JQUERY NOTES

❖ What is jQuery?

jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

- **The jQuery library contains the following features:**

1. HTML/DOM manipulation
2. CSS manipulation
3. HTML event methods
4. Effects and animations
5. AJAX
6. Utilities

❖ Why jQuery?

There are lots of other JavaScript libraries out there, but jQuery is probably the most popular, and also the most extendable.

Many of the biggest companies on the Web use jQuery, such as:

- Google
- Microsoft
- IBM
- Netflix

- **jQuery CDN**

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

❖ jQuery Syntax

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the

element(s).

Basic syntax is: **$(*selector*).*action*()**

- A $ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery *action*() to be performed on the element(s)

  Examples:

  $(this).hide() - hides the current element.

  $("p").hide() - hides all <p> elements.

  $(".test").hide() - hides all elements with class="test".

❖ jQuery Selectors

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

jQuery selectors are used to "find" (or select) HTML elements based on

their name, id,

classes, types, attributes, values of attributes and much more. It's based on

the existing CSS,

Selectors and in addition, it has some own custom selectors.

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: -

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

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: -

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

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 :-

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


  });
});
```

Output:-

This is a heading

This is a paragraph.

This is another paragraph.

Click me

❖ Functions In a Separate File:-

If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, you can put your jQuery functions in a separate .js file.

When we demonstrate jQuery in this tutorial, the functions are added directly into the <head> section. However, sometimes it is preferable to place them in a separate file,

like this (use the src attribute to refer to the .js file):

Example :-

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

<script src="my_jquery_functions.js"></script>
</head>
```

❖ Aspects of the DOM and jQuery

- **Identification:** how do I obtain areference to the node that I want.
- **Traversal:** how do I move around theDOM tree.
- **Node Manipulation:** how do I get orset aspects of a DOM node.
- **Tree Manipulation:** how do I changethe structure of the page.

❖ jQuery node identification

```
// id selector
 var elem = $("#myid");

// group selector

var elems = $("#myid, p");

// context selector

var elems = $("#myid < div p");

// complex selector

var elems = $("#myid < h1.special:not(.classy)");
```

❖ Selecting groups of DOM objects

name	description
getElementById	returns array of descendents withthe given tag, such as "div"
getElementsByTagName	returns array of descendents withthe given tag, such as "div"
getElementsByName	returns array of descendents with the given name attribute (mostly useful for accessing form controls)
querySelector *	returns the first element that wouldbe matched by the given CSS selector string
querySelectorAll *	returns an array of all elementsthat would be matched by the given CSS selector string

❖ Commonly Used jQuery Event Methods

All the different visitors' actions that a web page can respond to are called events. An event represents the precise moment when something happens.

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

- **$(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();
});
```

Output :-

```
If you click on me, I will disappear.
Click me away!
Click me too!
```

- **dblclick()**

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

The function is executed when the user double-clicks on the HTML element:

Example :-

```
<!DOCTYPE html>

<html>

<head>

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

<script>

$(document).ready(function(){

  $("p").dblclick(function(){

    $(this).hide();

  });

});

</script>
```

```
</head>

<body>

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

<p>Click me away!</p>

<p>Click me too!</p>

</body>

</html>
```

Output:-

```
If you double-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:

```
<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:

```
<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>
```

- **mousedown()**

  The mousedown() method attaches an event handler function to an HTML element.
  The function is executed, when the left, middle or right mouse button is pressed down,
  while the mouse is over the HTML element:

```
<script>
$(document).ready(function(){
  $("#p1").mousedown(function(){
    alert("Mouse down over p1!");
  });
});
</script>
</head>
<body>
<p id="p1">This is a paragraph.</p>
</body>
</html>
```

- **mouseup()**

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

```
<script>
$(document).ready(function(){
  $("#p1").mouseup(function(){
    alert("Mouse up over p1!");
  });
});
</script>
</head>
<body>
<p id="p1">This is a paragraph.</p>
</body>
</html>
```

- **hover()**

  The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.

```
<script>
$(document).ready(function(){
  $("#p1").hover(function(){
    alert("You entered p1!");
  },
  function(){
    alert("Bye! You now leave p1!");
  });
});
</script>
</head>
<body>
<p id="p1">This is a paragraph.</p>
```

- **focus()**

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

```
<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">
```