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

import { TutorialsComponent } from './tutorials.component';

@Component({

selector: 'my-app',

template: `<h1>Hello {{name}}</h1>

<my-tutorials></my-tutorials>`,

})

export class AppComponent { name = 'Angular 2'; }

**Data Binding, Two-way binding:**

*App.component.ts*

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

@Component({

selector: 'my-app',

template: `<h3 (click) = ”onSelect()”

[class.clicked] = “showDetail==true”>{{contact.firstName}} {{contact.lastName}}</h3>

<input [(ngmodel)]=”contact.firstName” type=”text”>

<div \*ngIf = “showDetail==true”>

Phone: {{contact.phone}}<br>

Email: {{contact.email}}

</div>`,

styleUrls: ["../src/styles.css "]

})

export class AppComponent {

public contact = { firstName: "Mahi", lastName: "Kaur", phone: "1234", email: "mk@gmail.com"

};

public showDetail = false;

onSelect(){

this.showDetail = true;

}

}

*Style.css*

h1 {

color: #369;

font-family: Arial, Helvetica, sans-serif;

font-size: 250%;

}

.clicked{  
color: red  
}

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

@Component({

selector: 'my-app',

template: ` <ul (click) = ”onSelect(contact)”

[class.clicked] = “showDetail==true”>

<li \*ngFor = “contacts of contact”>

{{contact.firstName}} {{contact.lastName}}

</li>

</ul>

<input [(ngmodel)] = “selctedContact.firstName” type=”text”>

<div>

Phone: {{selectedcontact.phone}}<br>

Email: {{selectedcontact.email}}

</div>`,

styleUrls: ["../src/styles.css "]

})

export class AppComponent {

public contact = [{ firstName: "Mahi", lastName: "Kaur", phone: "1234", email: "mk@gmail.com"}, { firstName: "Mohana", lastName: "Kanne", phone: "567", email: "mka@gmail.com"}, { firstName: "Mouni", lastName: "Kuchi", phone: "890", email: "mku@gmail.com"}, ];

public selectedContact{};

onSelect(contact){

this.selectedContact = contact;

}

}

**Multiple Components**

*app.component.ts*

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

import { ContactListComponent} from ‘../contacts/contact-listcomponent’;

@Component({

selector: 'my-app',

template: ` <contact-list></contact-list>

`,

Directives: [ContactListComponent],

})

export class AppComponent {

}

*contact-list.component.ts*

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

import {contactcomponent} from “./contact.component”;

@Component({

selector: 'contact-list',

template: ` <ul (click) = ”onSelect(contact)”

[class.clicked] = “selectedContact==contact”>

<li \*ngFor = “contacts of contact”>

{{contact.firstName}} {{contact.lastName}}

</li>

</ul>

<contact [contact]=”selectedcontact”></contact>`,

Directives: [ContactComponent],

styleUrls: ["../src/styles.css "]

})

export class Contact-ListComponent {

public contact = [{ firstName: "Mahi", lastName: "Kaur", phone: "1234", email: "mk@gmail.com"}, { firstName: "Mohana", lastName: "Kanne", phone: "567", email: "mka@gmail.com"}, { firstName: "Mouni", lastName: "Kuchi", phone: "890", email: "mku@gmail.com"}, ];

public selectedContact = {};

onSelect(contact){

this.selectedContact = contact;

}

}

*Contact.component.ts*

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

@Component({

selector: 'contact',

template: `

<input [(ngmodel)] = “contact.firstName” type=”text”>

<div>

Phone: {{contact.phone}}<br>

Email: {{contact.email}}

</div>`,

Inputs: [“contact”],

})

export class ContactComponent {

public contact = {}

}

**Services and constructor, ngonit**

*Contact.ts*

Export interface contact{

firstName string,

lastname string,

phone string,

email string

}

*Mock-contact.ts*

Import {contact} from “./contact”

Export const contacts: contact[] = [{ firstName: "Mahi", lastName: "Kaur", phone: "1234", email: "mk@gmail.com"}, { firstName: "Mohana", lastName: "Kanne", phone: "567", email: "mka@gmail.com"}, { firstName: "Mouni", lastName: "Kuchi", phone: "890", email: "mku@gmail.com"}, ];

