

## JQuery HTML

There are four types

1. GET
2. SET
3. ADD
4. REMOVE

### GET Method

Three simple, but useful, jQuery methods for DOM manipulation are:

- `text()` - Sets or returns the text content of selected elements
- `html()` - Sets or returns the content of selected elements (including HTML markup)
- `val()` - Sets or returns the value of form fields

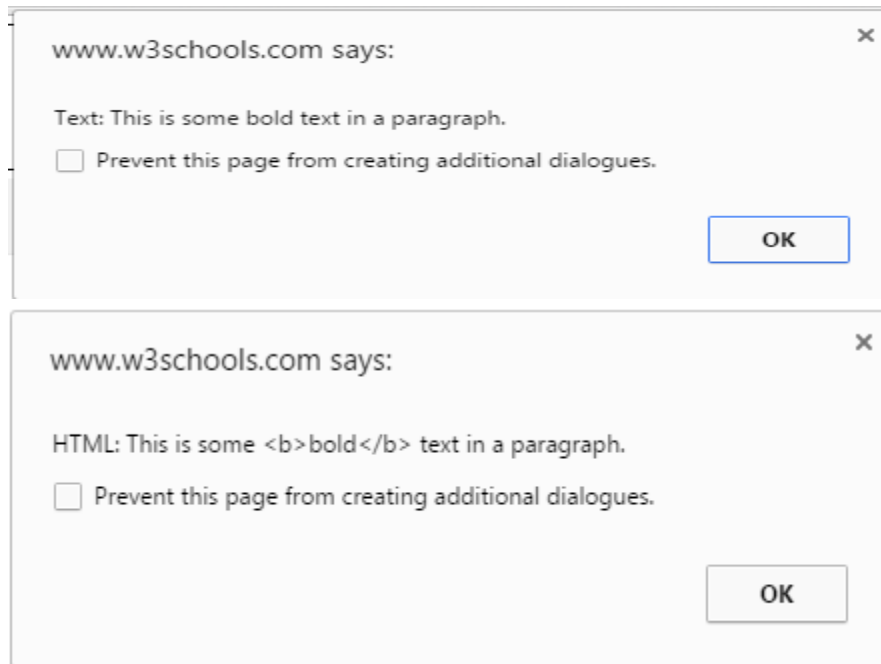
The following example demonstrates how to get content with the jQuery `text()` and `html()` methods:

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    alert("Text: " + $("#test").text());
  });
  $("#btn2").click(function(){
    alert("HTML: " + $("#test").html());
  });
});
</script>
</head>
<body>
<p id="test">This is some <b>bold</b> text in a paragraph.</p>
<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>
</body>
</html>
```

This is some **bold** text in a paragraph.

Show Text

Show HTML



The following example demonstrates how to get the value of an input field with the jQuery val() method:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

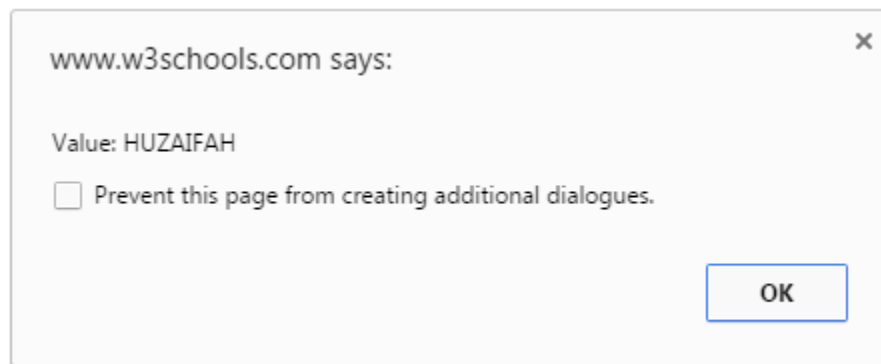
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Value: " + $("#test").val());
    });
});
</script>

</head>
<body>

<p>Name: <input type="text" id="test" value="Mickey Mouse"></p>
<button>Show Value</button>
```

```
</body>
```

```
</html>
```

Name: 

### Get Attributes - attr()

The jQuery attr() method is used to get attribute values.

