

# Get next / previous element using JavaScript?

Asked 15 years, 10 months ago    Modified 2 years, 6 months ago

Viewed 508k times



183



javascript

html

dom



How do I get the next element in HTML using JavaScript?

Suppose I have three `<div>` s and I get a reference to one in JavaScript code, I want to get which is the next `<div>` and which is the previous.

Share

Improve this question

Follow

edited Feb 6, 2018 at 14:54



[informatik01](#)

16.4k ● 11 ● 78 ● 108

asked Feb 22, 2009 at 13:14



[Amr Elgarhy](#)

68.8k ● 70 ● 192 ● 310

10 Answers

Sorted by:

Highest score (default)





334



use the `nextSibling` and `previousSibling` properties:

```
<div id="foo1"></div>
<div id="foo2"></div>
<div id="foo3"></div>
```

```
document.getElementById('foo2').nextSibling; // #foo3
document.getElementById('foo2').previousSibling; // #f
```

However in some browsers (I forget which) you also need to check for whitespace and comment nodes:

```
var div = document.getElementById('foo2');
var nextSibling = div.nextSibling;
while(nextSibling && nextSibling.nodeType !== 1) {
    nextSibling = nextSibling.nextSibling
}
```

Libraries like jQuery handle all these cross-browser checks for you out of the box.

Share Improve this answer

Follow

edited Jun 27, 2018 at 5:17



WD40

458 ● 1 ● 9 ● 20

answered Feb 22, 2009 at 13:24




Crescent Fresh

117k ● 27 ● 157 ● 140

---


`nextSibling` and `previousSibling` (should) always return all nodes, not just element nodes shouldn't they? This includes text nodes, comment nodes, etc. Your second example looks like a good solution for this! Maybe you could cut

down on the repetition with a `do {} while();` loop? (edit: or recursion perhaps) – [thomasrutter](#) Mar 27, 2010 at 11:04 

---

113 [nextElementSibling](#) is much more useful. – [Trisped](#) Mar 14, 2012 at 22:39

---

9 Note that neither `nextSibling` nor `nextElementSibling` are fully cross browser compatible. Firefox's `nextSibling` returns text nodes while IE doesn't and `nextElementsibling` is not implemented until IE9. – [Carl Onager](#) Jun 18, 2012 at 8:50 

---

3 This will not work if the elements are not siblings. – [rvighne](#) Jan 12, 2014 at 3:22

---

2 @Kloar: The question asked how to get the "next [div] element in html." The next `div` in the markup might not be adjacent to the element; it might be a level deeper or a level higher. – [rvighne](#) May 27, 2014 at 21:54

---



Its quite simple. Try this instead:

47



```
let myReferenceDiv = document.getElementById('mydiv');
let prev = myReferenceDiv.previousElementSibling;
let next = myReferenceDiv.nextElementSibling;
```



Share Improve this answer

answered Feb 10, 2021 at 12:40



Follow



[I'm Weverton](#)

680 ● 5 ● 5

---

this should be considered as the accepted answer

– [Sébastien M.](#) Nov 11 at 13:21

---



Really depends on the overall structure of your document.

33

If you have:



```
<div></div>
<div></div>
<div></div>
```



it may be as simple as traversing through using

```
mydiv.nextSibling;
mydiv.previousSibling;
```

However, if the 'next' div could be anywhere in the document you'll need a more complex solution. You could try something using

```
document.getElementsByTagName("div");
```

and running through these to get where you want somehow.

If you are doing lots of complex DOM traversing such as this I would recommend looking into a library such as jQuery.

Share Improve this answer

Follow

edited Oct 25, 2017 at 8:14



John R Perry

4,172 ● 2 ● 44 ● 69

answered Feb 22, 2009 at 13:20



Andy Hume

41.6k ● 10 ● 51 ● 59

---

4    `nextSibling()` .. should be `nextSibling`, I think – [paul\\_h](#) Jan 8, 2017 at 19:41

---



Well in pure javascript my thinking is that you would first have to collate them inside a collection.

32



```
var divs = document.getElementsByTagName("div");
//divs now contain each and every div element on the p
var selectionDiv = document.getElementById("MySecondDi
```



So basically with `selectionDiv` iterate through the collection to find its index, and then obviously `-1 = previous` `+1 = next` within bounds



```
for(var i = 0; i < divs.length;i++)
{
    if(divs[i] == selectionDiv)
    {
        var previous = divs[i - 1];
        var next = divs[i + 1];
    }
}
```

Please be aware though as I say that extra logic would be required to check that you are within the bounds i.e. you are not at the end or start of the collection.

This also will mean that say you have a div which has a child div nested. The next div would not be a sibling but a child, So if you only want siblings on the same level as the target div then definately use nextSibling checking the **tagName** property.

Share Improve this answer

edited Feb 22, 2009 at 13:27

Follow

answered Feb 22, 2009 at 13:19



REA\_ANDREW

10.8k ● 8 ● 50 ● 72

---

1 seems good solution but how to get the next element, as you wrote now i have the collection and the selected one, can you help me more? – [Amr Elgarhy](#) Feb 22, 2009 at 13:25

---

the next sibling solution answers more accurately to the question – [Sébastien M.](#) Nov 11 at 13:17

---



There is a attribute on every HTMLElement, "previousElementSibling".