*Contact.service.ts*

Import {injectable} from “angular2/core”;

Import {contacts} from “./mock-contact”;

@injectable()

Export class contactservice{

Getcontacts(){

Return promise.resolve(contacts);

}

*Contact-list.component.ts*

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

import { ContactComponent } from './contact.component';

import { ContactService } from './contact.service';

import { Contact } from './contact';

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

@Component({

selector: 'contact-list',

template: ` <ul (click) = "onSelect(contact)"

[class.clicked] = "selectedContact === contact">

<li \*ngFor = "contacts of contact">

{{contact.firstName}} {{contact.lastName}}

</li>

</ul>

<contact [contact] = "selectedContact" ></contact>`,

providers: [ContactService],

styleUrls: ["../src/styles.css "]

})

export class ContactListComponent implements OnInit {

public contacts: Contact[];

public selectedContact = {};

constructor(private \_ContactService: ContactService){}

onSelect(contact){

this.selectedContact = contact;

}

getContacts(){

this.\_ContactService.getContacts.then((contacts: Contact[])=>this.contacts = contact);

}

ngOnInit(): any {

this.getContacts();

}

}

**Inputs and Outputs**

*Tutorials.component.ts*

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

import { EventEmitter} from ‘@angular/core’;

@Component({

selector: 'my-tutorials',

template: ` <h2>Child TutorialsComponent value</h2>

<label>Enter Child TutorialsComponent Value</label>

<input type-=”text” #childtext (keyup)=”onChange(childtext.value)”>

<p>value from parent AppComponent is </p>

{{parentData}}`,

Inputs: [`parentData`],

Outputs: [`childEvent`]

})

export class TutorialsComponent {

public parentData: string;

childEvent = new EventEmitter<string>();

}

*App.component.ts*

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

import { TutorialsComponent} from ‘@angular/core’;

@Component({

selector: `my-app`,

template: ` <h1>Hello World</h1>

<h2>Parent App Component</h2>

<label>Enter Parent AppComponent Value</label>

<input type-=”text” #ptext (keyup)=”0”>

<p>value from parent AppComponent is </p>

<my-tutorials (childEvent)=”childData=$event”[parentData]=”ptext.value”></my-tutorials>`,

Directives: [TutorialsComponent]

})

export class AppComponent {

public childData: string;

}

**Why Angular 2?   
 *Performance:*** 5 times faster than angular js  
***Mobile Support:*** With angular 2 we can build single application that works across mobile and desktop devices.  
***Component-based:*** Components are building blocks of angular2.  
***More Language choices:*** ECMAScript5  
 ECMAScript6(also called ES2015)  
 Typescript  
 Dart  
 Purescript  
 Elm etc.

Angular2 itself is built using Typescript.  
Javascript language standard is officially called ECMAScript.

**Typescript:**  
Opensource and free programming language developed by Microsoft.  
Superscript of javascript.  
Transpilation compiles typescript to javascript.  
***Benefits:***  
Intellisense  
Autocompletion  
Code Navigation  
Advanced Refactoring  
Strong Typing  
Supports ES2015 features like classes, interfaces and inheritance.

Angular applications are modular in architecture.  
***Module*** is a block of code for specific purpose. We can export, import classes and fucntions.  
 **Component**A component in angular is a class with a template and decorator  
1. Template.  
2. Class.  
3. Decorator.  
*Template* defines the user interface. Contains HTML, directives and data bindings.  
*Class* contains code for the template.  
*Decorato*r adds metadata to the class making it an angular component.( Angular component is a javascript class with metadata).

**Directive**Directive is a sort of instruction present in HTML which allows to modify the DOM or extends it behavior.

* Stylings are restricted to components. They don’t off to and from html file.

**Constructor and ngOnInit**

* <https://www.dev6.com/ngOnInit-and-constructor-in-Angular2>
* https://angular.io/docs/ts/latest/guide/lifecycle-hooks.html