Name-Divya R Gavandi Name-Palak Davda

Roll no-74 Roll no-72

Se comps Se comps

Experiment no-4

Aim: Jquery

Design web pages  using Client side scripting-

jQuery Basics, jQuery Getters and Setters, Altering Document Structure, Handling events with jQuery, Animated Effects, Utility functions, jQuery Selectors and Selection Methods, Extending jQuery with Plugins, The jQuery UI Library.

Theory: jQuery

* jQuery is a JavaScript Library.
* jQuery greatly simplifies JavaScript programming.
* jQuery is easy to learn.

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 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

Tip: In addition, jQuery has plugins for almost any task out there.

Adding jQuery to Your Web Pages

There are several ways to start using jQuery on your web site. You can:

* Download the jQuery library from jQuery.com
* Include jQuery from a CDN, like Google

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)

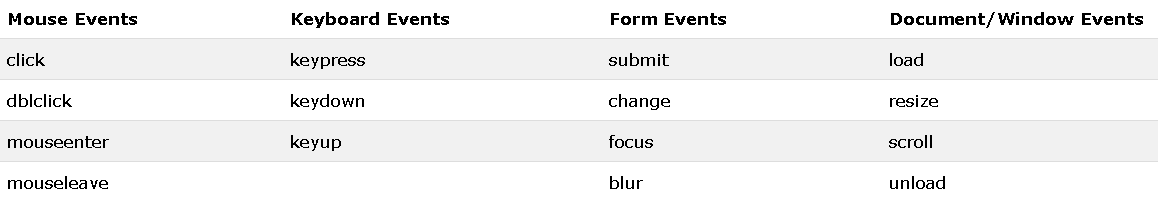
Examples:

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

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

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

$("#test").hide() - hides the element with id="test".



jQuery contains powerful methods for changing and manipulating HTML elements and attributes.

jQuery DOM Manipulation

One very important part of jQuery is the possibility to manipulate the DOM.

jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.

DOM = Document Object Model  
  
The DOM defines a standard for accessing HTML and XML documents:  
  
*"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*

**Get Content - text(), html(), and val()**

Three simple, but useful, jQuery methods for DOM manipulation are:

* text() - Sets or returns the text content of selected elements
* html() - Sets or returns the content of selected elements (including HTML markup)
* val() - Sets or returns the value of form fields

**Set Content - text(), html(), and val()**

We will use the same three methods from the previous page to set content:

* text() - Sets or returns the text content of selected elements
* html() - Sets or returns the content of selected elements (including HTML markup)
* val() - Sets or returns the value of form fields

**jQuery Animations - The animate() Method**

* The jQuery animate() method is used to create custom animations.
* Syntax:
* $(*selector*).animate({*params*}*,speed,callback*);
* The required params parameter defines the CSS properties to be animated.
* The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
* The optional callback parameter is a function to be executed after the animation completes.

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")

--------------------------------------------------------------------------------------------------------------------------------------

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")

The .class Selector

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

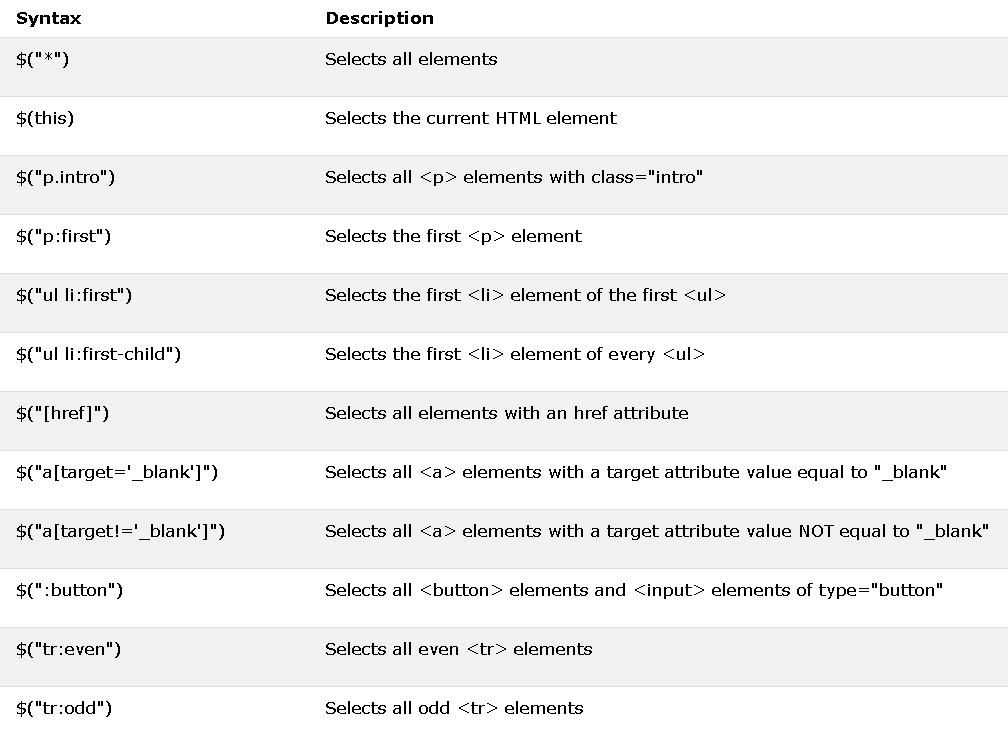
To find elements with a specific class, write a period character, followed by the name of the class:

$(".test")

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.

More Examples of jQuery Selectors



Plugins

A jQuery plugin is simply a new method that we use to extend jQuery's prototype object. By extending the prototype object you enable all jQuery objects to inherit any methods that you add. As established, whenever you call jQuery() you're creating a new jQuery object, with all of jQuery's methods inherited.

The idea of a plugin is to do something with a collection of elements. You could consider each method that comes with the jQuery core a plugin, like .fadeOut() or .addClass().

You can make your own plugins and use them privately in your code or you can release them into the wild. There are thousands of jQuery plugins available online. The barrier to creating a plugin of your own is so low that you'll want to do it straight away!

Conclusion :Hence I have implemented this experiment using jQuery.