27

Ex:



```
<div id="a">A</div>
<div id="b">B</div>
<div id="c">c</div>
```

```
<div id="result">Resultado: </div>
```

---

```
var b = document.getElementById("c").previousElementSi  
document.getElementById("result").innerHTML += b.innerHTML
```

Live: <http://jsfiddle.net/QukKM/>

Share Improve this answer

answered Jan 11, 2013 at 18:28

Follow



[user1970894](#)

271 ● 3 ● 2

This fails on IE8 (haven't checked on other IE versions).

`previousSibling` seems to be the cross browser solution.

– [Niks](#) Oct 8, 2013 at 7:05



This will be easy... its an pure javascript code

17



```
<script>
```

```
alert(document.getElementById("someElement").previousE  
</script>
```



Share Improve this answer

answered Jan 3, 2014 at 6:57

Follow



[smurf](#)

219 ● 2 ● 4



all these solutions look like an overkill. Why use my solution?

10

`previousElementSibling` supported from IE9



`document.addEventListener` needs a polyfill



`previousSibling` might return a text



Please note i have chosen to return the first/last element in case boundaries are broken. In a RL usage, i would prefer it to return a null.

```
var el = document.getElementById("child1"),
    children = el.parentNode.children,
    len = children.length,
    ind = [].indexOf.call(children, el),
    nextEl = children[ind === len ? len : ind + 1],
    prevEl = children[ind === 0 ? 0 : ind - 1];

document.write(nextEl.id);
document.write("<br/>");
document.write(prevEl.id);
```

```
<div id="parent">
  <div id="child1"></div>
  <div id="child2"></div>
</div>
```



Run code snippet



[Expand snippet](#)



Share Improve this answer

edited Aug 3, 2018 at 9:01

Follow

answered Jan 31, 2017 at 17:18



Rafael Herscovici

17.1k ● 19 ● 67 ● 94



8



You can use **nextElementSibling** or **previousElementSibling** properties

```
<div>
  <span id="elem-1">
    span
  </span>
</div>
```

```
<div data-id="15">
  Parent Sibling
</div>
```

```
const sp = document.querySelector('#elem-1');
let sibling_data_id = sp.parentNode.nextElementSibling
console.log(sibling_data_id); // 15
```

Share Improve this answer

answered Mar 2, 2022 at 13:29

Follow



Harut Enogyan

99 ● 1 ● 2





1



Tested it and it worked for me. The element finding me change as per the document structure that you have.

```
<html>
  <head>
    <script type="text/javascript" src="test.js"></script>
  </head>
  <body>
    <form method="post" id = "formId" action="action" onsubmit="return validateForm();">
      <table>
        <tr>
          <td>
            <label class="standard_text">E-mail</label>
          </td>
          <td><input class="textare" name="email" type="text" value="" placeholder="E-mail"></td>
        </tr>
        <tr>
          <td><input class="textare" name="password" type="password" value="" placeholder="E-mail"></td>
          <td><input class="textare" name="confirm_password" type="password" value="" placeholder="E-mail"></td>
        </tr>
        <tr>
          <td><div class="check_icon icon_yes" id="mail_ok_icon"></div></td>
          <td><div class="check_icon icon_no" id="mail_no_icon"></div></td>
        </tr>
        <tr>
          <td><div class="check_message" style="display: none;" id="mail_message"><label class="important_text">The email is not correct!</label></div></td>
        </tr>
      </table>
      <input class="button_submit" type="submit" value="Register"/>
    </form>
  </body>
</html>
```

```

var inputs;
document.addEventListener("DOMContentLoaded", function
var form = document.getElementById('formId');
inputs = form.getElementsByTagName("input");
for(var i = 0 ; i < inputs.length;i++) {
    inputs[i].addEventListener('keydown', function
        if(e.keyCode == 13) {
            var currentIndex = findElement(e.target
            if(currentIndex > -1 && currentIndex <
                inputs[currentIndex+1].focus();
            }
        }
    });
}
});

function findElement(element) {
    var index = -1;
    for(var i = 0; i < inputs.length; i++) {
        if(inputs[i] == element) {
            return i;
        }
    }
    return index;
}

```

Share Improve this answer

edited Nov 24, 2016 at 2:12

Follow



Alexis Tyler

1,031 ● 6 ● 32 ● 50

answered Sep 16, 2016 at 8:39



Nitin Misra

76 ● 6



that's so simple

-2



```
var element = querySelector("div")  
var nextelement = element.parentElement.querySelector(
```

Here is the browser supports

<https://caniuse.com/querySelector>

Share Improve this answer

edited Jun 20, 2022 at 16:51

Follow

answered Sep 13, 2020 at 10:23



**Mert Çelik**

382 ● 1 ● 11

- 
- 1 Should be `element.parentElement` , `ParentElement` doesn't seem to work. – [Thomas](#) Jun 8, 2022 at 17:23
- 

