Transforming Data

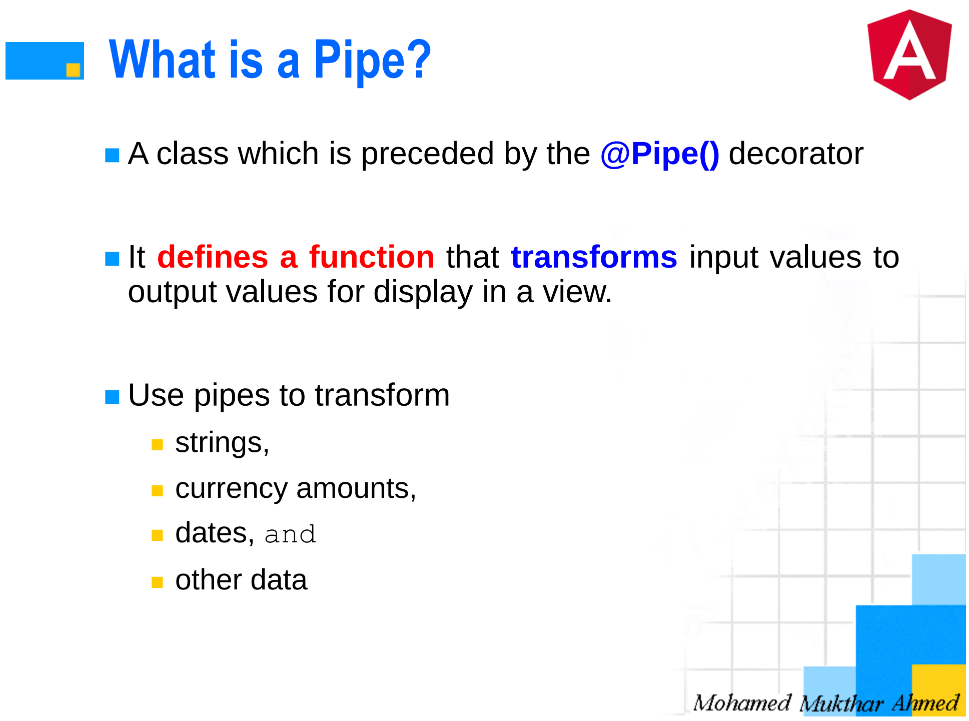
Mohamed Mukthar Ahmed

Transforming Data

Outline

- What is a Pipe?
- Transform Data Using Pipe
- Built-in Pipes
- Using a Pipe in Template
- Chaining Pipes
- Custom Pipes
  - Making a Class as Pipe
  - Using PipeTransform Interface
- Pure Pipes
- Impure Pipes
- Pipes and Precedence
- Q&A

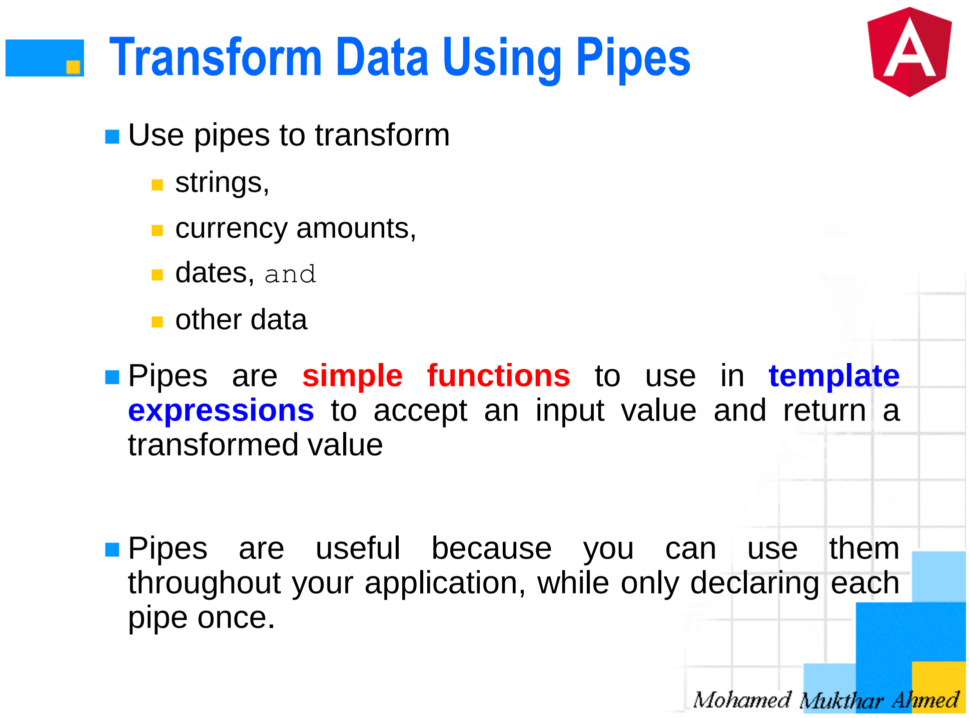
Mohamed Mukthar Ahmed



What is a Pipe?

