

Lab 2: Framework Angular (Directives & Pipes)

On s'intéresse dans ce lab aux éléments Angular suivants:

- Les directives structurales : permettent de modifier le DOM en ajoutant, enlevant ou remplaçant un élément du DOM (*ngIf et *ngFor)
- Les attribut directives : permettent de modifier l'apparence ou le comportement d'un élément ([ngStyle] et [ngClass])
- La création d'une nouvelle directive (la nôtre)
- Pipes (|) : formatage d'une Date

Liste des sessions de formation

Suivre les sessions que je veux participer.

Web

MEAN Stack
Prévue le Jun 13, 2018
pendant 3 jours à Lyon
Session complète!
20 Participants

[Details](#)[Inscription](#)[Partager](#)

Mobile

Angular et Ionic
Prévue le Aug 10, 2018
pendant 5 jours à Paris
Encore 15 places sont
disponibles!
5 Participants

[Details](#)[Inscription](#)[Partager](#)

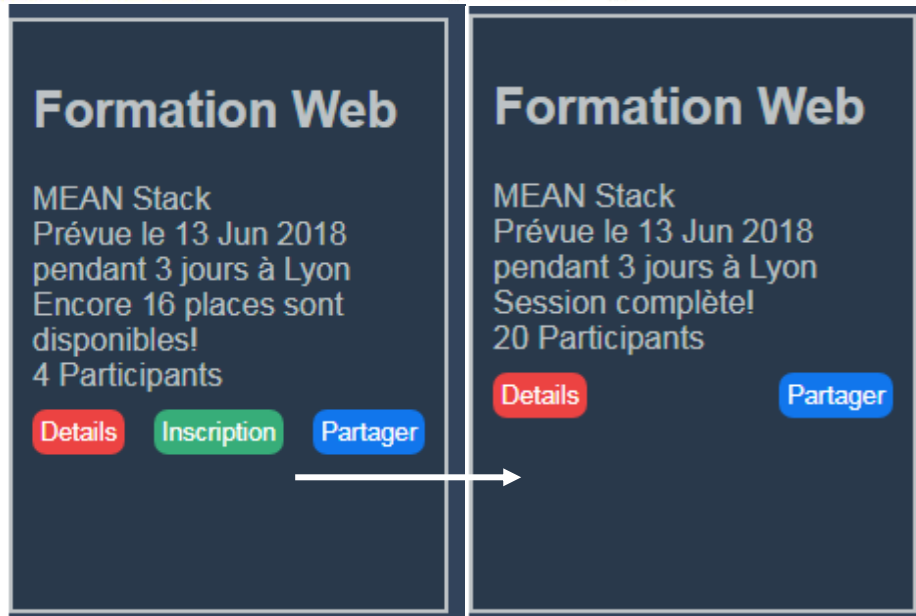
Web

NodeJS
Prévue le Jul 20, 2018
pendant 5 jours à Lyon
Session complète!
20 Participants

[Details](#)[Inscription](#)[Partager](#)

Etapes d'implémentations

1. Cette question s'intéresse à la Directive structurale (**ngIf**). On suppose que chaque session contient **20 participants**. Ou vous demande d'afficher le message « **Session Complete** » si on atteint 20 participants. Pensez aussi à une solution pour cacher/désactiver le lien « Inscription »



- a. Modifier la structure dans le composant « AppComponent » de la manière suivante ;

```
export class AppComponent {  
  firstSession = {  
    id: 1,  
    name: 'Formation Web',  
    track: 'MEAN Stack',  
    date: new Date('2018-06-13'),  
    duree: 3,  
    local: 'Lyon',  
    participants: 0  
  };  
}
```

- b. Modifier le template HTML (DOM) du composant SessionItemComponent de la manière suivante :

```
1 <h2>{{session.name}}</h2>  
2 <div>{{session.track}}</div>  
3 <div>Prévue le {{session.date | date:'d MMM y'}} pendant {{session.duree}} jours à {{session.local}}</div>  
4 <li *ngIf="session.participants < 20;then ok else notok"></li>  
5 <ng-template #ok>Encore {{20 - session.participants}} places sont disponibles!</ng-template>  
6 <ng-template #notok>Session complète!</ng-template>  
7 <div>{{session.participants}} Participants</div>  
8 <div class="tools">  
9   <a class="delete">  
10     Details  
11   </a>  
12   <a *ngIf="session.participants < 20"(click)="inscrire()" class="inscrire">  
13     Inscription  
14   </a>  
15   <a class="partager">  
16     Partager  
17   </a>  
18 </div>
```

