Angular Basics

Nota: Este ejemplo no está directamente resuelto. Simplemente hay un documento HTML des del cual trabajaremos. La intención es realizar una aplicación AngularJS paso a paso partiendo del HTML que os damos. Es decir, que vamos a convertir una página web estática, en una aplicación AngularJS 100% funcional.

#Introducción

¿Qué es AngularJS?

AngularJS es un proyecto de código abierto, realizado en Javascript que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un framework para el desarrollo, en esta caso sobre el lenguaje Javascript con programación del lado del cliente.

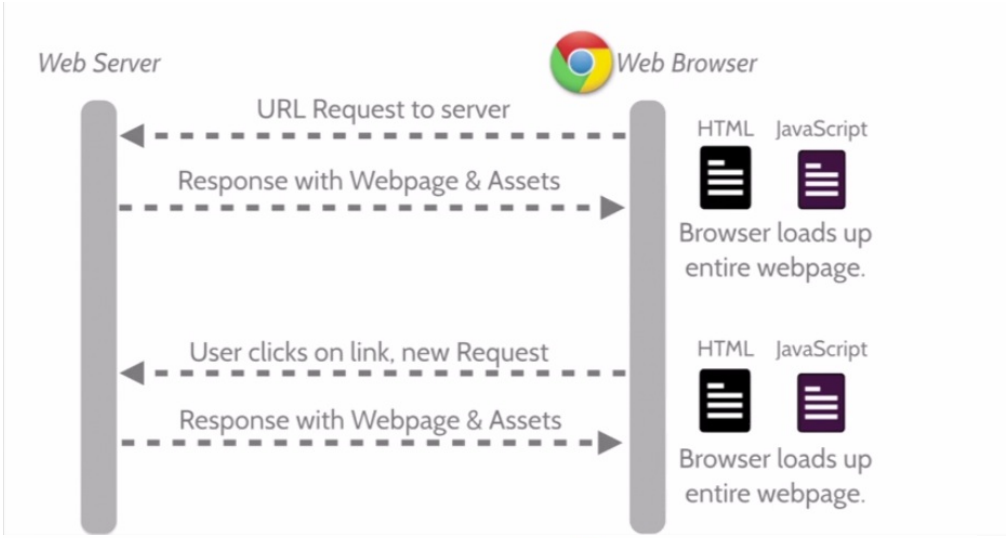
Puedes encontrar el proyecto de AngularJS en su propio sitio web: AngularJS, Superheroic JavaScript MVW Framework. Al ser un proyecto de código abierto cualquier persona con un poco de curiosidad echar un vistazo con profundidad y ver cómo se ha escrito, incluso admiten colaboraciones de desarrolladores que quiera aportar cosas.

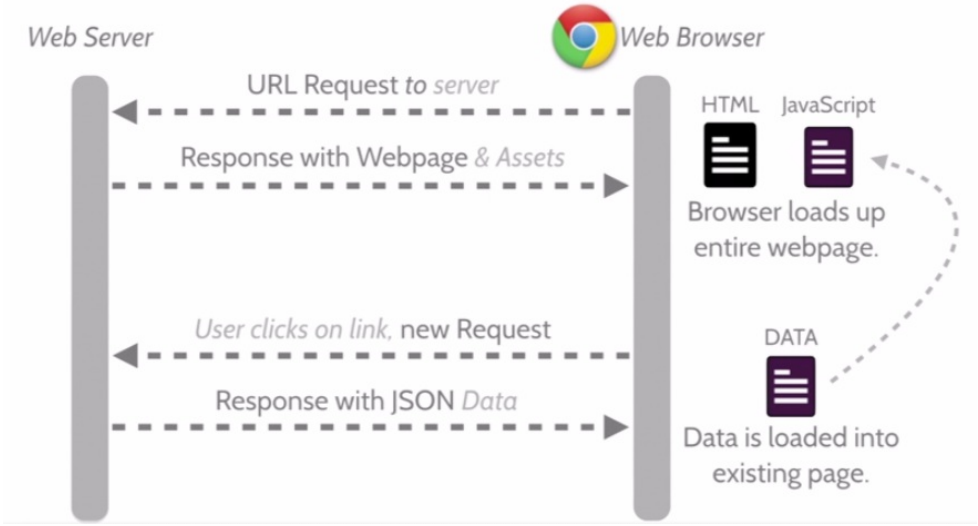
Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticos o dinámicos.

