

Appending HTML string to the DOM

How to append this HTML string

99 `var str = '<p>Just some text here</p>';`

to the DIV with the ID 'test' which is in the DOM?

★ (Btw `div.innerHTML += str;` is not acceptable.)

15 `javascript` `html` `dom` `browser`

share improve this question

asked Sep 6 '11 at 22:48

 **Sime Vidas**
127k ● 52 ● 243 ● 355

add a comment

9 Answers

active oldest votes

Use `insertAdjacentHTML` if it's available, otherwise use some sort of fallback. `insertAdjacentHTML` [is supported in all current browsers](#).

207 `div.insertAdjacentHTML('beforeend', str);`

✓ **Live demo:** <http://jsfiddle.net/euQ5n/>

share improve this answer

edited Jan 23 at 21:09  **mikemaccana**
48.4k ● 53 ● 254 ● 324

answered Sep 6 '11 at 22:58  **Neil**
46k ● 8 ● 47 ● 64

1 @alex It's the same concept: parsing a HTML string and putting it into the DOM. But the functionality is different - `innerHTML` puts the string into the element (replacing all children), whereas `insertAdjacentHTML` puts it (1) before the element, (2) after the element, (3) inside the element before the first child, or (4) inside the element after the last child. – [Sime Vidas](#) Sep 6 '11 at 23:14

3 @alex `insertAdjacentHTML` is faster than appending to `innerHTML`, because `insertAdjacentHTML` doesn't serialize and reparse the existing children of the element. – [hsivonen](#) Oct 21 '11 at 17:31

1 Also, `insertAdjacentHTML` maintains the values of altered form elements, whereas appending to `innerHTML` rebuilds the element and loses all form context. – [Brandon Gano](#) Apr 2 '13 at 17:20

add a comment

Is this acceptable?


18 `var child = document.createElement('div');
child.innerHTML = str;
child = child.firstChild;
document.getElementById('test').appendChild(child);`

[jsFiddle](#).

But, [Neil's answer](#) is a better solution.

share improve this answer

edited Aug 21 '17 at 7:34

answered Sep 6 '11 at 22:51  **alex**
360k ● 178 ● 795 ● 928

That's what I came up with too. But it feels like a hack... – [Sime Vidas](#) Sep 6 '11 at 22:52

1 @Sime Working with the DOM API *always* feels like a hack :P Would you prefer to unserialise the string *without* `innerHTML`? – [alex](#) Sep 6 '11 at 22:55

1 @Sime - I think there's only two ways to utilize the browser's HTML parser, `innerHTML` and `document.write`. And if you think `innerHTML` is a hack ... – [Alohci](#) Sep 6 '11 at 23:02

1 @Sime - Thanks. I found the WHATWG version. It's in the [DOM Parsing and Serialization](#) spec. It indicates that as well as `innerHTML` and `insertAdjacentHTML`, there's `outerHTML` and, I think, `createContextualFragment`. – [Alohci](#) Sep 6 '11 at 23:35

1 This will only work if `str` has a single top-level element. – [Chris Calo](#) Oct 24 '11 at 6:08

[show 7 more comments](#)

The idea is to use `innerHTML` on an intermediary element and then move all of its child nodes to where you really want them via `appendChild`.

```

var target = document.getElementById('test');
var str = '<p>Just some <span>text</span> here</p>';

var temp = document.createElement('div');
temp.innerHTML = str;
while (temp.firstChild) {
  target.appendChild(temp.firstChild);
}

```

This avoids wiping out any event handlers on `div#test` but still allows you to append a string of HTML.

share improve this answer

edited Jan 28 '18 at 22:36

answered Oct 24 '11 at 6:02



Chris Calo

4,578 ● 6 ● 35 ● 52

This is neat! Thanks! – CLOUGH Sep 13 '17 at 10:16

add a comment

Here is some performance test:

Set up environment (2019.07.10) MacOS High Sierra 10.13.4 on Chrome 75.0.3770 (64-bit), Safari 11.1.0 (13604.5.6), Firefox 67.0.0 (64-bit)

Testing in Chrome 75.0.3770 / Mac OS X 10.13.4		
	Test	Ops/sec
A	container.innerHTML += '<p>Just some text here</p>';	332 ±2.76% 100% slower
B	container.innerHTML = '<p>Just some text here</p>';	47,733 ±1.31% 66% slower
C	\$('#container').append('<p>Just some text here</p>');	14,926 ±1.32% 89% slower
D	var p = document.createElement("p"); p.innerHTML = 'Just some text here'; container.appendChild(p);	39,265 ±3.35% 72% slower
E	var p = document.createElement("p"); var s = document.createElement("span"); s.appendChild(document.createTextNode("text ")); p.appendChild(document.createTextNode("Just some ")); p.appendChild(s); p.appendChild(document.createTextNode(" here")); container.appendChild(p);	140,137 ±1.97% fastest
F	container.insertAdjacentHTML('beforeend', '<p>Just some text here</p>');	46,428 ±1.75% 67% slower

- on Chrome E (140k operations per second) is fastest, B (47k) and F (46k) are second, A (332) is slowest
- on firefox F (94k) is fastest, then B(80k), D (73k), E(64k), C (21k) slowest is A(466)
- on Safari E(207k) is fastest, then B(89k), F(88k), D(83k), C (25k), slowest is A(509)

Summary

`appendChild` (E) is more than 2x faster than other solutions on chrome and safari, `insertAdjacentHTML` (F) is fastest on firefox. The `innerHTML` (B) (do not confuse with `+=`) is second fast solution on all browsers and it is much more handy than E and F. You can replay test in your machine [here](#)

► [Show code snippet](#)

share improve this answer

edited Jul 10 at 14:35

answered Jul 2 '18 at 16:57



Kamil Kielczewski

22k ● 10 ● 101 ● 120

add a comment

The right way is using `insertAdjacentHTML`. In Firefox earlier than 8, you can fall back to using `Range.createContextualFragment`

if your `str` contains no `script` tags.

