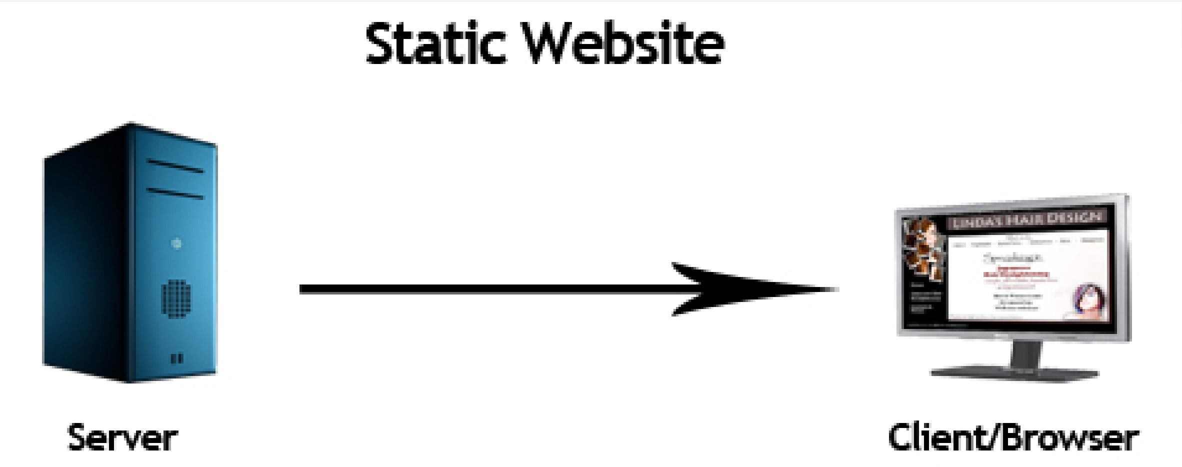
Angular Application Development

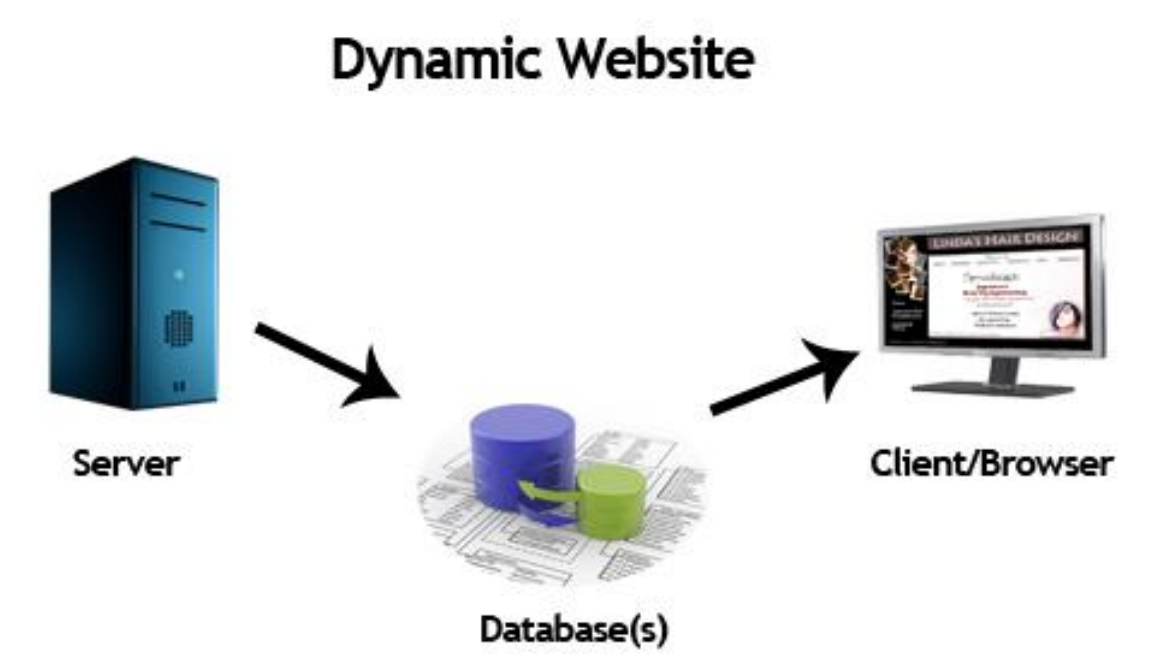
Website

COMPANY NAME
BRINGING THE FUTURE TO YOU
HOME
ABOUT US
LATEST NEWS
EMPLOYMENT
SERVICES
TESTIMONIALS
CONTACT
World Wide Web
BUILDING A BRIGHTER FUTURE
JOIN OUR CLUB
HEALTH WEB TEMPLATE
Art Website
WEB SHOP TEMPLATE
SHOP NOW & SAVE TIME!
WEB TEMPLATE ORANGE AND BLUE
FIND EVERYTHING YOU NEED!
WELCOME TO OUR WEBSITE

Static Vs Dynamic Website



Static Website

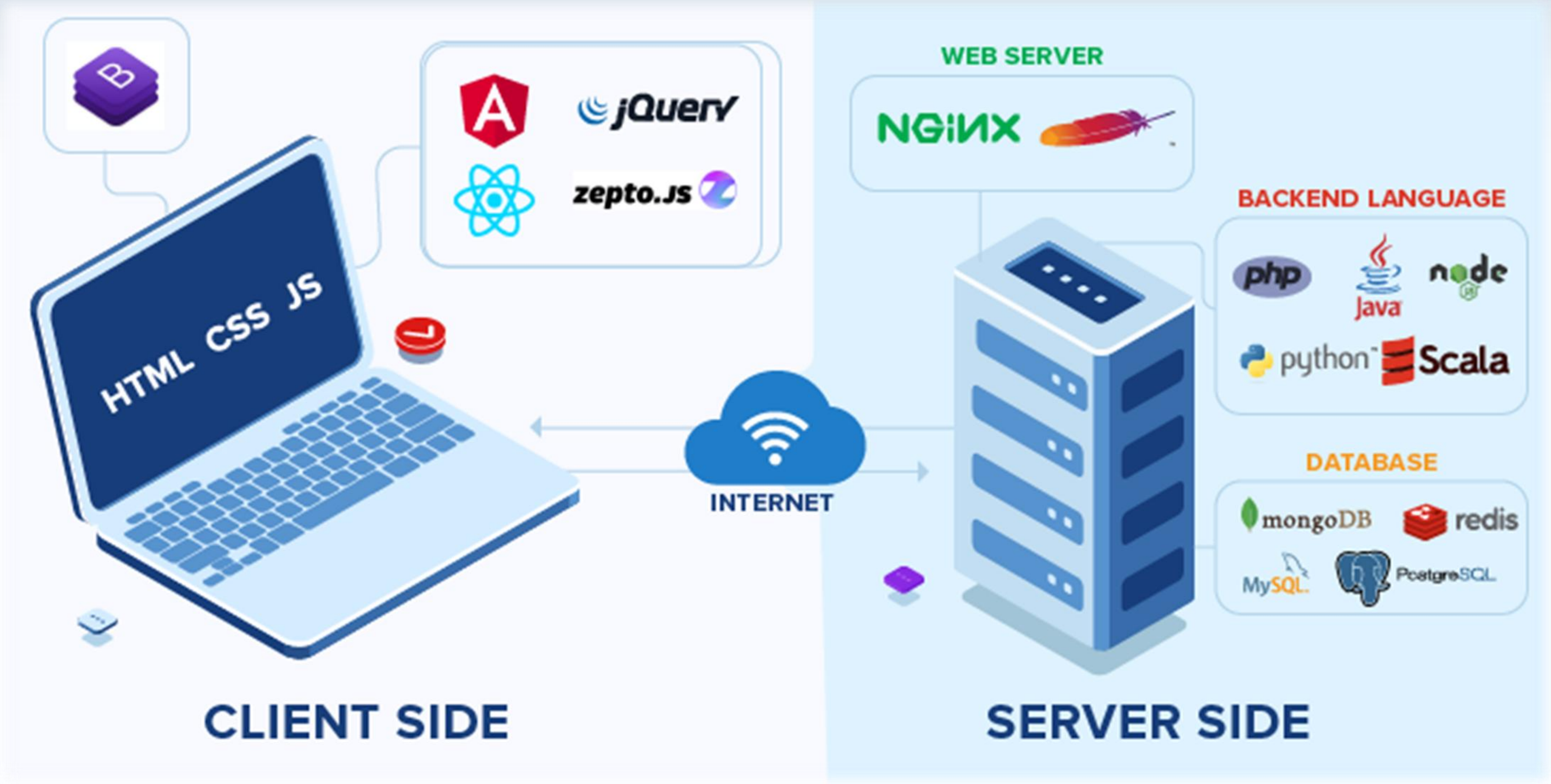


Dynamic Website

Websites Vs Web Applications

Responsive Web Design

Synchronous Vs Asynchronous



What is Angular..?

