

Instituto Politecnico Nacional

Escuela Superior de Cómputo

Tecnologías para la Web

Alejandro Hernández Hernández

PRÁCTICA 7

Introducción

Javascript es un lenguaje de programación en web, utilizado en el lado del cliente, implementado en navegadores web permitiendo vistas dinámicas y mejoras en la interfaz de usuario.

Es un lenguaje débilmente tipado y definido como orientado a objetos.

El implementar o agregar archivos *Javascript*, permite modificar elementos de la página html, esto es, agregar, quitar, ocultar, mostrar elementos en una página web **sin necesidad de recargar la página**, esto define el dinamismo en el lenguaje.

Jquery

Jquery es un framework de javascript, cuyo propósito es simplificar el uso y manipulación del árbol **DOM** de un documento html, manejar eventos y para el uso y desarrollo de animaciones.

Desarrollo

Como ya tenemos definido el documento xhtml, esto es, la pagina principal, y el archivo `.js` para la dinamica de la pagina, se modificará el archivo, incluyendo el *jquery* e implementandolo para conocer su funcionamiento.

Lo primero, es descargar el archivo de jquery, de la pagina:

<http://jquery.com/download/>, personalmente, descargue el **jQuery 3.2.1 slim built**

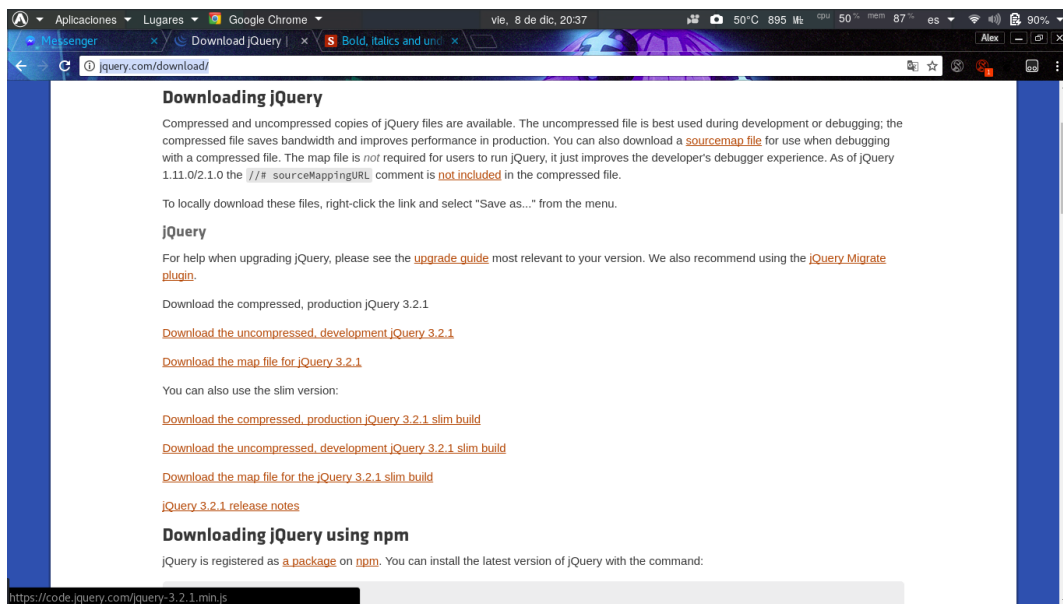


Figura 1: Descarga jQuery

una vez descargado, se mueve a la carpeta donde tenemos nuestra calculadora

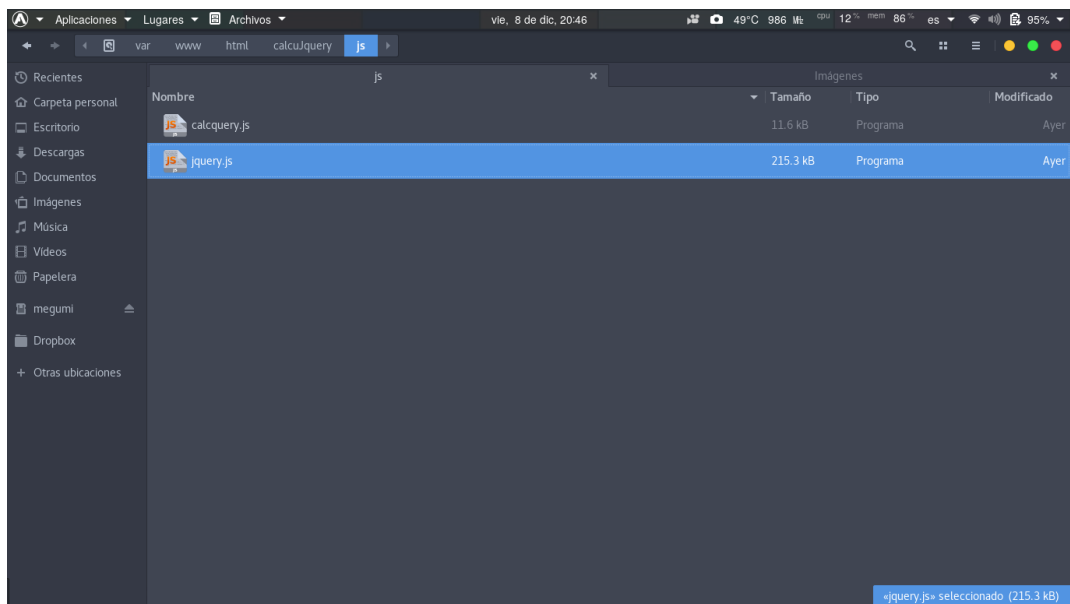
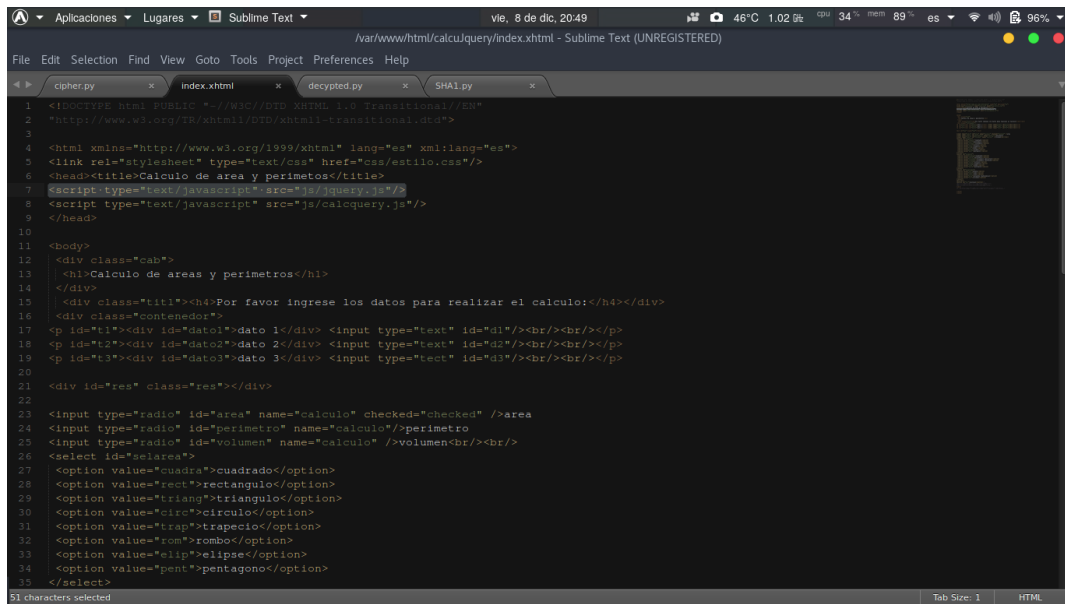


