

Svg + Angular

Fundamentos

- Es una especificación XML soportada por todos los navegadores modernos
- Se puede incrustar en una página HTML y...simplemente funciona

Fundamentos

- Y al igual que HTML, se puede interactuar con sus elementos utilizando CSS
- Es como HTML con su propio conjunto de elementos y atributos para dibujar cosas

Fundamentos

- Las imágenes en svg tienen muchas ventajas
- Son pequeñas en comparación con los
bitmaps y son dinámicas

Fundamentos

- En el desarrollo de aplicaciones web móviles con diseño “responsive”, los SVGs son una solución muy viable para crear imágenes dinámicas

Fundamentos

- Como sabemos, Angularjs permite encapsular lógica de interfaz de usuario compleja en directivas (HTML “personalizado”)
- Esto da módulos limpios y fáciles de mantener

Fundamentos

- Angular (directivas + doble data binding) + SVG son una combinación muy buena para la manipulación compleja de imágenes dinámicas

Al grano

- La idea es incluir de alguna manera el SVG de tal manera que pertenezca a mi scope
- Y tener elementos o propiedades SVG “bindados” en ese scope

Al grano

```
<body ng-controller="MainCtrl">  
  <svg><g ng-repeat="cell in row">  
    <rect width="50" height="50" x="{{cell.node.x}}"   
      y="{{cell.node.y}}"></rect>  
  </g></svg>  
</body>
```

Al grano

```
app.controller('MainCtrl', function($scope) {  
    $scope.row = [  
        { node: { x: 0, y : 0 } },  
        { node: { x: 50, y : 50 } }  
    ];  
});
```

Al grano

- Hay muchas maneras de poder incluir un SVG en nuestro DOM
- Vamos a ver una de ellas, muy apropiada para trabajar desde nuestra directiva

ng-include

- Obtiene, compila e incluye un fragmento de HTML externo

```
<div id="gna-svg"ng-include="\'/images/gna-display.svg\'"></div>
```

Ejercicios

- Añade un módulo **svg**
 - Duplica el módulo Promise
 - Haz los cambios oportunos (index, router, navbar, renombra módulo promise)

Ejercicios

- Objetivo:
 - Utilizar el **display** en SVG de la carpeta **images** para mostrar el **número aleatorio** el **mod** y el **interval**

Ejercicios

- Objetivo:
 - Hay que comprender esa imagen, localizar las variables a “bindar” en el scope
 - Y después...¿qué pasos seguirías?

Ejercicios

- Proposición
 - Nueva directiva gna-display que contiene y manipula el SVG
 - **Es un display**, sin lógica referente al cálculo de números

Ejercicios

- Proposición
 - Tiene “isolated scope” y recibe como parámetros los valores a mostrar

```
scope: { randomNumber: '@number', mod: '@', interval: '@' }
```

Ejercicios

- Proposición: ¿Te atreves?

Ejercicios

- ¿Siguen funcionando los tests?