevsimplified.com>

Add Elements

Name	Return Type	Description	Results
createElement <code>document.createElement("div")</code>	Element	Create a new element of the type passed to createElement but does not add it to the DOM <i>Create a new div element Only works on the document</i>	div
cloneNode <code>div.cloneNode(true)</code>	Node	Create a copy of the given node but does not add it to the DOM. If true is passed to cloneNode it will also copy all the nodes inside the copied node. <i>Create a copy of the div with all children included</i>	div b a p
prepend <code>div.prepend(a, "t")</code>	Undefined	Append the list of nodes or strings to the start of parent element <i>Append the a tag and the text "t" to the start of the div</i>	div a "t" p
append <code>div.append(a, "t")</code>	Undefined	Append the list of nodes or strings to the end of parent element <i>Append the a tag and the text "t" to the end of the div</i>	div p a "t"
appendChild <code>div.appendChild(a)</code>	Node Return added node	Append the node to the end of parent element. This can only take one node and does not accept strings. <i>Append the a tag to the end of the div</i>	div p a
after <code>p.after(a, "t")</code>	Undefined	Insert the list of nodes or strings after the specified node <i>Insert the a tag and the text "t" after the p tag</i>	div p a "t"
before <code>p.before(a, "t")</code>	Undefined	Insert the list of nodes or strings after the specified node <i>Insert the a tag and the text "t" after the p tag</i>	div a "t" p
insertBefore <code>div.insertBefore(a, p)</code>	Node Return added node	Insert the node passed as the first parameter before the node passed as the second parameter in the specified parent <i>Insert the a tag before the p tag inside the div</i>	div b a p

Remove Elements

Name	Return Type	Description	Results
remove <code>a.remove()</code>	Undefined	Remove the element from the DOM Remove the a tag element	div a b
removeChild <code>div.removeChild(a)</code>	Node Return removed node	Remove the node from the specified parent Remove the a tag from the div	div p a "t"
replaceWith <code>a.replaceWith(b, "t")</code>	Undefined	Replace the specified node with the nodes and strings passed to replaceWith Replace the a tag with the tag and the text "t"	div a b "t"
replaceChild <code>div.replaceChild(b, a)</code>	Node Return removed node	Replace the node passed as the second parameter with the node passed as the first parameter inside the specified parent. This can replace one node and can <i>not</i> accept strings. Replace the a tag with the b tag inside the div	div a b
replaceChildren <code>div.replaceChildren(b, "t")</code>	Undefined	Replace all children of the specified node with the nodes and strings passed to replaceChildren Replace the children of the div with the b tag and the text "t"	div a b "t"

Modify CSS

Name	Return Type	Description
style <code>a.style.backgroundColor = "red"</code>	Undefined	Set CSS styles on the given element. CSS properties must be declared in camelCase. Set the background-color CSS property of the a tag to red
classList.add <code>a.classList.add("active")</code>	Undefined	Add a class to the element Add the class active to the a tag
classList.remove <code>a.classList.remove("active")</code>	Undefined	Remove a class from the element Remove the class active to the a tag
classList.toggle <code>a.classList.toggle("active")</code> <code>a.classList.toggle("active", true)</code>	Boolean Return true if the class is on the element after toggle runs	Remove the class if it is already on the element or add the class if it is not on the element. If the second parameter is passed to toggle then it will always add the class when the second parameter is true and always remove when it is false. Toggle the class active on the a tag Add the class active to the a tag
classList.replace <code>a.classList.replace("open", "close")</code>	Boolean Return true if the class was replaced	Replace the first class with the second class on the element. If the first class does not exist on the element then do not do the replacement. Replace the class open with close on the a tag

Modify Content

Name	Return Type	Description
innerText <code>a.innerText = "Text"</code> <code>const text = a.innerText</code>	String	Replace the text of the element with the given text, or get the value of the text in the element. This only returns text that is visible on the screen since it takes into account CSS styles. Set the text of the a tag to "text" Get the visible text inside the a tag
textContent <code>a.textContent = "Text"</code> <code>const text = a.textContent</code>	String	Replace the text of the element with the given text, or get the value of the text in the element. This returns the raw text even if it is not visible on the screen. Set the text of the a tag to "text" Get the raw text inside the a tag
innerHTML <code>a.innerHTML = "Text"</code> <code>const html = a.innerHTML</code>	String	Replace the HTML of the element with the given HTML string, or get the value of the HTML in the element. This inserts new HTML in the page so can lead to cross site scripting vulnerabilities. Set the HTML of the a tag to a span with the text "text" inside of it Get the HTML inside the a tag

Modify Attributes

Name	Return Type	Description
dataset <code>a.dataset.color = "red"</code> <code>const color = a.dataset.color</code> <code>delete a.dataset.color</code>	String	Read, write, and delete the value of data attributes on an element Set the value of the data-color attribute on the a tag to red Get the value of the data-color attribute on the a tag Delete the data-color attribute from the a tag
getAttribute <code>a.getAttribute("title")</code>	String	Get the value of the specified attribute on the element Get the value of the title attribute on the a tag
setAttribute <code>a.setAttribute("title", "Text")</code>	Undefined	Set the value of the specified attribute on the element to the given value Set the value of the title attribute on the a tag to "Text"
removeAttribute <code>a.removeAttribute("title")</code>	Undefined	Remove the specified attribute from the element Remove the title attribute from the a tag
toggleAttribute <code>a.toggleAttribute("active")</code> <code>a.toggleAttribute("active", true)</code>	Boolean Return true if the attribute is on the element after toggle	Remove the attribute if it is already on the element or add the attribute if it is not on the element. If the second parameter is passed to toggleAttribute then it will always add the attribute when the second parameter is true and always remove when it is false. Toggle the active attribute on the a tag Add the active attribute to the a tag