ANIMATIONS ET ANGULARJS

SQLI GROUP

LES ANIMATIONS

ANIMATIONS

PRÉSENTATION

+ Basées sur les animations **CSS3**
+ Entièrement **class-based**
+ Support des directives **ng-class**, **class expressions** et **callbacks**
+ Support d'IE 10+
+ Arrivées avec la version **1.2** d'AngularJS

ANIMATIONS

DÉCLARATION DE L'API

+ Inclure le fichier JavaScript **angular-animate.js**

```
<script src="angular/angular-animate.js"></script>
```

+ Dépendre du module **ngAnimate**

```
var myApp = angular.module('myApp', ['ngAnimate']);
```

ANIMATIONS CSS

ENTER, LEAVE, MOVE

+ Toutes les animations sont **class-based** : des classes CSS peuvent être associées à des animations

+ Les animations sont résolues depuis les classes CSS des éléments

+ Des classes CSS spécifiques sont ajoutées à des moments clés
  - Lors de l'**insertion** d'un élément dans le DOM : **`ng-enter`**
  - Lors du **retrait** d'un élément du DOM : **`ng-leave`**
  - Lors du **repositionnement** d'un élément dans le DOM : **`ng-move`**

+ Application
  - ng-enter : ngRepeat, ngInclude, ngIf, ngView, ngSwitch
  - ng-leave : ngRepeat, ngInclude, ngIf, ngView, ngSwitch
  - ng-move : ngRepeat

ANIMATIONS CSS

{CLASS} ET {CLASS}-ACTIVE

+ **ng-enter** est appliquée immédiatement lors de l'ajout d'un nouvel élément au DOM
  - **style au démarrage de l'animation**

+ **ng-enter-active** est appliquée à la suite
  - **style à la fin de l'animation**

+ Les deux classes sont retirées une fois l'animation terminée

+ Principe appliqué également aux événements
  - **leave** : ng-leave et ng-leave-active
  - **move** : ng-move et ng-move-active
  - **classes custom** : {CLASS} et {CLASS}-active

ANIMATIONS CSS

{CLASS} ET {CLASS}-ACTIVE - EXEMPLE

```
.my-special-animation.ng-enter {
  -webkit-transition: 0.5s linear all;
  transition: 0.5s linear all;
  background: red;
}

.my-special-animation.ng-enter.ng-enter-active {
  background: blue;
}
```

ANIMATIONS CSS

ANIMATIONS SUR CLASSES CUSTOM

- L'ajout ou le retrait d'une classe sur un élément peut être animé
- **class-add** : appliqué lors de l'ajout d'une classe sur un élément
  - **class-add** et **class-add-active**
- **class-remove** : appliqué lors du retrait d'une classe sur un élément
  - **class-remove** et **class-remove-active**
- Applicables sur **ngClass** et **class={{}}**

ANIMATIONS CSS

ANIMATIONS SUR CHANGEMENT DE VISIBILITÉ D'UN ÉLÉMENT

+ La classe **ng-hide** est positionnée sur un élément quand il doit être masqué

```
.ng-hide {
  display:none !important;
}
```

+ Attention à prendre en compte le **!important** en cas de surcharge

+ L'animation peut être prise en compte avec les suffixes
  - –add
  - -add-active
  - -remove
  - –remove-active

ANIMATIONS CSS

ANIMATIONS SUR CHANGEMENT DE VISIBILITÉ D'UN ÉLÉMENT

```
.my-elm {
    -webkit-transition: 0.5s linear all;
    transition: 0.5s linear all;
    opacity: 1;
}

.my-elm.ng-hide {
    opacity: 0;
}

.my-elm.ng-hide-add,
.my-elm.ng-hide-remove {
    display: block!important;
}
```

ANIMATIONS PAR KEYFRAMES

KEYFRAMES

+ Méthode d'animation dans laquelle on indique plusieurs positions intermédiaires
  - Ici les positions intermédiaires sont calculées

