



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100
Department of MCA

INTERNAL ASSESSMENT TEST – 1 SOLUTION SET

Subject & Code : Advanced Web Programming (17MCA42) **Time** : 11:30 am – 1:00 pm

Note: Note: Answer any FIVE full questions, choosing one full question from each Part

1) What is jQuery? Explain the importance features of 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 website.

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:

HTML/DOM manipulation

CSS manipulation

HTML event methods

Effects and animations

AJAX

Utilities

- **DOM manipulation** – The jQuery made it easy to select DOM elements, negotiate them and modifying their content by using cross-browser open source selector engine called Sizzle.
- **Event handling** – The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
- **AJAX Support** – The jQuery helps you a lot to develop a responsive and feature rich site using AJAX technology.
- **Animations** – The jQuery comes with plenty of built-in animation effects which you can use in your websites.
- **Lightweight** – The jQuery is very lightweight library - about 19KB in size (Minified and zipped).
- **Cross Browser Support** – The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+
- **Latest Technology** – The jQuery supports CSS3 selectors and basic XPath syntax.



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

2) Explain jQuery selectors in detail with examples

jQuery Selector

- A jQuery Selector is a function which makes use of expressions to find out matching elements from a DOM based on the given criteria.
- Selectors are used to select one or more HTML elements using jQuery. Once an element is selected then we can perform various operations on that selected element.
- Syntax
\$(selector)
or
jQuery(selector)

1) Name - The element selector selects all the elements that have a tag name of T.

Syntax

\$('tagname')

tagname – Any standard HTML tag name like div, p, em, img, li etc.

Like any other jQuery selector, this selector also returns an array filled with the found elements.

Eg –

- \$('p') – Selects all elements with a tag name of p in the document.
- \$('div') – Selects all elements with a tag name of div in the document.

2) #ID - The element ID selector selects a single element with the given id attribute.

Syntax

\$('#elementid')

elementid – This would be an element ID. If the id contains any special characters like periods or colons you have to escape those characters with backslashes.

Like any other jQuery selector, this selector also returns an array filled with the found element.

Eg –

- \$('#myid') – Selects a single element with the given id myid.
- \$('div#yourid') – Selects a single division with the given id yourid.

3) .class - The element class selector selects all the elements which match with the given class of the elements.

Syntax

\$('.classid')

classid – This is class ID available in the document.

Eg –

\$('.big') – Selects all the elements with the given class ID big.

\$('p.small') – Selects all the paragraphs with the given class ID small.

\$('.big.small') – Selects all the elements with a class of big and small.

4) Universal(*) - selects all the elements available in the document.



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

Syntax

`$('.*)`

*– A symbolic star.

- Like any other jQuery selector, this selector also returns an array filled with the found elements.

Eg-

- `$('.*)` selects all the elements available in the document.

5) Multiple elements E,F,G - selects the combined results of all the specified selectors E, F or G.

You can specify any number of selectors to combine into a single result. Here order of the DOM elements in the jQuery object aren't necessarily identical.

Syntax

`$('E, F, G,...')`

where E,F,G are any valid selectors

Eg-

- `$('div, p')` – selects all the elements matched by div or p.
- `$('p strong, .myclass')` – selects all elements matched by strong that are descendants of an element matched by p as well as all elements that have a class of myclass.
- `$('p strong, #myid')` – selects a single elements matched by strong that is descendant of an element matched by p as well as element whose id is myid.

3) Explain the utility functions in jQuery

Utility functions

Jquery provides several utilities in the format of `$(name space)`. These methods are helpful to complete the programming tasks.