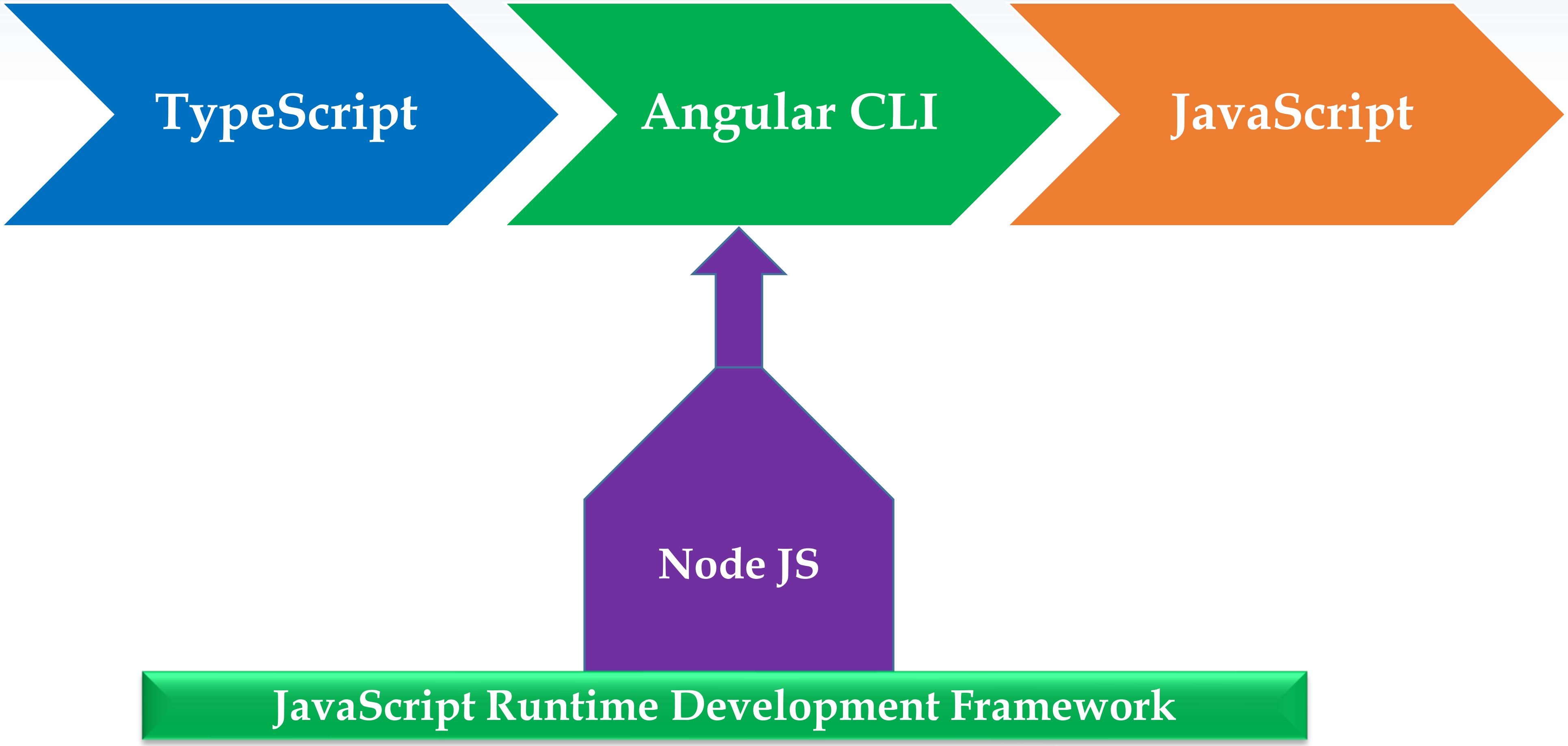
Front – End Web Development Framework

Single Page Web Applications (SPA)

Typescript | Superscript Language of JavaScript

Open Source Language, Supported by Google.

How Angular works..?



TypeScript

Angular CLI

JavaScript

Node JS

JavaScript Runtime Development Framework

Angular CLI | Node.js

- Command Line Interface for Angular Programming
- Create, Build, Run, Deploy Angular Projects
- Node.js | JavaScript Runtime Environment
- Angular CLI 19 | Node.js 23.3.0

Angular CLI | Installation

Download & Install Node.js:

Download Node.js from the following link: https://nodejs.org/en/download/

After successful installation, verify the Node.js existence in the system,

open cmd, type the following command:

```
node –v
V20.17.0
```

In this Node.js installation, NPM is installed with this. *verify the NPM existence in the system,* type the following command:

```
npm –v
10.8.2
```

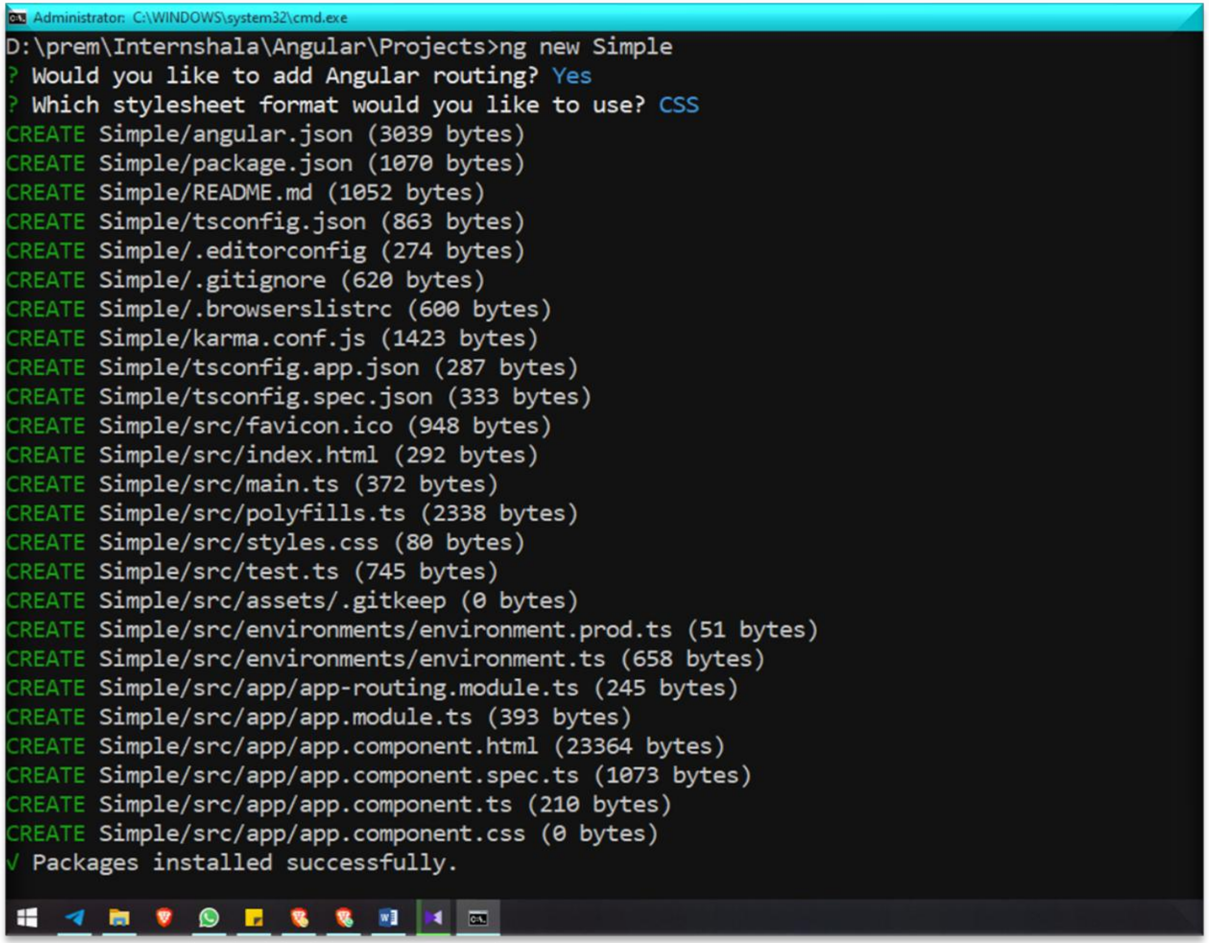
In the CMD, type the following command to install Angular CLI,

```
npm install –g @angular/cli
```