+ Utilisation de l'annotation **@keyframes**

+ Définition des styles de début **from {}** et de fin **to {}**

```
@keyframes red-to-blue {
    from { background:red; }
    to { background:blue; }
}

@-webkit-keyframes red-to-blue {
    from { background:red; }
    to { background:blue; }
}
```

ANIMATIONS PAR KEYFRAMES

KEYFRAMES

- La « keyframe » est référencée dans les classes associée à une animation
- Permet de se passer des classes **–active**
- Utilisation de l'attribut **animation** pour référencer le « keyframe » et indiquer sa durée

```
.my-special-animation.ng-enter {
    animation: 0.5s red-to-blue;
    -webkit-animation: 0.5s red-to-blue;
}
```

ANIMATIONS ÉCHELONNÉES

STAGGERING

+ Collection d'animations affichées avec un léger délai entre chaque opération = effet **rideau**

+ Classe CSS **ng-EVENT-stagger**

+ Peut utiliser **transition-delay** ou **animation-delay**

+ Utilisation sur **ngRepeat**

```
.my-animation.ng-enter-stagger {
    -webkit-transition-delay: 0.1s;
    transition-delay: 0.1s;

    -webkit-transition-duration: 0s;
    transition-duration: 0s;
}
```

ANIMATIONS DEPUIS JS

DÉCLARATION

+ Fonction **animation()** disponible sur le module pour enregistrer une fonction d'animation associée à une classe CSS

```
myApp.animation('.fade', function() {
  return {
    enter : function(element, done) { },
    leave : function(element, done) { },
    move : function(element, done) { },
    beforeAddClass : function(element, className, done) { },
    addClass : function(element, className, done) { },
    beforeRemoveClass : function(element, className, done) { },
    removeClass : function(element, className, done) { },
  };
});
```

ANIMATIONS DEPUIS JS

DÉCLARATION

- Fonctions disponibles pour l'animation
  - **enter** : déclenchée quand l'élément contenant le style est ajouté au DOM
  - **leave** : déclenchée quand l'élément contenant le style est retiré du DOM
  - **move** : déclenchée quand l'élément contenant le style est repositionné dans le DOM
  - **beforeAddClass** : déclenchée avant que le style ne soit ajouté à un élément
  - **addClass** : déclenchée après que le style ait été ajouté à un élément
  - **beforeRemoveClass** : déclenchée avant que le style ne soit retiré d'un élément
  - **removeClass** : déclenchée après que le style ait été retiré d'un élément

- Fonction **done()** en paramètre appelée quand l'animation est terminée

ANIMATIONS DEPUIS JS

DÉCLARATION – BONNES PRATIQUES

+ Préférer l'implémentation sous la forme du **design pattern Revealing Module**

```
myApp.animation('.scroll', function() {

   var animateUp = function(element, className, done) {  };
   var animateDown = function(element, className, done) { };

   return {
    addClass : animateUp,
    removeClass : animateDown
   }
});
```

ANIMATIONS DEPUIS JS

DÉCLARATION – BONNES PRATIQUES

+ **jQuery** à considérer pour simplifier le code

```
var animateUp = function(element, className, done) {
   element.css({
      position: 'absolute',
      top: 500,
      left: 0,
      display: 'block'
    });

jQuery(element).animate({
  top: 0
}, done);
```

ANIMATIONS DEPUIS JS

SERVICE $ANIMATE

- Service en charge des animations
  - Détection des animations déclarées, des opérations du DOM, des états CSS
  - Exécution des animations enregistrées

- Utilisable « en direct »
  - Pour déclencher une animation : ajouter/supprimer une classe à un élément

```
$animate.addClass(element, 'ng-hide');
$animate.removeClass(element, 'mon_anim');
```

- Possibilité d'intercepter la fin de l'animation avec une fonction de callback

```
var doneCallback = function() { . . . };
$animate.removeClass(element, 'mon_anim', doneCallback);
```

SQLI GROUP

EXERCICES PRATIQUES

TP13. ANIMATIONS

FIN