

Introduction to Angular

Week 5 Discussion

Angular

- Angular is an open source JavaScript framework that is used to build single page based web applications.
- Developed by Google
- Release Date - March 2017
- One framework. Mobile & Desktop.

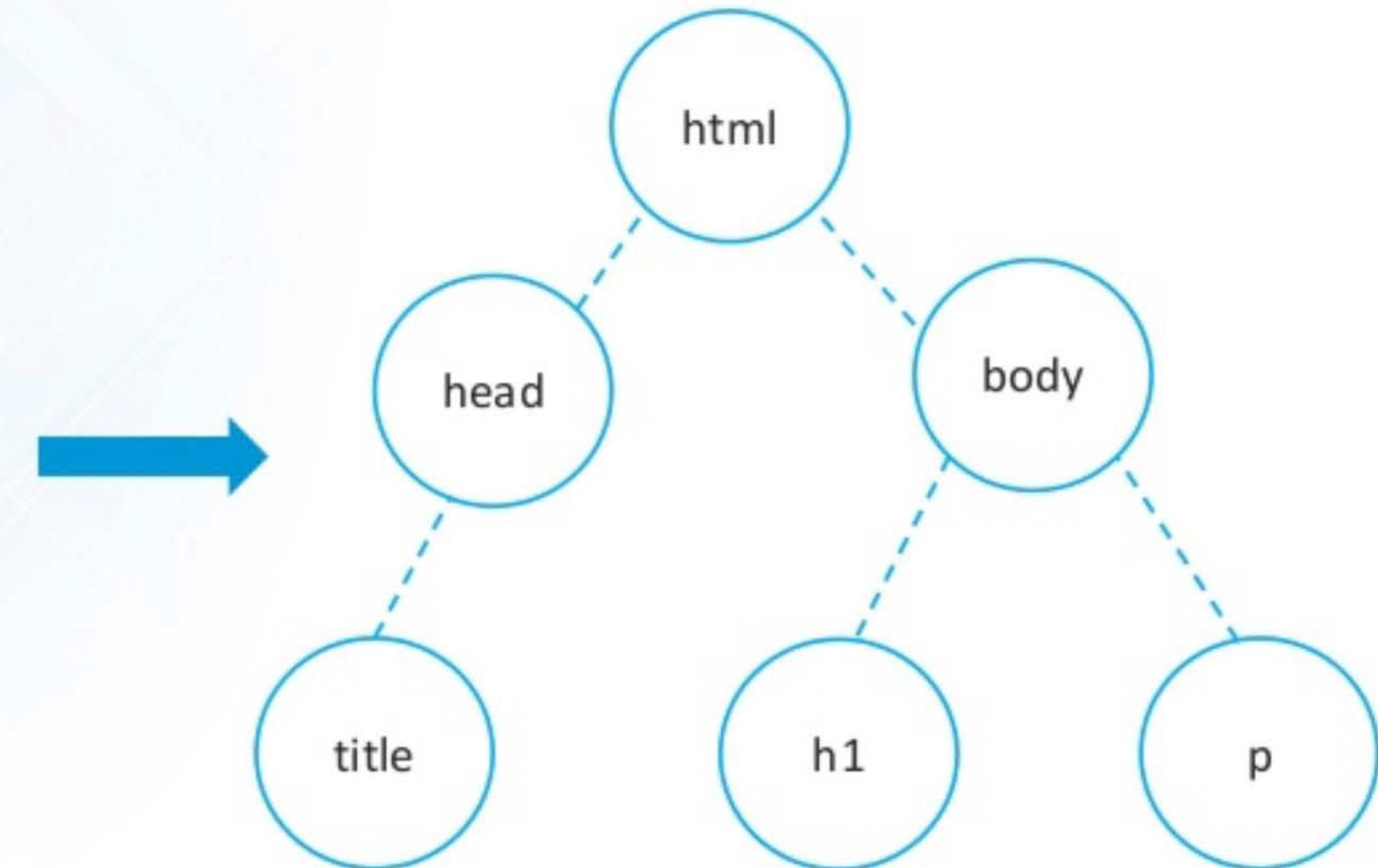
Revisit the traditional Web Dev ...

Webpage and DOM

Webpage and DOM

```
<html>
<head>
<title> Angular 2 Tutorial </title>
</head>
<body>
  <h1> Welcome to Angular 2 Tutorial </h1>
  <p>Angular is a development platform for creating
    applications using modern web standards.</p>
</body>
</html>
```

HTML Markup

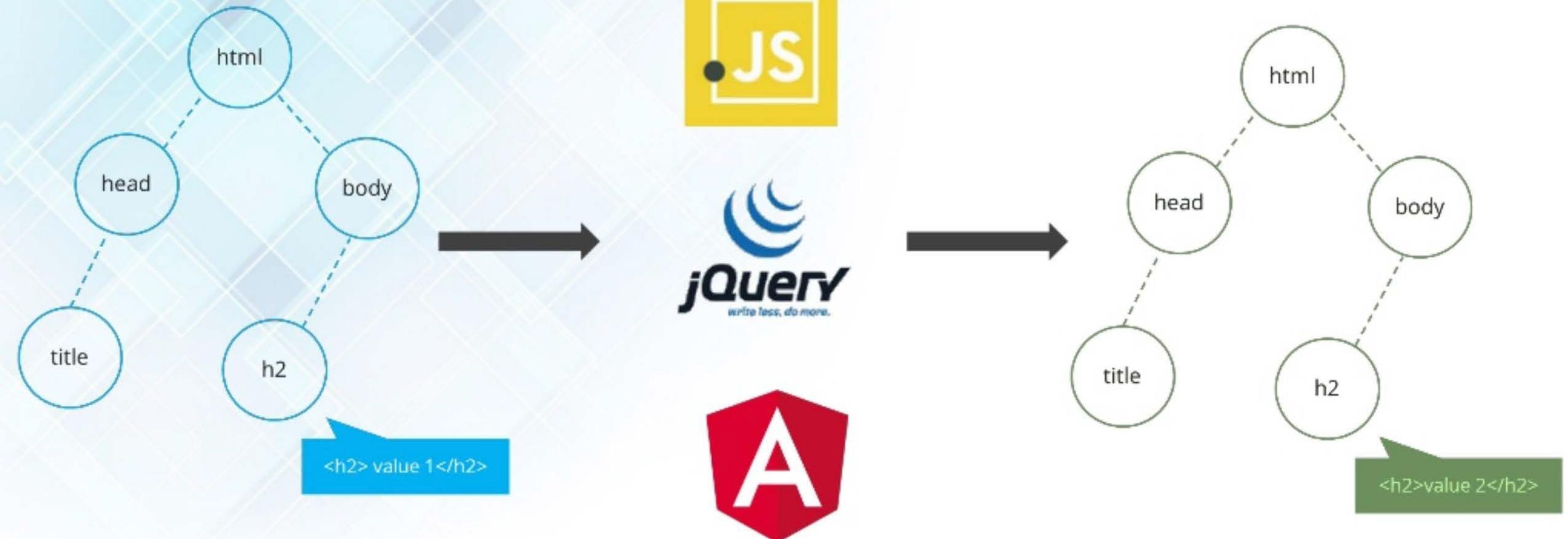


DOM Tree of the HTML document

DOM Manipulation

DOM Manipulation

edureka!



JavaScript

JavaScript & jQuery

edureka!