1)`$.trim()` - used to Removes leading and trailing whitespace

Eg-

```
$.trim( " lots of extra whitespace " );
```

2) `$.each()` - used to Iterates over arrays and objects

Eg-

```
$.each([ "foo", "bar", "baz" ], function( idx, val )  
    { console.log( "element " + idx + " is " + val ); });  
$.each({ foo: "bar", baz: "bim" }, function( k, v )  
    { console.log( k + " : " + v ); });
```

3) `$.isArray()`- used to return a value's index in an array, or -1 if the value is
array.

not in the



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

Eg-

```
var myArray = [ 1, 2, 3, 5 ];  
if ( $.isArray( 4, myArray ) !== -1 )  
{ console.log( "found it!" ); }
```

4) \$.extend()- used to change the properties of the first object using the subsequent objects.

properties of

Eg-

```
var firstObject = { foo: "bar", a: "b" };  
var secondObject = { foo: "baz" };  
var newObject = $.extend( firstObject, secondObject );  
console.log( firstObject.foo );  
console.log( newObject.foo );
```

5) \$.fn.extend - used to extends the jQuery prototype

4) List and explain attribute selectors in jQuery

1) The [attribute] selector selects each element with the specified attribute.

Syntax

```
$("#[attribute]")
```

Eg- \$("#[id]")

2) The [attribute=value] selector selects each element with the specified attribute and value.

Syntax

```
$("#[attribute=value]")
```

3) The [attribute\$=value] selector selects each element with a specific attribute, with a value ending in a specific string.

Syntax

```
$("#[attribute$='value']")
```

Eg – \$("#a[href\$='.org']")

4) The [attribute^=value] selector selects each element with a specific attribute, with a value beginning in a specific string.

Syntax

```
$("#[attribute^='value']")
```

Eg- \$("#input[name^='nation']")

Select all <input> elements with a name attribute that starts with "nation":

5) The [attribute*=value] selector selects each element with a specific attribute, with a value containing a string.

Syntax

```
$("#[attribute*='value']")
```

Eg –



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

```
$("#input[name*='nation']")
```

Select all <input> elements with a name attribute that contains the word "nation"

5)A. Write a jQuery program to add a 'p' element to 'h1' element. And add a CSS style – background color: yellow

```
<!DOCTYPE html>
<html>
<head>
<script src=" jquery-3.2.1.min.js "></script>
<script>
$(document).ready(function(){
    $("#h1").add("p").add("span").css("background-color", "yellow");
});
</script>
</head>
<body>
<h1>Welcome</h1>
<p>A p element.</p>
<p>Another p element.</p>
<span>A span element.</span>
<span>A span element.</span><br><br>
<div>This example adds the same css style for both p and span elements, as the existing h1
element.</div>
</body>
</html>
```

b. What is Query chaining. Give example.

Let's say that we want to apply a thick border to all elements with alt attributes, and the apply a level of transparency to all elements with either alt or title attributes. The comma operator (,) of CSS selectors won't help us with this one because we want to apply an operation to a wrapped set and *then* add more elements to it. We could easily accomplish this with multiple statements, but it would be more efficient and elegant to use the power of jQuery chaining to accomplish the task in a single statement, such as

```
$('#img[alt]').addClass('thickBorder').add('img[title]').addClass('seeThrough')
```

- In this statement, we create a wrapped set of all elements with alt attributes, apply a predefined class that applies a thick border, add the elements that have title attributes, and finally apply a class that applies transparency to the newly augmented set.

6) How do you obtain the subsets of the wrapped set? Explain the functions

Obtaining subsets of the wrapped set



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

jQuery slice() method - provides a method to obtain a subset of the wrapped set, based on the position of elements within the set. This command creates and returns a *new* set from any contiguous portion, or a slice, of an original wrapped set.

Syntax

slice(begin, end)

Eg-

\$('*').slice(2,3);

selects all elements on the page and then generates a new set containing the third element in the matched set.

\$('*').slice(0,4);

selects all elements on the page and then creates a set containing the first four elements.

This method is used to limit the selection of elements in a group, by a start and end point: the *start* parameter is a starting index (starts at 0) from which to create the subset, and the *stop* parameter is an optional ending point

* find() method -returns descendant elements of the selected element.

