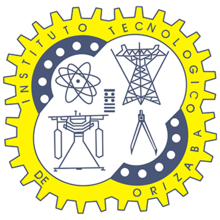
****

**TECNOLÓGICO NACIONAL DE MÉXICO**

**INSTITUTO TECNOLÓGICO DE ORIZABA**

**Programación Web**

**Ing. Sistemas Computacionales**

**Tarea:**

*Cuadro Comparativo*

**Alumno:**

*García Durán Yordi*

**Docente:**

**Fecha de entrega**

04/04/2020

|  |  |  |
| --- | --- | --- |
| **Cuadro Comparativo** | | |
| **Criterios** | **JQuery** | **AngularJS** |
| **Sintaxis** | La sintaxis de jQuery se basa en el objeto jQuery o $(). Este objeto tiene sus métodos, por lo que la forma más común sería la siguiente:  $("selector").metodo(argumentos);  En esta sintaxis básica observamos las siguientes partes:   * El objeto jQuery al principio $(), indicado por el signo "dolar" seguido de un paréntesis. * Al objeto jQuery se le pasa como argumento (dentro del paréntesis) un "selector", que buscará en la página uno o varios elementos para trabajar con ellos. * le sigue un método del objeto jQuery. para ello escribimos primero un punto, después el nombre del método, y luego entre paréntersis los argumentos. * los argumentos (dentro del paréntesis) que se le pasan al método pueden ser muy variados, desde algunos métodos que no requieren argumentos a otros en los que podemos poner funciones anónimas que incluyan otros métodos. * La misma estructura "document ready" se basa en este esquema. En el argumento del método "ready" ponemos una función anónima, en la cual escribimos el resto del código, volviendo a repetir este esquema. | La sintaxis de AngularJS se base en lo siguiente.   * Incluir Angular JS:   Agregando el siguiente código en el head de nuestro HTML.  <script src = "<https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js>"></script>   * Asociar nuestra aplicación:   Debemos indicar que parte del HTML está asociada a nuestra aplicación en Angular, para ello colocamos **ng-app = «miApp»**, con el nombre del módulo que luego procederemos a crear.   * Parte de la Vista:   <div ng-controller = "HomeController" >  <h2>Bienvenido al tutorial de {{helloTo.title}} en Guiadev!</h2>  </div>   * Controlador:   La creación del controlador es la siguiente  <script>  angular.module("miApp", [])    .controller("HomeController", function($scope) {  $scope.helloTo = {};  $scope.helloTo.title = "Angular JS";  });  </script>  En el código podemos ver que se registra una nueva función llamada **«HomeController»** en el módulo **«miApp»**.  Luego, en el scope, asignamos el Objeto helloTo, que luego tendrá el campo title, con el valor **«Angular JS»**.  Es decir, que como hemos mencionado anteriormente, si en nuestra vista deseamos mostrar **«Angualar JS»**, debemos llamar al campo title, del modelo helloTo. |
| **Funcionamiento** | El funcionamiento de jQuery utiliza la programación orientada a objetos de javascript para crear nuevos objetos, los cuales tienen a su vez propiedades y métodos que son los que crean una nueva manera de programar.  El principal objeto utilizado es el objeto jQuery (), el cual podemos escribirlo también de manera abreviada: $().  Este objeto actúa directamente sobre los elementos de la página (elementos del DOM), seleccionándolos y modificándolos con sus métodos y propiedades.  Dentro del paréntesis, el objeto admite varios parámetros, el primero de ellos indicará el lugar de la página o elemento que queremos modificar, por ejemplo $("h1"), nos indicará que queremos modificar todos los elementos de título 1 (con html <h1> ...</h1>).  a partir de ahí le añadimos al código las propiedades o métodos que indiquen lo que queremos hacer con este elemento o elementos, o también podemos indicárselo mediante una función. | El funcionamiento de AngularJS está pensado para aplicaciones de una sola página por lo que el cliente no tendrá que estar cargando las páginas solicitadas como una página web tradicional, sino que estás se cargaran en forma de templetes por detrás para mejorar la experiencia de usuario.  angularjs1 |
| **Aplicación** | La aplicación de jQuery es una biblioteca de JavaScript. Dicho de otro modo, es un simple fichero con extensión .js que contiene objetos y funciones escritas en JavaScript. Este código pone a tu disposición un montón de funciones prefabricadas que podemos utilizar en nuestros proyectos. | La aplicación de AngularJS es logra un buen equilibrio entre estos dos marcos al brindarle una estructura que fomenta la separación de preocupaciones, con construcciones para organizar su código en controladores, servicios, constantes, directivas y filtros, al tiempo que mantiene la flexibilidad y extensibilidad de la implementación. |