- JavaScript is a programming language designed for use in a web browser.
- Used for manipulating DOM
- Example:

Document.body.style.background = red;



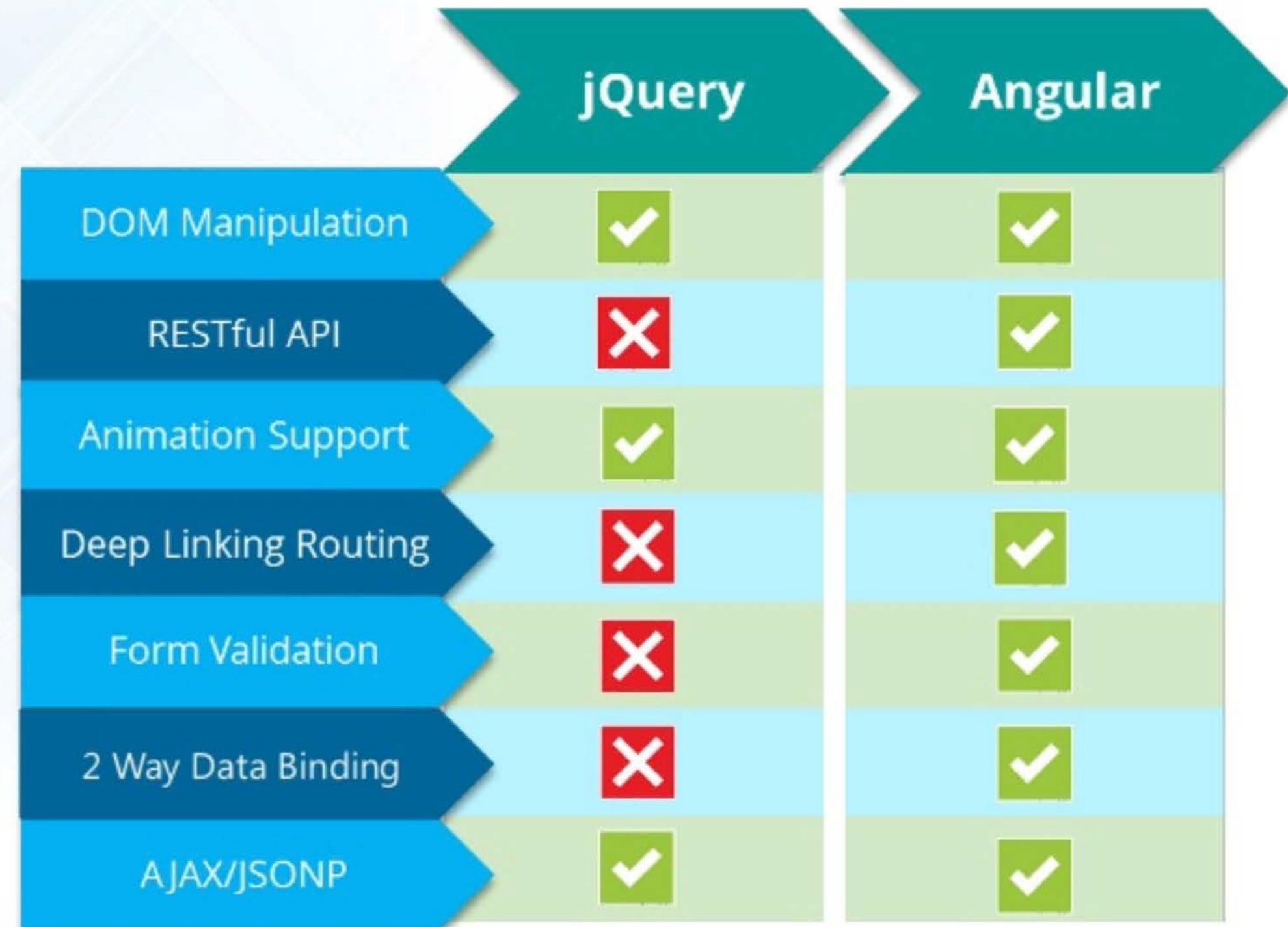
- jQuery is a library built in JavaScript to automate and simplify common tasks.
- Used for manipulating DOM
- Example:

\$(‘body’).css(‘background’, ‘#ccc’);

Why Angular?

Why Angular?

edureka!

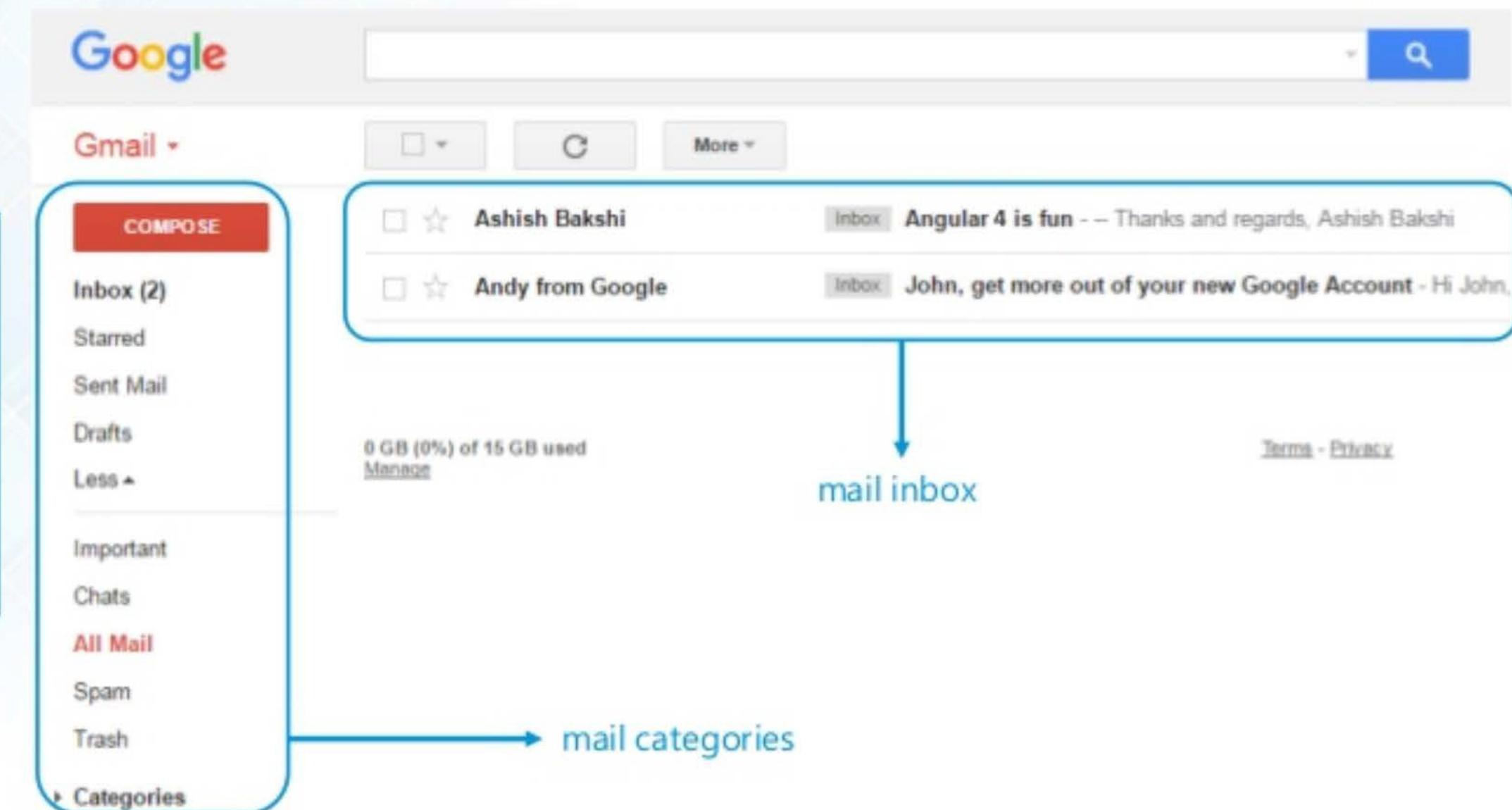


What is a Single Page Application?

What is SPA?

A Single Page Application is a web application that requires only a single page load in a web browser.

