CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

Internal Assessment Test1 Answer Key– – Feb. 2019

Sub:	Advanced Web Programming					Sub Code:	17MCA42	Branch:	MCA
Date:	5.03.2019	Duration:	90 min's	Max Marks:	50	Sem	IV		OBE

1 Explain about manipulating element properties. [10]

jQuery doesn't possess a specific command to obtain or modify the properties of elements. Rather, we use the native JavaScript notation to access the properties and their values. The trick is in getting to the element references in the first place. The easiest way to inspect or modify the component elements of a matched set is with the each() command. The syntax of this command is as follows:

Command syntax: each

`each(iterator)`

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

Parameters

`iterator` (Function) A function called for each element in the matched set. The parameter passed to this function is set to the zero-based index of the element within the set, and the element itself is available as the `this` property of the function.

Returns

The wrapped set.

Example :

```
$('img').each(function(n){
        this.alt='This is image['+n+'] with an id of '+this.id;
});
```

Fetching attribute values

The attr() command can be used to either fetch the value of an attribute from the first element in the matched set or set attribute values onto all matched elements.
The syntax for the fetch variant of the attr() command is as follows:

Command syntax: attr

`attr(name)`

Obtains the values assigned to the specified attribute for the first element in the matched set.

Parameters

`name` (String) The name of the attribute whose value is to be fetched.

Returns

The value of the attribute for the first matched element. The value `undefined` is returned if the matched set is empty or the attribute doesn't exist on the first element.

The jQuery attr() command is much more than a wrapper around the JavaScript getAttribute() and setAttribute() methods. In addition to allowing access to the set of element attributes, jQuery provides access to some commonly used properties that, traditionally, have been a thorn in the side of page authors everywhere due to their browser dependency.

Normalized name	Source name
class	className
cssFloat	styleFloat for IE, cssFloat for others (when used with .css)
float	styleFloat for IE, cssFloat for others (when used with .css)
for	htmlFor

Normalized name	Source name
maxlength	maxLength
readonly	readOnly
styleFloat	styleFloat for IE, cssFloat for others (when used with .css)

Setting attribute values

There are two ways to set attributes onto elements in the wrapped set with jQuery.

Command syntax: attr

`attr(name,value)`

Sets the named attribute onto all elements in the wrapped set using the passed value.

Parameters

`name` (String) The name of the attribute to be set.

`value` (String|Object|Function) Specifies the value of the attribute. This can be any JavaScript expression that results in a value, or it can be a function. See the following discussion for how this parameter is handled.

Returns

The wrapped set.

Command syntax: attr

`attr(attributes)`

Sets the attributes and values specified by the passed object onto all elements of the matched set

Parameters

`attributes` (Object) An object whose properties are copied as attributes to all elements in the wrapped set

Returns

The wrapped set

Removing attributes

In order to remove an attribute from DOM elements, jQuery provides the removeAttr() command. Its syntax is as follows:

Command syntax: removeAttr

`removeAttr(name)`

Removes the specified attribute from every matched element

Parameters

`name` (String) The name of the attribute to be removed

Returns

The wrapped set

2 Explain the basics selectors of jQuery with suitable examples[10]

For applying styles to page elements, web developers have become familiar with a small, but powerful and useful, group of selection methods that work across all browsers. Those methods include selection by an element's ID, CSS class name, tag name, and the DOM hierarchy of the page elements.

Here are some examples to give you a quick refresher.

- a—This selector matches all link (<a>) elements.
- #specialID—This selector matches elements that have an id of specialID.
- .specialClass—This selector matches elements that have the class of specialClass.
- a#specialID.specialClass—This selector matches links with an id of specialID and a class of specialClass.
- p a.specialClass—This selector matches links with a class of specialClass declared within <p> elements

We can mix and match the basic selector types to select fairly fine-grained sets of elements. In fact, the most fancy and creative websites use some combination of these basic options to create their dazzling displays

To select elements using jQuery, we wrap the selector in $(), as in

$("p a.specialClass")

With a few exceptions, jQuery is fully CSS3 compliant, so selecting elements this way will come with no surprises; the same elements that would be selected in a style sheet by a standards-compliant browser will be selected by jQuery's selector engine.

3 Write jQuery program to solve the following: a) Limit character input in the textarea including count. b) Based on check box, disable/enable the form submit button.

```
<!DOCTYPE html>
<html>
<head>
	<title>lab1a</title>
	<script src="jquery-1.2.1.js"></script>
	<script>
		var maxlength = 15;
		$(document).ready(function(){
			$('textarea').keyup(function() {
			  var textlen = maxlength - $(this).val().length;
			  $('#rchars').text(textlen);
			});
		});

	</script>

</head>
<body align="center">
	<form>
		<label>Maximum 15 Characters</label> <br/><br/>
		<textarea id="textarea" maxlength="15"></textarea><br/><br/>
		<label><span id="rchars">15</span> Character(s) Remaining</label>
	</form>
</body>
</html>

<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">
	<title>lab1b</title>
	<script src="jquery-1.2.1.js"></script>
	<script>
```

```
            $(document).ready(function(){
                $('#accept').click(function() {
                    if ($('#submitbtn').is(':disabled')) {
                        $('#submitbtn').removeAttr('disabled');
                    } else {
                        $('#submitbtn').attr('disabled', 'disabled');
                    }
                });
            });
        </script>

</head>
<body>
    <input id="accept" name="accept" type="checkbox" value="y"/>I accept<br>
    <input id="submitbtn" disabled="disabled" name="Submit" type="submit"
value="Submit" />
</body>
</html>
```

4 a) Explain any two methods to fade an element in and out of visibility. [5]

fadeIn() and fadeOut(), only affect the opacity of the elements.

