

Angular

- Templates

```
templateUrl: './app.component.html'
```

template have ur own !!

using ES6 template syntax

can not have in multiple lines

use back-ticks for multilines

- Styles

styles have ur own !!

Angular

- Data Binding
 - use component data in the template
- Component Interaction with templates
 - expressions `{{variable}}`
 - directives `//*ngFor, *ngIf ..`
 - commands used in templates
 - constructor
 - initialise values for variables
 - ts to declare variable types
 - events
 - properties
 - bi-directional

Angular

- Events

(<name_of_event>) //in template

- a component method can be bound to event
- use \$event var. to pass the event as a parameter to the method
- template data can also be passed

[<prop_name>] //in template

- directive
- on any HTML tag property
- can be bound to component code
- sometimes more safer than using expressions
 - no loading issues
 - processed after the component is initialized

Angular

- Bi-directional binding
 - mostly useful with forms

`[(ngModel)]` `// monitor the value of any input field`

`[(ngControl)]` `//non input field form elements`

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

`//in app.module`

`//include in imports`

Angular

- External data and life cycle method

onInit

modular

```
import { HttpClientModule } from '@angular/common/http';
```

```
//in app.module
```

```
//include in imports
```

```
import { HttpClient } from '@angular/common/http';
```

```
//in app.component
```

subscriptions

Angular

- External data and life cycle method

onInit

//in app.component

export class AppComponent implements OnInit {

..

constructor(private http: HttpClient) {

subscriptions

```
ngOnInit(): void {  
  this.http.get<Object>('../assets/emp.json').subscribe(  
    empInfo => {  
      this.employee = empInfo;  
    })  
}
```

Angular

- Creating more components
 - CLI

ng generate component <component_name>

- - flat to have the the created component in the 'app' level

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-emp-pics',
  templateUrl: './emp-pics.component.html'
})
export class EmpPicsComponent implements OnInit {

  constructor() { }

  ngOnInit() {
  }

}
```

Angular

- Creating more components
 - CLI

updates the app.module.ts

```
import { EmpPicsComponent } from './emp-pics/emp-pics.component';
```

```
@NgModule({  
  declarations: [  
    AppComponent,  
    EmpPicsComponent  
  ],  
})
```


Angular

- Creating more components
 - sub-components inputs to get data from the main component

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-emp-pics',
  templateUrl: './emp-pics.component.html',
  inputs: ['emp']
})
export class EmpPicsComponent implements OnInit {

  constructor() { }

  ngOnInit() {
  }

}
```

Angular

- Creating more components
 - sub-components inputs to get data from the main component

```
<div class="media">  
  <!--   
  <!-- alt="{{e.name}}" -->  
  
  <!--   
  <!-- alt="{{e.name}}" -->  
  
  <div class="media-body align-self-center">  
    <h5 class="m-0">{{ emp.name }}</h5>  
    {{emp.desig}} <br>  
    {{emp.dept}}  
  </div>  
</div>
```

Angular

- Creating more components
 - props for communication between components

```
<app-emp-pics [emp]="e"></app-emp-pics>
```

Angular

- Pipes
 - filter & search
 - text-modifications
 - date formatting
 - CLI
 - transform(data_2b_transformed, modifier)

Angular

- Pipes

```
ng generate pipe find-employee
```

app.module gets updated

```
import { FindEmployeePipe } from './find-employee.pipe';

@NgModule({
  declarations: [
    AppComponent,
    EmpPicsComponent,
    FindEmployeePipe
  ],
  .
```

Angular

- Pipes

TS *find-employee.pipe.ts* ✕

```
1  import { Pipe, PipeTransform } from '@angular/core';
2
3  @Pipe({
4    | name: 'findEmployee'
5  | })
6  export class FindEmployeePipe implements PipeTransform {
7    |
8    |   transform(value: any, args?: any): any {
9    |   |   return null;
10   |   }
11   |
12 }
```