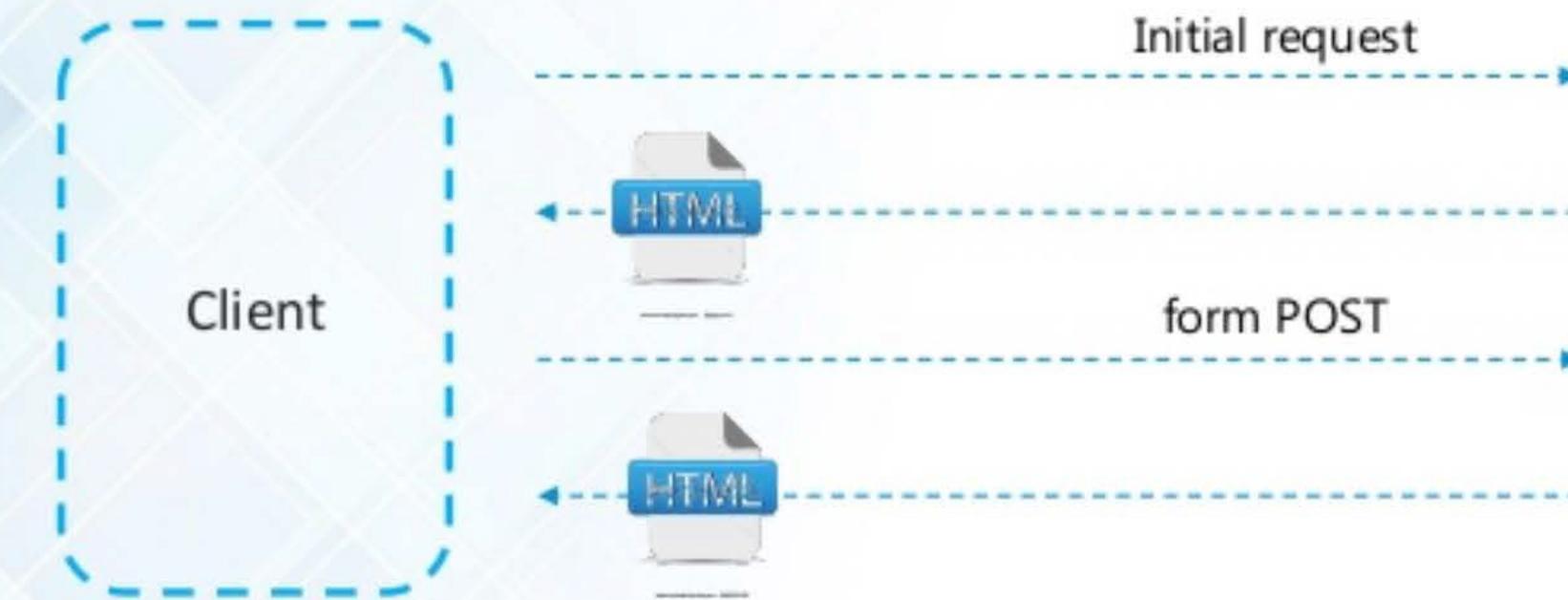
- Whole page is not reloaded every time
- Your browser fully renders the DOM once
- Later any server interactions is performed by JavaScript which modifies the view



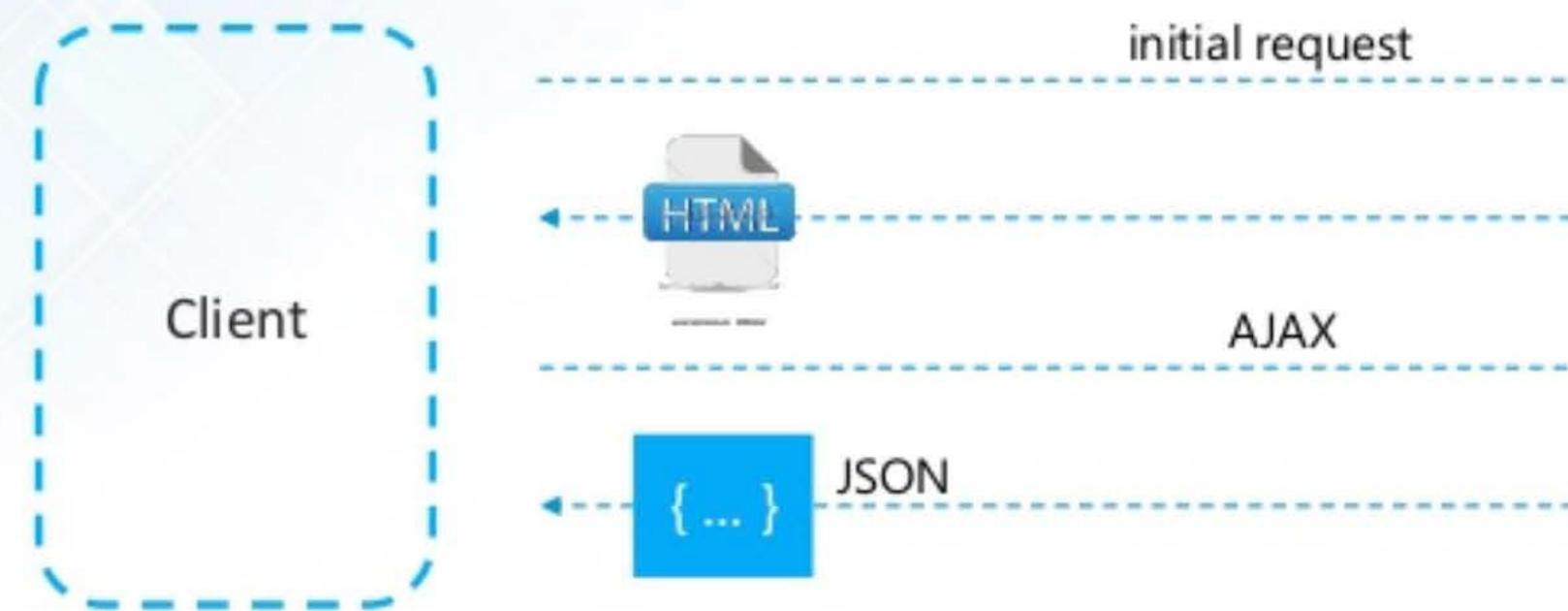
Traditional Way Vs Single Page Application

edureka!