En pocas palabras:
`Es un framework del lado del cliente, escrito en Javascript con la finalidad de añadir interactividad al HTML`

Método tradicional de comunicación con el servidor vs método que usa AngularJS





En la imagen de la izquierda podemos ver el método tradicional de comunicación con el servidor. Básicamente cuando tu entrabas en una pagina web el servidor te devolvia todo el HTML de la página y cuando hacias click en otro enlace, el servidor te devolvia otra vez todo un HTML.

En la imagen de la derecha vemos el método que usa Angular. Cuando haces una petición al servidor te devuelve todo un HTML, igual que con el método tradicional. Pero cuando haces click en un enlace, en vez de devolverte toda la página, simplemente de devuelve JSON con la información que se debe modificar y el navegador muestra el trozo que se ha cambiado. El resto de la página sigue intacta. De esta manera ahorramos datos a recibir y tiempo de carga.

Como incorporar AngularJS en mi HTML

Simplemente debes poner el CDN que nos proporciona la web de Angular con toda la libreria de Angular. Así de simple.

```
  <html>
    <head>
      <title>My Angular App</title>
      <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.js">
      </script>
    </head>
    <body>
    </body>
  </html>
```

Tambien podemos descarganos el .js de la web, si lo queremos tener en local.

```
<script src="ruta_del_fichero/angular.js">
```

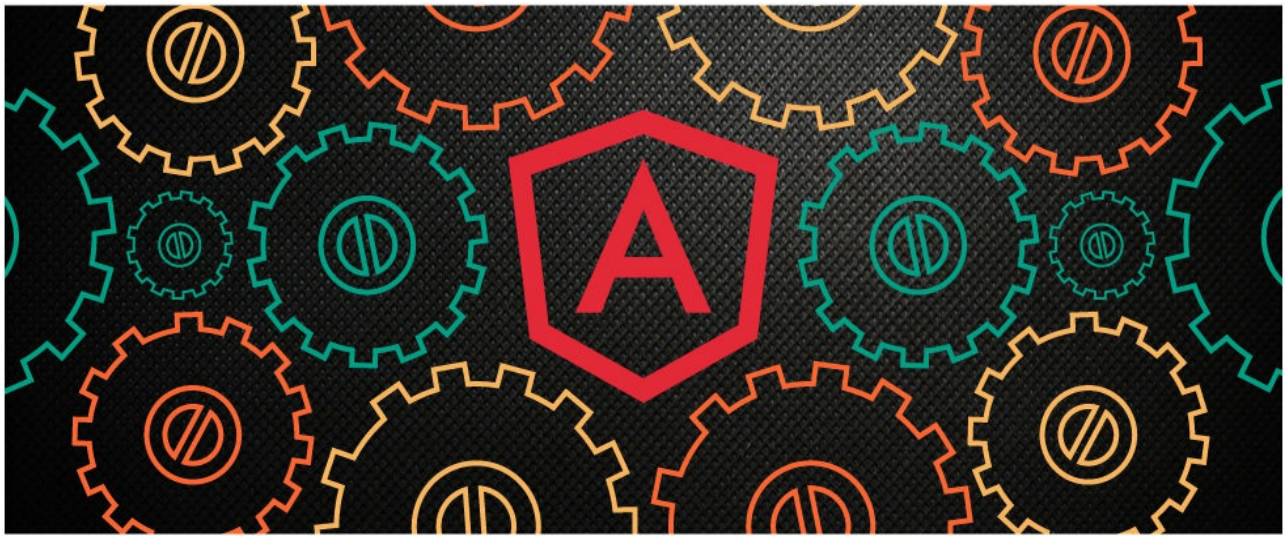
Hello world!