Create an Angular App

Check the installation of Angular CLI ng version

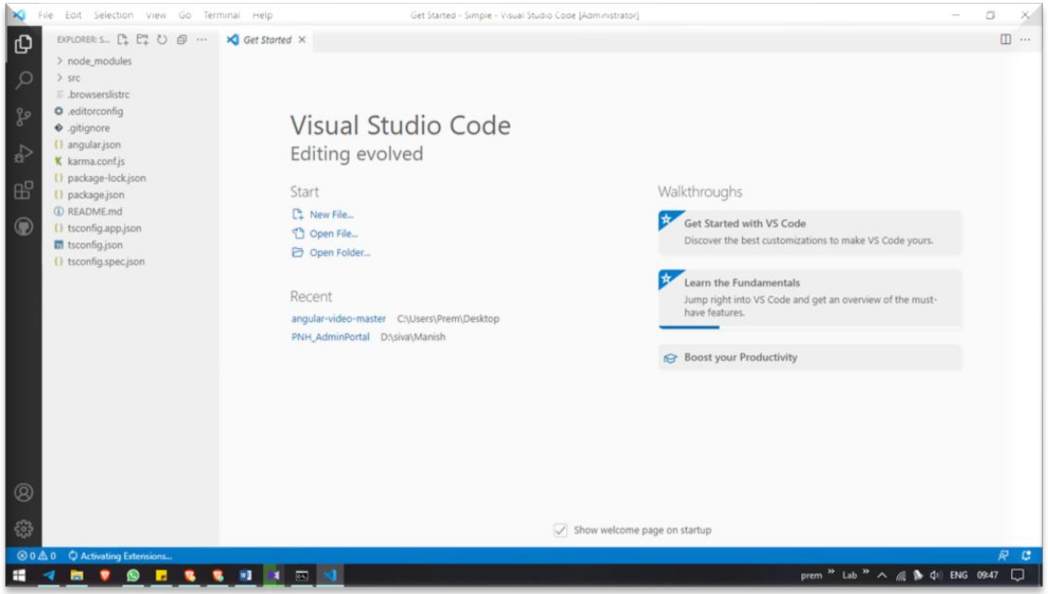
create a new app, use the following command: ng new Simple



Administrator: C:\WINDOWS\system32\cmd.exe

```
D:\prem\Internshala\Angular\Projects>ng new Simple
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE Simple/angular.json (3039 bytes)
CREATE Simple/package.json (1070 bytes)
CREATE Simple/README.md (1052 bytes)
CREATE Simple/tsconfig.json (863 bytes)
CREATE Simple/.editorconfig (274 bytes)
CREATE Simple/.gitignore (620 bytes)
CREATE Simple/.browserslistrc (600 bytes)
CREATE Simple/karma.conf.js (1423 bytes)
CREATE Simple/tsconfig.app.json (287 bytes)
CREATE Simple/tsconfig.spec.json (333 bytes)
CREATE Simple/src/favicon.ico (948 bytes)
CREATE Simple/src/index.html (292 bytes)
CREATE Simple/src/main.ts (372 bytes)
CREATE Simple/src/polyfills.ts (2338 bytes)
CREATE Simple/src/styles.css (80 bytes)
CREATE Simple/src/test.ts (745 bytes)
CREATE Simple/src/assets/.gitkeep (0 bytes)
CREATE Simple/src/environments/environment.prod.ts (51 bytes)
CREATE Simple/src/environments/environment.ts (658 bytes)
CREATE Simple/src/app/app-routing.module.ts (245 bytes)
CREATE Simple/src/app/app.module.ts (393 bytes)
CREATE Simple/src/app/app.component.html (23364 bytes)
CREATE Simple/src/app/app.component.spec.ts (1073 bytes)
CREATE Simple/src/app/app.component.ts (210 bytes)
CREATE Simple/src/app/app.component.css (0 bytes)
√ Packages installed successfully.
```

code Simple



Get Started - Simple - Visual Studio Code [Administrator]

Visual Studio Code

Editing evolved

Start

New File...

Open File...

Open Folder...

Recent

angular-video-master C:\Users\Prem\Desktop

PNH_AdminPortal D:\siva\Manish

Walkthroughs

Get Started with VS Code

Discover the best customizations to make VS Code yours.

Learn the Fundamentals

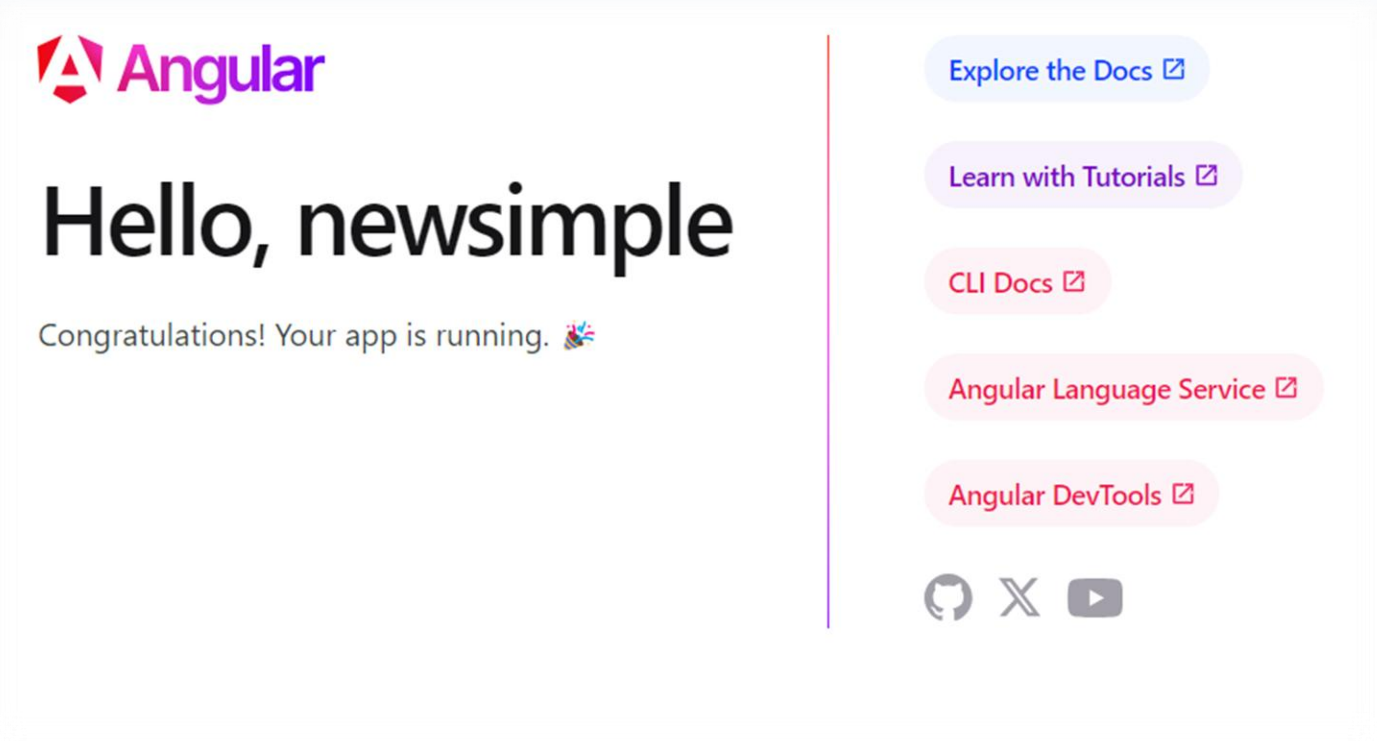
Jump right into VS Code and get an overview of the must-have features.

Boost your Productivity

☑ Show welcome page on startup

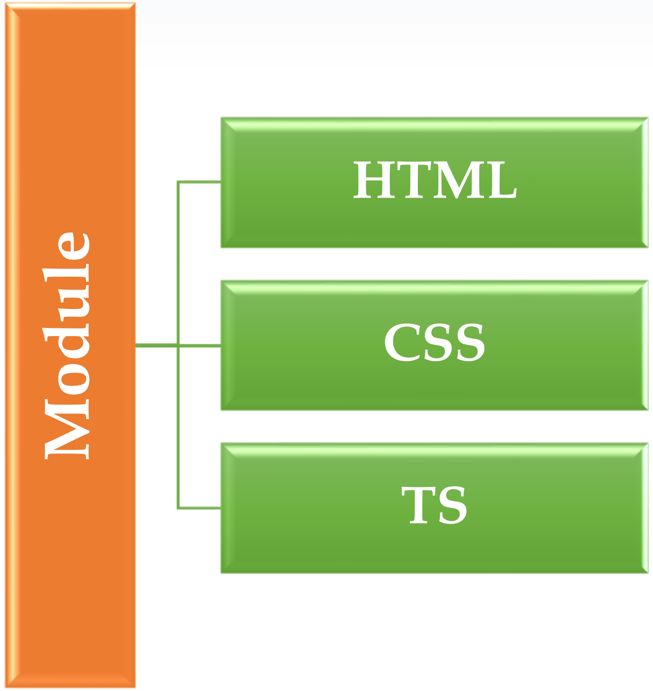
Run the Angular App

we can use either cmd or terminal (ctrl+`) in Visual Code to type the command: ng serve --open



Hello, newsimple

Congratulations! Your app is running. 🎉



Anatomy of an Angular App

node_modules – consists of node modules for angular projects. (Node packages).

src – source code for an app.

editor config – configuration file for visual code.