- c. On veut afficher la date de la formation sous la forme *Mois Jour, Année*. En utilisant une pipe, modifier l'affichage de l'attribut date (la date de la session) dans le template HTML (DOM) du composant SessionItemComponent:

```
<div>Prévue le {{session.date | date}} pendant {{session.duree}} jours à {{session.local}}</div>
Ou bien
<div>Prévue le {{session.date | date:'d MMM y'}} pendant {{session.duree}} jours à {{session.local}}</div>
```

2. Cette question s'intéresse à la Directive structurale (**ngFor**). On suppose maintenant qu'on a une liste de formations à afficher. On vous demande d'ajouter un nouveau composant « session-item-list » qui s'occupera de toutes les sessions de formations.
- a. A l'aide d'Angular CLI, créer le composant « session-item-list » avec le contenu suivant :
- i. Style CSS (session-item-list.component.css)

```
:host {
  display: flex;
  min-height: 90vh;
  flex-direction: column;
  padding: 10px;
}
header {
  color: #ffffff;
  padding: 10px;
}
header > h1 {
  font-size: 2em;
}
header .description {
  font-style: italic;
}
section {
  flex: 1;
  display: flex;
  flex-flow: row wrap;
  align-content: flex-start;
}
section > app-session-item {
  margin: 10px;
}
footer {
  text-align: right;
}
footer .icon {
  width: 64px;
  height: 64px;
  margin: 15px;
}
```

- ii. Composant SessionItemListComponent (session-item-list.component.ts)

```
import { Component, OnInit } from '@angular/core';
@Component({
  selector: 'app-session-item-list',
  templateUrl: './session-item-list.component.html',
  styleUrls: ['./session-item-list.component.css']
})
export class SessionItemListComponent implements OnInit {
  sessionItems = [
    {
      id: 1,
      name: 'Web',
      track: 'MEAN Stack',
      date: new Date('2018-06-13'),
      duree: 3,
      local: 'Lyon',
      participants: 0
    },
    {
      id: 2,
      name: 'Mobile',
      track: 'Angular et Ionic',
      date: new Date('2018-08-10'),
      duree: 5,
      local: 'Paris',
      participants: 0
    },
    {
      id: 3,
      name: 'Web',
      track: 'NodeJS',
      date: new Date('2018-07/20'),
      duree: 5,
      local: 'Lyon',
      participants: 0
    }
  ];
  constructor() { }
  ngOnInit() {}
}
```

iii. Document HTML (session-item-list.component.html)

```
1 <section>
2   <app-session-item *ngFor="let sessionItem of sessionItems" [session]="sessionItem">
3   </app-session-item>
4 </section>
```

- b. Charger le nouveau composant SessionItemListComponent dans le composant root AppComponent :

i. Document HTML (app.component.html)

```
1 <section>
2   <header>
3     <h1>Liste des sessions de formation </h1>
4     <p class="description">Suivre les sessions que je veux participer.</p>
5   </header>
6   <app-session-item-list></app-session-item-list>
7 </section>
```

3. Cette question se focalise sur le deuxième type de directives *Attribute Directive*. Sachant qu'on a deux types de formation (**Web** et **Mobile**), on vous demande alors d'utiliser la directive `[ngClass]` afin d'afficher chaque session de formation selon la classe CSS définie par le fichier ***session-item.component.css***.

i. Document HTML (session-item-list.component.html)

```
1 <section>
2   <app-session-item *ngFor="let sessionItem of sessionItems" [session]="sessionItem"
3     [ngClass]="{'session-web': sessionItem.name === 'Web', 'session-mobile': sessionItem.name
4       === 'Mobile'}">
5 </app-session-item>
</section>
```

4. Cette question s'intéresse à la création d'une nouvelle directive. On voudrait **désactiver le lien « Inscription » une fois on atteint les 20 participants**.