Angular

- Pipes

TS find-employee.pipe.ts ✕

```
1  import { Pipe, PipeTransform } from '@angular/core';
2
3  @Pipe({
4    name: 'findEmployee'
5  })
6  export class FindEmployeePipe implements PipeTransform {
7
8    transform(data_2b_transformed, modifier): any {
9      return data_2b_transformed.filter(eachEmp => {
10        return (
11          eachEmp['name'].toLowerCase().includes(modifier.toLowerCase()) ||
12          eachEmp['desig'].toLowerCase().includes(modifier.toLowerCase()) ||
13          eachEmp['dept'].toLowerCase().includes(modifier.toLowerCase())
14        )
15      });
16    }
17  }
18 }
```

Angular

- Pipes

```
*ngFor="let e of (employee | findEmployee: greet)"
```

- another sub-component

ng generate component about-emp

<> about-emp.component.html x

```
1  <div class="container mt-3">
2    <div class="row justify-content-center">
3      <div class="col-sm-4 col-lg-3">
4        <img class="img-fluid rounded" [src]=" './assets/images/' + emp.imglocn + '.png'"
5        alt="{{emp.name}} image">
6      </div>
7      <div class="col-sm-8 col-lg-4">
8        <h2 class="mt-3 mt-sm-0 mb-0">{{emp.name}}</h2>
9        <h4 class="mt-0">{{emp.desig}}</h4>
10       <p>{{emp.dept}}</p>
11     </div>
12   </div>
13 </div>
```


Angular

- another sub-component

TS *about-emp.component.ts* ✕

```
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-about-emp',
5    templateUrl: './about-emp.component.html',
6    inputs: ['emp']
7  })
8  export class AboutEmpComponent implements OnInit {
9
10     constructor() { }
11
12     ngOnInit() {
13     }
14
15 }
```

```

3 <div class="row justify-content-center">
4   <div class="card bg-success card mt-sm-3 col-md-6">
5     <div class="card-body">
6       <h3 class="mb-0 text-center text-warning">SSN Employee Repo.</h3>
7       <div class="form-group">
8         <div class="form-label text-center text-light mt-sm-3"
9           *ngIf="greet"><strong>{{ greet + ' !!!' }}</strong></div>
10        <input class="form-control mt-3" type="text"
11          [(ngModel)]="greet">
12      </div>
13    </div>
14  </div>
15</div>
16
17 <div class="position-absolute container" style="left: 0; z-index:10">
18   <div class="row justify-content-center">
19     <div class="col-sm-10 col-md-8 col-lg-6 col-xl-5">
20       <div class="list-group" *ngIf = "greet">
21         <a href="#" class="list-group-item list-group-item-action flex-column align-items-start"
22           *ngFor="let e of (employee | findEmployee: greet )"
23           (click)="displayEmployee($event, e)"
24           [style.backgroundColor]="e.bg ? '#CDE': '#FFF'">
25           <app-emp-pics [emp]="e"></app-emp-pics>
26
27         </a>
28       </div>
29     </div>
30   </div>
31</div>
32 <app-about-emp *ngIf = "this_e" [emp]="this_e"></app-about-emp>
33
34 </div>

```

Angular

- another sub-component

TS app.component.ts ✕

```
1  import { Component, OnInit } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3
4  @Component({
5    selector: 'app-root',
6    templateUrl: './app.component.html',
7  })
8  {
9    export class AppComponent implements OnInit {
10      greet: string;
11      employee: object;
12      this_e: object;
13
14      displayEmployee(evt, emp) {
15        // console.log(evt);
16        this.greet = 'Welcome ' + emp.name;
17        emp.bg = !emp.bg;
18        this.this_e = emp;
19      }
20
21      constructor(private http: HttpClient) {
22        // this.greet = 'Welcome';
23        this.greet = '';
24      }
25
26      ngOnInit(): void {
27        this.http.get<Object>('../assets/emp.json').subscribe(
28          empInfo => {
29            this.employee = empInfo;
30          }
31        )
32      }
33    }
34  }
```

<https://angular.io>