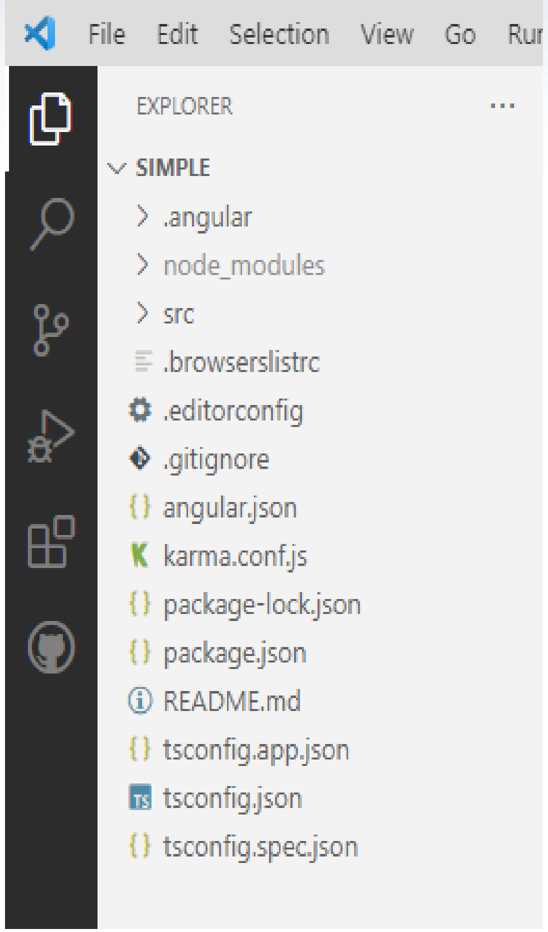
.gitignore – version control system.

angular.json – configuration for Angular CLI. (Version, Name, src folder, Prefix, etc)

package.json – defines the packages that angular project depends on.

tsconfig.json – contains configuration details for typescript compiler.

tslint.json – configuration for tslint. – code analytics tool that checks typescript code for readability, maintainability and functionality errors. It provides warnings to optimize the code. eg: unused variables in the declaration.



SIMPLE
- .angular
- node_modules
- src
- .browserslistrc
- .editorconfig
- .gitignore
- angular.json
- karma.conf.js
- package-lock.json
- package.json
- README.md
- tsconfig.app.json
- tsconfig.json
- tsconfig.spec.json

Installation & configure Bootstrap in Angular App

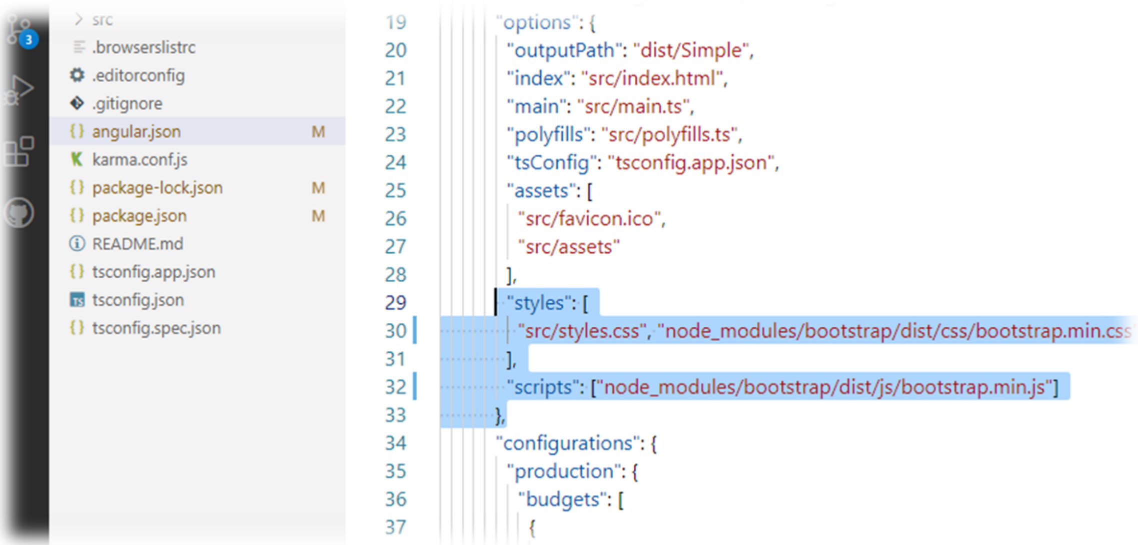
npm install bootstrap jquery --save

** Bootstrap depends on jQuery, so need to install jQuery also with this.*

Configure this bootstrap file in angular app

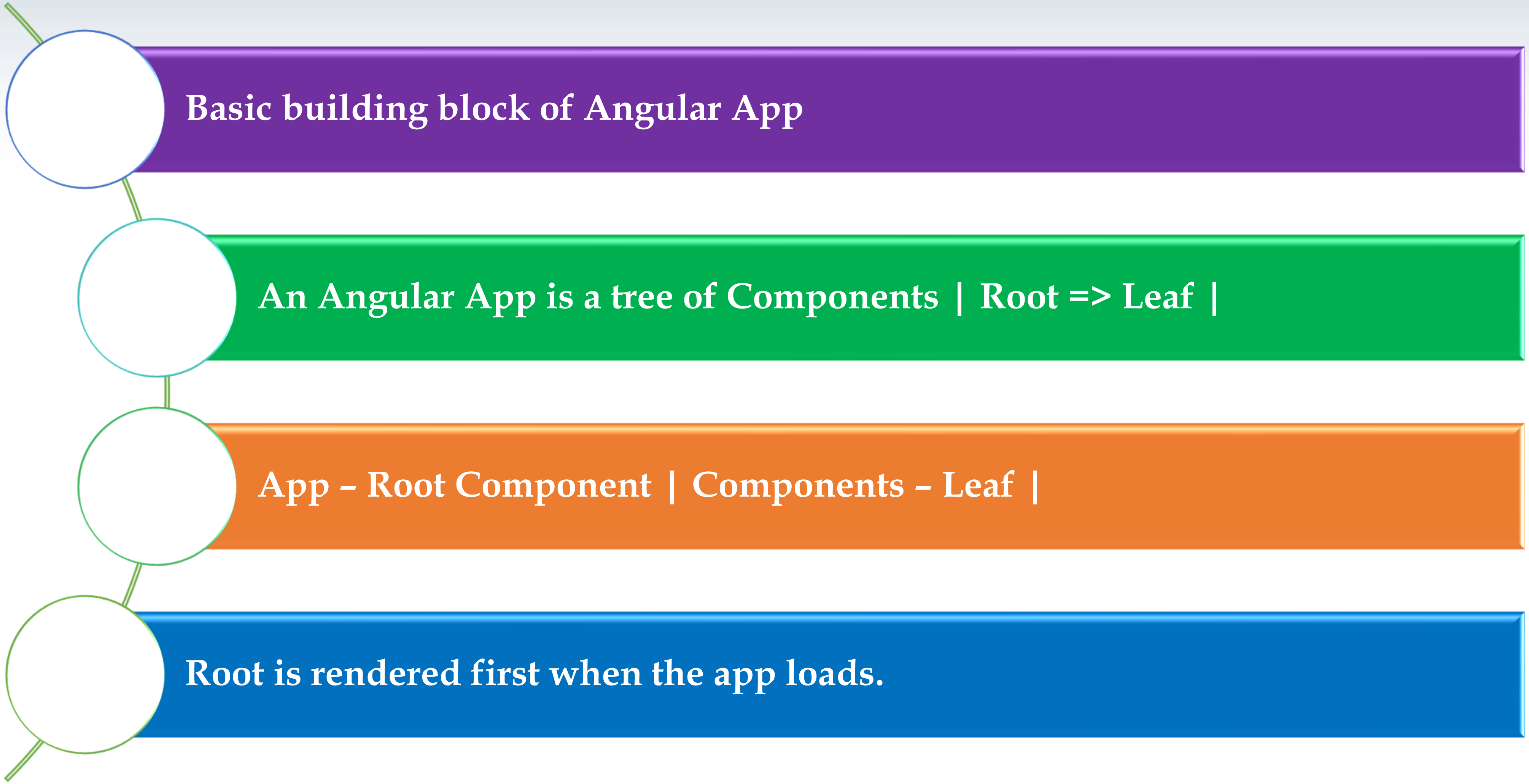
open angular.json file and add the following details.

```
   "styles": [

         "src/styles.css", "node_modules/bootstrap/dist/css/bootstrap.min.css"

               ],

  "scripts": ["node_modules/bootstrap/dist/js/bootstrap.min.js"]
```



```
      "options": {
        "outputPath": "dist/Simple",
        "index": "src/index.html",
        "main": "src/main.ts",
        "polyfills": "src/polyfills.ts",
        "tsConfig": "tsconfig.app.json",
        "assets": [
          "src/favicon.ico",
          "src/assets"
        ],
        "styles": [
          "src/styles.css", "node_modules/bootstrap/dist/css/bootstrap.min.css
        ],
        "scripts": ["node_modules/bootstrap/dist/js/bootstrap.min.js"]
      },
      "configurations": {
        "production": {
          "budgets": [
            {
```