a. Dans le composant *SessionItemListCompoent*, ajouter un nouvel attribut « *isCompleted* » de type booléen, par défaut initialisé à *False*. La valeur « *False* » signifie que la session est encore ouverte pour des nouvelles inscriptions. Par contre la valeur *True* signifie que la session est complétée.

b. Créer une nouvelle directive appelée « *InscriptionDisabledDirective* »

```
import {Directive, HostBinding, Input} from '@angular/core';

@Directive({
  selector: '[appInscriptionDisabled]'
})
export class InscriptionDisabledDirective {
  @HostBinding('class.link-is-disabled') status: boolean;
  @Input()
  set appInscriptionDisabled(value) {
    this.status = value;
  }
  constructor() { }
}
```

c. Ajouter la classe le fichier CSS du composant *SessionItemComponent* comme suit :

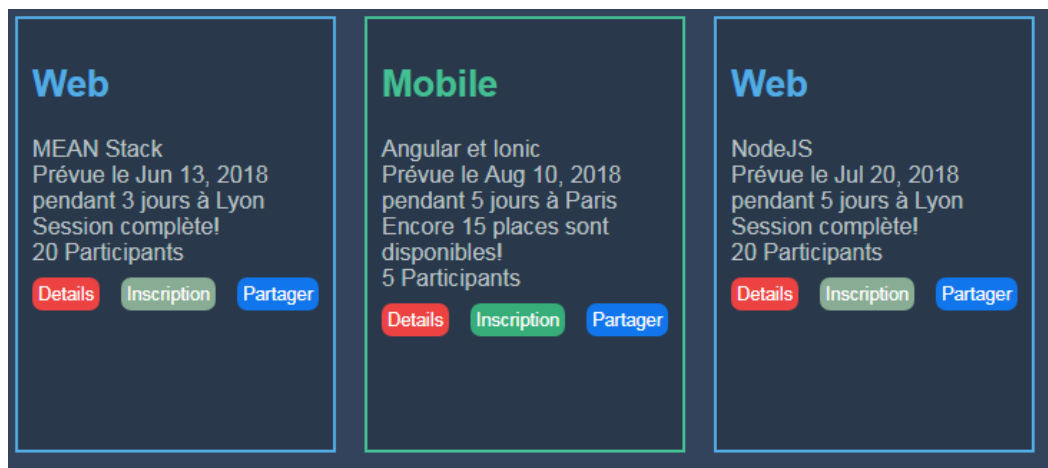
```
.inscrire.link-is-disabled {
  pointer-events: none;
  background-color: #88ad94;
}
```

d. Utiliser la nouvelle directive dans le fichier Template *session-item.compenent.html* :

```
12 <a [appInscriptionDisabled]="session.isCompleted" (click)="inscrire()" class="inscrire">
13   Inscription
14 </a>
```

- e. Modifier la fonction `inscrire()` dans le fichier TS « session-item.component.ts » afin de mettre à jour l'attribut `isCompleted` (devient True) de la session.

```
18  inscrire() {  
19      console.log('Nouvelle Inscription...');  
20      this.session.participants = +this.session.participants + 1;  
21      console.log(this.session.participants + ' Participants');  
22      this.participantsChange.emit({  
23          value: this.session.participants  
24      });  
25      if (this.session.participants >= 20) {  
26          this.session.isCompleted = true;  
27      }  
28  }  
29 }
```



Commentaires sur b):

Angular propose l'annotation **@HostBinding** qui permet de définir la classe CSS qu'on doit utiliser dans l'élément DOM (template HTML) d'un composant.

Pour définir la classe CSS, on utilise l'annotation **@HostBinding** ('class. link-is-disabled') sur une propriété booléenne de la Directive (*status*). La valeur de *status* détermine si la classe CSS est définie ou non. Si la valeur de *status* retournée est vraie, la classe sera appliquée à l'élément DOM du composant.

Angular propose aussi le décorateur (annotation) **@HostListener** permettant à une directive de réagir aux événements (par exemple lorsque le pointeur de la souris est sur/quitte

l'élément). Le décorateur *@HostListener* permet de positionner des Listners sur l'élément. Il prend en paramètre l'évènement que l'on veut écouter.