- A class which is preceded by the **@Pipe()** decorator
- It **defines a function** that **transforms** input values to output values for display in a view.
- Use pipes to transform
  - strings,
  - currency amounts,
  - dates, and
  - other data

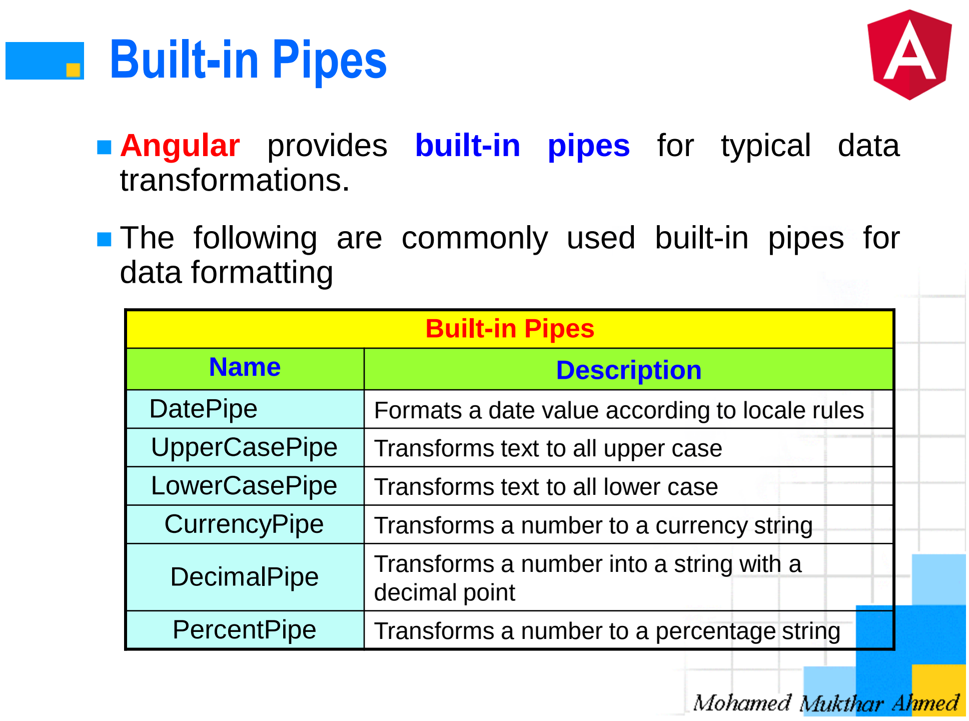
Mohamed Mukthar Ahmed



Transform Data Using Pipes

- Use pipes to transform
  - strings,
  - currency amounts,
  - dates, and
  - other data
- Pipes are **simple functions** to use in **template expressions** to accept an input value and return a transformed value
- Pipes are useful because you can use them throughout your application, while only declaring each pipe once.

