Basics of JQuery:

1. On load event

window.**onload** = function(){

// ……

}

Unfortunately, the code doesn't run until all images are finished downloading, including banner ads.

To run code as soon as DOM is ready we can use:

$(document).ready(function()

{

//…..

});

// **Shorthand** for $( document ).ready()

$(function() {

console.log( "ready!" );

});

1. event.preventDefault()
2. addClass(‘class\_name’) and removeClass(‘class\_name’)
3. Special effects : show(‘fast’); hide(‘slow’)
4. Callback and functions:

$.get(“myHtml.html”,myCallback);

Where myCallback is function name which will get execute after execution of it’s parent function.

**Callback function with arguments:**

$.get(“myHtml.html”,function(){

myCallback(param1,param2);

});

1. Most jQuery methods are called on jQuery objects e.g. $(“h1”).remove

* Methods called on jQuery selections are in the **$.fn namespace**, and automatically receive and return the selection as **this**. E.g. *.each()*
* Methods in the **$ namespace** are generally utility-type methods, and do not work with selections; they are not automatically passed any arguments, and their return value will vary. E.g.- *$.each()*

1. **Namespaces** allow us to group code and help us to avoid name-collisions.

// not safe, if there's another object with this name we will overwrite it

*var MYAPPLICATION = {};*

// We need to do a check before we create the namespace

*if (typeof MYAPPLICATION === "undefined") {*

*var MYAPPLICATION = {};*

*}*

// or a shorter version

*var MAYAPPLICATION = MYAPPLICATION || {};*

1. **Attributes**

$( "a" ).attr( "href", "allMyHrefsAreTheSameNow.html" );

$( "a" ).attr({

title: "all titles are the same too!",

href: "somethingNew.html"

});

1. **Element Selection:**

If you made selection and use it again then use store selection and then use it:

var $myDiv = $("#myId .myDiv");

Create variable using $ e.g. $myDiv - it's used to indicate that the variable contains a jQuery object. $ is not mandatory

* $(“\*”) // Select All Elements
* $(“#myId”) //By id
* $(“.myClass”) //By class
* $(" a[lang|='en'] ") //By attribute – Select element lang having value ‘en’
* $( "a[lang|='en']" ) //Attribute Contains Prefix Selector [name|="value"]
* $( "a[lang$='en']" ); //lang value ends with ‘en’
* $("tr:odd")
* $("div:visible") // jQuery tests the actual visibility of the element, not its

CSS visibility or display properties

* $( "div:animated" ) //Animated selection
* $("div: gt(2) ") // Except first 3 divs
* $( "div.foo" ).has( "p" ); // div.foo elements that contain <p> tags
* $( "h1" ).not( ".bar" ); // h1 elements that don't have a class of bar
* $( "ul li" ).filter( ".current" ); // unordered list items with class of current
* $( "ul li" ).first(); // just the first unordered list item
* $( "ul li" ).eq( 1 ); // the 2nd item
* $(“div”).hasClass(‘myClass’); // check the class present
* $(“div”).is(“myClass.mydiv”); //For more advanced matching
* $(“div”).is(“:hidden”); // We can check whether an element is hidden
* $(“#myDiv”).length // Check whether element exist
* $( "#some**\\**:id" ) // select an element by an ID that has characters used in CSS notation
* $( "#x" ).prop( "disabled", true ); // disable form elements
* $( "#x" ).prop( "checked", true ); // check or uncheck checkbox or radio button
* $(“#myselect”).val() // It will return selected value of value attr
* $(“#myselect option:selected”).text() // will return selected options text.
* $(“div:nth- child(3)”) ; // 3rd div element of it’s parent
* $(“div:nth-last-child(3)”) ; // 3rd last div element