Así se construye la sintaxis de jQuery

Con el código jQuery puedes acceder a los elementos HTML de tu proyecto web. Puedes seleccionar estos elementos por medio de un selector orientado sintácticamente a selectores CSS. A esto le sigue, generalmente, una acción que describe de qué manera se debe modificar el elemento seleccionado. El **símbolo del dólar** (*$*) se antepone al selector y a la acción para distinguir el código como función jQuery. Esto ocurre para evitar confusiones cuando se utilizan diferentes bibliotecas. El símbolo *$()*actúa como abreviatura para la función de jQuery *().*La sintaxis básica tiene la siguiente apariencia:

$(Selector).Acción()

Una línea de código esencial que debe incorporar cualquier descripción de jQuery en su documento HTML es el evento **document ready**. Este código garantiza que todos los comandos que contiene se ejecutarán después de que todos los elementos HTML se hayan cargado.

$(document).ready(function(){

//código jQuery adicional

});

el siguiente código jQuery te permite, por ejemplo, definir que se oculten todos los elementos *<p>*, es decir, todos los bloques de texto, cuando el visitante de la página web haga clic en un botón:

$(document).ready(function(){

$("button").click(function(){

$("p").hide();

});

Posibilidad de responder a las acciones de los usuarios con jQuery

Los usuarios que visitan tu página o tu aplicación web interactúan de muchas maneras con tu proyecto, ya sea por medio de un clic de ratón, de una entrada de teclado, rellenando un formulario o agrandando una ventana. Estas acciones son también conocidas como **eventos DOM**y pueden definirse en jQuery como **elementos desencadenantes de acciones**.

Por ejemplo, puedes utilizar el movimiento del puntero del ratón como elemento desencadenante de una acción, lo que tendría lugar por medio de *mouseenter()* o *mouseleave()*. La primera función reacciona cuando el usuario desplaza el **cursor del ratón sobre el elemento HTML seleccionado**. La segunda, por su parte, entra en acción cuando el cursor del ratón se aleja del elemento. El siguiente código jQuery muestra al usuario una ventana de diálogo informativa cuando este señala con el puntero del ratón un bloque de texto:

<script type="text/javascript">

$(document).ready(function(){

$("p").mouseenter(function(){

alert("Has seleccionado un bloque de texto");

});

});

Otros elementos desencadenantes importantes son:

|  |  |
| --- | --- |
| **Elementos desencadenantes** | **Descripción** |
| focus() | Se acciona al centrarse en un elemento (por clic de ratón o etiqueta) |
| blur() | Se acciona al dejar de centrarse en un elemento |
| keydown() | Se acciona cuando se mantiene una tecla pulsada |
| keyup() | Se acciona cuando se suelta una tecla |
| change() | Se acciona cuando tiene lugar una entrada de teclado o cuando se hace una selección |
| scroll() | Se acciona cuando uno se desplaza por el elemento seleccionado |
| select() | Se acciona cuando se marca un texto en el campo del formulario |

Con ayuda de *on()*podrás, además, añadir más de un elemento desencadenante a un elemento seleccionado y **combinar**, así, **diferentes acciones**de manera sencilla. Con el siguiente código jQuery puedes hacer, por ejemplo, que el color de fondo de los elementos *<p>*varíe tanto cuando los usuarios lo seleccionen (*blue*), como cuando estos vuelvan a elegirlo (*red*) o a hacer clic en él (*yellow*):

$("p").on({

mouseenter: function(){

$(this).css("background-color", "blue");

},

mouseleave: function(){

$(this).css("background-color", "red");

},

click: function(){

$(this).css("background-color", "yellow");

}

});

[jQuery Cheat Sheet](https://htmlcheatsheet.com/) jQuery Cheat Sheet

<https://htmlcheatsheet.com/jquery/>

**Events**

$(".demo").**click**(function(){

$(*this*).hide(*200*);

});

**Mouse**

scroll, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, mouseenter, mouseleave, load, resize, scroll, unload, error

**Keyboard**

keydown, keypress, keyup

**Form**

submit, change, focus, blur

**DOM Element**

blur, focus, focusin, focusout, change, select, submit

**Browser**

load, resize, scroll, unload, error

**.bind()**

$(**document**).ready(function() { // attach a handler to an event for the elements

$("#demo").bind('blur', function(e) {

//dom event fired

});

});

**Basics**

**Include**

[Download](https://code.jquery.com/)

<**script** *src*="https://code.jquery.com/jquery-latest.min.js" *type*="text/javascript"></**script**>

<**script** *src*="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></**script**>

**Syntax**

$(**document**).ready(function(){

$(".demo").**click**(function(){

$(*this*).hide(*200*);

});

});

$(function(){

// Short Document Ready

});

**Each**

$(".demo").each(function() { // parse each .demo element

**document**.**write**($(*this*).**text**() + "\n"); // output their text

});

**Trigger**

$("a#mylink").trigger("click"); // triggers event on an element

**noConflict**

var jq = $.noConflict(); // avoid conflict with other frameworks also using the dollar sign

jq(**document**).ready(function(){

jq("#demo").**text**("Hello World!");

});

$("\*") // all elements

$("p.demo") // <p> elements with class="demo"

$("p:first") // the first <p> element

$("p span") // span, descendant of p

$("p > span") // span, direct child of p

$("p + span") // span immediately proceeded by a p

$("p ~ span") // strong element proceeded by p

$("ul li:first") // the first <li> element of the first <ul>

$("ul li:first-child") // the first <li> element of every <ul>

$("ul li:nth-child(3)") // third child

$("[href]") // any element with an href attribute

$("a[target='\_blank']") // <a> elements with a target "\_blank" attribute

$("a[target!='\_blank']") // <a> elements with a target attribute value other than "\_blank"