The following example demonstrates how to get the value of the href attribute in a link:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
```

```
<script>
```

```
$(document).ready(function(){
```

```
  $("button").click(function(){
```

```
    alert($("#w3s").attr("href"));
```

```
  });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<p><a href="http://www.w3schools.com" id="w3s">Huzaifah.com</a></p>
```

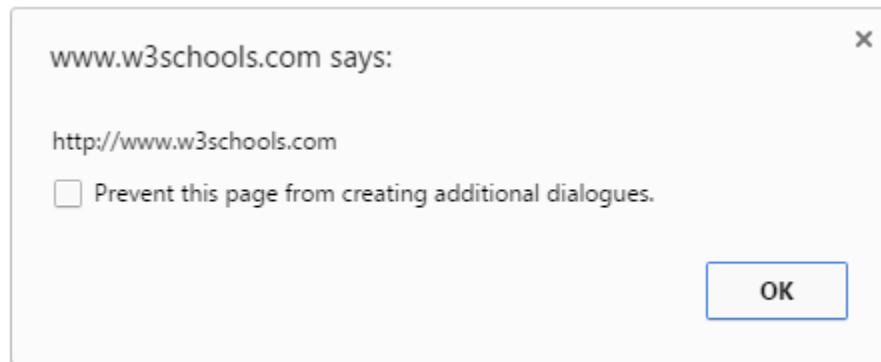
```
<button>Show href Value</button>
```

```
</body>
```

```
</html>
```

[HuzaiFah.com](http://HuzaiFah.com)

Show href Value



## jQuery - Add Elements

With jQuery, it is easy to add new elements/content.

Add New HTML Content

We will look at four jQuery methods that are used to add new content:

- `append()` - Inserts content at the end of the selected elements
- `prepend()` - Inserts content at the beginning of the selected elements
- `after()` - Inserts content after the selected elements
- `before()` - Inserts content before the selected elements

### jQuery `append()` Method

- The jQuery `append()` method inserts content AT THE END of the selected HTML elements.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        $("p").append(" <b>Appended text</b>.");
    });

    $("#btn2").click(function(){
```

```

    $("ol").append("<li>Appended item</li>");
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>

<button id="btn1">Append text</button>
<button id="btn2">Append list items</button>

</body>
</html> (OUTPUT)

```

This is a paragraph. **Appended text. Appended text. Appended text. Appended text. Appended text.**

This is another paragraph. **Appended text. Appended text. Appended text. Appended text. Appended text.**

1. List item 1
2. List item 2
3. List item 3
4. Appended item
5. Appended item
6. Appended item
7. Appended item
8. Appended item
9. Appended item

Append text

Append list items

## jQuery prepend() Method

The jQuery prepend() method inserts content AT THE BEGINNING of the selected HTML elements.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

```

```
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#p").prepend("<b>Prepended text</b>. ");
  });
  $("#btn2").click(function(){
    $("#ol").prepend("<li>Prepended item</li>");
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>

<button id="btn1">Prepend text</button>
<button id="btn2">Prepend list item</button>

</body>
</html>
```

## OUTPUT

**Prepended text.** This is a paragraph.

**Prepended text.** This is another paragraph.

1. Prepended item
2. List item 1
3. List item 2
4. List item 3

Prepend text

Prepend list item

### Add Several New Elements With append() and prepend()

In both examples above, we have only inserted some text/HTML at the beginning/end of the selected HTML elements.

However, both the append() and prepend() methods can take an infinite number of new elements as parameters. The new elements can be generated with text/HTML (like we have done in the examples above), with jQuery, or with JavaScript code and DOM elements.

In the following example, we create several new elements. The elements are created with text/HTML, jQuery, and JavaScript/DOM. Then we append the new elements to the text with the append() method (this would have worked for prepend() too) :

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
function appendText() {
    var txt1 = "<p>Text.</p>";           // Create text with HTML
    var txt2 = $("<p></p>").text("Text."); // Create text with jQuery
    var txt3 = document.createElement("p");
    txt3.innerHTML = "Text.";           // Create text with DOM
    $("body").append(txt1, txt2, txt3); // Append new elements
}
</script>
</head>
<body>

<p>This is a paragraph.</p>
<button onclick="appendText()">Append text</button>

</body>
</html>
```

## OUTPUT

This is a paragraph.