Figura 2: archivo jquery.js

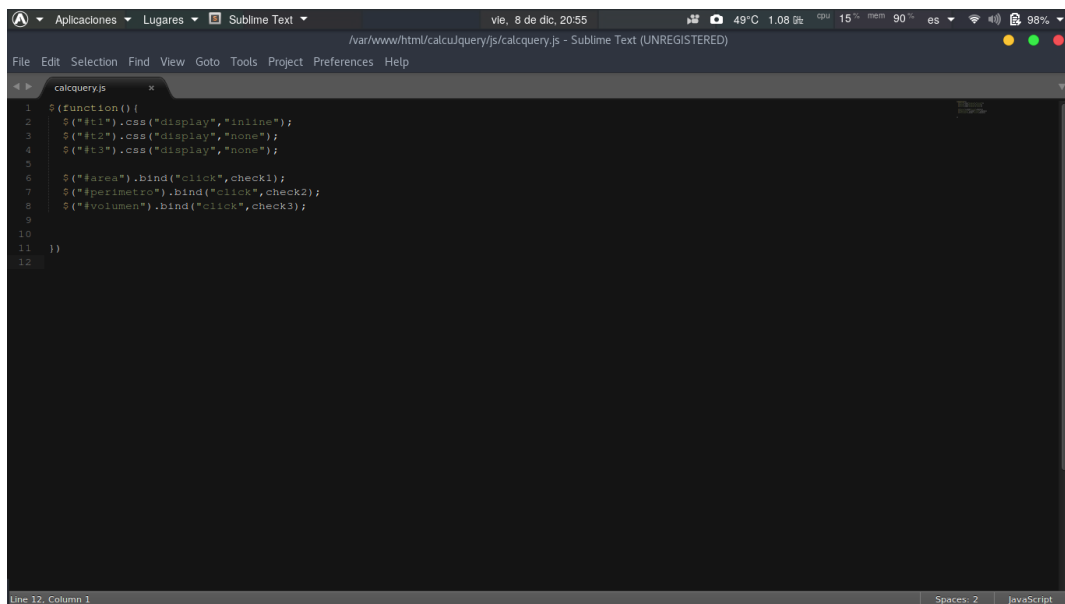
se incluye en la cabecera como un archivo javascript cualquiera; para fines prácticos, cambié el nombre a *jquery.js*



```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3
4 <html xmlns="http://www.w3.org/1999/xhtml" lang="es" xml:lang="es">
5 <link rel="stylesheet" type="text/css" href="css/estilo.css"/>
6 <head><title>Calculo de area y perimetros</title>
7 <script type="text/javascript" src="js/jquery.js"/>
8 <script type="text/javascript" src="js/calquery.js"/>
9 </head>
10
11 <body>
12 <div class="cab">
13 |<h1>Calculo de areas y perimetros</h1>
14 </div>
15 |<div class="titl"><h4>Por favor ingrese los datos para realizar el calculo:</h4></div>
16 <div class="contenedor">
17 <p id="t1"><div id="dato1">dato 1</div> <input type="text" id="d1"/><br/><br/></p>
18 <p id="t2"><div id="dato2">dato 2</div> <input type="text" id="d2"/><br/><br/></p>
19 <p id="t3"><div id="dato3">dato 3</div> <input type="text" id="d3"/><br/><br/></p>
20
21 <div id="res" class="res"></div>
22
23 <input type="radio" id="area" name="calculo" checked="checked" />area
24 <input type="radio" id="perimetro" name="calculo"/>perimetro
25 <input type="radio" id="volumen" name="calculo" />volumen<br/><br/>
26 <select id="selarea">
27 <option value="cuadra">cuadrado</option>
28 <option value="rect">rectangulo</option>
29 <option value="triang">triangulo</option>
30 <option value="circ">circulo</option>
31 <option value="trap">trapecio</option>
32 <option value="rom">rombo</option>
33 <option value="elip">elipse</option>
34 <option value="pent">pentagono</option>
35 </select>
```