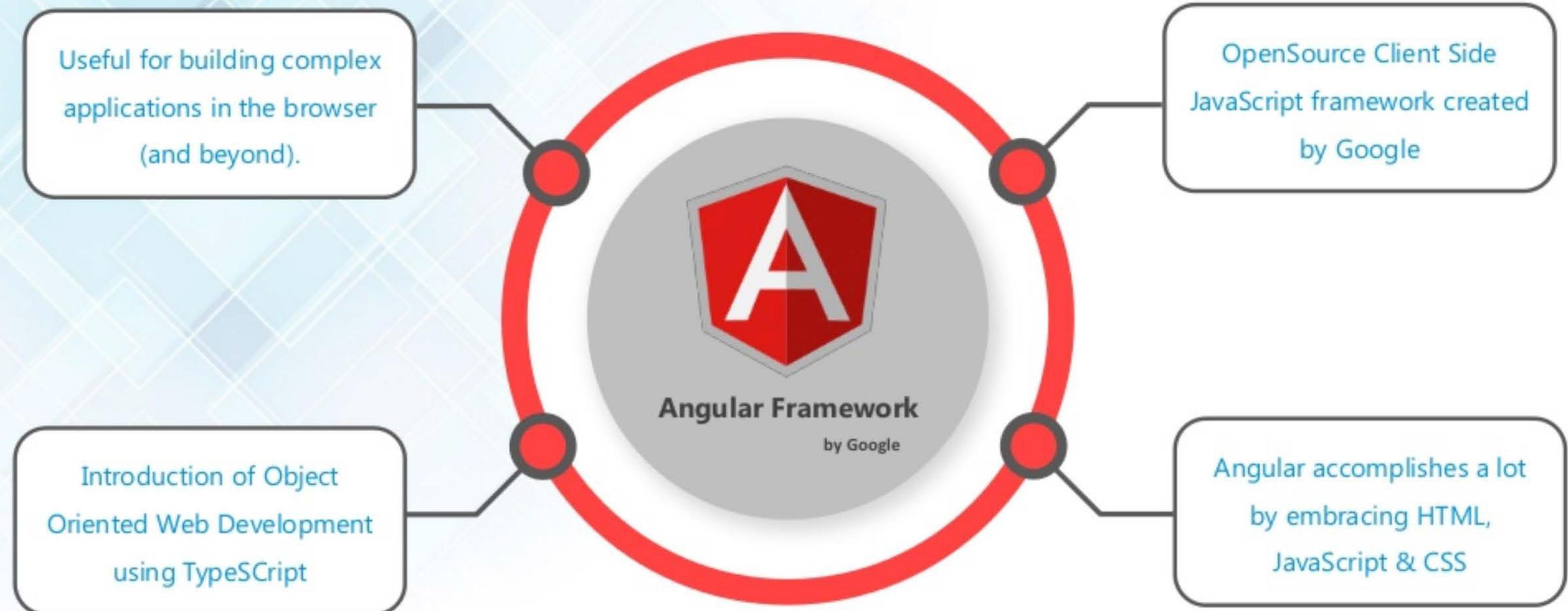
Traditional Way
Life Cycle



Single Page Application
Life Cycle

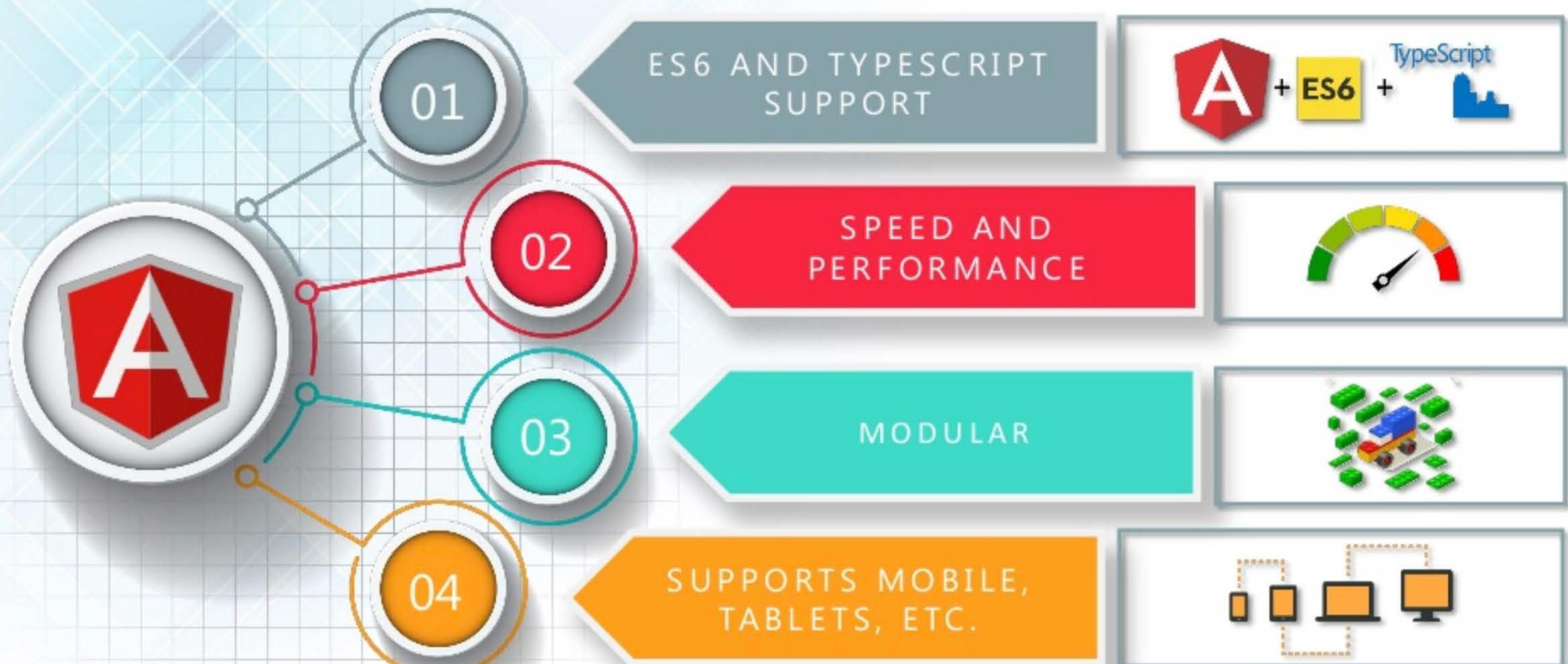


Angular Introduction

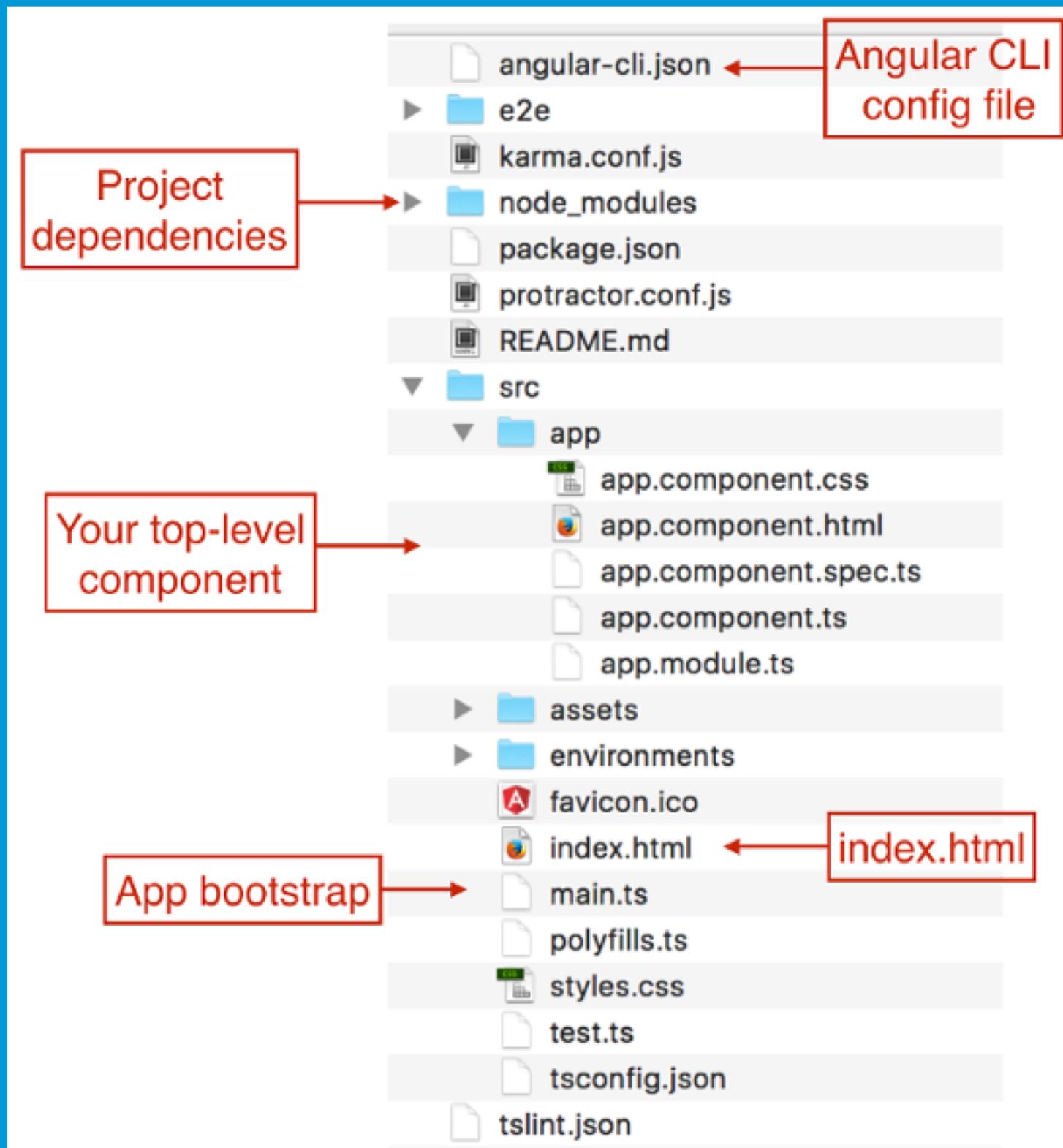


Angular Features

Angular Features



Angular project Structure



Application Starting Order

main.ts

- Main.ts file is entry point of our application.
- Main.ts file bootstrap app.module.ts file.

app.module.ts

- This file bootstraps app.component.ts.

app.component.ts

- It renders app.component.html.

app.component.html

- Final HTML template

Building Blocks of Angular

Module

Component

Metadata

Template

Data Binding

Services

Directives

Building Blocks of Angular

Module

Module is a class with
@NgModule metadata

Every Angular app has at
least one root module

Encapsulation of different
similar functionalities

Component

Metadata

Template

Data Binding

Services

Directives

*Similar
Functionalities*

Components

Directives

⋮

Pipes

export

Single Module

Building Blocks of Angular

edureka!

