

Q.1) What is jQuery? Explain features supported by jQuery?

⇒ jQuery is a small, light-weight and fast JavaScript library. It is cross-platform & supports different type of browser. It is designed to simplify the client-side scripting of HTML. It makes things like HTML document traversal and manipulation, animation, event handling, and AJAX very simple with an easy-to-use API that work on a lot of different type of browsers.

- jQuery is platform-independent.
- jQuery means "write less do more"

Features of jQuery:-

- 1> DOM manipulation
- 2> Event Handling
- 3> Ajax support
- 4> Animations and effects.

1> DOM manipulation:-

jQuery uses Document object model manipulation library to traverse and select objects. All the elements of a webpage are represented in the form of tree where in jQuery syntax is used for finding, selecting, and traversing the elements of the DOM.

Ex:-

```
$ (document).ready(() => {
  $("hi").htm("Hello World");
})
```

## 2) Event Handling :-

Action or event handlers are functions that are activated by user interactions on your website / web page. For suppose your website has a text input box and a button where the user can enter his details & click on the button to submit the data.

Ex:-

```
$ (document).ready(() => {
  $("button").click ((event) => {
    alert ("You clicked me");
  })
})
```

## 3) Ajax support:-

If it is support ajax, you can develop a responsive and feature-rich site using AJAX technology.

## 4) Animation:-

The jquery animate() method is used to create custom animation by changing the CSS numerical properties of DOM element, for example width, height, margin, padding, opacity, top, left etc.

Ex:- \$ (selector).animate ({ properties, speed, callback })

Q.2) Write a short note on jquery ready () function.

⇒ The ready () function in jquery executes the code only when the DOM is fully loaded. It is an inbuilt function in jquery. It can fire before loading of all images, etc. but only when the DOM is ready. The code inserted between \$(document).ready() is executed only when the page is ready for Javascript code to execute.

Syntax:-

The commonly used Syntax of the ready() function is as follow  
\$(document).ready(function).

Example:-

```
<script>
  $(document).ready(function() {
    $("button").click(function() {
      $("#P1").css({"font-size": "30px",
                    "color": blue});
    });
  });
</script>
<body>
  <p id = "P1"> Welcome to the DYP-ATU </p>
  <button> Click me </button>
</body>
```

Function:- It is the mandatory parameter that specifies the function which execute after the document is loaded.

Ex:-

\$ (document).ready() function.

Q3) List jquery actions and explain any two:

- All the different visitor's action that can respond it called a web page event.
- jquery event are the action that can be detected by your web application.
- They are used to create dynamic web page.

1) Mouse Events

2) Keyboard Events

3) Form Events

4) Document / window Events.

1) Mouse Event :

1) click Event

2) dblclick

3) mouseenter

4) mouseleave

3) Click :-

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

Ex:-

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

## 2) dblclick():

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

Ex:-

```
$("p").dblclick(function() {  
    alert("You double clicked me!");  
});
```

## 3) mouseenter()

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

Ex:-

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

## 4) mousedown():

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

Ex:-

```
$("#p1").mousedown(function() {  
    alert("Mouse down over p1!");  
});
```

key board Event: 3 types

1> key press

2> key down

3> key up:-

1> Key press:-

Ex:-

<input type = "text" ...>

onkeypress = "my function ()>

2> Key down:-

Ex:-

<input type = "text" ...>

onkeydown = "my function ()>

3>

key up:-

<input type = "text" ...>

onkeyup = "myfunction ()>

Explain in brief : ~~jQuery Validation~~ ~~Validation~~ ~~jQuery~~ plugin.

plugins:

A plug-in is piece of code written in a standard JS file. These file provide useful jQuery methods which can be used along with jQuery library methods.

validation plugin:-

You can add these file to your projects via package managers like Bower or NPM. You can also just directly get a CDN link to the file and add them to a script tag on your webpage. Since this is a jQuery-based plugin, you'll also need to add a link to the jQuery library.

Ex:-

```
$ (function () {  
    $ ("#register-form").validate ({
```

rules: {

email: {

required : true, email: true  
},

password : " required ",

password2: {

required : true,

equalTo : "#password"

y};

message : {

email: {

required : " Please enter an email address ",

email: 'Please enter a `<input type="text">` element valid email address.'

```
{});  
y);
```

Q) Write categories of jQuery UI and explain.

JQuery UI Accordion.

→ JQuery UI Categories:-

- 1> Interaction
- 2> Widgets
- 3> Effect
- 4> Utilities

JQuery UI provides various features for creating highly interactive web application. In the term, jquery UI, the UI stand for User Interface.

• Feature:-

- 1> Open-Source
- 2> Free to use
- 3> Facilitates a powerful theme mechanism
- 4> Stable and long duration
- 5> Helpful documentation.

JQuery UI Accordion:-

The jquery UI Accordion is broken into sections. It probably resembles like tabs.

Syntax:-

```
$ (selector, context).accordion(option)
Method
```

1> First method:-

\$(selector, context).accordion(options) Method.

Ex:-

```
<script>
$(function () {
  $("#accordion").accordion();
})
```

```
<body>
<div id="accordion">
  <h3> what is WT? </h3>
  <div>
    <p> ----- </p>
  </div>
</div>
```

2) Second method: - first method is not

```
$( selector, context ).accordion( "action", params )
```

Ex:-

```
<script>
$(function () {
  $("#accordion-5").accordion({
    disabled: false
  });
  $("input").each(function () {
    $(this).change(function () {
      if ($(this).attr("id") == "disabled") {
        $("#accordion-5").accordion("option",
          "disabled", true);
      } else {
        $("#accordion-5").accordion("option", "disabled",
          false)
      }
    })
  })
})
```

Describe

JQuery

UI Tabs with example

Ent that allows you to create tabbed content section on a web page. It enables you to organize and present information in a tabbed format, where each tab represents a different content section.

Ex:-

```
<div id="tabs">
  <ul>
    <li><a href="#tab1">Tab1</a></li>
    <li><a href="#tab2">Tab2</a></li>
    <li><a href="#tab3">Tab3</a></li>
  </ul>
  <div id="tab1">
    <h2>Tab1 Content</h2>
    <p>This is the content for Tab1</p>
  </div>
  <div id="tab2">
    <h2>Tab2 Content</h2>
    <p>This is the content for Tab2</p>
  </div>
  <div id="tab3">
    <h2>Tab3 Content</h2>
    <p>This is the content for Tab3</p>
  </div>

```

**Java script Code:-**

```
$ (function () {
  $ ("#tabs").tabs();
});
```

In the HTML structure, we have a `<div>` element with id of "tabs" that serves as the container for the tabs. Inside the container, we have an unordered list (`<ul>`) containing individual list items (`<li>`) for each tab.

In the javascript code, we select the element with an id of "tabs" using the jquery selector `$("#tabs")` and apply the `tabs()` method to initialize the tabs functionality.

The result is a visually appealing and interactive tabbed interface. Users can click different tabs to switch between content sections, viewing the relevant information without leaving the page.

**Q.7) Describe jquery UI Tooltip with example**

- To allow customization and to add new themes, the jquery UI tooltip is used.
- To display a title in the title box next to any element, on hovering the element with a mouse, the tooltips are used with the element.

Tooltip can be used by adding the `title` attribute to input element. The value of the `title` attribute will then be used as a tooltip.

**Syntax:-**

The `tooltip()` method can be used to add forms.

`$ (selector, context).tooltip(options) method`

OR `$(selector, context).tooltip("action", [param])`

`Method.name(selector, context, options)`

↳ first method:-

`content, disabled, hide, items, position, show, tooltipClass and track`

↳ second method:-

`close(), destroy(), disable(), enable(), open(), option(optionName), option(), option(optionName, value), option(options), widget()`

Ex:- HTML structure :-

```
<span title="Tooltip text" class="tooltip">
    Hover over me </span>
```

Java script code:-

```
$ (function() {
    $('.tooltip').tooltip();
})
```

The jquery UI Tooltip component offers various customization options, including:

↳ positioning:-

You can specify where the tooltip should be positioned relative to the element such as above below to the left or to the right.

↳ Styling:-

You can customize the appearance of

the tooltip by applying CSS style to match your website design and layout.

3) Delay:- You can control the delay the tooltip appears and how long it stays visible.

4) content:- In addition to the title attribute you can provide a tooltip content dynamically using AJAX or other methods.

5) events and callbacks:- You can handle tooltip-related event and customize the behavior using callback function.