jQuery documentation.

Why jQuery:

* Write less amount of code and get maximum benefit.

Features of jQuery:

* Dynamically manipulate the DOM. ( DOM is like., ‘P’ , ‘Div’ , ‘Span’ etc.,)
* Dynamically manipulate the CSS.
* Easily integrate effects and animations.
* Ajax calls.
* Easily integrate HTML events.

Integrating jQuery in your page.

* Include CDN path directly into your web page.
* Download the jQuery file into your system and integrate.

jQuery Syntax:

* A ‘$’ sign will be used to access jQuery
  + Ex:
    - $(“#demo”).hide();
      * Demo element will be hidden.

jQuery basic event.

* Document.ready.
  + Ex:
    - $(document).ready(function(){

});

jQuery selectors.

* jQuery selectors are HTML DOM elements. Ex: $(“#test”)

<div id = “progresscard”>My progress card </div>

Manju.

$(“#progresscard”).html(‘Manju progress card’);

<div class =”redfont id=”firstdiv”>1</div>

<div class =”redfont >2</div>

<div class =”redfont >3</div>

$(“. redfont”).style.css = “greenfont;

$(“.#firstdiv”).style.css = “yellowfont;

jQuery events.

|  |  |  |  |
| --- | --- | --- | --- |
| Mouse Event | Keyboard Event | Form Event | Document Event. |
| Click | keypress | Submit | Load |
| dbClick | keydown | Change | resize |
| Mouseenter | keyup | focus | scroll |
| mouseleave |  | blur |  |

Jquery Events.

* Initially to hide a div ( any DOM) we can use CSS like “display:none;” or if we want to show the div ( any DOM) we can use CSS like “display:block”.
* To achieve the same behaviour through jQuery we have two in-built methods hide() and show().
* Ex.
  + <div id=”demo”>Hello All</div>
    - $(“#demo”).hide();
    - $(“#demo”).show();

We can use toggle method to show hidden elements and hide shown elements

$(“#demo”).toggle();

Fade – in.

Get Content.

In Javascript, if you want to get the content from the DIV you can use document.getElementById(‘demo’).value. In the similar way from jQuery we can use $(‘#demo’).val();

In Javascript, if you want to push any the content to the DIV you can use document.getElementById(‘demo’).innerHTML. In the similar way from jQuery we can use $(‘#demo’).html();

JQuery add content dynamically.

In the jQuery we can use the below default methods to add dynamic HTML conent.

**Append**: Insert content at the end of the selected elements.

Ex: $("#demo").append('<h1>After demo</h1>');

**Prepend** : Insert content at the beginning of the selected elements.

Ex: $("#demo").prepend('<h1>After demo</h1>');

jQuery remove content dynamically.

In the jQuery we can use the below methods to accommodate the feature.

Remove() : Removes the selected element and it’s child elements.

Empty() : Removes the child elements from the selected elements.

jQuery Manipulating CSS.

We can use the below in-build methods to manipulate the CSS dynamically.

addClass() : Add one or more classes to the selected element.

removeClass():Remove one or more classes from the selected element.

toggleClass():Toggles between adding / removing classes from the selected element.

Css(): Set dynamic styles to the selected element.

jQuery Traversing Up.

Finding the parent DOMs and start with one selection and move through that selection until you reach the element which you desire.

Below are methods to traversing up the DOM tree.

1. Parent
   1. The DOM element traverse up to the immediate parent.

Ex: $(“span”).parent().css({“color”:”red”});

1. Parents
   1. The DOM element traverse upto the top parent element.

Ex: $(“span”).parents().css({“color”:”red”});

1. ParentUntil.
   1. The DOM element traverse upto the element which you desire.
      1. $(“span”).parentsUntil(“Div”).css({“color”:”red”});

Traversing Bottom

Finding the child DOMs and start with one selection and move through that selection until you reach the element which you desire.

Below are methods to traversing down the DOM tree.

1. Children

* The Children() method can use to find all the Childrens of a DOM
* To apply all the dynamic CSS to all the children of DIV.

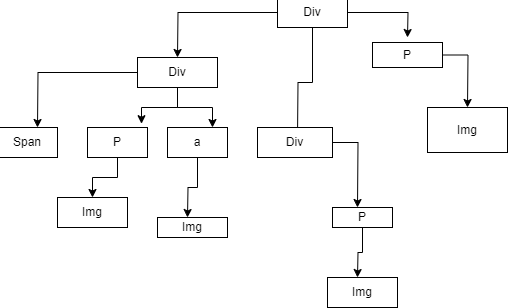
$("div").children().css({"color": "red", "border": "2px solid red"});

* To apply all the dynamic CSS to all the span elements under a DIV.
* $("div").children('span').css({"color": "red", "border": "2px solid red"});
* To apply all the dynamic CSS to specific class elements under a DIV.

$("div").children('p.second').css({"color": "red", "border": "2px solid red"});

1. Find
   1. By using find() method we can get the children until you reach the element which you desire.

* To apply dynamic CSS to all the Spans under a DIV regardless of primary child or not.
* $("div").find('span').css({"color": "red", "border": "2px solid red"});
* To apply dynamic CSS to all the Spans which are having class name as ‘first’ regardless of primary child or not.
* $("div").find('span.first').css({"color": "red", "border": "2px solid red"});
* To apply dynamic CSS to a particular DOM element by using ID.
* $("div").find('#span3').css({"color": "red", "border": "2px solid red"});



To apply dynamic CSS to multi level DOM.

$("div").find('p').find('img').css({"color": "red", "border": "2px solid red"});

Traversing All.

* Sibling : This method is used to apply to find all the siblings of the one DOM element. Below is the example to add dynamic CSS to all siblings.
* $("h2").siblings().css({"color": "red", "border": "2px solid red"});

* Next : To find the next DOM element we can use next method.

Ex:

$(document).ready(function(){

  $("h2").next().css({"color": "red", "border": "2px solid red"});

});

* Nextall : To find all the next DOMs from the target element we can use nextall method.

$(document).ready(function(){

  $("h2").nextAll().css({"color": "red", "border": "2px solid red"});

});

* nextUntil: To find next sibling element between two given DOMS / arguments
* $(document).ready(function(){
* $("h2").nextUntil('span').css({"color": "red", "border": "2px solid red"});
* });
* Prev : To find the before DOM element we can use prev method

$(document).ready(function(){

  $("h2").prev().css({"color": "red", "border": "2px solid red"});

});

* prevAll : To find all the next DOMs from the target element we can use prevall method.
* $(document).ready(function(){
* $("h2").prevAll().css({"color": "red", "border": "2px solid red"});
* });
* prevUntil : To find before sibling element between two given DOMS / arguments