Append text

Text.

Text.

Text.

Text.

Text.

Text.

**jQuery after() and before() Methods:-**

The jQuery after() method inserts content AFTER the selected HTML elements.

The jQuery before() method inserts content BEFORE the selected HTML elements.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        $("img").before("<b>Before</b>");
    });

    $("#btn2").click(function(){
        $("img").after("<i>After</i>");
    });
});
</script>
</head>
<body>

<br><br>

<button id="btn1">Insert before</button>
```



```
<button id="btn2">Insert after</button>
```

```
</body>
```

```
</html>
```

OUTPUT



### Add Several New Elements With after() and before()

Also, both the after() and before() methods can take an infinite number of new elements as parameters. The new elements can be generated with text/HTML (like we have done in the example above), with jQuery, or with JavaScript code and DOM elements.

In the following example, we create several new elements. The elements are created with text/HTML, jQuery, and JavaScript/DOM. Then we insert the new elements to the text with the after() method (this would have worked for before() too) :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
```

```
<script>
```

```
function afterText() {
```

```
    var txt1 = "<b>I </b>";           // Create element with HTML
```

```
    var txt2 = $("<i></i>").text("love "); // Create with jQuery
```

```
    var txt3 = document.createElement("b"); // Create with DOM
```

```
    txt3.innerHTML = "jQuery!";
```

```
    $("img").after(txt1, txt2, txt3);    // Insert new elements after img
```

```
}
```

```
</script>
```

```
</head>
```

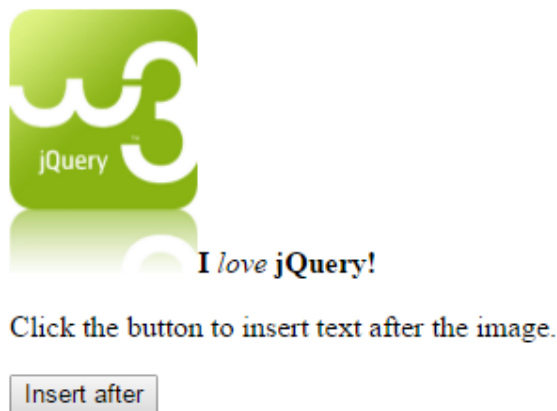
```
<body>



<p>Click the button to insert text after the image.</p>

<button onclick="afterText()">Insert after</button>

</body>
</html>
OUTPUT
```



## jQuery - Set Content and Attributes

Set Content - text(), html(), and val()

We will use the same three methods from the previous page to **set content**:

- text() - Sets or returns the text content of selected elements
- html() - Sets or returns the content of selected elements (including HTML markup)
- val() - Sets or returns the value of form fields

The following example demonstrates how to set content with the jQuery text(), html(), and val() methods:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
```

```
$("#btn1").click(function(){
    $("#test1").text("Hello world!");
});
$("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
    $("#test3").val("Huzaif");
});
});
</script>
</head>
<body>

<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>

<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>

<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>

</body>
</html>
```

This is a paragraph.

This is another paragraph.

Input field:

Set Text

Set HTML

Set Value

### A Callback Function for text(), html(), and val()

All of the three jQuery methods above: text(), html(), and val(), also come with a callback function. The callback function has two parameters: the index of the current element in the list of elements selected and the original (old) value. You then return the string you wish to use as the new value from the function.

The following example demonstrates text() and html() with a callback function:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        $("#test1").text(function(i, origText){
            return "Old text: " + origText + " New text: Hello world! (index: " + i + ")";
        });
    });

    $("#btn2").click(function(){
        $("#test2").html(function(i, origText){
            return "Old html: " + origText + " New html: Hello <b>world!</b> (index: " + i + ")";
        });
    });
});
</script>
</head>
<body>

<p id="test1">This is a <b>bold</b> paragraph.</p>
<p id="test2">This is another <b>bold</b> paragraph.</p>

<button id="btn1">Show Old/New Text</button>
<button id="btn2">Show Old/New HTML</button>

</body>
</html>
```

OUTPUT

Old text: This is a bold paragraph. New text: Hello world! (index: 0)

Old html: This is another bold paragraph. New html: Hello world! (index: 0)

Show Old/New Text

Show Old/New HTML

## jQuery - Remove Elements

With jQuery, it is easy to remove existing HTML elements.