Components



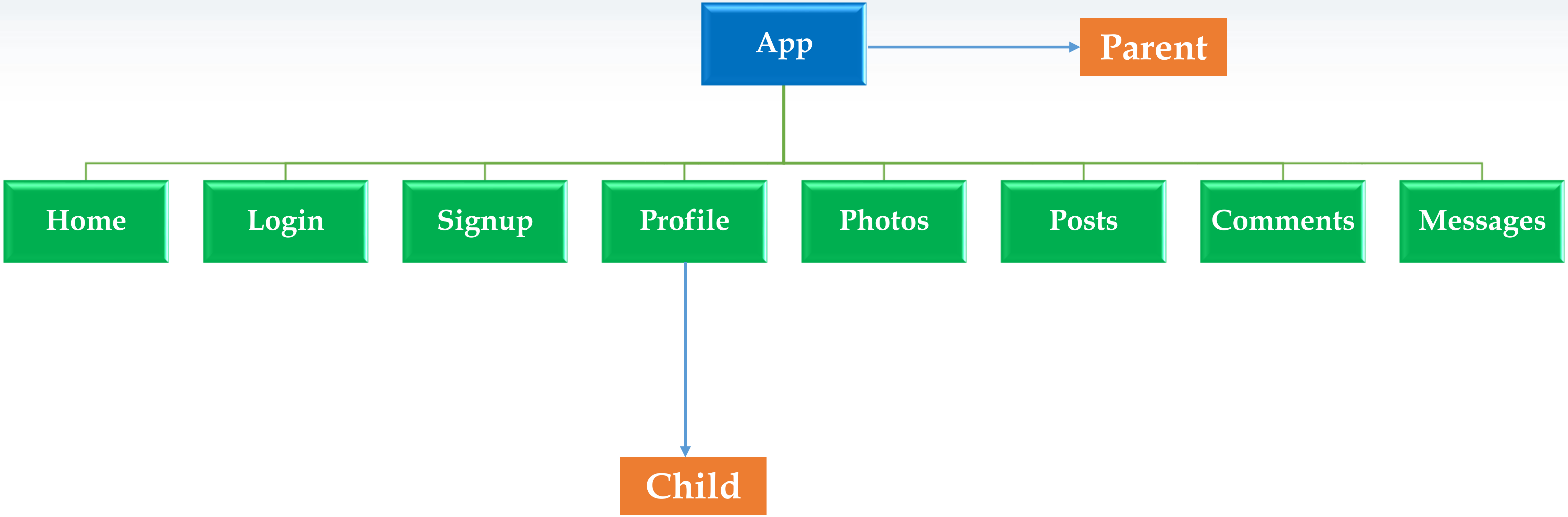
Basic building block of Angular App

An Angular App is a tree of Components | Root => Leaf |

App – Root Component | Components – Leaf |

Root is rendered first when the app loads.

Components - Example



Creating a Component

ng generate component home

* ***home -> Component Name***

- Delete home.component.spec.ts file (ie. used for testing)

home.component.html

```
<h1 align="center"><B>Welcome to All..!</B></h1>

<h6 align="center">Computer Applications<br/>

V..H.N.S.N.College (Autonomous), Virudhunagar.</h6>
```

app.component.ts

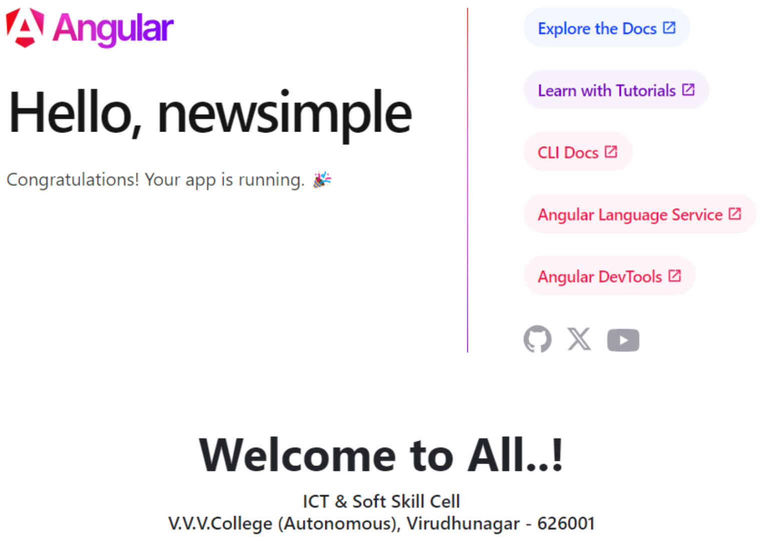
! - import { HomeComponent } from './home/home.component';

! - imports: [RouterOutlet, HomeComponent],

app.component.html

```
<main class="main">

…

</main>
```

! - <app-home></app-home>



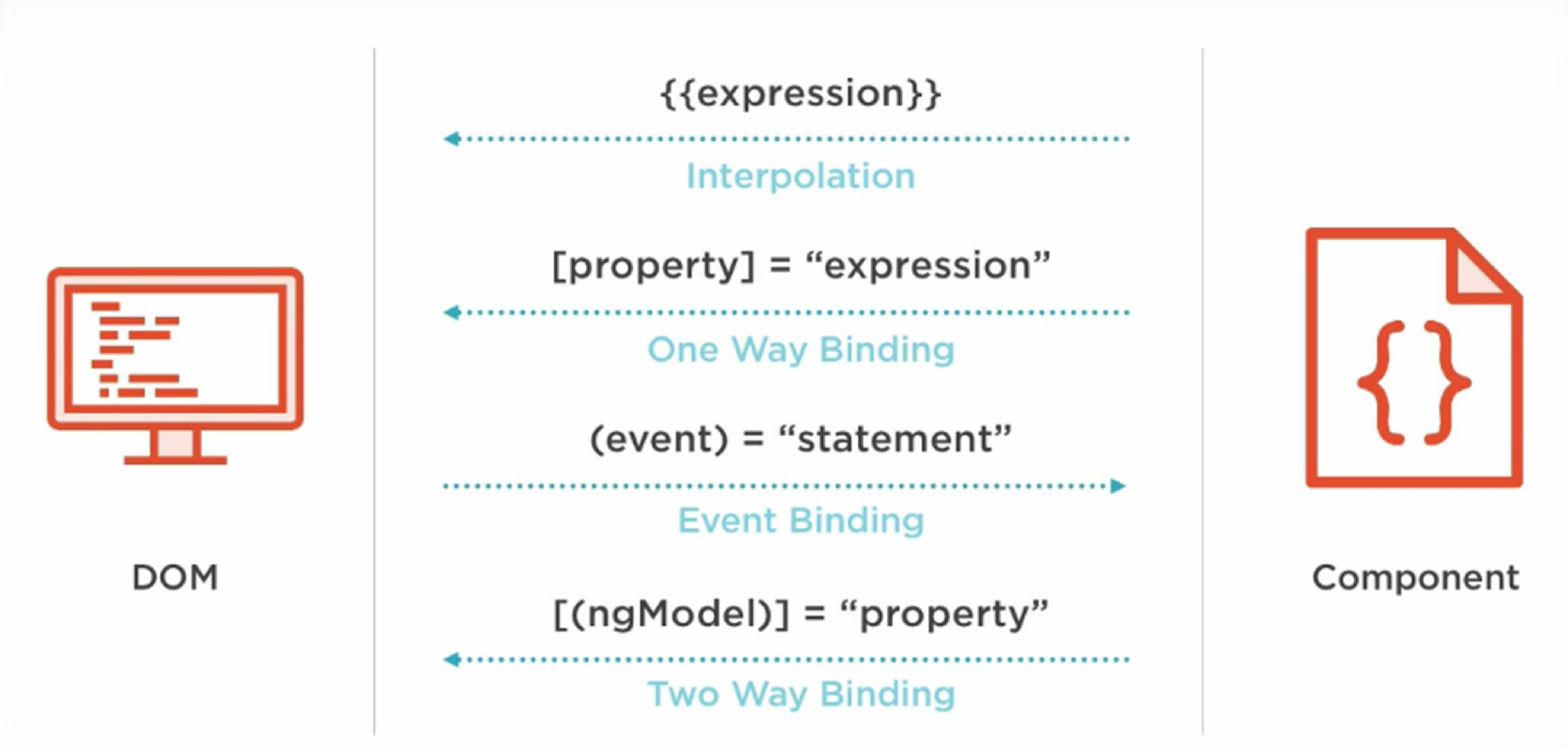
Hello, newsimple

Congratulations! Your app is running. 🎉

Welcome to All..!

