**Pipes in Angular**

* Transform data before display
* Built in pipes include lowercase, uppercase, decimal, date, percent, currency etc
* To apply a pipe on a bound property use the pipe character " | "  
  <td>{{employee.code | uppercase}}</td>
* We can also chain pipes<td>{{employee.dateOfBirth | date:'fullDate' |uppercase }}</td>
* Pass parameters to pipe using colon " : "

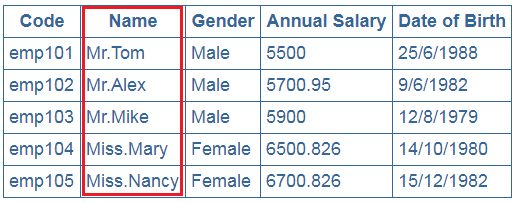
<td>{{employee.annualSalary | currency:'USD':true:'1.3-3'}}</td>

<td>{{employee.dateOfBirth | date:'fullDate'}}</td>

<td>{{employee.dateOfBirth | date:'dd/MM/y'}}</td>

Angular custom pipe

Depending on the gender of the employee, we want to display Mr. or Miss. prefixed to the employee name as shown below.



**Step 1 :** To achieve this let's create a custom pipe called employeeTitlePipe. Right click on the "employee" folder and add a new TypeScript file. Name it "employeeTitle.pipe.ts".

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

@Pipe({

    name: 'employeeTitle'

})

export class EmployeeTitlePipe implements PipeTransform {

    transform(value: string, gender: string): string {

        if (gender.toLowerCase() == "male")

            return "Mr." + value;

        else

            return "Miss." + value;

    }

}

**Step 2 :** declarations: [ AppComponent,EmployeeTitlePipe ]

**AppModule.ts**

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

import { EmployeeTitlePipe } from './empTitle.pipe';

@NgModule({

declarations: [ AppComponent,EmployeeTitlePipe ],

imports: [ BrowserModule ],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

**Step 3** : apply the pipe

Component.html

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1>

Welcome to {{title}}!

</h1>

</div>

<table>

<thead>

<tr>

<th>Code</th>

<th>Name</th>

<th>Gender</th>

<th>Annual Salary</th>

<th>Date of Birth</th>

</tr>

</thead>

<tbody>

<tr \*ngFor='let employee of employeeList'>

<td>{{employee.code | uppercase}}</td>

<td>{{employee.name | employeeTitle:employee.gender}}</td>

<td>{{employee.gender}}</td>

<td>{{employee.annualSalary}}</td>

<td>{{employee.dateOfBirth | date:'fullDate' |uppercase }}</td>

</tr>

<tr \*ngIf="!employeeList || employeeList.length==0">

<td colspan="5">

No Employee List to Display

</td>

</tr>

</tbody>

</table>

Nested Component Using Input / Output Properties

**AppModule.ts**

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

import { EmplistComponent } from './emplist/emplist.component';

import { EmpTitlePipe } from './emplist/empTitle.pipe';

import { EmpcountComponent } from './empcount/empcount.component';

import { FormsModule } from '@angular/forms';

@NgModule({

declarations: [AppComponent,EmplistComponent,EmpTitlePipe, EmpcountComponent ],

imports: [BrowserModule,FormsModule],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

EmplistComponent.ts (Parent Component)

import { Component} from '@angular/core';

import { IEmployee } from './IEmployee';

@Component({

selector: 'app-emplist',

templateUrl: './emplist.component.html',

styleUrls: ['./emplist.component.css']

})

export class EmplistComponent {

employeeList: IEmployee[];

// This property keeps track of which radio button is selected

// We have set the default value to All, so all the employees

// are displayed in the table by default

//selectedEmployeeCountRadioButton: string = 'All';

constructor() {

this.employeeList = [

{

code: 'emp101', name: 'Tom', gender: 'Male',

annualSalary: 5500, dateOfBirth: '6/25/1988'

},

{

code: 'emp102', name: 'Alex', gender: 'Male',

annualSalary: 5700.95, dateOfBirth: '9/6/1982'

},

{

code: 'emp103', name: 'Mary', gender: 'Female',

annualSalary: 6500.826, dateOfBirth: '10/14/1980'

},

];

}

getEmployees(): void {

this.employeeList = [

{

code: 'emp101', name: 'Tom', gender: 'Male',

annualSalary: 5500, dateOfBirth: '2/6/1988'

},

{

code: 'emp102', name: 'Alex', gender: 'Male',

annualSalary: 5700.95, dateOfBirth: '9/6/1982'

