

jQuery Notes

- ❖ jQuery Introduction
- ❖ jQuery Syntax
- ❖ jQuery Selectors
- ❖ jQuery Event Methods

1. What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It simplifies tasks like HTML document traversal and manipulation, event handling, and animation. It also provides an easy-to-use API that works across different browsers. jQuery was designed to make things like DOM manipulation and event handling easier, especially for developers who need to support older browsers.

2. Why use jQuery?

- **Cross-browser compatibility:** jQuery automatically handles inconsistencies between different browsers.
- **Simplified syntax:** It provides a more concise syntax than vanilla JavaScript for common operations like DOM manipulation, animations, etc.
- **Powerful features:** jQuery includes methods for AJAX requests, animations, event handling, and DOM traversal.

3. Basic jQuery Syntax

The basic syntax of jQuery is:

```
$(selector).action()
```

- **\$:** This is the jQuery function or alias.
- **selector:** Specifies the element(s) you want to work with.
- **action:** Specifies the action you want to perform (e.g., hide, show, animate).

4. Including jQuery in Your Project

You can include jQuery in your HTML file using a CDN (Content Delivery Network) or by downloading it and linking it locally. Below are both methods:

Using CDN:

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

Using Local File:

1. Download jQuery from jquery.com.
2. Link to the downloaded file in your HTML.

```
<script src="jquery.min.js"></script>
```

The Document Ready Event:-

You might have noticed that all jQuery methods in our examples, are inside a document ready event:

```
$(document).ready(function(){  
  
    // jQuery methods go here...  
  
});
```

This is to prevent any jQuery code from running before the document is finished loading (is ready).

It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.

Here are some examples of actions that can fail if methods are run before the document is fully loaded:

- Trying to hide an element that is not created yet
- Trying to get the size of an image that is not loaded yet

Tip: The jQuery team has also created an even shorter method for the document ready event:

```
$(function(){  
    // jQuery methods go here...  
});
```

Syntax with Html Code :-

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>  
  
<script>  
  
$(document).ready(function(){  
  
    // jQuery methods go here...  
  
});  
  
</script>  
  
</head>  
  
<body>  
  
</body>  
  
</html>
```

jQuery Selectors

jQuery selectors allow you to select and manipulate HTML element(s).

Query selectors are used to "find" (or select) HTML elements based on their name, id, classes.

All selectors in jQuery start with the dollar sign and parentheses: \$().

The element Selector

The jQuery element selector selects elements based on the element name.

You can select all `<p>` elements on a page like this:

```
$("#p")
```

Example

When a user clicks on a button, all `<p>` elements will be hidden:

Example

```
$(document).ready(function(){  
  $("#button").click(function(){  
    $("#p").hide();  
  });  
});
```

Example 1:-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
```

```
<script>
```

```
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<h2>This is a heading</h2>
```

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

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

```
<button>Click me to hide paragraphs</button>
```

```
</body>
```

```
</html>
```

Output:-

This is a heading

This is a paragraph.

This is another paragraph.

Click me to hide paragraphs

The #id Selector

The jQuery `#id` selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.

To find an element with a specific id, write a hash character, followed by the id of the HTML element:

```
$("#test")
```

Example

When a user clicks on a button, the element with id="test" will be hidden:

Example

When a user clicks on a button, the element with id="test" will be hidden:

Example

```
$(document).ready(function(){
  $("button").click(function(){
    $("#test").hide();
  });
});
```

Example 1:-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
```

```
<script>
```

```
$(document).ready(function(){
  $("button").click(function(){
    $("#test").hide();
```

```
    });  
  });  
</script>  
</head>  
<body>  
  
<h2>This is a heading</h2>  
  
<p>This is a paragraph.</p>  
<p id="test">This is another paragraph.</p>  
  
<button>Click me</button>  
  
</body>  
</html>
```

Output:-

This is a heading

This is a paragraph.

This is another paragraph.

Click me

The .class Selector

The jQuery *.class* selector finds elements with a specific class.

To find elements with a specific class, write a period character, followed by the name of the class:

```
$(".test")
```

Example

When a user clicks on a button, the elements with class="test" will be hidden:

Example

```
$(document).ready(function(){  
  $("button").click(function(){  
    $(".test").hide();  
  });  
});
```

Example 1:-

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>  
  
<script>  
  
$(document).ready(function(){  
  $("button").click(function(){  
    $(".test").hide();  
  });  
});  
  
</script>  
  
</head>  
  
<body>  
  
<h2 class="test">This is a heading</h2>  
  
<p class="test">This is a paragraph.</p>  
  
<p>This is another paragraph.</p>  
  
<button>Click me</button>
```

```
</body>
```

```
</html>
```

Output:-

This is a heading

This is a paragraph.

This is another paragraph.