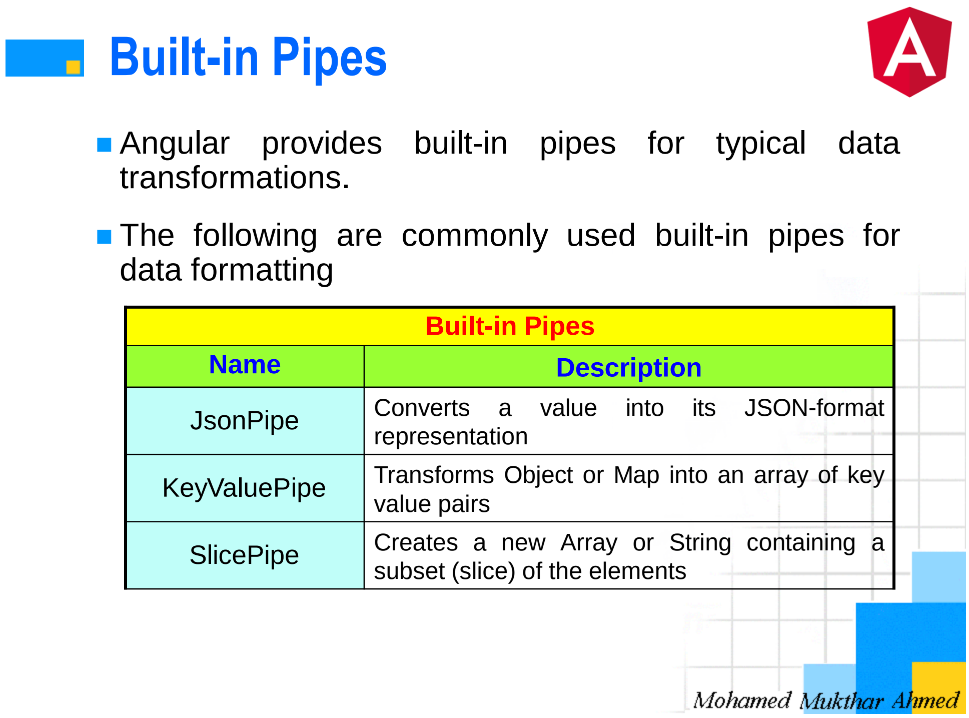
Mohamed Mukthar Ahmed



Built-in Pipes

- **Angular** provides **built-in pipes** for typical data transformations.
- The following are commonly used built-in pipes for data formatting

Built-in Pipes	
Name	**Description**
DatePipe	Formats a date value according to locale rules
UpperCasePipe	Transforms text to all upper case
LowerCasePipe	Transforms text to all lower case
CurrencyPipe	Transforms a number to a currency string
DecimalPipe	Transforms a number into a string with a decimal point
PercentPipe	Transforms a number to a percentage string



Built-in Pipes

- Angular provides built-in pipes for typical data transformations.
- The following are commonly used built-in pipes for data formatting

Built-in Pipes	
Name	**Description**
JsonPipe	Converts a value into its JSON-format representation
KeyValuePipe	Transforms Object or Map into an array of key value pairs
SlicePipe	Creates a new Array or String containing a subset (slice) of the elements

Using a Pipe in Template

- To apply a pipe, use the **pipe | symbol** within a template expression
  - <p>Current date: {{ today | date }}</p>
- Use **optional parameters** to fine-tune a pipe's output.
  - Use the CurrencyPipe with a country code such as EUR as a parameter
  - {{ amount | currency:'EUR’ }}
  - {{ today | date:"dd/MM/yy" }}
- Some pipes require at least one parameter and allow more optional parameters, such as SlicePipe.
  - {{ slice:1:5 }}

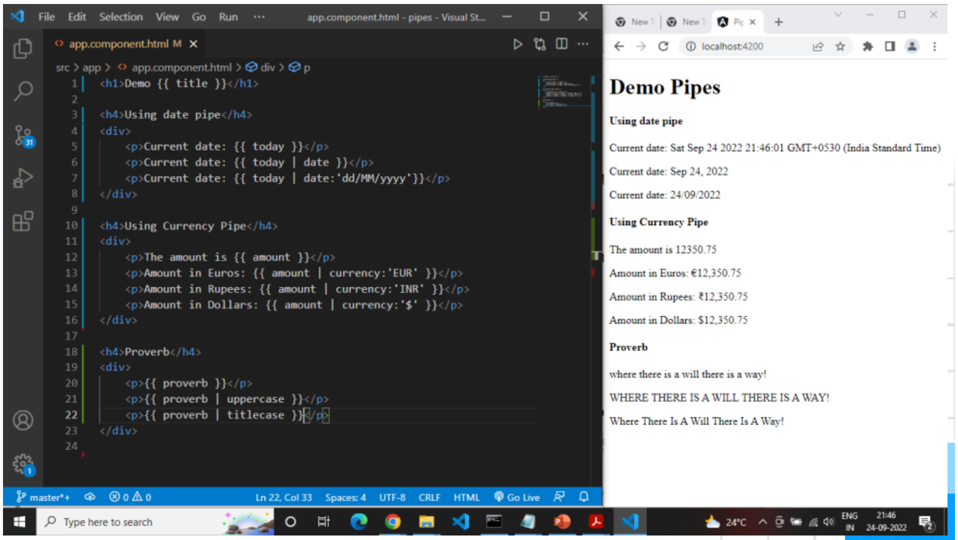
116

Chaining Pipes

- Applying two formats by **chaining** pipes
- Chain pipes so that the output of one pipe becomes the input to the next
  - {{ today | date | uppercase }}

117

Example



```html
<h1>Demo {{ title }}</h1>

<h4>Using date pipe</h4>
<div>
    <p>Current date: {{ today }}</p>
    <p>Current date: {{ today | date }}</p>
    <p>Current date: {{ today | date:'dd/MM/yyyy'}}</p>
</div>

<h4>Using Currency Pipe</h4>
<div>
    <p>The amount is {{ amount }}</p>
    <p>Amount in Euros: {{ amount | currency:'EUR' }}</p>
    <p>Amount in Rupees: {{ amount | currency:'INR' }}</p>
    <p>Amount in Dollars: {{ amount | currency:'$' }}</p>
</div>

<h4>Proverb</h4>
<div>
    <p>{{ proverb }}</p>
    <p>{{ proverb | uppercase }}</p>
    <p>{{ proverb | titlecase }}</p>
</div>
```

