



Sébastien Lafontaine
Version 1.2.1
19/01/2018
AngularJS

Node : logiciel qui s'occupe de la partie serveur (back).

AngularJS : framework qui s'occupe de la partie client (front).

- Toujours avoir un « index.html » pour mettre les scripts et les liens css
- Mettre les fichiers « bootstrap-theme.min.css » et « bootstrap.min.css »
- Placer les fichiers javascript en bas de la page

Databinding : lien entre les données et leur affichage (bind : lien)

Directive : fonctionnalité qui exécute les actions (**ng-controller=** 'trucCtrl')

\$scope.unnom = qqc ;

J'ajoute une nouvelle fonction à mon controller : **\$scope.nouvellefonction = fonction{\$scope qqc ++ ;}** (ici la fonction ajoute automatiquement +1 à chaque fois)

ng-repeat='truc in trucs'

→ Répète autant de fois que nécessaire l'information.

[] : dépendances associées à la fonction

```
(var app = angular.module('truc', []);
```

Module : définit l'application, il contient les différents controller

```
ng-app='uniqueapplication'
```

→ Il faut déclarer obligatoirement un seul ng-app dans son html pour situer son application

HTML

```
<div ng-app='cequejeveux'>untext</div>
```

JS

```
var app = angular.module('cequejeveux', []);
```

Controller : il contrôle les données qui sont créées

HTML

```
<body ng-controller='untruc as truc'>
  <div class='des trucs'>
    <h3>{{7eleven.produits.prix}}
      <em>{{7eleven.produits.nom}}</em>
    </h3>
  </div>
```

JS

```
(function(){
  var gem = {nom : 'topaz', prix : 80};
  var app = angular.module('7eleven', []);
  app.controller('untruc', function($scope){
    this.produits = gem;
  });
})();
```

```
ng-show='7eleven.produits.achetable'
```

→ peut montrer l'élément si achetable = true dans le js

```
ng-show='!7eleven.produits.épuisé'
```

→ affiche les valeurs inverses grâce au !

```
ng-hide='7eleven.produits.épuisé'
```

→ cache quand épuisé=true dans le js

ng-repeat='produit in 7elevenproduits'

→ répète données d'une arraylist

Filter : ajout d'une « pipe » dans les crochets

{{produits.prix | **currency**}} (currency ajoute des virgules sur des prix par exemple)

| **date** : 'MM/dd/yyyy h : mma'

→ spécifie un format d'affichage date et heure

| **limitTo** : 11

→ met une limite à 11 (ex : caractère | limitTo : 11)

| **orderBy** : '-prix'

→ organise par prix décroissant sinon normalement en « ascending » : croissant

ng-src

→ reprend les fichiers du js et les affiche

ex :

```
var produits = [{  
  nom : mochi  
  prix : 4  
  images : [  
    'images/gem-02.png',  
    'images/gem-03.png'  
  ],  
}
```

Tabs :

ng-click="

HTML

```
<div class="container">  
  <section ng-app="myApp" ng-controller="TabController as tab">  
    <ul class="nav nav-pills">  
      <li ng-class="{active:tab.isSet(1)}">  
        <a href ng-click="tab.setTab(1)">Tab 1</a></li>  
      <li ng-class="{active:tab.isSet(2)}">  
        <a href ng-click="tab.setTab(2)">Tab 2</a></li>  
      <li ng-class="{active:tab.isSet(3)}">  
        <a href ng-click="tab.setTab(3)">Tab 3</a></li>  
    </ul>  
    <div ng-show="tab.isSet(1)">  
      <h4>Tab 1</h4>  
    </div>
```

```
<div ng-show="tab.isSet(2)">
  <h4>Tab 2</h4>
</div>
<div ng-show="tab.isSet(3)">
  <h4>Tab 3</h4>
</div>
</section>
</div>
```

JS

```
(function () {
  var app = angular.module('monApp', []);

  app.controller('TabController', function () {
    this.tab = 1;

    this.setTab = function (tabId) {
      this.tab = tabId;
    };

    this.isSet = function (tabId) {
      return this.tab === tabId;
    };
  });
})();

$scope.isSet = function(tabNum){
  return $scope.tab === tabNum;
};
});
```

Method :

```
this.setCurrent = function(value) {
  this.current = value || 0;
```