Vamos a realizar nuestra primera aplicación con AngularJS. En primer lugar vemos que la etiqueta html contiene la directiva `ng-app`.
A continuación veremos que són las directivas y otros elementos de Angular. De momento nos debemos quedar con la idea de que `ng-app`sirve para inicializar una aplicación de AngularJS. Es decir, que ya podremos usar todas las funciones que nos proporciona AngularJS.

Lo siguiente que vemos son estos elementos `{{ }}`. Todo lo que pongamos entre estos dos elementos, Angular lo interpretará. En este caso hemos escrito `(2 + 3)*2` para que veáis como Angular realiza la operación lógica.

```
  <html ng-app>
  <head> ... </head>
  <body>
      <h1 align="center">Hello world! {{ (2 + 3)*2 }} = 10</h1>
  </body>
  </html>
```

Modulos, controladores, $scope, directivas y filtros



Modulos

Los sirven básicamente para organizar tu código javascript. La idea es muy simple, tienes diferentes módulos para tener ordenado el código y así que sea más fácil mantenerlo y debugarlo.

Para declarar un módulo lo hacemos de la siguiente manera.

```
//Código JS:
angular.module('Modulo', [])

//Codigo HTML:
<html ng-app="Modulo">
```

Fijaros que usamos el ng-app que hemos visto antes. En él indicamos que modulo queremos usar.

Controladores

Los módulos contienen controladores. Los controladores sirven para unir la vista con el modelo.

Para declarar un controlador lo hacemos de la siguiente manera.

```
//Código JS:
var App = angular.module('nuevaApp', []);

App.controller('nuevaAppCtrl', ['$scope', function($scope) {

    [Contenido del controlador]

}]);

//Codigo HTML:
<html ng-app="nuevaApp" ng-controller="nuevaAppCtrl">
```

$scope

El $scope nos permite mantener una relación entre el html y el javascript. Es la via que tienen para compartir información la vista y el controlador. El $scope se declara en el Javascript siguiendo el formato `$scope.miVariable`. En el html se usan las `curly braces` `{{miVariable}}` para mostrar la información de una variable declarada en el $scope.

```
```
//Código JS:
var App = angular.module('nuevaApp', []);

App.controller('nuevaAppCtrl', ['$scope', function($scope) {

    $scope.miVariable = 'Hola mundo!';

}]);

//Codigo HTML:
<body ng-app="nuevaApp" ng-controller="nuevaAppCtrl">
    {{miVariable}}
</body>
```
```

Para modificar el contenido del $scope desdel html se una una directiva llamada `<ng-model>` que a continuación la veréis.

Directivas

Las directivas son componentes de angular que se insertan junto a las etiquetas html para añadir funcionalidades a HTML. Es como si le estuvieramos dando superpoderes al HTML para que realize nuevas funciones.

Cada directiva tiene un uso y tu puedes programar tus propias directivas para añadir funcionalidades o fragmentos de HTML. A continuación veremos algunas directivas básicas que AngularJS nos aporta.

- ng-app: sirve para inicializar una aplicación de angular y seleccionar el modulo que queremos usar.
- ng-controller: sirve para seleccionar el controlador que queremos usar.

```
  `<html ng-app="nuevaApp" ng-controller="nuevaAppCtrl>`
```

- ng-model: Sirve para relacionar el valor de un elemento html con una propiedad del $scope. Se usa principalmente en los inputs para obtener o modificar el valor de éste. Por ejemplo si inicialmente tenemos `$scope.propietario = 'Joaquin'`, en el input veriamos escrito `Joaquin`. Si nosotros borrásemos el contenido del input y escribieramos `Marcos`, el valor de la varible `$scope.propietario` pasaria a ser 'Marcos'. Poder modificar el valor de una variable tanto en Javascript como en el html, es lo que se conoce como `Data Binding`.

  ```
  `<input type="text"  class="form-control" placeholder="Propietario" ng-model="propietario"/>`

  $scope.propietario = 'Joaquin';
  ```