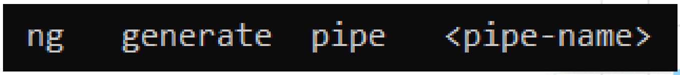
```
Demo Pipes

Using date pipe

Current date: Sat Sep 24 2022 21:46:01 GMT+0530 (India Standard Time)

Current date: Sep 24, 2022

Current date: 24/09/2022

Using Currency Pipe

The amount is 12350.75

Amount in Euros: €12,350.75

Amount in Rupees: ₹12,350.75

Amount in Dollars: $12,350.75

Proverb

where there is a will there is a way!

WHERE THERE IS A WILL THERE IS A WAY!

Where There Is A Will There Is A Way!
```

Custom Pipes

- Create **custom pipes** to **encapsulate** transformations that are not provided with the built-in pipes
- Done with the following steps:
- [a] Marking a class as a pipe
  - To mark a class as a pipe and supply configuration metadata, apply the **@Pipe** decorator to the class
  - Use **UpperCamelCase** (the general convention for class names)
  - **camelCase** for the corresponding name string
  - **Do not use hyphens** in the name
- [b] Use the **PipeTransform** interface

Custom Pipes

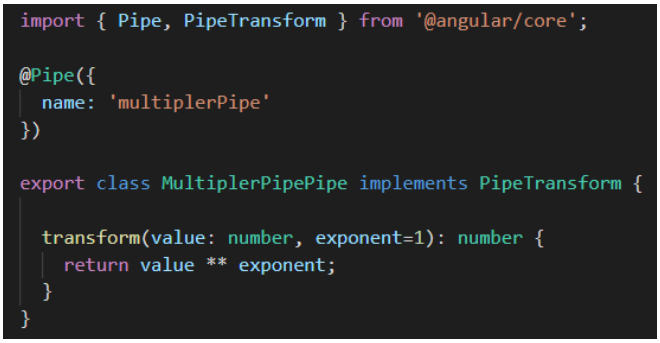
- Create custom pipes to encapsulate transformations that are not provided with the built-in pipes
- Done with the following steps:
- [b] Use the **PipeTransform** interface
  - Implement the **PipeTransform** interface in your custom pipe class to perform the transformation
  - Angular invokes the transform method with the value of a binding as the first argument, and any parameters as the second argument in list form, and returns the transformed value
- Use the following command:



```
ng   generate  pipe   <pipe-name>
```

Custom Pipe - Example

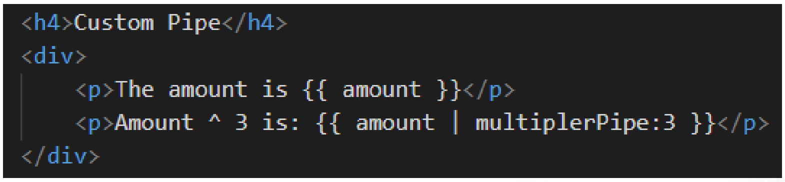
- Sample Code



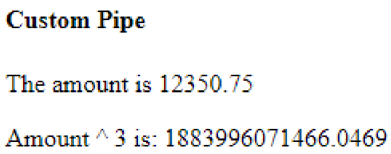
```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'multiplerPipe'
})

export class MultiplerPipePipe implements PipeTransform {

  transform(value: number, exponent=1): number {
    return value ** exponent;
  }
}
```

Custom Pipe - Example

- Sample Code



```
<h4>Custom Pipe</h4>
<div>
    <p>The amount is {{ amount }}</p>
    <p>Amount ^ 3 is: {{ amount | multiplerPipe:3 }}</p>
</div>
```



Custom Pipe

The amount is 12350.75

Amount ^ 3 is: 1883996071466.0469

Pure Pipes

- A **pure pipe** must use a pure function, which is one that processes inputs and returns values without side effects.

- In other words, given the same input, a pure pipe should always return the same output.

- With a pure pipe, **Angular** **ignores changes** within composite objects

Impure Pipes

- If a pipe results in a **change** in the composite objects, then we term such pipes as **impure pipes**.
- Angular executes an impure pipe every time it detects a change with every keystroke or mouse movement
- While an impure pipe can be useful, be careful using one.
- A long-running impure pipe could dramatically slow down your application.
- Make a pipe impure by setting its **pure** flag to **false**

Pipes and precedence

- The pipe operator has a **higher precedence** than the **ternary operator** (**? :**)
- Which means: a ? b : c | x is parsed as a ? b : (c | x)
- If you want a pipe to apply to the result of a ternary, wrap the entire expression in **parentheses**.
- Example:
  - {{ (true ? 'true' : 'false') | uppercase }}

Mohamed Mukthar Ahmed