

.prepend()

Categories: [Manipulation](#) > [DOM Insertion](#), [Inside](#)

.prepend(content [, content])

Returns: [jQuery](#)

Description: Insert content, specified by the parameter, to the beginning of each element in the set of matched elements.

.prepend(content [, content])

version added: 1.

content

Type: [htmlString](#) or [Element](#) or [Text](#) or [Array](#) or [jQuery](#)

DOM element, text node, array of elements and text nodes, HTML string, or jQuery object to insert at the beginning of each element in the set of matched elements.

content

Type: [htmlString](#) or [Element](#) or [Text](#) or [Array](#) or [jQuery](#)

One or more additional DOM elements, text nodes, arrays of elements and text nodes, HTML strings, or jQuery objects to insert at the beginning of each element in the set of matched elements.

.prepend(function)

version added: 1.

function

Type: [Function](#)([Integer](#) elementOfArray, [String](#) html) => [htmlString](#) or [Element](#) or [Text](#) or [jQuery](#)

A function that returns an HTML string, DOM element(s), text node(s), or jQuery object to insert at the beginning of each element in the set of matched elements. Receives the index position of the element in the set and the old HTML value of the element as arguments. Within the function, `this` refers to the current element in the set.

The `.prepend()` method inserts the specified content as the first child of each element in the jQuery collection (To insert it as the *last* child, use [.append\(\)](#)).

The `.prepend()` and `.prependTo()` methods perform the same task. The major difference is in the syntax: specifically, in the placement of the content and target. With `.prepend()` , the selector expression preceding the method is the container into which the content is inserted. With `.prependTo()` , on the other hand, the content precedes the method, either as a selector expression or as markup created on the fly, and it is inserted into the target container.

Consider the following HTML:

```
1 <h2>Greetings</h2>
2 <div class="container">
3   <div class="inner">Hello</div>
4   <div class="inner">Goodbye</div>
5 </div>
```

You can create content and insert it into several elements at once:

```
1 | $( ".inner" ).prepend( "<p>Test</p>" );
```

Each `<div class="inner">` element gets this new content:

```
1 | <h2>Greetings</h2>
2 | <div class="container">
3 |   <div class="inner">
4 |     <p>Test</p>
5 |     Hello
6 |   </div>
7 |   <div class="inner">
8 |     <p>Test</p>
9 |     Goodbye
10 |   </div>
11 | </div>
```

You can also select an element on the page and insert it into another:

```
1 | $( ".container" ).prepend( $( "h2" ) );
```

If a *single element* selected this way is inserted into a single location elsewhere in the DOM, it will be moved into the target (*not cloned*):

```
1 | <div class="container">
2 |   <h2>Greetings</h2>
3 |   <div class="inner">Hello</div>
4 |   <div class="inner">Goodbye</div>
5 | </div>
```

Important: If there is more than one target element, however, cloned copies of the inserted element will be created for each target except for the last one.

Additional Arguments

Similar to other content-adding methods such as `.append()` and `.before()`, `.prepend()` also supports passing in multiple arguments as input. Supported input includes DOM elements, jQuery objects, HTML strings, and arrays of DOM elements.

For example, the following will insert two new `<div>`s and an existing `<div>` as the first three child nodes of the body:

```
1 | var $newdiv1 = $( "<div id='object1'></div>" ),
2 |   $newdiv2 = document.createElement( "div" );
```

```
4
5 $( "body" ).prepend( $newdiv1, [ newdiv2, exist
```

Since `.prepend()` can accept any number of additional arguments, the same result can be achieved by passing in the three `<div>`s as three separate arguments, like so: `$("body").prepend($newdiv1, newdiv2, existingdiv1)`. The type and number of arguments will largely depend on how you collect the elements in your code.

Additional Notes:

By design, any jQuery constructor or method that accepts an HTML string — [jQuery\(\)](#), [.append\(\)](#), [.after\(\)](#), etc. — can potentially execute code. This can occur by injection of script tags or use of HTML attributes that execute code (for example, ``). Do not use these methods to insert strings obtained from untrusted sources such as URL query parameters, cookies, or form inputs. Doing so can introduce cross-site-scripting (XSS) vulnerabilities. Remove or escape any user input before adding content to the document.

jQuery doesn't officially support SVG. Using jQuery methods on SVG documents, unless explicitly documented for that method, might cause unexpected behaviors. Examples of methods that support SVG as of jQuery 3.0 are `addClass` and `removeClass`.

Examples:

Prepends some HTML to all paragraphs.

```
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>prepend demo</title>
6    <style>
7      p {
8        background: yellow;
9      }
10   </style>
11   <script src="https://code.jquery.com/jquery
12 </head>
13 <body>
14
15 <p>there, friend!</p>
16 <p>amigo!</p>
17
18 <script>
19 $( "p" ).prepend( "<b>Hello </b>" );
20 </script>
21
22 </body>
23 </html>
```

Demo:

Prepends a DOM Element to all paragraphs.

```
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>prepend demo</title>
6    <style>
7      p {
8        background: yellow;
9      }
10   </style>
11   <script src="https://code.jquery.com/jquery"
12 </head>
13 <body>
14
15 <p>is what I'd say</p>
16 <p>is what I said</p>
17
18 <script>
19 $( "p" ).prepend( document.createTextNode( "I
20 </script>
21
22 </body>
23 </html>
```

Demo:

Prepends a jQuery object (similar to an Array of DOM Elements) to all paragraphs

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>prepend demo</title>
6   <style>
7     p {
8       background: yellow;
9     }
10  </style>
11  <script src="https://code.jquery.com/jquery
12 </head>
13 <body>
14
15 <p> is what was said.</p><b>Hello</b>
16
17 <script>
18 $( "p" ).prepend( $( "b" ) );
19 </script>
20
21 </body>
22 </html>
```

