JS History

JS HTML DOM

DOM Intro
DOM Methods
DOM Document
DOM Elements
DOM HTML
DOM Forms
DOM CSS
DOM Animations
DOM Events
DOM Event Listener
DOM Navigation
DOM Nodes
DOM Collections
DOM Node Lists

JS Browser BOM

JS Window
JS Screen
JS Location

JavaScript HTML DOM Node Lists

‹ Previous Next ›

The HTML DOM NodeList Object

A `NodeList` object is a list (collection) of nodes extracted from a document.

A `NodeList` object is almost the same as an `HTMLCollection` object.

Some (older) browsers return a NodeList object instead of an HTMLCollection for methods like `getElementsByClassName()`.

All browsers return a NodeList object for the property `childNodes`.

Most browsers return a NodeList object for the method `querySelectorAll()`.

The following code selects all `<p>` nodes in a document:

Example

```
const myNodeList = document.querySelectorAll("p");
```

The elements in the NodeList can be accessed by an index number.

To access the second <p> node you can write:

```
myNodeList[1]
```

Try it Yourself »

Note: The index starts at 0.

HTML DOM Node List Length

The `length` property defines the number of nodes in a node list:

Example

```
myNodelist.length
```

Try it Yourself »

The `length` property is useful when you want to loop through the nodes in a node list:

Example

Change the color of all <p> elements in a node list:

```
const myNodelist = document.querySelectorAll("p");
for (let i = 0; i < myNodelist.length; i++) {
  myNodelist[i].style.color = "red";
}
```

Try it Yourself »

The Difference Between an HTMLCollection and a NodeList

An `HTMLCollection` (previous chapter) is a collection of HTML elements.

A `NodeList` is a collection of document nodes.

A NodeList and an HTML collection is very much the same thing.

Both an HTMLCollection object and a NodeList object is an array-like list (collection) of objects.

Both have a length property defining the number of items in the list (collection).

Both provide an index (0, 1, 2, 3, 4, ...) to access each item like an array.

HTMLCollection items can be accessed by their name, id, or index number.

NodeList items can only be accessed by their index number.

Only the NodeList object can contain attribute nodes and text nodes.

A node list is not an array!

A node list may look like an array, but it is not.

You can loop through the node list and refer to its nodes like an array.

However, you cannot use Array Methods, like valueOf(), push(), pop(), or join() on a node list.

‹ Previous Next ›