Click me

Common jQuery Methods and Examples

a. DOM Manipulation

- `$(selector).html()`: Get or set the inner HTML of an element.

```
// Get HTML content
```

```
var content = $('#example').html();
```

```
// Set HTML content
```

```
$('#example').html('<p>New content</p>');
```

- `$(selector).text()`: Get or set the text content of an element.

```
// Get text content  
var text = $('#example').text();
```

```
// Set text content  
$('#example').text('New text content');
```

- `$(selector).css()`: Get or set CSS properties.

```
// Get CSS value  
  
var color = $('#example').css('color');  
  
// Set CSS value  
  
$('#example').css('color', 'red');
```

b. Event Handling

- `$(selector).click()`: Attach a click event handler.

```
$('#example').click(function() {  
    alert('Element clicked!');  
});
```

- `$(selector).hover()`: Attach mouse hover events.

```
$('#example').hover( function()  
{  
    $(this).css('background-color', 'yellow');  
},  
function() {  
    $(this).css('background-color', 'white');  
}  
);
```

jQuery Event Methods

jQuery is tailor-made to respond to events in an HTML page.

What are Events?

All the different visitors' actions that a web page can respond to are called events.

An event represents the precise moment when something happens.

Examples:

- moving a mouse over an element
- selecting a radio button
- clicking on an element

The term "**fires/fired**" is often used with events. Example: "The keypress event is fired, the moment you press a key".

jQuery Syntax For Event Methods

In jQuery, most DOM events have an equivalent jQuery method.

To assign a click event to all paragraphs on a page, you can do this:

```
$("#p").click();
```

The next step is to define what should happen when the event fires. You must pass a function to the event:

```
$("#p").click(function(){
  // action goes here!!
});
```

Commonly Used jQuery Event Methods

\$(document).ready()

The `$(document).ready()` method allows us to execute a function when the document is fully loaded. This event is already explained in the jQuery Syntax chapter.

click()

The `click()` method attaches an event handler function to an HTML element.

The function is executed when the user clicks on the HTML element.

The following example says: When a click event fires on a `<p>` element; hide the current `<p>` element:

Example

```
$("#p").click(function(){
  $(this).hide();
});
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
```

```
<script>
```

```
$(document).ready(function(){
  $("#p").click(function(){
    $(this).hide();
  });
```

```
});  
  
</script>  
  
</head>  
  
<body>  
  
<p>If you click on me, I will disappear.</p>  
  
<p>Click me away!</p>  
  
<p>Click me too!</p>  
  
</body>  
  
</html>
```

Output:-

If you click on me, I will disappear.

Click me away!

Click me too!

mouseenter()

The `mouseenter()` method attaches an event handler function to an HTML element.

The function is executed when the mouse pointer enters the HTML element:

Example

```
$("#p1").mouseenter(function(){  
    alert("You entered p1!");  
});  
  
<!DOCTYPE html>  
  
<html>  
  
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

    $("#p1").mouseenter(function(){
        alert("You entered p1!");
    });

});

</script>

</head>

<body>

<p id="p1">Enter this paragraph.</p>

</body>

</html>
```

mouseleave()

The `mouseleave()` method attaches an event handler function to an HTML element.

The function is executed when the mouse pointer leaves the HTML element:

Example

```
$("#p1").mouseleave(function(){
    alert("Bye! You now leave p1!");
});
```

Example 1:-

```
<!DOCTYPE html>
```

```
<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#p1").mouseleave(function(){
    alert("Bye! You now leave p1!");
});

});

</script>

</head>

<body>

<p id="p1">This is a paragraph.</p>

</body>

</html>
```

focus()

The `focus()` method attaches an event handler function to an HTML form field.

The function is executed when the form field gets focus:

Example

```
$("#input").focus(function(){
    $(this).css("background-color", "#cccccc");
});
```

```
<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

  $("input").focus(function(){

    $(this).css("background-color", "yellow");

  });

  $("input").blur(function(){

    $(this).css("background-color", "green");

  });

});

</script>

</head>

<body>

Name: <input type="text" name="fullname"><br>

Email: <input type="text" name="email">

</body>

</html>
```

blur()

The `blur()` method attaches an event handler function to an HTML form field.

The function is executed when the form field loses focus:

Example

```
$("#input").blur(function(){
    $(this).css("background-color", "#ffffff");
});
```

The on() Method

The `on()` method attaches one or more event handlers for the selected elements.

Attach a click event to a `<p>` element:

Example

```
$("#p").on("click", function(){
    $(this).hide();
});
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
```

```
<script>
```

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

```
    $("#p").on("click", function(){
        $(this).hide();
    });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<p>If you click on me, I will disappear.</p>
```

```
<p>Click me away!</p>
```

```
<p>Click me too!</p>
```

```
</body>
```

```
</html>
```

Attach multiple event handlers to a `<p>` element:

Example

```
$("p").on({  
  mouseenter: function() {  
    $(this).css("background-color", "lightgray");  
  },  
  mouseleave: function() {  
    $(this).css("background-color", "lightblue");  
  },  
  click: function() {  
    $(this).css("background-color", "yellow");  
  }  
});
```

Example 1:-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
```

```
<script>
```

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

```
  $("p").on({  
    mouseenter: function() {  
      $(this).css("background-color", "lightgray");  
    },  
    mouseleave: function() {
```

```
    $(this).css("background-color", "lightblue");
  },
  click: function(){
    $(this).css("background-color", "yellow");
  }
});

});

</script>

</head>

<body>

<p>Click or move the mouse pointer over this paragraph.</p>

</body>

</html>
```