},

{

code: 'emp103', name: 'Mary', gender: 'Female',

annualSalary: 6500.826, dateOfBirth: '10/14/1980'

},

{

code: 'emp104', name: 'John', gender: 'Male',

annualSalary: 6500.826, dateOfBirth: '4/10/1980'

},

{

code: 'emp105', name: 'Nancy', gender: 'Female',

annualSalary: 6700.826, dateOfBirth: '5/12/1982'

},

];

}

// For Parent To Chield

/\* \*/

getTotalEmployeesCount(): number {

return this.employeeList.length;

}

getMaleEmployeesCount(): number {

return this.employeeList.filter(e => e.gender === 'Male').length;

}

getFemaleEmployeesCount(): number {

return this.employeeList.filter(e => e.gender === 'Female').length;

}

/\* \*/

// For chield to parent

/\* \*/

// This property keeps track of which radio button is selected

// We have set the default value to All, so all the employees

// are displayed in the table by default

selectedEmployeeCountRadioButton: string = 'All';

// Depending on which radio button is selected, this method updates

// selectedEmployeeCountRadioButton property declared above

// This method is called when the child component (EmployeeCountComponent)

// raises the custom event - countRadioButtonSelectionChanged

// The event binding is specified in employeeList.component.html

onEmployeeCountRadioButtonChange(selectedRadioButtonValue: string): void {

this.selectedEmployeeCountRadioButton = selectedRadioButtonValue;

}

/\* \*/

}

Emplist.component.html

<app-empcount [all]="getTotalEmployeesCount()"

[male]="getMaleEmployeesCount()"

[female]="getFemaleEmployeesCount()"

(countRadioButtonSelectionChanged)="onEmployeeCountRadioButtonChange($event)">

</app-empcount>

<br /><br />

<table>

<thead>

<tr>

<th>Code</th>

<th>Name</th>

<th>Gender</th>

<th>Annual Salary</th>

<th>Date of Birth</th>

</tr>

</thead>

<tbody>

<ng-container \*ngFor="let employee of employeeList;">

<tr \*ngIf="selectedEmployeeCountRadioButton=='All' ||

selectedEmployeeCountRadioButton==employee.gender">

<td>{{employee.code | uppercase}}</td>

<td>{{employee.name | employeeTitle:employee.gender }}</td>

<td>{{employee.gender}}</td>

<td>{{employee.annualSalary | currency:'USD':true:'1.3-3'}}</td>

<td>{{employee.dateOfBirth | date:'dd/MM/y'}}</td>

</tr>

</ng-container>

<tr \*ngIf="!employeeList || employeeList.length==0">

<td colspan="5">

No employees to display

</td>

</tr>

</tbody>

<br />

<button (click)='getEmployees()'>Refresh Employees</button>

</table>

**Empcount.component.ts (Chield component)**

//For parent to Chield Import Input

// For Chield to parent Import Output and EventEmitter

import { Component, Input, Output, EventEmitter } from '@angular/core';

@Component({

selector: 'app-empcount',

templateUrl: './empcount.component.html',

styleUrls: ['./empcount.component.css']

})

export class EmpcountComponent {

// @Input() For Parent To Chield

@Input()

all: number = 10;

@Input()

male: number = 5;

@Input()

female: number = 5;

/\* \*/// For Chield to parent

// Holds the selected value of the radio button

selectedRadioButtonValue: string="All";

// The Output decorator makes the property an Output property

// EventEmitter class is used to create the custom event

// When the radio button selection changes, the selected

// radio button value which is a string gets passed to the

// event handler method. Hence, the event payload is string.

@Output()

countRadioButtonSelectionChanged: EventEmitter<string> =

new EventEmitter<string>();

// This method raises the custom event. We will bind this

// method to the change event of all the 3 radio buttons

onRadioButtonSelectionChange() {

this.countRadioButtonSelectionChanged

.emit(this.selectedRadioButtonValue);

}

/\* \*/

}

Empcount.component.html

<span class="radioClass">Show : </span>

<input name='options' type='radio' value="All"

[(ngModel)]="selectedRadioButtonValue"

(change)="onRadioButtonSelectionChange()">

<span class="radioClass">{{'All(' + all + ')'}}</span>

<input name="options" type="radio" value="Male"

[(ngModel)]="selectedRadioButtonValue"

(change)="onRadioButtonSelectionChange()">

<span class="radioClass">{{"Male(" + male + ")"}}</span>

<input name="options" type="radio" value="Female"

[(ngModel)]="selectedRadioButtonValue"

(change)="onRadioButtonSelectionChange()">

<span class="radioClass">{{"Female(" + female + ")"}}</span>