Module

Component

Metadata

Template

Data Binding

Services

Directives

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/http';

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

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

Decorator

Declaring all the components

Importing Modules

Provide Services to all
module's component

Building Blocks of Angular

edureka!

Module

Component

Metadata

Template

Data Binding

Services

Directives



→ Nav Bar

→ News Feed

Building Blocks of Angular

edureka!

Module

Component

Metadata

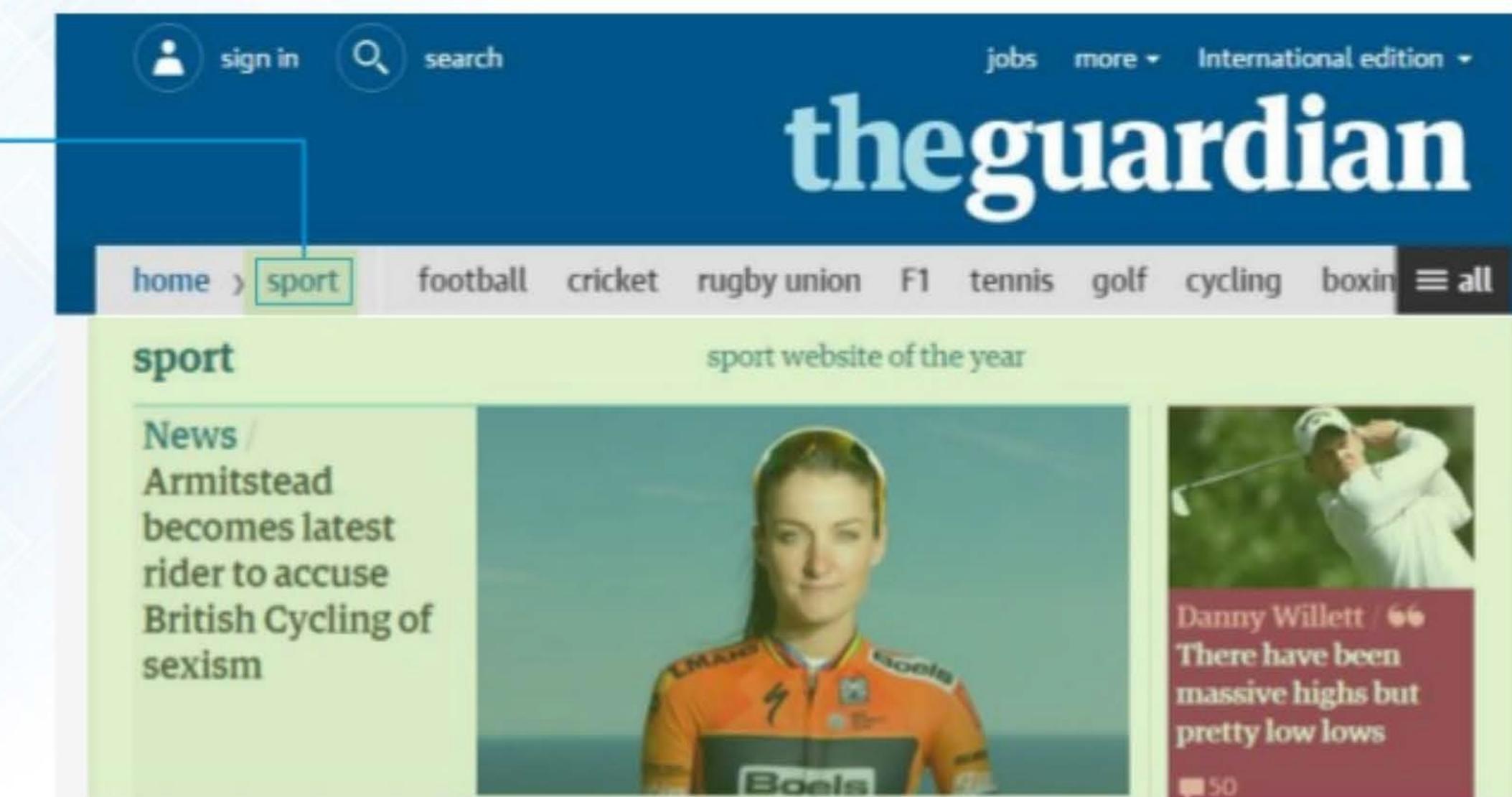
Template

Data Binding

Services

Directives

Sports
Component



Building Blocks of Angular

Module

Component

Metadata

Template

Data Binding

Services

Directives

```
import { Component, OnInit } from '@angular/core';  
  
@Component({  
  selector: 'app-example',  
  templateUrl: './example.component.html',  
  styleUrls: ['./example.component.css']  
})  
  
export class ExampleComponent implements OnInit {  
  
  constructor() {}  
  
  ngOnInit() {}  
  
}
```

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

Decorator → `@Component({`

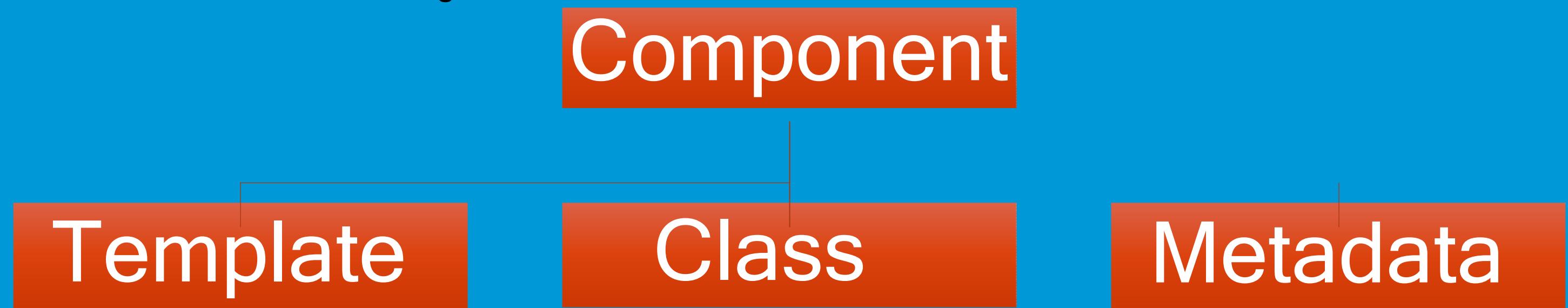
Meta Data → `selector: 'app-example',`
`templateUrl: './example.component.html',`
`styleUrls: ['./example.component.css']`

Exporting Component Class → `export class ExampleComponent implements OnInit {`

Components

- ❖ Components are a logical piece of code for Angular application.

A Component consists of the following –



- ✓ Template is used to render the view for the application.
- ✓ This contains the HTML that needs to be rendered in the application.

- ✓ This is like a class defined in any language such as C#.
- ✓ This has the code which is used to support the view.
- ✓ It is defined in TypeScript.

- ✓ This has the extra data defined for the Angular class.
- ✓ It is defined with a decorator.