$(":input") // all form elements

$(":button") // <button> and <input> elements of type="button"

$("tr:even") // even <tr> elements

$("tr:odd") // odd <tr> elements

$("span:parent") // element which has child element

$("span:contains('demo')") // element conaining the specified text

**Actions**

**$(selector).action()**

$(*this*).hide() // the current element

$("div").hide() // all <div> elements

$(".demo").hide() // all elements with class="demo"

$("#demo").hide() // the element with id="demo"

**DOM Manipulation**

**Content**

$("#demo").**text**(); // returns text content

$("#demo").html(); // returns content, including HTML markup

$("#demo").val(); // returns field value

$("#demo").html('Hey <em>yo</em>'); // sets HTML content

**Attributes**

$("#link").attr("href"); // get an attribute

$("#link").attr("href",'https://html6.com'); // set attribute

$("#link").attr({

"href" : "https://html6.com", // setting multiple attributes

"title" : "HTML Editor"

});

$("#link").attr("href", function(i, origValue){

**return** origValue + "/help"; // callback function gets and changes the attribute

});

**Add**

$(".demo").prepend("Yo!"); // adds content at the beginning in the selected elements

$(".demo").append("<em>Hey!</em>"); // adds content at the end in the selected elements

$(".demo").before("Cheers"); // adds content before the selected elements

$(".demo").after("<em>Peace</em>"); // adds content after the selected elements

**Remove**

$("#demo").**remove**(); // removes the selected element

$("#demo").empty(); // removes children

$("div").**remove**(".cl1, .cl2"); // removes divs with the listed classes

**Classes**

$("#demo").addClass("big red"); // add class

$("h1, p").removeClass("red"); // remove class

$("#demo").toggleClass("big"); // toggle between adding and removing

**CSS**

$("#demo").css("background-color"); // returns CSS value

$("#demo").css("color", "blue"); // sets CSS rule

$("#demo").css({"color": "blue", "font-size": "20px"}); // sets multiple CSS properties

**Dimensions**

width, height, innerWidth, innerHeight, outerWidth, outerHeight  
***inner****- includes padding****outer****- includes padding and border*

**Traversing**

$("#demo").**parent**(); // accessing direct parent

$("span").**parent**().hide(); // changing parent color

$("#demo").parents(); // all ancestors of the element

$("#demo").parentsUntil("#demo2"); // all ancestors between two - demo is inside demo2

$("#demo").children(); // all direct children

$("#demo").children(".first"); // all direct children having a specified class

$("#demo").**find**("span"); // all span elements inside #demo

$("#demo").**find**("\*"); // all descendants

$("#demo").siblings("span"); // span siblings of #demo

$("#demo").**next**(); // the next sibling

$("p").nextAll(); // all next siblings

$("#demo").nextUntil("#demo2"); // siblings between two arguments

$("#demo").prev(); // the previous sibling

$("p").prevAll(); // all previous siblings

$("#demo").prevUntil("#demo2"); // previous siblings between two arguments

**Filtering**

$("span strong").first(); // first strong in first span

$("span strong").last(); // last strong in last span

$("div").eq(*9*); // element with a specific index

$("div").filter(".big"); // all div elements with .big class

$("div").not(".big"); // opposite of filter

**Effects**

**Hide / Show**

$("#demo").hide(); // sets to display: none

$("#demo").show(*200*); // shows hidden elemnt with animation (speed)

$("#demo").toggle(); // toggle between show and hide

$( "#element" ).hide( "slow", function() { // hide with callback function

*console***.log**( "Animation complete." );

});

**Fade**

fadeIn, fadeOut, fadeToggle, fadeTo

$("#demo").fadeIn(); // fade in a hidden element

$("#demo").fadeOut(*300*); // fade out

$("#demo").fadeToggle("slow"); // toggle between fadeIn and hadeOut

$("#demo").fadeTo("slow", *0.25*); // fades to 0.25 opacity

**Slide**

slideDown, slideUp, slideToggle

$("#demo").slideDown();

$("#demo").slideUp("slow");

$("#demo").slideToggle();

**Animate**

$(selector).animate({params},speed,callback);

$("div").animate({

opacity: '0.5',

left: '200px',

height: '200px'

});

**stop() Method**

$("#demo").**stop**();

$('#demo').mouseleave(function(**event**) { // hover end

$('.tab').**stop**().animate({ // stop() method

opacity : '0.5',

marginTop: '10px'

}, *500*, function() { // animation complete

$('#demo').removeClass('hovered'); // callback function

});

});

$('#demo').mouseover(function(**event**) { // hover begin

$('.tab').**stop**().animate({ // stop() method

opacity : '1',

marginTop: '0px'

}, *300*, function() { // animation complete

$('#demo').addClass('hovered'); // callback function

});

});

**Chaining**

$("#demo").css("backgroundColor", "green").slideUp(*500*).

**Ajax**

$("#demo").**load**("file.txt h1.main"); // returns the h1 tag in the text file

$("#demo").**load**("file.txt", function(responseTxt, statusTxt, xhr){ // callback function

**if**(statusTxt == "success") {

**document**.**write**("Content loaded successfully!");

} **else** {

**document**.**write**("Error: " + xhr.**status** + ": " + xhr.**statusText**);

}

});

**$.get()**

$.get("demo.asp", function(data, status){ //$.get(URL,callback);

**document**.**write**("Data: " + data + "\nStatus: " + status);

});

**$.post()**

$.post("demo.asp", // send HTTP POST request to a page and get the answer

{

name: "John", // send data

age: *30*

},

function(data, status){ //retreive response

*console***.log**("Data: " + data + "\nStatus: " + status);

});