What is an Angular?

Angular is a platform and framework for building client applications in HTML and TypeScript.

https://angular.io

How to start

Download and install node.js if you don’t have one (https://nodejs.org/en/download/)

Install Angular CLI

```
npm install @angular/cli -g
```

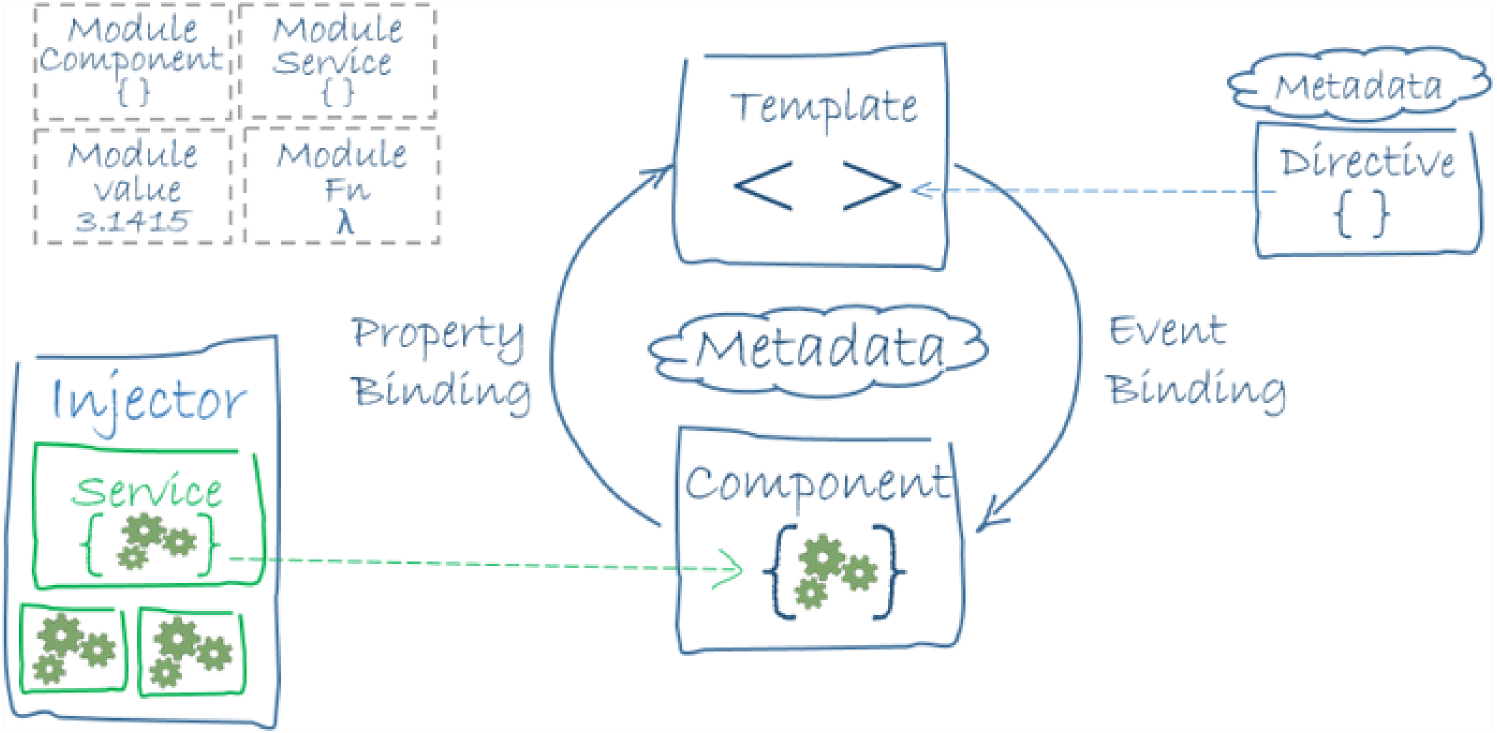
Create a new Project

```
ng new yourApp
```

Start your project

```
ng serve -o   //ng serve --port 4401
```

Angular Architecture



Source: https://angular.io/guide/architecture

Modules

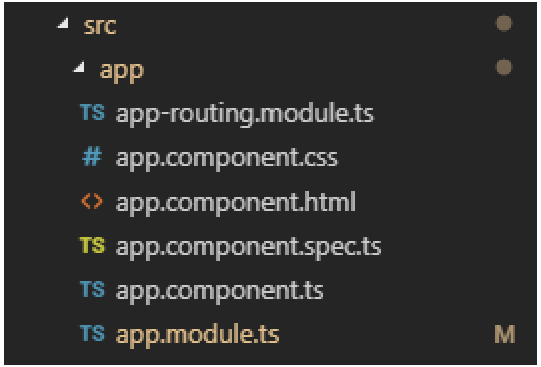
Module in Angular refers to a place where you can group the components, directives, pipes, and services, which are related to the application.