ICT & Soft Skill Cell
V.V.V.College (Autonomous), Virudhunagar - 626001

Basics of Angular



DOM

{{expression}}

Interpolation

[property] = “expression”

One Way Binding

(event) = “statement”

Event Binding

[(ngModel)] = “property”

Two Way Binding

Component

Interpolation

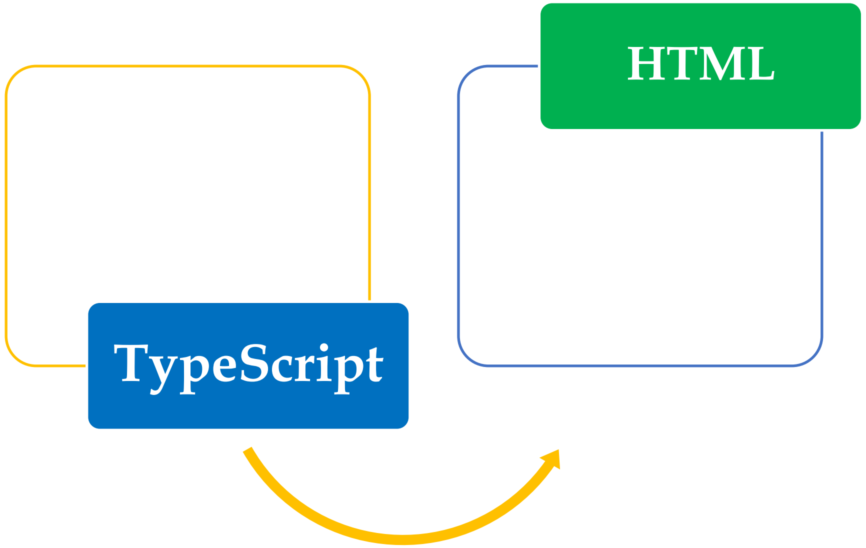
Way to share information between a ts class and HTML file of the component.

home.component.ts

```
export class HomeComponent implements OnInit {
 fs:string="";
 ngOnInit(): void {
  this.fs=“Computer Applications";
 }
```

home.component.html

```
<h1 align="center"><B>Welcome to {{fs}}</B></h1>
```



Property Binding

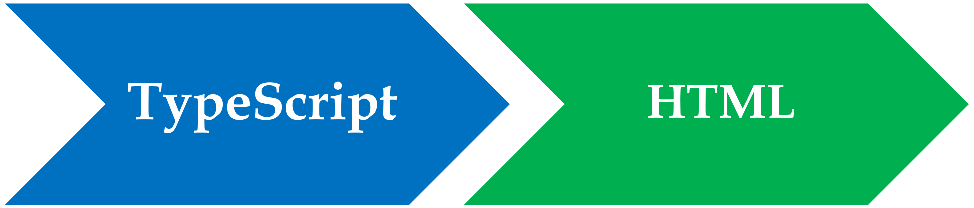
Allows us to **bind values to properties (Attribute) of an element (Tag)** to modify their behaviour or appearance.

home.component.ts

```
export class HomeComponent implements OnInit {
  fs:string="";
  g:string="";
  ngOnInit(): void {
    this.fs="ICT & Soft Skill Cell";
 ! -  this.g="http:\/\/\google.com"
  }
}
```

home.component.html

```
<a role="button" class="btn btn-warning" [href]="g"
target="new">Visit Google</a>
```



Event Binding

Binding HTML Events to functions is called Event Binding.

HTML

TypeScript

home.component.ts

```
alerthello()

{

   alert("Welcome to All..!");

}
```

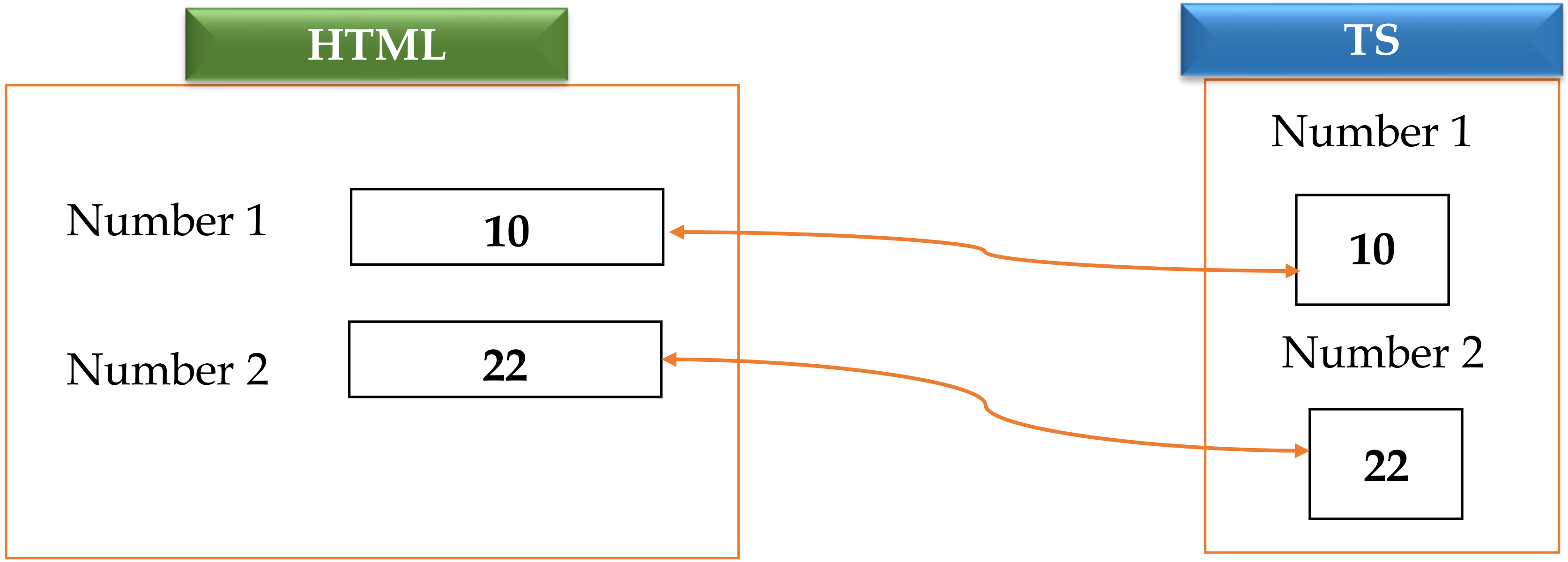
home.component.html

```
<button type="button" class="btn btn-success" (click)="alerthello()">Welcome</button>
```

Two way Binding

Allows the view (**HTML**) and class (**TS**) to maintain the data in sync. HTML ↔ TS

Interpolation + Property Binding + Event Binding = Two Way Binding



HTML

Number 1 10

Number 2 22

TS

Number 1

10

Number 2

22

Two way Binding

home.component.ts

```
! -  import { FormsModule } from '@angular/forms';
imports: [
    BrowserModule,
    AppRoutingModule,
! -    FormsModule
  ],
```