If your `str` contains `script` tags, you need to remove `script` elements from the fragment returned by `createContextualFragment` before inserting the fragment. Otherwise, the scripts will run. (`insertAdjacentHTML` marks scripts unexecutable.)

share improve this answer

edited Oct 24 '11 at 5:24

answered Oct 21 '11 at 17:39



hsiivonen

7,037 ● 1 ● 26 ● 33

Thank you for mentioning `createContextualFragment`. I was searching for a way to make some html includes with scripts run only if the user agrees to set a cookie. – [schliifo](#) May 9 '18 at 15:40

add a comment

Why is that not acceptable?

```
document.getElementById('test').innerHTML += str
```

would be the textbook way of doing it.

share improve this answer

answered Sep 6 '11 at 22:56



Nick Brunt

5,603 ● 4 ● 41 ● 75

4 It would [kill event handlers](#) attached to `#test` though. That is generally not desirable. – [alex](#) Sep 6 '11 at 22:57

Interesting that it does that. Doesn't seem logical really. – [Nick Brunt](#) Sep 6 '11 at 23:04

2 It's logical I guess if you think about the serialising of the HTML to a string and then setting the HTML back. – [alex](#) Sep 6 '11 at 23:05

I guess the JavaScript would have to be run again to reset the event handler. Fair enough. – [Nick Brunt](#) Sep 6 '11 at 23:08

1 @Nick You don't want to stringify (serialize) a part of the DOM just so that you can concatenate it with another string and then parse the whole thing back into the DOM. As alex said, the serialization won't record bound event handlers (among other things). Even if he serialization could capture everything, you still wouldn't want to do it. – [Sime Vidas](#) Sep 6 '11 at 23:23

add a comment

Quick Hack:

```
<script>
document.children[0].innerHTML="<h1>QUICK_HACK</h1>";
</script>
```

Use Cases:

1: Save as .html file and run in chrome or firefox or edge. (IE wont work)

2: Use in <http://js.do>

In Action: http://js.do/HeavyMetalCookies/quick_hack

Broken down with comments:

```
<script>

//: The message "QUICK_HACK"
//: wrapped in a header #1 tag.
var text = "<h1>QUICK_HACK</h1>";

//: It's a quick hack, so I don't
//: care where it goes on the document,
//: just as long as I can see it.
//: Since I am doing this quick hack in
//: an empty file or scratchpad,
//: it should be visible.
var child = document.children[0];

//: Set the html content of your child
//: to the message you want to see on screen.
child.innerHTML = text;

</script>
```

Reason Why I posted:

JS.do has two must haves:

1. No autocomplete
2. Vertical monitor friendly

But doesn't show console.log messages. Came here looking for a quick solution. I just want to see the results of a few lines of scratchpad code, the other solutions are too much work.

share improve this answer

answered Aug 20 '18 at 9:10



J.M.I. MADISON
1,707 • 15 • 12

add a comment

This can solve

0

```
document.getElementById("list-input-email").insertAdjacentHTML('beforeend', '<div class=""><input type="text" na
```

share improve this answer

answered Feb 6 '18 at 20:59



Rafael Senne
1

3

Welcome to Stack overflow. Please explain your code as part of an answer to provide some context rather than just pasting a long single line of code. Although you understand it others may not understand what its doing. – [loan.burger](#) Feb 7 '18 at 0:57

add a comment

InnerHTML clear all data like event for existing nodes

0

append child with firstChild adds only first child to innerHTML. For example if we have to append:

```
<p>text1</p><p>text2</p>
```

only text1 will show up

What about this:

adds special tag to innerHTML by append child and then edit outerHTML by deleting tag we've created. Don't know how smart it is but it works for me or you might change outerHTML to innerHTML so it doesn't have to use function replace

```
function append(element, str)
{
    var child = document.createElement('someshittyuniquetag');
    child.innerHTML = str;
    element.appendChild(child);
    child.outerHTML = child.outerHTML.replace(/<\/?someshittyuniquetag/, '');
    // or Even child.outerHTML = child.innerHTML
}
```

```
<div id="testit">
This text is inside the div
<button onclick="append(document.getElementById('testit'), '<button>dadasasdas</button>')">To div</button>
<button onclick="append(this, 'some text')">to this</button>
</div>
```

Run code snippet

Expand snippet

Run code snippet

Expand snippet

share improve this answer

answered Aug 2 '18 at 8:36




Łukasz Szpak
1


add a comment


Your Answer

B	<i>I</i>								
B	<i>I</i>								

Sign up or [log in](#)

 Sign up using Google

 Sign up using Facebook

 Sign up using Email and Password

Post as a guest

Name

Email
Required, but never shown

Post Your Answer

By clicking "Post Your Answer", you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#)

Not the answer you're looking for? Browse other questions tagged [javascript](#) [html](#) [dom](#) [browser](#) or [ask your own question](#).