Building Blocks of Angular

Module

Component

Metadata

Template

Data Binding

Services

Directives

Metadata describes how
to process the class

Decorator is used to
attach metadata

Example:



@Component({
.....
})

Decorator



Component
{ }



@NgModule({
.....
})

Decorator



Module
{ }

Building Blocks of Angular

edureka!

Module

Component

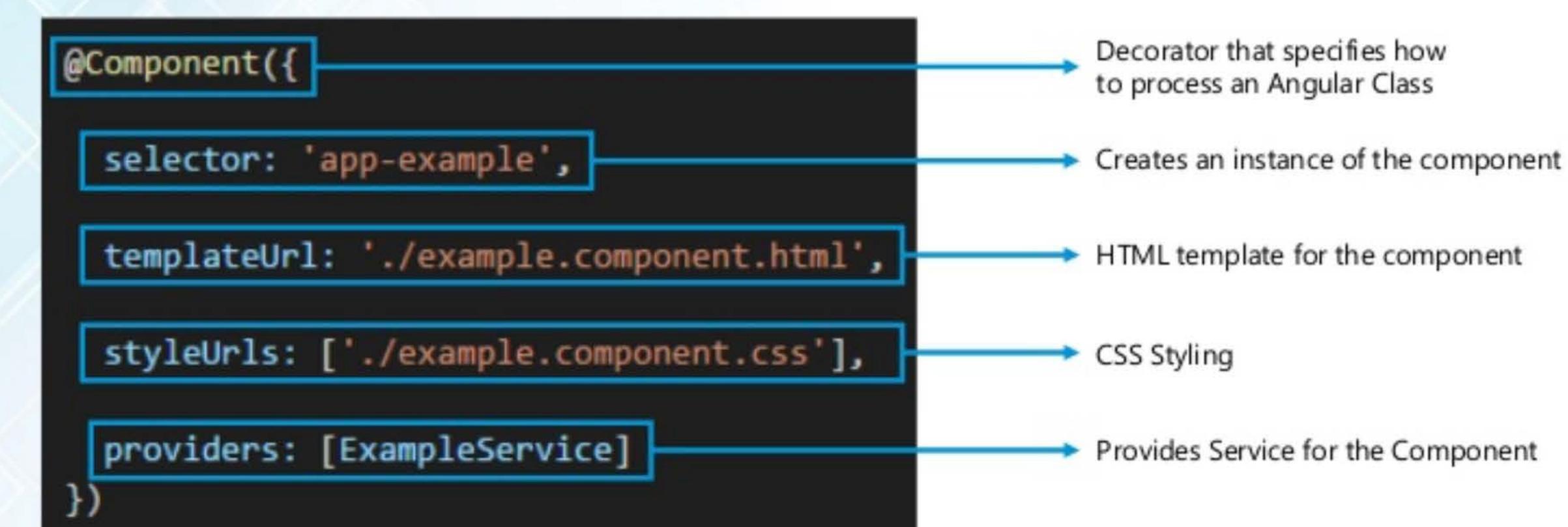
Metadata

Template

Data Binding

Services

Directives



Building Blocks of Angular

Module

Component

Metadata

Template

Data Binding

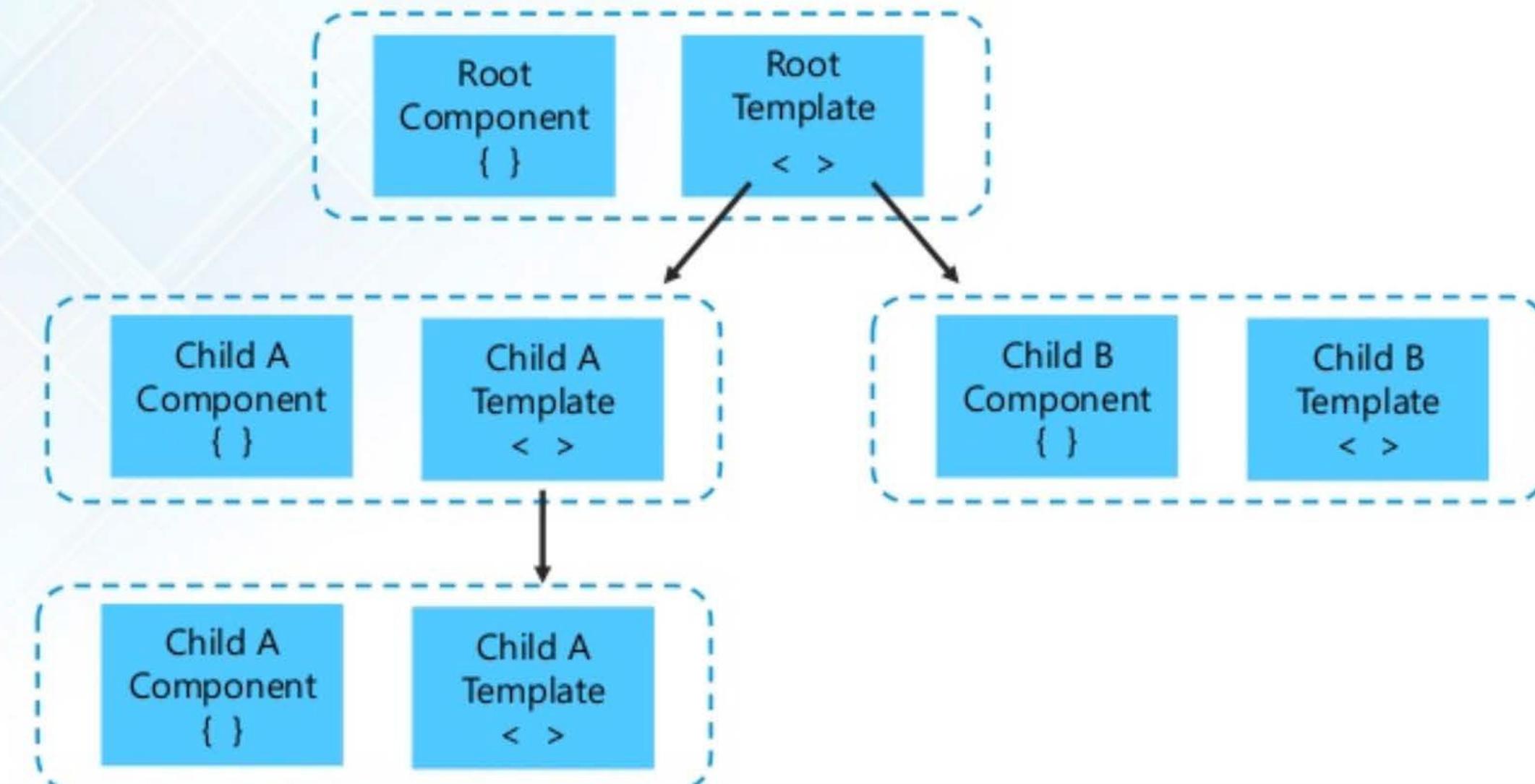
Services

Directives

Used to define view of a component

Looks like HTML, except for a few differences.

Describes how the component is rendered on the page.



Building Blocks of Angular

edureka!