- ng-click: Cuando se pulsa un elemento html, por ejemplo un botón, realiza las operaciones que indiques.
  `<button type="button" ng-click="añade()" class="btn btn-success"> Añadir </button>`.

  ```
  `$scope.afegeix = function(){
        $scope.armes.push({ model: $scope.model, propietari:
  $scope.propietari,
        cost: $scope.cost});
  }`
  ```

- ng-repeat: Repite el contenido de un array o un fragmento de HTML
  ```
  $scope.armas = [
  {
  modelo: 'AK-47', propietario: 'Mario', coste: 400
  },
  {
  modelo: 'M16', propietario: 'Lucia', coste: 500
  }
  ];
  ```

```
<tbody ng-repeat="arma in armas">
  <tr>
      <td>{{arma.modelo}}</td>
      <td>{{arma.propietario}}</td>
      <td>{{arma.coste}}</td>
  </tr>
</tbody>
```

- ng-hide y ng-show: si `ng-hide=1` oculta el contenido de la etiqueta donde está. Si `ng-hide=0`lo oculta. `ng-show` funciona al reves, si vale 1 muestra el contenido y si vale 0 lo oculta.

```
`<h1 ng-hide=1>Hola mundo!</h1>  //oculta el contenido`
`<h1 ng-hide=0>Hola mundo!</h1>  //muestra el contenido`
`<h1 ng-show=1>Hola mundo!</h1>  //muestra el contenido`
`<h1 ng-show=0>Hola mundo!</h1>  //oculta el contenido`
```

- ng-init: Inicializa una variable del `$scope`. Si ya esta inicializada cambia su valor.

  ```
  `<h1 ng-init="miVariable= 'Hola mundo!'">{{miVariable}}</h1>`
  ```

Filtros

Los filtros son una herramienta que nos proporciona el framework de AngularJS para ahorrarnos trabajo recurrente. Permite que dados unos datos de entrada, mediante unos parámetros obtengamos una salida depurada según nuestras necesidades. Ya sean mostrar los datos de un modo que al usuario le resulten más útiles como formatear una fecha, limitar el número máximo de resultados para no sobrecargar una vista de información o cosas tan mundanas como convertir un texto a minúsculas.

Para que un filtro interactue con una expresion simplemente es necesario colocarlo dentro de una doble llave y separarlo de la expresión a filtrar por una barra vertical de esta forma `{{expresion | filtro }}`.

- Filtro currency:

  Sirve para cuando queremos indicar que un valor es una moneda. Por defecto cuando ponemos el filtro `currency`la moneda prederminada es el `$`. Pero podemos indicar que moneda queremos `currency:"€"`.

  ```
  <p>{{500 | currency:'€'}}</p>
  ```

- Filtro number:

  Con este filtro podemos indicar cuanto decimales queremos que tenga un numero.

  {{500 | currency:'€'| number:2}}

  El resultado seria `€500.00`.

- Filtro date:

  Pone un `date` en el formato que queramos de fecha.

  ```
  {{miFecha | date:'dd/MM/yyyy'}}

  $scope.miFecha = new Date();
  ```

- Filtro filter:
  Sirve para filtrar el contenido de un array en función del nombre que pongamos. Se usa en los ng-repeat para mostrar solo los resultados que contengan el nombre especificado en el filtro.

  `<tbody ng-repeat="alumno in alumnos | filter:Marc">`

  Solo muestra los resultados que contengan la palabra Marc.

- Filtro lowercase y uppercase:

  Las variables salen representadas en minúscula (lowercase) o mayúscula (uppercase).

  ```
  <h1>{{miVariable | uppercase}}:</h1>

  <h1>{{miVariable | uppercase}}:</h1>
  ```

- Filtro limitTo:

  Muestra sólo un número concreto de elementos de un array. Se usa en los ng-repeat para deliminar el número de resultados

```
 //Delimita el resultado a 2
 <tbody ng-repeat="alumno in alumnos | limitTo:2 ">
```

- Filtro orderBy:

  Ordena alfabeticamente un array en funcion de un parámetro.

```
<tbody ng-repeat="alumno in alumnos | orderBy: 'nota'">
```