Figura 3: inclusion jquery.js

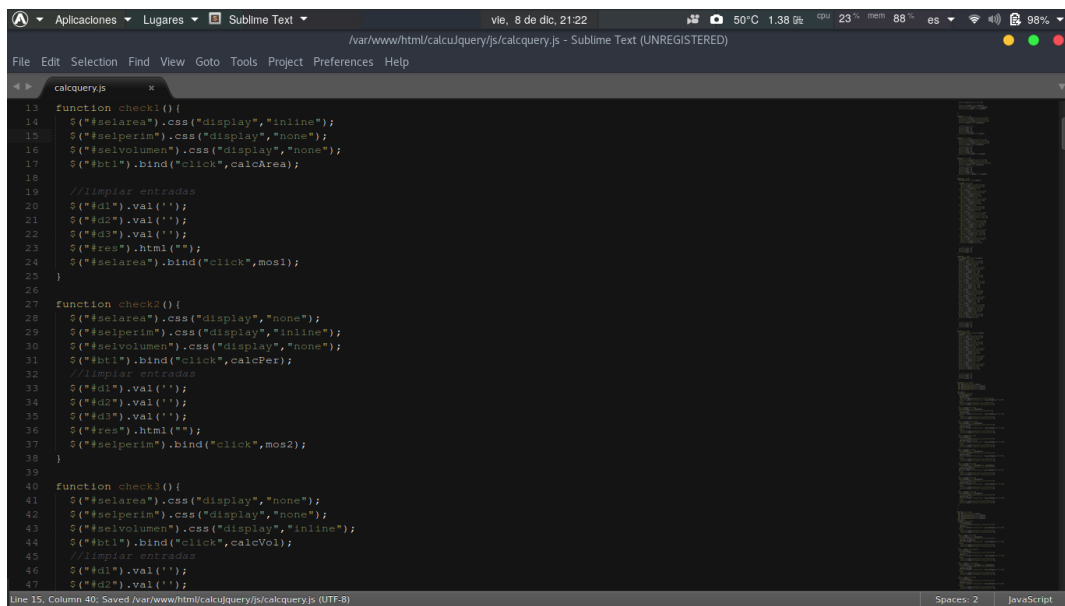
Luego, en el archivo *calquery.js*, agregamos la funcion ‘principal’ que se cargará al mismo tiempo de cargar la pagina



```
1 $(function(){
2   $('#t1').css("display","inline");
3   $('#t2').css("display","none");
4   $('#t3').css("display","none");
5
6   $('#area').bind("click",check1);
7   $('#perimetro').bind("click",check2);
8   $('#volumen').bind("click",check3);
9
10
11 })
12
```

Figura 4: funcion principal

Luego, sustituimos los llamados o usos de *document* en el archivo original de javascript, por `$(#identificador)`, y dependiendo de la accion, le seguirá, para estilo: `$(#identificador).css('propiedad','valor');`; para texto: `$(#identificador).html("texto a mostrar");`; para valores en las cajas de entrada: `$(#identificador).val();`



```
13 function check1(){
14     $("#selarea").css("display","inline");
15     $("#selperim").css("display","none");
16     $("#selvolumen").css("display","none");
17     $("#bt1").bind("click",calcArea);
18 }
19 //limpiar entradas
20 $("#d1").val("");
21 $("#d2").val("");
22 $("#d3").val("");
23 $("#res").html("");
24     $("#selarea").bind("click",mos1);
25 }
26
27 function check2(){
28     $("#selarea").css("display","none");
29     $("#selperim").css("display","inline");
30     $("#selvolumen").css("display","none");
31     $("#bt1").bind("click",calcPer);
32 //limpiar entradas
33     $("#d1").val("");
34     $("#d2").val("");
35     $("#d3").val("");
36     $("#res").html("");
37     $("#selperim").bind("click",mos2);
38 }
39
40 function check3(){
41     $("#selarea").css("display","none");
42     $("#selperim").css("display","none");
43     $("#selvolumen").css("display","inline");
44     $("#bt1").bind("click",calcVol);
45 //limpiar entradas
46     $("#d1").val("");
47     $("#d2").val("");
```

Figura 5: modificación

y al comprobar el funcionamiento de la pagina, era exactamente el mismo que con el anterior javascript.

Conclusión

El uso de jquery, personalmente, me facilitó el entendimiento de mi anterior código, así como reducción en el tamaño de las líneas, es decir, el código quedó mejor, y se ‘optimizó’ el uso de variables en jquery que en javascript, además de que no me bombardeé de tanto `document.getElementById` para cambiar estilos de las etiquetas y elementos en el html.

En resumen, jquery me permitió manipular el árbol DOM mejor que javascript solo, mandando a llamar solamente el elemento a modificar con el nombre de su identificador.