Remove Elements/Content

To remove elements and content, there are mainly two jQuery methods:

- `remove()` - Removes the selected element (and its child elements)
- `empty()` - Removes the child elements from the selected element

### jQuery `remove()` Method

The jQuery `remove()` method removes the selected element(s) and its child elements.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
```

```
<script>
```

```
$(document).ready(function(){
```

```
    $("button").click(function(){
```

```
        $("#div1").remove();
```

```
    });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">
```

This is some text in the div.

```
<p>This is a paragraph in the div.</p>
```

```
<p>This is another paragraph in the div.</p>
```

```
</div>
```

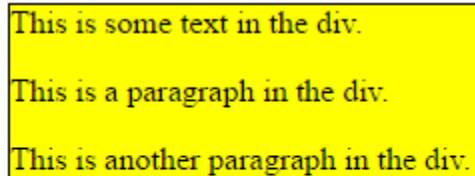
```
<br>
```

```
<button>Remove div element</button>
```

```
</body>
```

```
</html>
```

(OUT PUT)



This is some text in the div.  
This is a paragraph in the div.  
This is another paragraph in the div.

Remove div element

Remove div element

### jQuery empty() Method

The jQuery empty() method removes the child elements of the selected element(s).

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

    $("button").click(function(){

        $("#div1").empty();

    });

});

</script>

</head>

<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">

This is some text in the div.

<p>This is a paragraph in the div.</p>

<p>This is another paragraph in the div.</p>

</div>

<br>
```

<button>Empty the div element</button>

</body>

</html>

This is some text in the div.

This is a paragraph in the div.

This is another paragraph in the div.

Empty the div element

Empty the div element

### Filter the Elements to be Removed

The jQuery remove() method also accepts one parameter, which allows you to filter the elements to be removed.

The parameter can be any of the jQuery selector syntaxes.

The following example removes all <p> elements with class="test":

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

\$(document).ready(function(){

  \$("button").click(function(){

    \$("p").remove(".test");



```
});

});

</script>

<style>

.test {

    color: red;

    font-size: 20px;

}

</style>

</head>

<body>

<p>This is a paragraph.</p>

<p class="test">This is another paragraph.</p>

<p class="test">This is another paragraph.</p>

<button>Remove all p elements with class="test"</button>

</body>

</html>
```

## Output

&lt;!DOCTYPE html&gt;

```
<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

    $("button").click(function(){

        $("p").remove(".test");

    });

});

</script>

<style>

.test {

    color: red;

    font-size: 20px;

}

</style>

</head>

<body>

<p>This is a paragraph.</p>
```

```
<p class="test">This is another paragraph.</p>
```

```
<p class="test">This is another paragraph.</p>
```

```
<button>Remove all p elements with class="test"</button>
```

```
</body>
```

```
</html>
```

Output

This is a paragraph.

This is another paragraph.

This is another paragraph.

This is a paragraph.

Remove all p elements with class="test"

Remove all p elements with class="test"

This example removes all `<p>` elements with `class="test"` and `class="demo"`:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
```

```
<script>
```

```
$(document).ready(function(){
```

```
  $("button").click(function(){
```

```
    $("p").remove(".test, .demo");

});

});

</script>

<style>

.test {

    color: red;

    font-size: 20px;

}

.demo {

    color: green;

    font-size: 25px;

}

</style>

</head>

<body>

<p>This is a paragraph.</p>

<p class="test">This is p element with class="test".</p>
```

```
<p class="test">This is p element with class="test".</p>
```

```
<p class="demo">This is p element with class="demo".</p>
```

```
<button>Remove all p elements with class="test" and class="demo"</button>
```

```
</body>
```

```
</html>
```

## OUTPUT

This is a paragraph.

This is p element with class="test".

This is p element with class="test".

This is p element with class="demo".

Remove all p elements with class="test" and class="demo"

This is a paragraph.

Remove all p elements with class="test" and class="demo"