Module

Component

Metadata

Template

Data Binding

Services

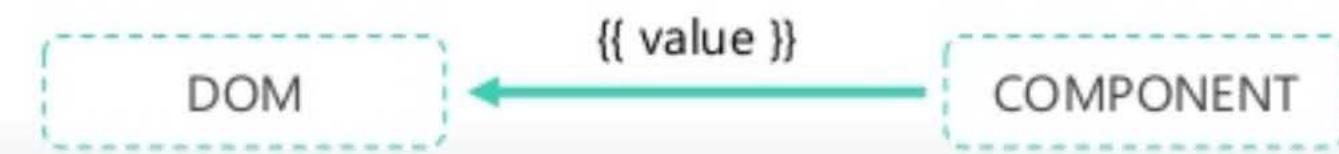
Directives

TYPES OF DATA BINDING

Data binding plays an important role in communication between a template and its component

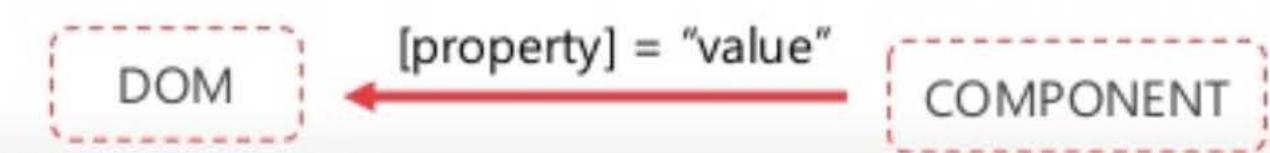
INTERPOLATION

01



PROPERTY BINDING

02



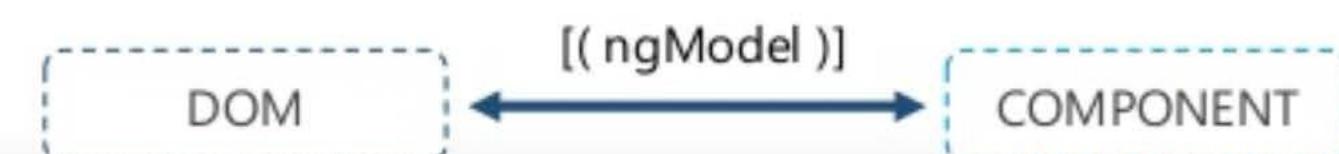
EVENT BINDING

03

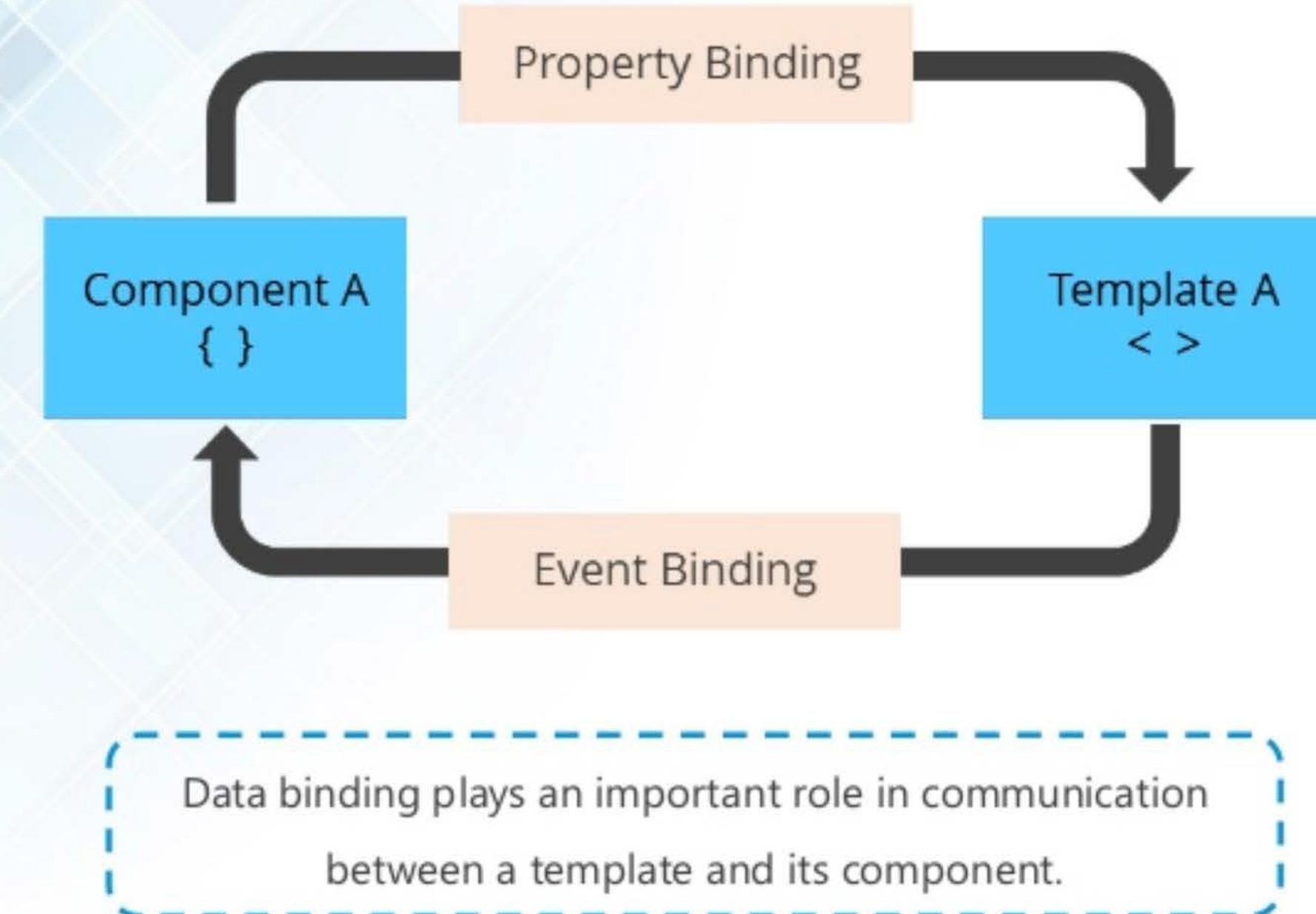


2 WAY DATA BINDING

04



Building Blocks of Angular



Building Blocks of Angular

edureka!