Command syntax: fadeOut

`fadeOut(speed,callback)`

Causes any matched elements that aren't hidden to be removed from the display by gradually changing their opacity to 0% and then removing the element from the display. The duration of the change in opacity is determined by the `speed` parameter. Any elements that are already hidden aren't affected.

Parameters

`speed`	(Number|String) Specifies the duration of the effect as a number of milliseconds or as one of the predefined strings: *slow*, *normal*, or *fast*. If omitted, the default is *normal*.
`callback`	(Function) An optional function invoked when the animation completes. No parameters are passed to this function, but the function context (`this`) is set to the element that was animated.

Returns

The wrapped set.

Command syntax: fadeIn

`fadeIn(speed,callback)`

Causes any matched elements that are hidden to be shown by gradually changing their opacity to their natural value. This value is either the opacity originally applied to the element, or 100%. The duration of the change in opacity is determined by the `speed` parameter. Any elements that aren't hidden aren't affected.

Parameters

`speed`	(Number|String) Specifies the duration of the effect as a number of milliseconds or as one of the predefined strings: *slow*, *normal*, or *fast*. If omitted, the default is *normal*.
`callback`	(Function) An optional function invoked when the animation completes. No parameters are passed to this function, but the function context (`this`) is set to the element that was animated.

Returns

The wrapped set.

b) Explain html() and text() methods for replacing html and text contents [5]

html() command, which allows us to retrieve the HTML contents of an element when used without parameters or, to set its contents when used with a parameter.

Command syntax: html

`html()`

Obtains the HTML content of the first element in the matched set.

Parameters
none

Returns
The HTML content of the first matched element. The returned value is identical to accessing the `innerHTML` property of that element.

Command syntax: html

`html(text)`

Sets the passed HTML fragment as the content of all matched elements

Parameters
`text` (String) The HTML fragment to be set as the element content

Returns
The wrapped set

We can also set or get only the text contents of elements. The text()command, when used without parameters, returns a string that's the concatenation of all text.

Command syntax: text

`text()`

Concatenates all text content of the wrapped elements and returns it as the result of the command

Parameters
none

Returns
The concatenated string

Command syntax: text

`text(content)`

Sets the text content of all wrapped elements to the passed value. If the passed text contains angle brackets (< and >), these characters are replaced with their equivalent HTML entities.

Parameters
`content` (String) The text content to be set into the wrapped elements. Any angle bracket characters are escaped as HTML entities.

Returns
The wrapped set.

5 a) What is jquery? explain document ready handler with syntax and example. [5]

- jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax.
- jQuery is a lightweight, "write less, do more", JavaScript library.
- Using raw JavaScript can result in dozens of lines of code.
- The creators of jQuery specifically created the library to make common tasks trivial.
- The real power in this jQuery statement comes from the selector, an expression for identifying target elements on a page that allows us to easily identify and grab the elements we need.

jQuery provides a simple means to trigger the execution of code once the DOM tree, but not external image resources, has loaded. The formal syntax to define such code is as follows:

Syntax

```
$(document).ready(function() {
        //jquery statements
});
```

Example

```
$(document).ready(function() {
        $("table tr:nth-child(even)").addClass("even");
});
```

First, we wrap the document instance with the jQuery() function, and then we apply the ready() method, passing a function to be executed when the document is ready to be manipulated.
We called that the formal syntax for a reason; a shorthand form used much more frequently is as follows:

```
$(function() {
        $("table tr:nth-child(even)").addClass("even");
});
```

By passing a function to $(), we instruct the browser to wait until the DOM has fully loaded (but only the DOM) before executing the code. Even better, we can use this technique multiple times within the same HTML document, and the browser will execute all of the functions we specify in the order that they are declared within the page.

b) Explain wrap() and unwrap() in jQuery with example; [5]

Another type of DOM manipulation that we'll often need to perform is to wrap an element (or series of elements) in some markup. For example, we might want to wrap all links of a certain class inside a . We can accomplish such DOM modifications by using jQuery's wrap() command. Its syntax is as follows:

Command syntax: wrap

`wrap(wrapper)`

Wraps the elements of the matched set with the passed HTML tags or a clone of the passed element.

Parameters

`wrapper` (String|Element) The opening and closing tags of the element with which to wrap each element of the matched set, or an element to be cloned and server as the wrapper.

