

## Angular 6 Training Course

### Exercise F-pipe

- Pipes are used to transform data in cases where we only need that transformation to happen in the **template/view**.
- PIPEs apply transformations to expressions and can be **chained together** to achieve compound effects.

### Setup

- Create a new project

```
ng new f-pipe
cd f-pipe
ng serve --open
// Review the project in localhost:4200
```

- Create a minimal template in

### Built-in Pipes

- Edit the main component class **src/app/app.component.html**.
- Create some properties that will be used in this example

```
export class AppComponent {

    conference = {
        title: "AngularCon",
        year: 2017,
        price: 745.50,
        byline:"the only Angular conference"
    };

    today = new Date();
}
```

- Edit the main template and add some basic styling.

```
<section class="demoPipe"></section>

.pipeDemo{
  font-family: tahoma;
  font-size: 2rem;
}
```

- Add some **built-in Angular Pipes**.
- These examples use currency, json, data and lowercase.

```
<p>The price was {{ conference.price | currency:"GBP":"£" }}</p>

<p>{{ conference | json }}</p>

<p>{{ today | date:"fullDate" }}</p>

<p>{{ today | date:"fullDate" | lowercase }}</p>
```

## Capitalise Pipe

- We will create a pipe that capitalises the first letter of each word called TitleCase.
- Here is a Javascript function to do this using a regular expression. *Try this out in the Chrome console.*

```
function toTitleCase(s) {
  return str.replace(/\w\S*/g,
    function(word){
      return word(0).toUpperCase() +
word(1).toLowerCase();
    });
}
```

- Use the Angular-CLI to **generate a new Pipe**.

```
ng generate pipe pipe/titleCase --dry-run
ng generate pipe pipe/titleCase
```

- Review **title-case.pipe.ts**

```
@Pipe({
  name: 'titleCase'
})
export class TitleCasePipe implements PipeTransform {
  transform(value: any): any {
    return null;
  }
}
```

- Our custom logic will go inside the **transform** function.

```
transform(value: any): any {

  return value.replace(/\w\S*/g, function(word) {
    return word.charAt(0).toUpperCase() +
word.substr(1).toLowerCase();
  });
}
```

- Use the custom pipe in the template.

```
<p>{{ conference.byline | titleCase }}
```

## Hyphenated Pipe

- Create a pipe which transforms a phrase into a lowercase hyphenated url.

```
"The History Of Oxford" becomes "the-history-of-oxford.html"
```

- Using the Angular CLI to **create a new pipe**.

```
ng generate pipe hyphen --dry-run
```

```
ng generate pipe hyphen
```

- **NgModule** has been updated by the CLI tools:

```
import { HyphenPipe } from './hyphen.pipe';  
declarations: [ AppComponent, HyphenPipe ],
```

- Make use of `split`, `join`, `toLowerCase` and add a file extension.
- Edit the return statement in the `transform()` function in **src/app/hyphen.pipe.ts**

```
return value.split(" ").join("-").toLowerCase() + ".html";
```

- Define a property and test the pipe

```
newsStory:string = "Mortgage Base Rate Rises";
```

- Test the new pipe in the template

```
<p>{{ newsStory | hyphen }}</p>
```

## PIPE arguments

- Add a string parameter which allows you to add a site prefix.

```
transform(value: any, site:string ): any {  
    value = site ? site + value : value;  
    return value.split(" ").join("-").toLowerCase() +  
    ".html";  
}
```

- Pass an argument in to the pipe:

```
<p>{{ newsStory | hyphen:"http://news.com/" }}</p>
```