In case you are developing a website, the header, footer, left, center and the right section become part of a module.

To define module, we can use the **NgModule**.

Angular Modules - app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppRoutingModule } from './app-routing.module';

@NgModule({
declarations: [                 // components, directives, pipes
AppComponent,
],
imports: [                      // modules
BrowserModule,
AppRoutingModule
],
providers: [],                  // services
bootstrap: [AppComponent]       // root component
})
export class AppModule { }
```



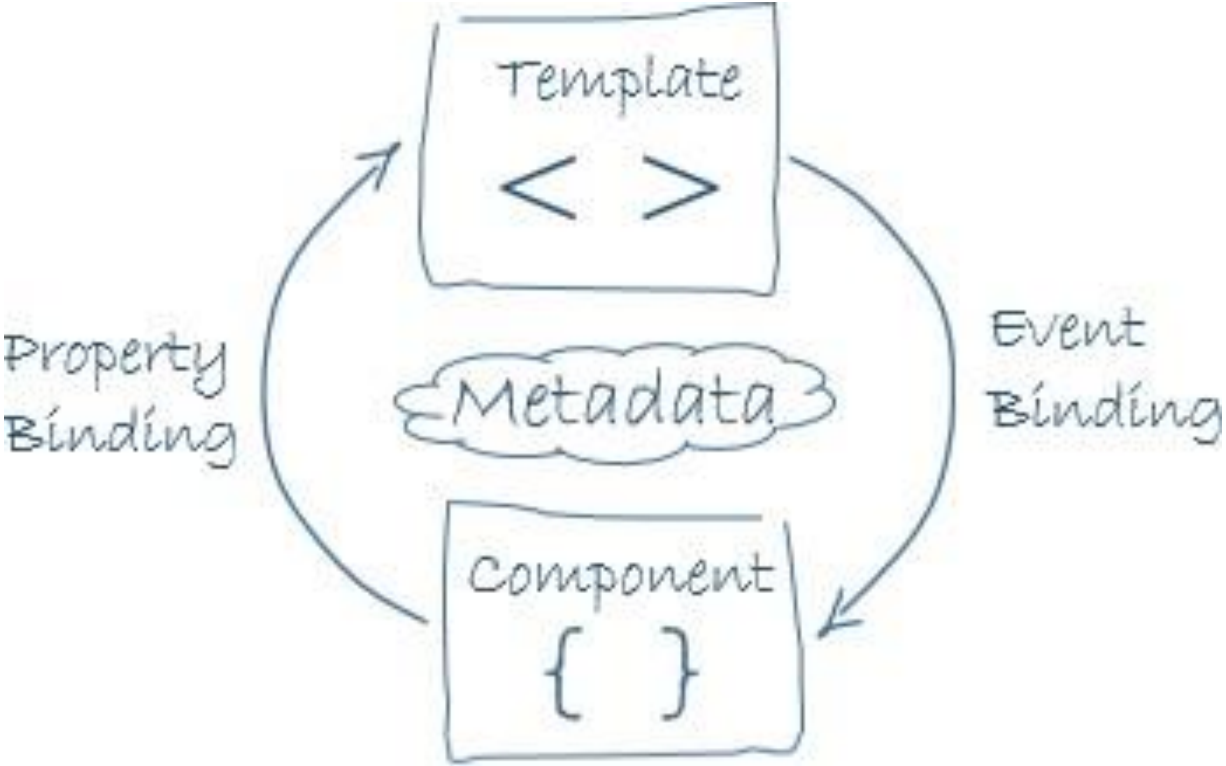
```
◢ src                          ●
  ◢ app                        ●
    TS app-routing.module.ts
    #  app.component.css
    <> app.component.html
    TS app.component.spec.ts
    TS app.component.ts
    TS app.module.ts           M
```

Components

Components are basically classes that interact with the .html file of the component, which gets displayed on the browser.

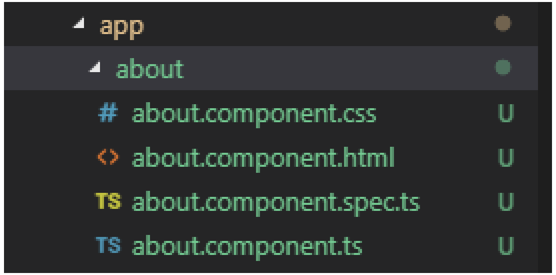
https://www.tutorialspoint.com/angular7/index.htm

Components


Template
< >
Property Binding
Metadata
Event Binding
Component
{ }

Create components

- ng g c [name]
- //ng g c about



```
◢ app                          ●
  ◢ about                      ●
    # about.component.css      U
    <> about.component.html    U
    TS about.component.spec.ts U
    TS about.component.ts      U
```

```
//about.component.ts
import {  Component, OnInit }  from '@angular/core';

@Component({
selector:  'app-about',
templateUrl:  './about.component.html',
styleUrls:  ['./about.component.css']
})
export class AboutComponent implements OnInit {

constructor() { }

ngOnInit()  {
}

}
```

Changes in the app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { AboutComponent } from './about/about.component';

@NgModule({
declarations: [
AppComponent,
AboutComponent
],
imports: [
BrowserModule,
AppRoutingModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }
```

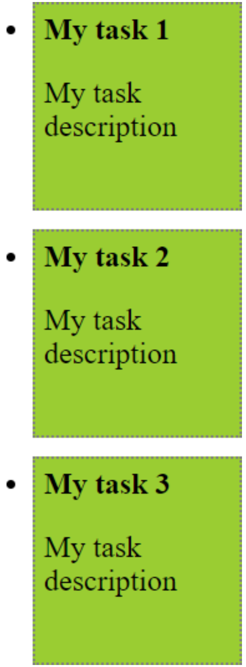
Component Class [app.component.ts]

```
                      import { Component} from '@angular/core';
                      @Component({
                      selector: 'app-root',
                      templateUrl: './app.component.html',
                      styleUrls: ['./app.component.css']
                      })
// public property    export class AppComponent{
                      myProp:string ='Go ahead, click that button’;

                      myArr = [
                      {'title':'My task 1', 'desc':'My task description'},
                      {'title':'My task 2', 'desc':'My task description'},
                      {'title':'My task 3', 'desc':'My task description'},
                      ];
// private property
                      private name: string;
                      myMethod() {
// public method      this.myProp ='That button above mewas clicked';
                      }
```

app.component.html

- <div class="container">
    - <button (click)="myMethod()">Press
  - me</button>
    - <p>My prop is {{ myProp }}</p>
    - <ul>
      - <li *ngFor="let arr of myArr">
        - <strong>{{ arr.title }}</strong>
          - <p>{{ arr.desc }}</p>
      - </li>
    - </ul>
- </div>
- <router-outlet></router-outlet>

app.component.css



- **My task 1**

  My task description

- **My task 2**

  My task description

- **My task 3**

  My task description

```
li {
    width: 100px;
    height: 100px;
    padding: 5px;
    margin: 10px;
    border: 1px dotted grey;
    background: yellowgreen;
}
```

Simple Routing & Navigation (Example)

src/app/app-routing.module.ts

```
const appRoutes: Routes = [
  { path: 'crisis-center', component: CrisisListComponent },
  { path: 'heroes', component: HeroListComponent },
  { path: '', redirectTo: '/heroes', pathMatch: 'full' },
  { path: '**', component: PageNotFoundComponent }
 ];
```

https://angular.io/guide/router

app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import {AboutComponent} from './about/about.component';

const routes: Routes = [
{
path: 'about/:name',
component: AboutComponent
}
];

@NgModule({
imports: [RouterModule.forRoot(routes)],
exports: [RouterModule]
})
export class AppRoutingModule { }
```

about.component.ts

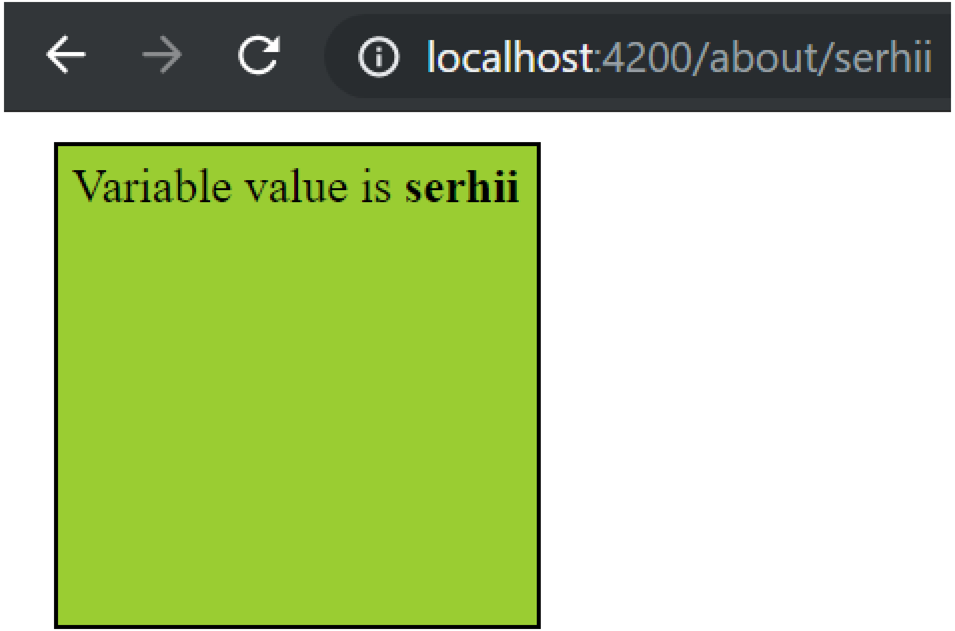
```
import { Component, OnInit } from '@angular/core';
import {ActivatedRoute} from '@angular/router';
import { from } from 'rxjs';

@Component({
selector: 'app-about',
templateUrl: './about.component.html',
styleUrls: ['./about.component.css']
})
export class AboutComponent implements OnInit {

name: string = '';
constructor(private router: ActivatedRoute) { }

ngOnInit() {
this.router.params.subscribe((params) =>{
this.name = params['name'];
})
}}
```

about.component.html

```
<div class="box">
Variable value is <strong>{{name}}</strong>
</div>
```

about.component.css

```
.box {
        width: 150px;
        height: 150px;
        padding: 5px;
        margin: 10px;
        border: 1px solid black;
        background: yellowgreen;
}
```



localhost:4200/about/serhii

Variable value is **serhii**

Services

- An angular service is simply a class that allows you to access its' defined properties and methods.

Used for features that:

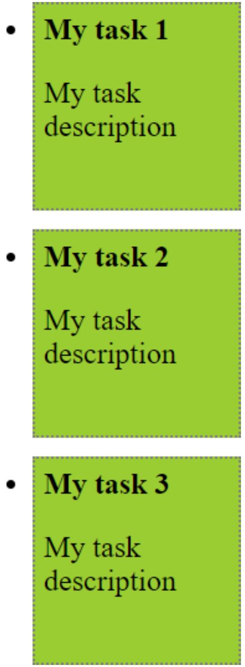
- Provide shared data and/or logic across components
- Encapsulate external interactions

How to create services

```
ng g s [name]
//ng g s data
```

```
//data.service.ts

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

@Injectable({
providedIn: 'root'
})
export class DataService{

getAll(): any[] { //Add a new method getAll()
  return [
{'title':'My task 1', 'desc':'My task description'},
{'title':'My task 2', 'desc':'My task description'},
{'title':'My task 3', 'desc':'My task description'},
];
}
constructor() { }
}
```

How to use services



- **My task 1**

  My task description
- **My task 2**

  My task description
- **My task 3**

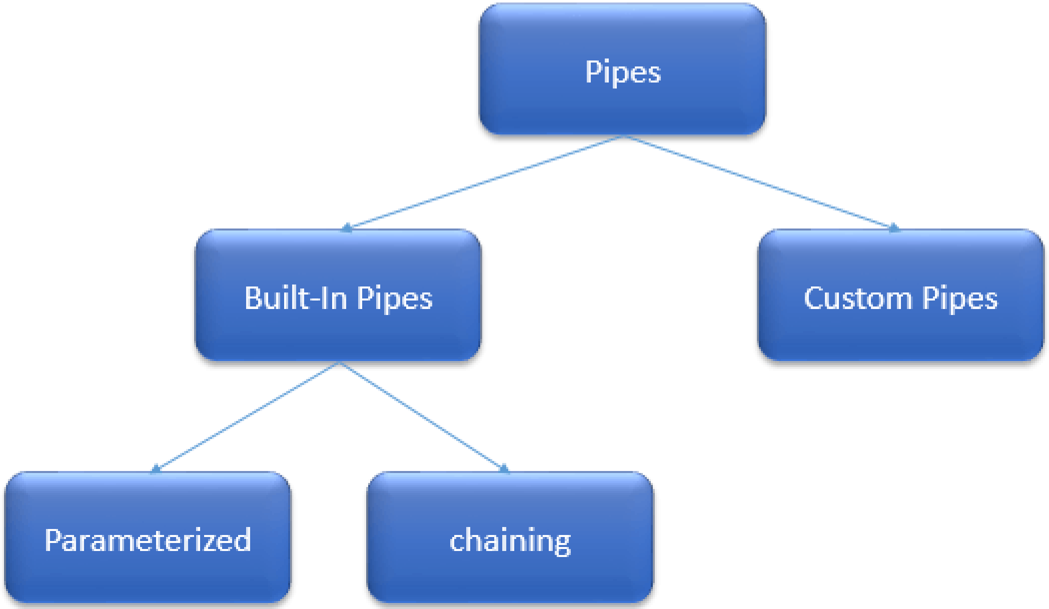
  My task description

//app.components.ts

```
import { Component, OnInit } from '@angular/core';
import { DataService } from './data.service';

@Component({
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
myProp: string ='Go ahead, click that button’;
myArr = [ ] ;
private name: string;
constructor(private data: DataService) { }
ngOnInit(): void {
this.myArr = this.data.getAll();
}
myMethod() {
this.myProp ='That button above mewas clicked’;
}
```

Pipes

Pipes allows us to change data inside of template or in a code.



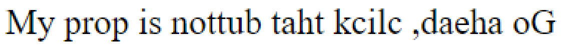
How to create Pipe

- ng g p [name]
- //ng g p *reverse-str*

```
// reverse-str.pipe.ts
import { Pipe, PipeTransform } from
'@angular/core';

@Pipe({
name: 'reverseStr'
})
export class ReverseStrPipe implements
PipeTransform {

transform(value: string): string {

let newStr: string = "";
for (let i = value.length - 1; i >= 0; i--) {
newStr += value.charAt(i);
}
return newStr;
}
}
```

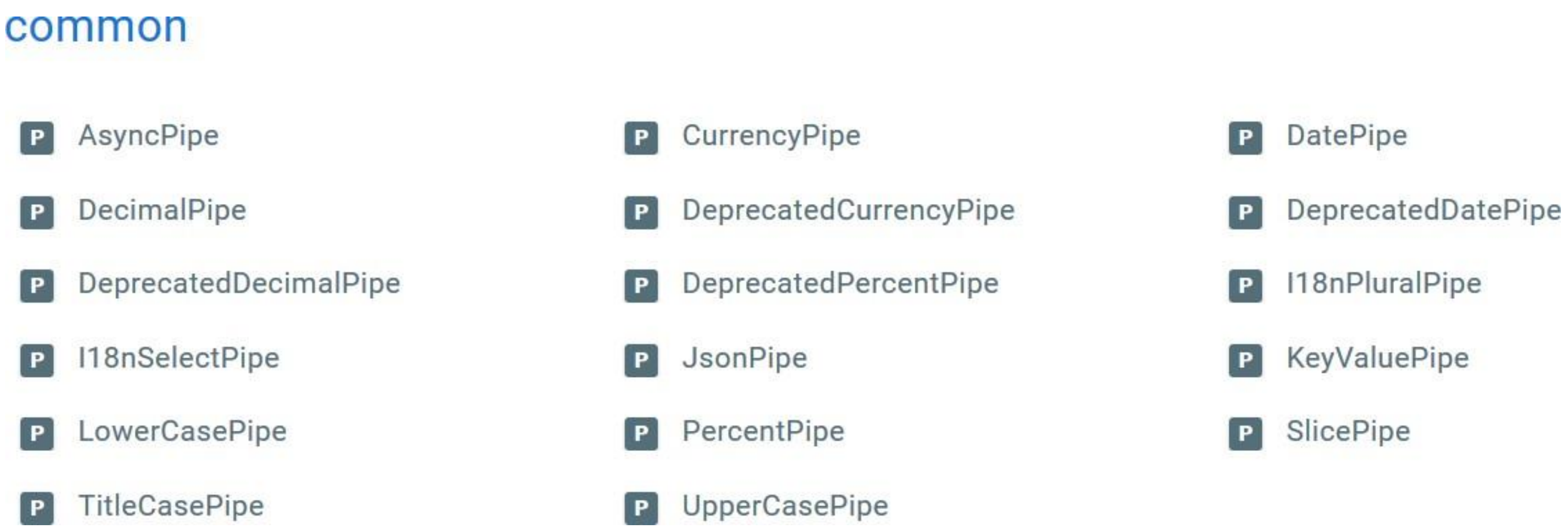
How to use Pipe

```
<!-- app.component.html -->

<div class="container">
        <button (click)="myMethod()">Press
    me</button>

        <p>My prop is {{ myProp | reverseStr
    }}</p>

        <ul>
            <li *ngFor="let arr of myArr">
                <strong>{{ arr.title
    }}</strong>
                     <p>{{ arr.desc }}</p>
            </li>
        </ul>
</div>

<router-outlet></router-outlet>
```



My prop is nottub taht kcilc ,daeha oG

Built-in Pipes

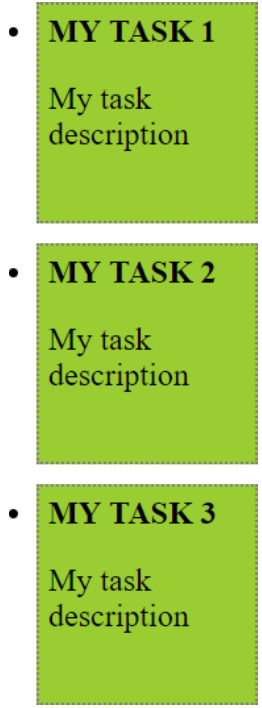


common

- AsyncPipe
- CurrencyPipe
- DatePipe
- DecimalPipe
- DeprecatedCurrencyPipe
- DeprecatedDatePipe
- DeprecatedDecimalPipe
- DeprecatedPercentPipe
- I18nPluralPipe
- I18nSelectPipe
- JsonPipe
- KeyValuePipe
- LowerCasePipe
- PercentPipe
- SlicePipe
- TitleCasePipe
- UpperCasePipe

https://angular.io/api?type=pipe

How to use built-in pipes



- **MY TASK 1**

  My task description

- **MY TASK 2**

  My task description

- **MY TASK 3**

  My task description

```
<!-- app.component.html -->

<div class="container">
            <button (click)="myMethod()">Press
        me</button>

            <p>My prop is {{ myProp| reverseStr }}</p>

            <ul>
                <li *ngFor="let arr of myArr">
                    <strong>{{ arr.title | uppercase
        }}</strong>
                         <p>{{ arr.desc }}</p>
                </li>
            </ul>
</div>

<router-outlet></router-outlet>
```

Directives

An Attribute directive changes the appearance or behavior of a DOM element.

There are three kinds of directives in Angular:

- Components—directives with a template.
- Structural directives—change the DOM layout by adding and removing DOM elements.
- Attribute directives—change the appearance or behavior of an element, component, or another directive.

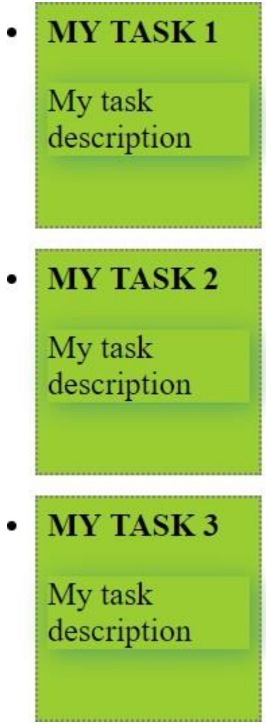
https://angular.io/guide/attribute-directives

How to create directives

- ng g d [name]
- //ng g d shadow

```
//shadow.directive.ts

import { Directive, ElementRef, Renderer2 } from
'@angular/core';

@Directive({
selector: '[appShadow]'
})
export class ShadowDirective {

constructor(elem: ElementRef, renderer: Renderer2)
{
renderer.setStyle(elem.nativeElement, 'box-shadow',
'2px 2px 12px #58A362');
}

}
```

How to use directives



- MY TASK 1

  My task description

- MY TASK 2

  My task description

- MY TASK 3

  My task description

```
<!-- app.component.html -->

<div class="container">
<button (click)="myMethod()">Press
me</button>

<p>My prop is {{ myProp| reverseStr}}</p>

<ul>
<li *ngFor="let arr of myArr">
<strong>{{ arr.title | uppercase}}</strong>

<p appShadow>{{ arr.desc }}</p>

</li>
</ul>
</div>

<router-outlet></router-outlet>
```