A descendant is a child, grandchild, great-grandchild, and so on. This method traverse downwards along descendants of DOM elements, all the way down to the last descendant.

Syntax

\$(selector).find(filter)

Filter- is a required selector expression, element or jQuery object to filter the search for descendants

*contains() method – jQuery also provides a method to find elements that contain a specified string. The contains() method will return a new wrapped set that consists of all elements that contain the passed string anywhere within its body content.

Syntax

contains(text)

Returns a new wrapped set composed of elements that contain the text string passed as the text parameter

Eg –

\$('p').contains('Lorem ipsum')

This expression yields a wrapped set containing all paragraphs that contain the text *Lorem ipsum*.

*is () method- The is() method checks if one of the selected elements matches the *selectorElement*.

Syntax

is(selector)

Determines if any element in the wrapped set matches the passed selector expression



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

7) A. How do you manage query chains using clone () and andself() methods .Explain

By chaining jQuery wrapper methods together we can perform a lot of activity in a single statement. This chaining ability not only allows us to write powerful operations in a concise manner, but it also improves efficiency because wrapped sets do not have to be recomputed in order to apply multiple commands to them.

*clone() method - makes a copy of selected elements, including child nodes, text and attributes.

Syntax

`$(selector).clone(true|false)`

*andself() method –

The andSelf() method adds the previous selection to the current selection. The method is useful when you have multiple traversals in your script and then adding something that was matched before the last traversal.

Syntax

`selector.andSelf()`

b. Write a jQuery program to Limit character input in the textarea including count

```
<!DOCTYPE html>
<html>
<head>
<title>Limit character input in the textarea including count</title>
<script src="jquery-3.2.1.min.js"></script>
<style type="text/css">
textarea {
  display:block;
  margin: 1em 0;
}
</style>
</head>
<body>
<script>
  var maxLength = 15;
  $('textarea').keyup(function() {
    var textlen = maxLength - $(this).val().length;
    $('#rchars').text(textlen);
  });
</script>
<form>
<label>Maximum 15 characters</label>
```



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

```
<textarea id="textarea" maxlength="15"></textarea>
  <span id="rchars">15</span> Character(s) Remaining
</form>
</body>
</html>
```

8) Explain the method used for manipulating element properties in jQuery

Manipulating element properties

The easiest way to inspect or modify the component elements of a matched set is with the each() command.

Syntax

```
each(iterator)
```

Traverses all elements in the matched set invoking the passed iterator function for each.

```
<!DOCTYPE html>
<html>
<head>
<script src=" jquery-3.2.1.min.js "></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("li").each(function(){
      alert($(this).text())
    });
  });
});
</script>
</head>
<body>
<button>Alert the value of each list item</button>
<ul>
  <li>Coffee</li>
  <li>Milk</li>
  <li>Soda</li>
</ul>
</body>
</html>
```



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100
Department of MCA

9) Explain the DOM Event Models

The DOM Level 0 Event Model

Under this event model, event handlers are declared by assigning a reference to a function instance to properties of the DOM elements.

These properties are defined to handle a specific event type;

Eg-

A click event is handled by assigning a function to the onclick property, and a mouseover event by assigning a function to the onmouseover property of elements that support these event types.

The browsers allow us to specify the body of an event handler function as attribute values in the DOM elements' HTML, providing a shorthand for creating event handlers

When an event is triggered on an element in the DOM tree, the event-handling mechanism of the browser checks to see if a handler has been established. If it has established, then its parent is checked, then its parent, then its parent, and on and on, all the way up to the top of the DOM tree. Because the event handling propagates upward like the bubbles in a champagne flute (assuming we view the DOM tree with its root at the top), this process is termed *event bubbling*.

