

## Animacions en AngularJS

### Instal·lació

AngularJS proporciona enllaços d'animació per a directives comunes com ngRepeat, ngSwitch i ngView, així com directives personalitzades a través del servei d'animació \$animate. Aquests enllaços d'animació s'estableixen per activar animacions durant el cicle de vida de diverses directives i, quan s'executin, intentaran realitzar una transició CSS, o una animació amb un fotograma concret amb CSS o una animació de devolució de trucada de JavaScript (depenent de si una animació es col·loca a la directiva indicada). Les animacions es poden col·locar amb CSS Vainilla seguint les convencions de nomenclatura establertes per Angular JS o amb el codi JavaScript, definit com a factory.

Les animacions no estan disponibles a menys que inclogueu el mòdul ngAnimate com a dependència de la vostra aplicació.

Per instal·lar el mòdul d'animacions, podem afegir la següent etiqueta per cridar el CDN oportú:

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-animate.js"></script>
```

O podem descarregar la llibreria y sol·licitarla de forma local.

També tenim que fer referència al modul ngAnimate dins de l'aplicació.

```
<body ng-app="ngAnimate">
```

O si l'aplicació ja te un nom personalitzat, s'ha d'afegir **ngAnimate** com a dependència en el modul de l'aplicació.

## Com funciona?

### Què fa ngAnimate?

El modul ngAnimate afegeix i elimina classes.

El mòdul ngAnimate no anima els vostres elements HTML, però quan ngAnimate observa certs esdeveniments, com ocultar o mostrar un element HTML, l'element obté algunes classes predefinides que es poden utilitzar per fer animacions.

Les directives en AngularJS que afegeixen o eliminen classes són:

- **ng-show**
- **ng-hide**
- **ng-class**
- **ng-view**
- **ng-include**

- `ng-repeat`
- `ng-if`
- `ng-switch`

Les directives `ng-show` i `ng-hide` afegeixen o eliminen un valor de la classe `ng-hide`. Les altres directives afegeixen un valor de classe `ng-enter` quan ingressen al DOM i un atribut `ng-leave` quan són eliminades del DOM. La directiva `ng-repeat` també afegeix un valor de classe `ng-move` quan l'element HTML canvia de posició. A més, durant l'animació, l'element HTML tindrà un conjunt de valors de la classe, que s'eliminarà quan hagi finalitzat l'animació. Exemple: la directiva `ng-hide` poden ser afegides a aquests valors de classe:

- `ng-animate`
- `ng-hide-animate`
- `ng-hide-add` (si l'element s'oculta)
- `ng-hide-remove` (si l'element es mostra)
- `ng-hide-add-active` (si l'element s'oculta)
- `ng-hide-remove-active` (si l'element es mostra)

## Transicions utilitzant CSS3

Les transicions CSS us permeten canviar els valors de propietat CSS sense problemes, d'un valor a un altre, durant una durada determinada.

Veure `Ejercici1.html`.

## Animacions utilitzant CSS3

### Animacions utilitzant CSS3

Les animacions en AngularJS es basen completament en classes CSS. Sempre que tingueu una classe CSS adjunta a un element HTML dins de la vostra aplicació, podeu aplicar-hi animacions. Posem per cas, per exemple, que tenim una plantilla HTML amb un bucle:

```
<div ng-repeat="item in items" class="repeated-item">
  {{ item.id }}
</div>
```

Com podeu veure, la classe `repeated-item` està present a l'element que es repeteix i aquesta classe s'utilitzarà com a referència en el codi d'animació de CSS i / o JavaScript de la nostra aplicació per indicar-li a AngularJS que realitzi una animació.

Aquí mentre `ngRepeat` s'executa, cada vegada que s'afegeix un nou element a la llista, `ngRepeat` afegirà una classe `ng-enter` a l'element que s'està afegint. Quan s'elimini s'aplicarà una classe `ng-leave` i quan es mogui al voltant s'aplicarà una classe `ng-move`.

Mirem el següent codi CSS, podem veure un codi d'animació de transició i un codi keyframe animation, per a cadascun d'aquests esdeveniments que es produeixen quan ngRepeat els desencadena:

```
/*
    Estem utilitzant transicions CSS per a l'hora d'introduir i moure esdeveniments
    i que es desencadenen per l'element que té la classe `repeats-item`
*/
.repeated-item.ng-enter, .repeated-item.ng-move {
    transition: all 0.5s linear;
    opacity: 0;
}

/*
    `.ng-enter-active` i `.ng-move-active` són on es troba la destinació de la transició
    Les propietats s'estableixen perquè l'animació sàpiga què te d'animar*/
.repeated-item.ng-enter.ng-enter-active,
.repeated-item.ng-move.ng-move-active {
    opacity: 1;
}

/*
    Nosaltres estem utilitzant CSS keyframe animations durant l'esdeveniment `leave` I que
    es desencadena per l'element que té la classe `repeats item`
*/
.repeated-item.ng-leave {
    animation: 0.5s my_animation;
}

@keyframes my_animation {
    from { opacity: 1; }
    to   { opacity: 0; }
}
```

El mateix enfocament de l'animació es pot utilitzar el codi JavaScript (per simplicitat, confiem en el framework jQuery per realitzar animacions):

```
myModule.animation('.repeated-item', function() {
  return {
    enter: function(element, done) {
      // Inicialitzem el element opacity
      element.css('opacity', 0);

      // Animem el element opacity
      // (`element.animate()` esta proporcionat per jQuery)
      element.animate({opacity: 1}, done);

      // Opcionalment en `onDone`/`onCancel` utilitzem la funció callback
      // per emprar-la en calsevol operació post-animation
      return function(isCancelled) {
        if (isCancelled) {
          // Aturem la animacio si es cancela
          // (`element.stop()` esta proporcionat per jQuery)
          element.stop();
        }
      };
    },
    leave: function(element, done) {
      // Inicialitzem el element opacity
      element.css('opacity', 1);

      // Animem el element opacity
      // (`element.animate()` esta proporcionat per jQuery)
      element.animate({opacity: 0}, done);

      // Opcionalment en `onDone`/`onCancel` utilitzem la funció callback
      // per emprar-la en calsevol operació post-animation
      return function(isCancelled) {
        if (isCancelled) {
          // Aturem la animacio si es cancela
          // (`element.stop()` esta proporcionat per jQuery)
          element.stop();
        }
      };
    }
  };
});
```

```
    }  
    };  
  },  
  
  // També podem capturar els següents esdeveniments d'animació:  
  move: function(element, done) {},  
  addClass: function(element, className, done) {},  
  removeClass: function(element, className, done) {}  
}  
});
```

Amb aquests noms de classe CSS generats presents en l'element del temps, AngularJS calcula automàticament si s'ha de realitzar una animació de CSS i / o JavaScript. Tingueu en compte que no podeu tenir animacions CSS i JavaScript basades en la mateixa classe CSS.

## class i ngClass per animacions

AngularJS també presta atenció als canvis de classe CSS en els elements mitjançant l'us d'afegir i eliminar enllaços. Això vol dir que si una classe CSS s'afegeix o s'elimina d'un element, llavors es pot executar una animació en el medi, abans que finalitzi l'addició o eliminació de la classe CSS. (Tingueu en compte que AngularJS només serà capaç de capturar canvis de classe si s'utilitza una expressió interpolada o la directiva ng class en l'element).

Veure Ejercici2.html i Ejercici2.css.