| **Similitudes** | jQuery   * Abstracts the DOM * Unit Test Runner * Deferred Promises * Cross-Module Communication * Animation Support * AJAX/JSONP | AngularJS   * Abstracts the DOM * Unit Test Runner * Deferred Promises * Cross-Module Communication * Animation Support * AJAX/JSONP |
| **Diferencias** | jQuery cuenta con muchas características. La API de jQuery funciona muy bien en la mayoría de los navegadores. Si quieres manipular y modificar documentos HTML, jQuery es perfecto para ello. La animación, la gestión de eventos, etc.… se benefician muy mucho de jQuery. Ajax también sale ganando con la presencia de jQuery, ya que hace que su implementación sea coser y cantar. Si deseas manipular el contenido de una página web, puedes confiar fácilmente en jQuery.   * jQuery hace que sea muy sencillo utilizar JavaScript. * Es rápido y ligero. * Permite tener estilos atractivos en la interfaz del usuario. * Se te permite llamar a funcionalidades como métodos siempre que lo desee. * Simplifica mucho todo lo que suele ser complicado de implementar con JavaScript, como por ejemplo las llamadas Ajax. * Las manipulaciones DOM son mucho más fáciles con la ayuda de jQuery. * jQuery cuenta con grandes características, incluyendo la manipulación del CSS. * Sin alterar la estructura de la página web, puedes implementar muchísimas cosas con la ayuda de jQuery. | Es un framework basado en JavaScript. Puedes añadir fácilmente el framework a los distintos documentos HTML. Las directivas enriquecen muy mucho los atributos de HTML. Con la ayuda de las Expresiones, puedes incluso detectar una fecha en HTML. O en otras palabras, permite que HTML tenga nuevos atributos totalmente emocionantes. Se puede decir que Angular JS funciona como una capa sofisticada. Puedes agregar fácilmente conceptos de MVC a tus proyectos con la ayuda de AngularJS.   * Proporciona muchas características interesantes, como los data-bindings cuando se trata de aplicaciones web. * Es un framework de código abierto que está basado en MVC, lo cual ya hemos explicado. * Cubre la estructura de jQuery. AngularJS puede utilizar jQuery para sus propias manipulaciones DOM. Esto hace que sea mucho más emocionante desarrollar nuevas aplicaciones. * AngularJS es un framework muy robusto. Como resultado, se ha vuelto mucho más flexible para crear aplicaciones web. * AngularJS es como una caja de herramientas, ya que es una mezcla de diferentes herramientas. Muchos de los problemas de programación se pueden resolver de manera efectiva con la ayuda de dichas herramientas. * El núcleo de AngularJS es Javascript, ya que está implementado en dicho lenguaje. Como resultado, puedes incorporarlo fácilmente en un sitio web con una etiqueta script de las de siempre. |
| **Ejemplos1** | **Mostrar y Ocultar**  //Ocultar bloque  $("#c1").click(function() {  $("#cont1").hide(1500);  });  //Mostrar bloque  $("#c2").click(function() {  $("#cont1").show("slow");  });  //Ocultar / Mostrar bloque  $("#c1b").click(function() {  $("#cont1").toggle(1500);  },function(){  $("#cont1").toggle(1500);  }); | **IIFE**   * Es una buena práctica empaquetar los componentes en funciones que se invocan inmediatamente. * De ese modo, evitamos que las variables y funciones estén más tiempo de lo esperado en el global scope.   (function() {  'use strict';  angular  .module('app')  .factory('logger', logger);  function logger() { ... }  })(); |
| **Ejemplos2** | **Ocultar con FadeOut y Mostrar con FadeIn**  //Ocultar con FadeOut  $("#c4").click(function() {  $("#cont1").fadeToggle(3000);  });  //Mostrar con FadeIn  $("#c5").click(function() {  $("#cont1").fadeIn(3000);  });  //Mover  $("#c6").click(function(){    $("#cont1").animate({opacity: "0.1", left: "+=400",fontSize:'1em',width: "200"}, 1200)  .animate({opacity: "0.4", top: "+=160", height: "20", width: "80",fontSize:'0.5em'}, "slow")  .animate({opacity: "1", left: "0", width: "260"}, "slow")  .animate({top: "0",width: "260",fontSize:'1.2em'}, "fast")  .slideUp()  .slideDown(1800)  return false; }); | **Módulos**   * Declararemos los módulos sin asignarlos a una variable y lo haremos utilizando la sintaxis de los **setters**. * Sintaxis setter: angular.module(‘app’, [depen.]) para declarar un módulo y especificar sus dependencias. * Sintaxis getter: angular.module(‘app’) para recuperar un módulo.   angular  .module('app', [  'ngAnimate',  'ui.router',  'app.shared'  ]) |