The DOM Level 2 Event Model

One severe shortcoming of the DOM Level 0 Event Model is that, because a property is used to store a reference to a function that's to serve as an event handler, only one event handler per element can be registered for any specific event type at a time. DOM Level 2 event handlers—also termed *listeners*—are established via an element *method*. Each DOM element defines a method named **addEventListener()** that's used to attach event handlers (listeners) to the element.

Syntax

```
addEventListener( eventType, listener, useCapture)
```

eventType - string that identifies the type of event to be handled like click, mouseover, keydown etc .

listener - is a reference to the function (or inline function) that's to be established as the handler for the named event type on the element.

useCapture – Boolean operation which makes the event first propagates from the root of the DOM tree down to the target element

Under the DOM Level 2 Event Model, when an event is triggered, the event first propagates from the root of the DOM tree down to the target element and then propagates again from the target element up to the DOM root. The former phase (root to target) is called *capture phase*, and the latter (target to root) is called *bubble phase*.

The Internet Explorer Event Model



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

Internet Explorer (both IE6 and, most disappointingly, IE7) doesn't provide support for the DOM Level 2 Event Model. Both these versions of Microsoft's browser provide a proprietary interface that closely resembles the bubble phase of the standard model.

The Internet Explorer Model defines a method named `attachEvent()` for each DOM element.

Syntax

```
attachEvent(eventName,handler)
```

`eventName` - names the event type to be attached like `onclick`, `onmouseover`, `onkeydown`(Dom 0 level)

`Handler` - function to be established as the handler, and as in the Basic Model, the Event instance must be fetched from the `window.event` property.

The jQuery Event Model

The jQuery Event Model, exhibits the following features:

- Provides a unified method for establishing event handlers
- Allows multiple handlers for each event type on each element
- Uses standard event-type names: for example, `click` or `mouseover`
- Makes the Event instance available as a parameter to the handlers
- Normalizes the Event instance for the most often used properties
- Provides unified methods for event canceling and default action blocking

The feature set of the jQuery Event Model closely resembles that of the Level 2 Model while supporting both standards-compliant browsers and Internet Explorer with a single API.

10) Write notes on

a) Document Ready handler

The document ready handler

Syntax

```
$( document ).ready()
```

A page can't be manipulated safely until the document is "ready." jQuery detects this state of readiness for you.

Code included inside `$(document).ready()` will only run once the page Document Object Model (DOM) is ready for JavaScript code to execute.

Code included inside `$(window).on("load", function() { ... })` will run once the entire page (images or iframes), not just the DOM, is ready.

Eg –

```
// A $( document ).ready() block.  
$( document ).ready(function() {  
    console.log( "ready!" );  
});
```



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

```
});
```

Shorthand for `$(document).ready()` –

```
$(function() {  
    console.log( "ready!" );  
});
```

But it's best to use the long form.

* You can also pass a named function to `$(document).ready()` instead of passing an anonymous function

Eg –

// Passing a named function instead of an anonymous function.

```
function readyFn( jQuery ) { // Code to run when the document is ready.  
    $( document ).ready( readyFn );  
// or:  
$( window ).on( "load", readyFn );
```

b) Methods to retrieve and set form values

1) jQuery's `val()` command

Takes care of the most common cases, returning the value attribute of a form element for the first element in the wrapped set. The `val()` command is also used for this purpose by supplying a value.

Syntax

```
val()
```

Returns the value property of the first element in the matched set. When the element is a multi-select element, the returned value is an array of all selections.

Eg –

```
$('#[name=radioGroup]:checked').val()
```

val(value) command

This method sets the value of the value attribute for ALL matched elements.

Syntax

```
$(selector).val(value)
```

Eg- Set the value of the `<input>` field.

```
$("#button").click(function(){  
    $("#input:text").val("Glenn Quagmire");  
});
```

val(values) command

Causes any check boxes, radio buttons, or options of `<select>` elements in the wrapped set to become checked or selected if their values match any of the values passed in the values array.

Eg- `$('#input,select').val(['one','two','three']);`



PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA