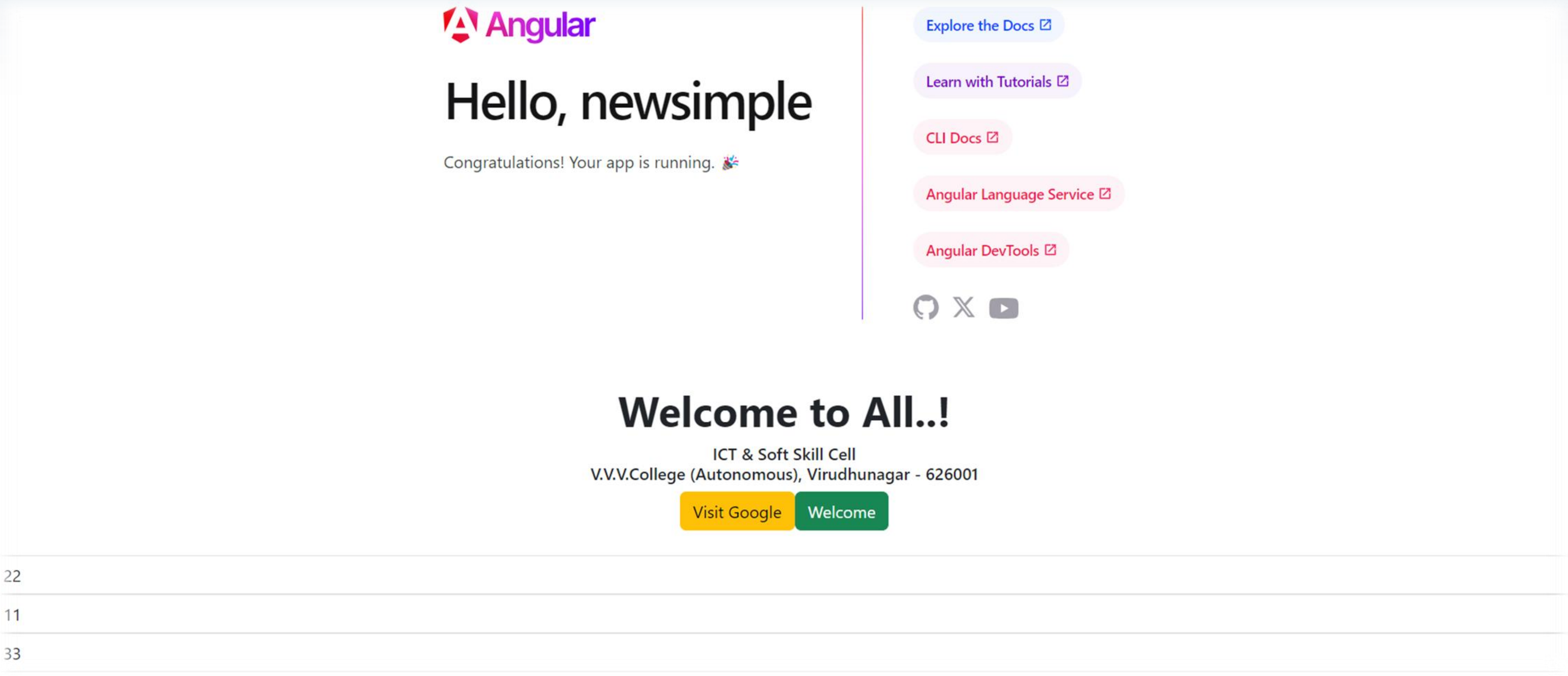
home.component.ts

```
n1:number=0;
n2:number=0;
sum:number=0;
add()
{
    this.sum=this.n1+this.n2;
}
```

home.component.html

```
<input type="number" class="form-control" placeholder="Number 1" [(ngModel)]="n1" (input)="add()">
<input type="number" class="form-control" placeholder="Number 2" [(ngModel)]="n2" (input)="add()">
<input type="number" class="form-control" placeholder="Result" [(ngModel)]="sum">
```

Two way Binding



Hello, newsimple

Congratulations! Your app is running. 🎉

Welcome to All..!

ICT & Soft Skill Cell
V.V.V.College (Autonomous), Virudhunagar - 626001

```
22
11
33
```

ngIf | Structural Directives

Directive which allows you to render components on the page based on a certain condition.

home.component.ts

```
!- import { CommonModule } from '@angular/common';

!- imports: [FormsModule, CommonModule],
```

home.component.ts

```
!- sgb:boolean=false;

this.g="http:\/\/\google.com"
```

home.component.html

```
<a role="button" class="btn btn-warning" [href]="g" *ngIf="sgb" target="new">Visit Google</a>
```

ngFor | Structural Directives

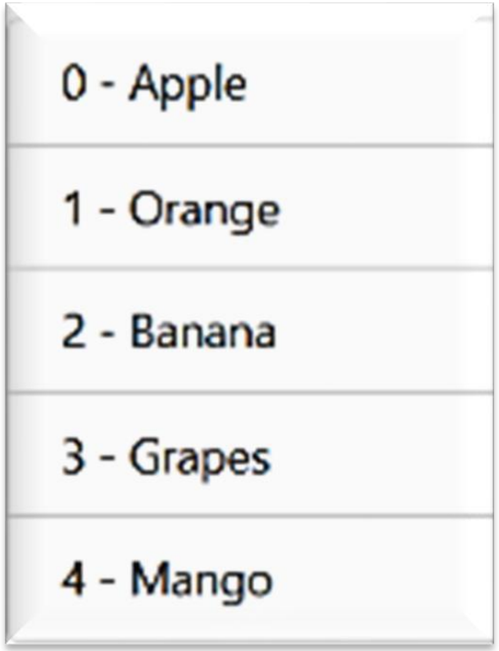
Special Directive that helps us to use arrays in angular to render components in page.

home.component.ts

```
fruits:string[]=[];

constructor() {

   this.fruits.push("Apple");

   this.fruits.push("Orange");

   this.fruits.push("Banana");

   this.fruits.push("Grapes");

   this.fruits.push("Mango");

 }
```

home.component.html

```
<ul class="list-group">

   <li class="list-group-item" *ngFor="let fruit of fruits let i=index">{{i}} - {{fruit}}</li>

</ul>
```



```
0 - Apple
1 - Orange
2 - Banana
3 - Grapes
4 - Mango
```

ngSwitch | Structural Directives

It specifies an expression to match against. The ngSwitchCase directive defines the expressions to match.

- It renders every view that matches.
- If there are no matches, the view with the ngSwitchDefault directive is rendered.

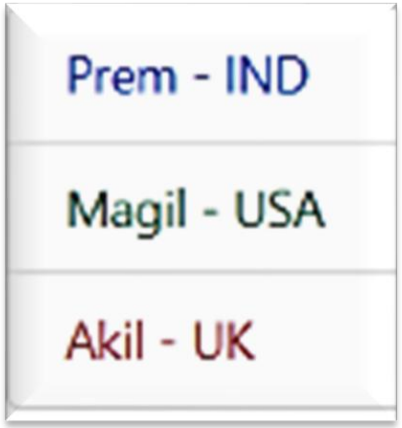
home.component.ts

```
people:any[]=[{}];
constructor() {
   this.people=[{
     "name":"Prem",
     "age":31,
     "country":"IND"
   },
   {
     "name":"Magil",
     "age":1,
     "country":"USA"
   },{
     "name":"Akil",
     "age":7,
     "country":"UK"
   }]
}
```

ngSwitch | Structural Directives

home.component.html

```
<ul class="list-group">
   <li class="list-group-item" *ngFor="let person of people" [ngSwitch]="person.country">
<div *ngSwitchCase="'IND'" class="text-primary">{{person.name}} – {{person.country}}  </div>
<div *ngSwitchCase="'USA'" class="text-success">{{person.name}} - {{person.country}}  </div>
<div *ngSwitchDefault class="text-danger">{{person.name}} - {{person.country}}</div>
   </li>
</ul>
```



Prem - IND

Magil - USA

Akil - UK

Reactive Forms

- ✓ Larger forms are easier to manage
- ✓ Validations are easier to implement
- ✓ More Flexible

Create New Component for Login form : ng generate component login

login.component.ts

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { FormBuilder, FormGroup, FormControl, Validators } from '@angular/forms';
imports: [
   BrowserModule,
   AppRoutingModule,
   FormsModule,
   ReactiveFormsModule
 ],
```

Angular Reactive Forms

Reactive Forms

login.component.ts

```
import { FormBuilder, FormGroup, FormControl, Validators } from '@angular/forms';
myForm:FormGroup;
 email:string="";
 pwd:string="";
constructor(public fb: FormBuilder) {
    this.myForm=this.fb.group({
   email:['',[Validators.email,Validators.required]],
   pwd:['',[Validators.minLength(8),Validators.required]],
  })
 }
```

```
onSubmit(form: FormGroup)
{
   alert("Login checking");
}
```

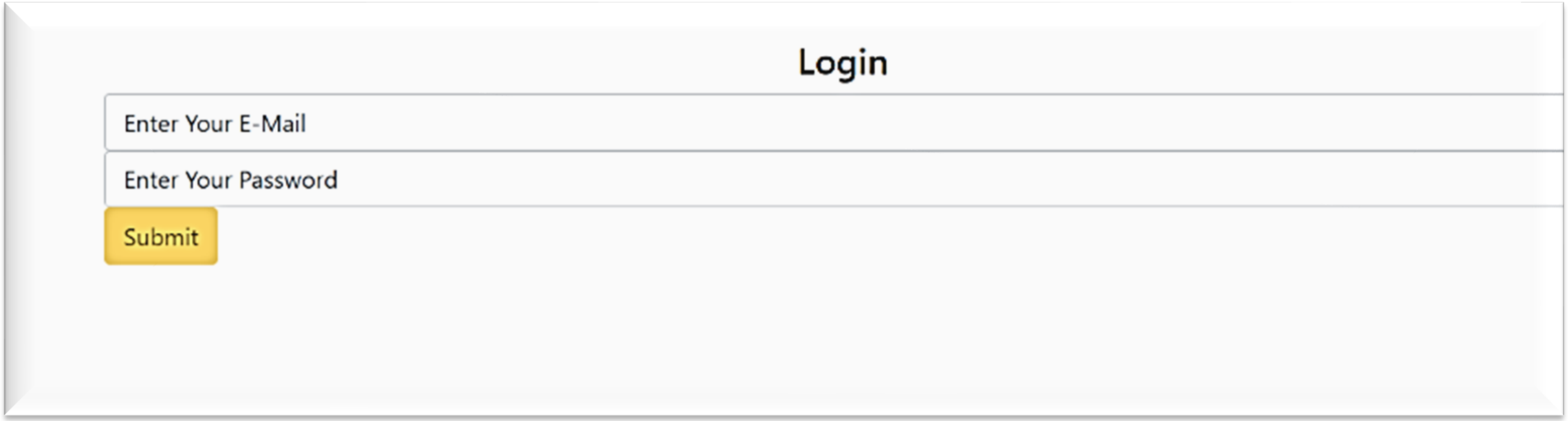
Angular Reactive Forms

Reactive Forms

```
<div class="row">
  <div class="offset-sm-2 col-sm-8">
    <div class="card text-blank border-0">
      <div class="card-body">
        <div class="card-title text-center">
          <h4>Login</h4>
        </div>
        <div class="card-text">
          <form [formGroup]="myForm" (ngSubmit)="onSubmit(myForm)">
            <div class="form-group">
         <input type="email" placeholder="Enter Your E-Mail" class="form-control"
    [formControl]="$any(myForm.controls['email'])">
            </div>
<div class="alert alert-danger" *ngIf="(!myForm.controls['email'].valid) && (myForm.controls['email'].touched)">E-Mail ID
Required.</div>
```

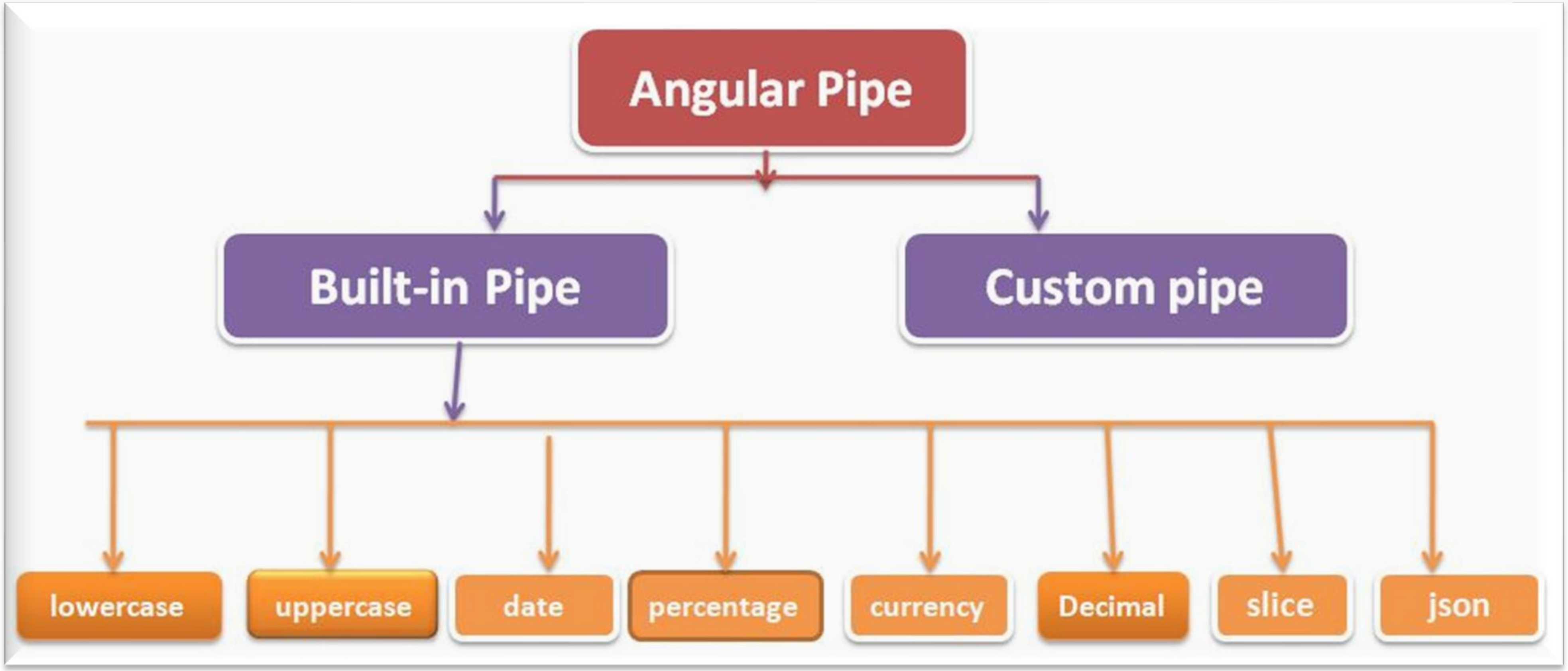
Reactive Forms

```
 <div class="form-group">
  <input type="password" placeholder="Enter Your Password" class="form-control“
                                                              [formControl]="$any(myForm.controls['pwd'])">
  </div>
<div class="alert alert-danger" *ngIf="(!myForm.controls['pwd'].valid) && (myForm.controls['pwd'].touched)">
                                                                              Password Required.</div>
     <button type="submit" class="btn btn-block btn-warning" [disabled]="!myForm.valid"> Submit </button>
     </form>
     </div>
      </div>
    </div>
</div>
</div>
```



Pipes

Pipes are functions that are used to format the data before displaying it to the user.



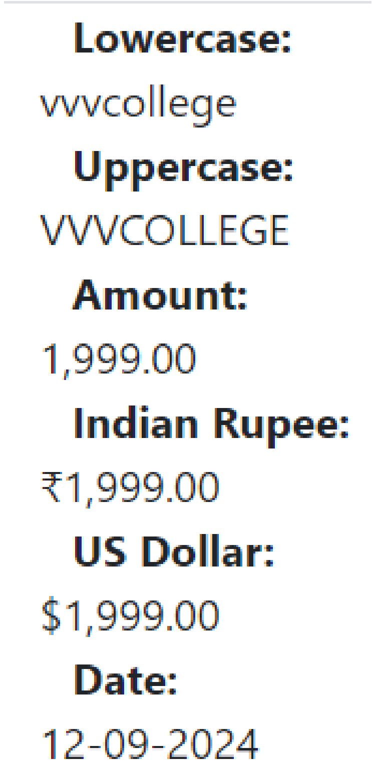
Pipes

home.component.html

```
<div class="container">
    <div class="row">
        <b>Lowercase:</b> {{'VVVCollege' | lowercase}} <br />
        <b>Uppercase:</b> {{ pname | uppercase}} <br />
        <b>Amount:</b> {{ amt | number:'0.2'}} <br />
        <b>Indian Rupee:</b> {{ amt | currency : 'INR'}} <br />
        <b>US Dollar:</b> {{ amt | currency : 'USD'}}
        <b>Date:</b>{{dob | date: ‘dd-MM-yyyy'}}
    </div>
</div>
```

home.component.ts

```
pname:string="vvvcollege";
amt:number=1999;
dob=new Date();
```



```
Lowercase:
vvvcollege
Uppercase:
VVVCOLLEGE
Amount:
1,999.00
Indian Rupee:
₹1,999.00
US Dollar:
$1,999.00
Date:
12-09-2024
```

Routing with Router Module

- Route is just a set of path & their corresponding components.
- It is used for navigating between the components.
- Angular Router checks the path in the browser based on it navigates and replaces the destination component.

app.route.ts

```
const routes: Routes = [

{ path:'',component: AppComponent},

{ path:'home',component: HomeComponent},

{ path:'login',component: LoginComponent}

];
```

** Routes is a JSON Array where each object in the array defined a route.*

Routing with Router Module

app.component.html

```
<a class="nav-link" routerLink="/login"> | Login | </a>

<a class="nav-link" routerLink="/home"> | Home | </a>
```

app.component.ts

```
!- imports: [HomeComponent, LoginComponent, RouterModule],
```