Returns

The wrapped set.

Example

To wrap each link with the class surprise in a with the class hello, we write

```
$("a.surprise").wrap("<div class='hello'></div>")
```

The .unwrap() method removes the element's parent and returns the unwrapped content. This is effectively the inverse of the .wrap() method. The matched elements (and their siblings, if any) replace their parents within the DOM structure

.unwrap()

This signature does not accept any arguments

unwrap([selector])

A selector to check the parent element against. If an element's parent does not match the selector, the element won't be unwrapped.

6 Discuss all the general syntactic characteristics of php[10]

PHP scripts are either embedded in XHTML documents or are in files that are referenced by XHTML documents. PHP code is embedded in XHTML documents by enclosing it between the `<?php` and `?>` tags.

If a PHP script is stored in a different file, it can be brought into a document with the `include` construct, which takes the filename as its parameter. For example:

```
include("table2.inc");
```

This construct causes the contents of the file `table2.inc` to be copied into the document where the call appears. The included file can contain XHTML markup or client-side script, as well as PHP code, but any PHP script it includes must be the content of a `<?php` tag, even if the `include` appears in the content of a `<?php` tag. The PHP interpreter changes from interpret to copy mode when an `include` is encountered.

All variable names in PHP begin with dollar signs ($). The part of the name after the dollar sign is like the names of variables in many common programming languages: a letter or an underscore followed by any number (including zero) of letters, digits, or underscores. PHP variable names are case sensitive.

Table 11.1 lists the PHP reserved words. Although variable names in PHP are case sensitive, neither reserved words nor function names are. For example, there is no difference between `while`, `WHILE`, `While`, and `wHiLe`.

Table 11.1 The reserved words of PHP

and	else	global	require	virtual
break	elseif	if	return	xor
case	extends	include	static	while
class	false	list	switch	
continue	for	new	this	
default	foreach	not	true	
do	function	or	var	

PHP allows comments to be specified in three different ways. Single-line comments can be specified either with #, as in Perl, or with //, as in JavaScript. Multiple-line comments are delimited with /* and */, as in many other programming languages. Perl-style comments are most frequently used in PHP scripts.

PHP statements are terminated with semicolons. Braces are used to form compound statements for control structures. Unless used as the body of a function definition, compound statements cannot be blocks (they cannot define locally scoped variables).

7 Explain following terms [10]

i) IsSet() ii) unset iii)error_reporting() iv) unbound variable v) interpolation

IsSet()

he isset () function is used to check whether a variable is set or not. If a variable is already unset with unset() function, it will no longer be set. The isset() function return false if testing variable contains a NULL value.

isset(variable1, variable2.....)

unset

unset (mixed $var [, mixed $...])

unset() destroys the specified variables.

The behavior of unset() inside of a function can vary depending on what type of variable you are attempting to destroy.

If a globalized variable is unset() inside of a function, only the local variable is destroyed. The variable in the calling environment will retain the same value as before unset() was called.

error_reporting()

If you want to be informed when an unbound variable is referenced, use the `error_reporting` function to change the error-reporting level of the PHP interpreter to 15. The following call is placed at the beginning of the script in the document file:

```
error_reporting(15);
```

The default error-reporting level is 7, which does not require the interpreter to report the use of an unbound variable.

unbound variable

Because PHP is dynamically typed, it has no type declarations. In fact, there is no way or need to ever declare the type of a variable.[3] The type of a variable is set every time it is assigned a value. An unassigned variable, sometimes called an *unbound variable*, has the value `NULL`, which is the only value of the NULL type.

interpolation

String literals are defined with either single (') or double quotes (") delimiters. In single-quoted string literals, escape sequences, such as `\n`, are not recognized as anything special, and the values of embedded variables are not substituted. (This substitution is called *interpolation*.) In double-quoted string literals, escape sequences are recognized, and embedded variables are replaced by their current values. For example, the value of

```
'The sum is: $sum'
```

is exactly as it is typed. However, assuming the current value of `$sum` is `10.2`, the value of

```
"The sum is: $sum"
```

is

```
The sum is: 10.2
```

8 List and explain commonly used string functions in php[10]

Function	Parameter Type	Returns
`strlen`	A string	The number of characters in the string
`strcmp`	Two strings	Zero if the two strings are identical, a negative number if the first string belongs before the second (in the ASCII sequence), or a positive number if the second string belongs before the first
`strpos`	Two strings	The character position in the first string of the first character of the second string, if the second string is in the first string; `false` if it is not there
`substr`	A string and an integer	The substring of the string parameter, starting from the position indicated by the second parameter; if a third parameter is given (an integer), it specifies the length of the returned substring
`chop`	A string	The parameter with all whitespace characters removed from its end
`trim`	A string	The parameter with all whitespace characters removed from both ends
`ltrim`	A string	The parameter with all whitespace characters removed from its beginning
`strtolower`	A string	The parameter with all uppercase letters converted to lowercase
`strtoupper`	A string	The parameter with all lowercase letters converted to uppercase