Module

Component

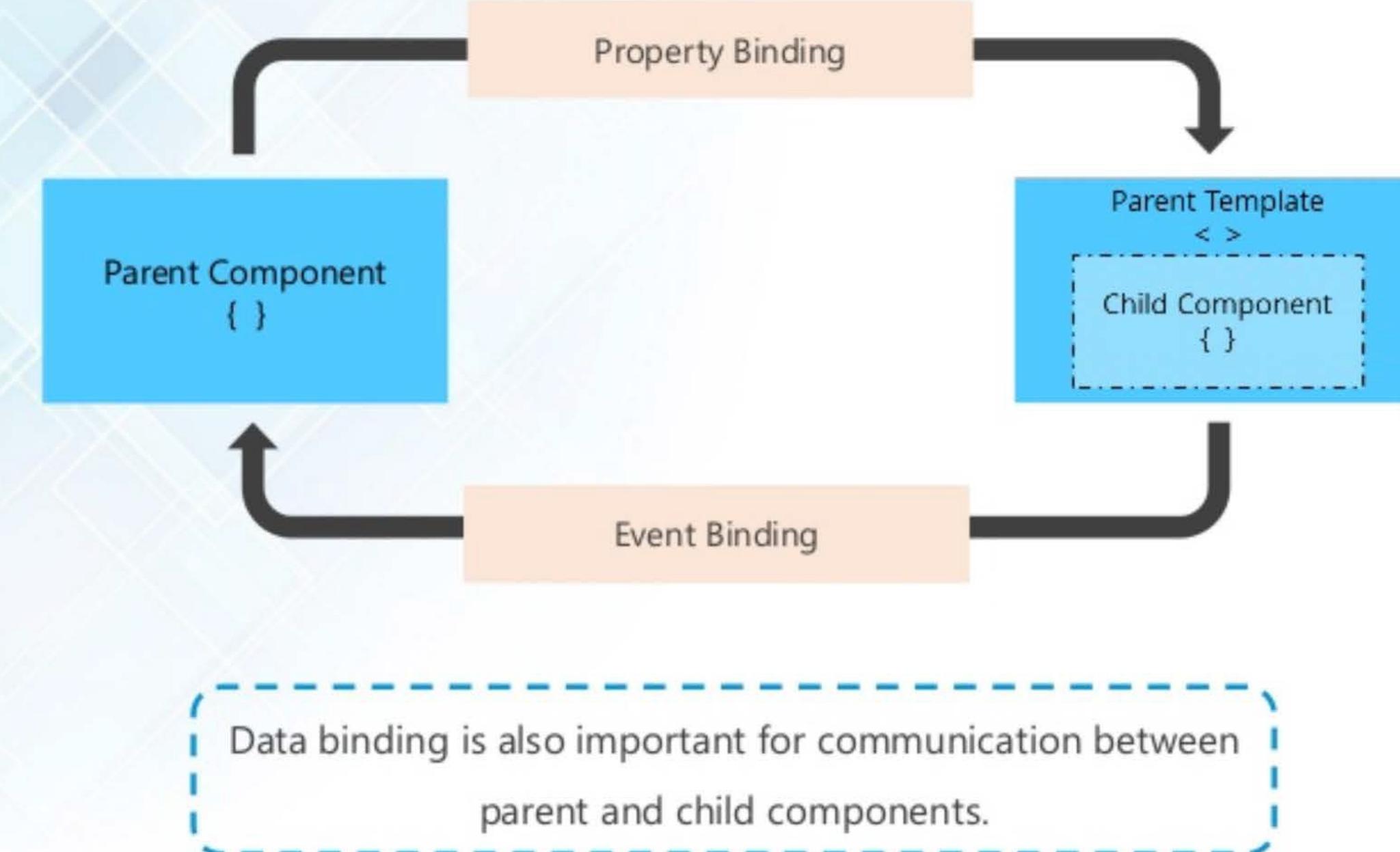
Metadata

Template

Data Binding

Services

Directives



Building Blocks of Angular

edureka!

Module

Component

Metadata

Template

Data Binding

Services

Directives

Service is a broad category encompassing any value, function, or feature that your application needs.

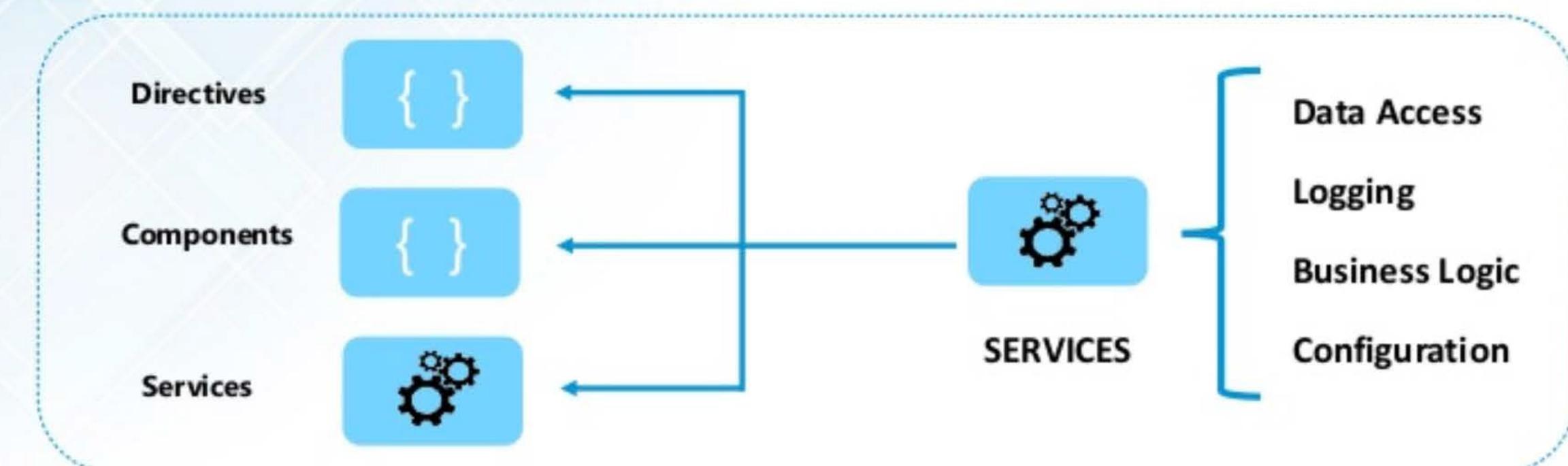
Example:

logging service

data service

message bus

tax calculator



Building Blocks of Angular

Module

Component

Metadata

Template

Data Binding

Services

Directives

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

@Injectable()
export class ExampleService {

  movies: string[] = ["Inception", "Dark Knight", "Shutter Island"];

  constructor() { }

  getMovies(): string[]
  {
    return this.movies;
  }
}
```

Service Class

Service Method for
retrieving data

Building Blocks of Angular

Module

Component

Metadata

Template

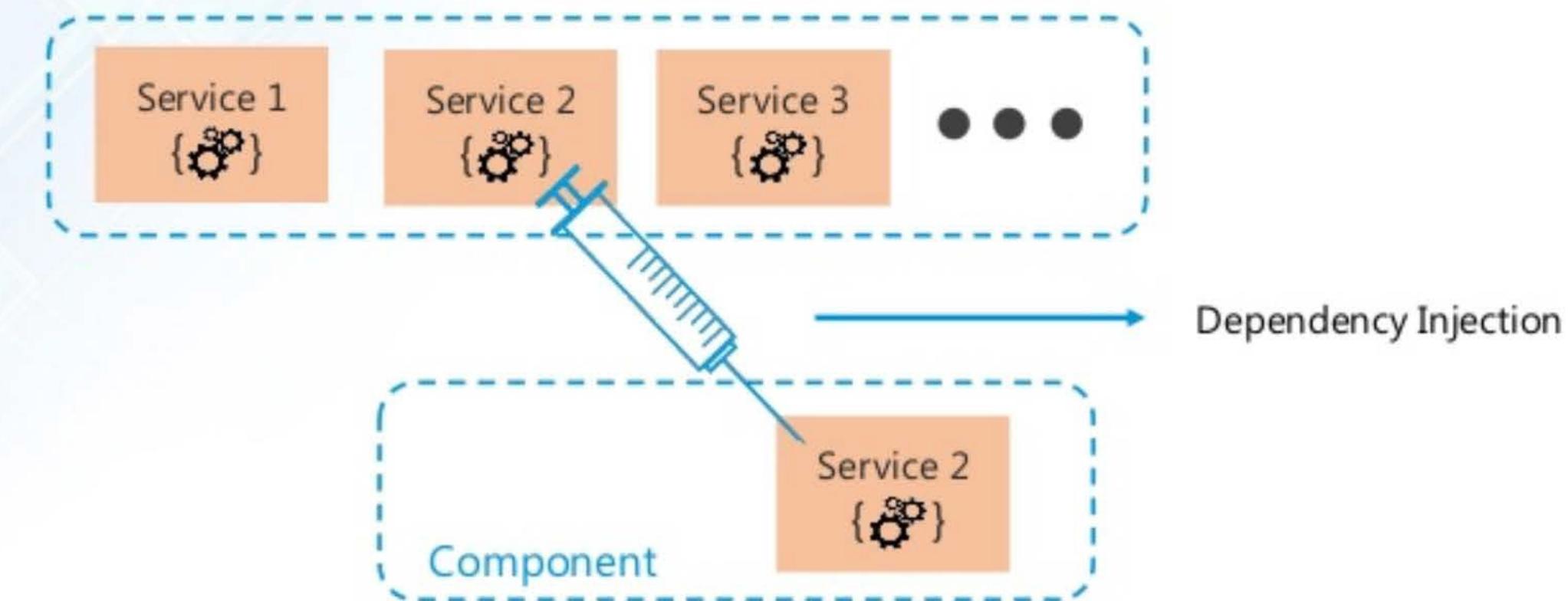
Data Binding

Services

Directives

Creates a new instance of class along with its required dependencies

Used to provide services to a component



Building Blocks of Angular

edureka!

Module

Component

Metadata

Template

Data Binding

Services

Directives

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

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

  movies: string[];
  constructor(private exampleService: ExampleService) {}

  ngOnInit() {
    this.movies = this.exampleService.getMovies();
  }
}
```

Importing Service Class

Injecting Service into the Component

Retrieving data

Building Blocks of Angular

edureka!

Module

Component

Metadata

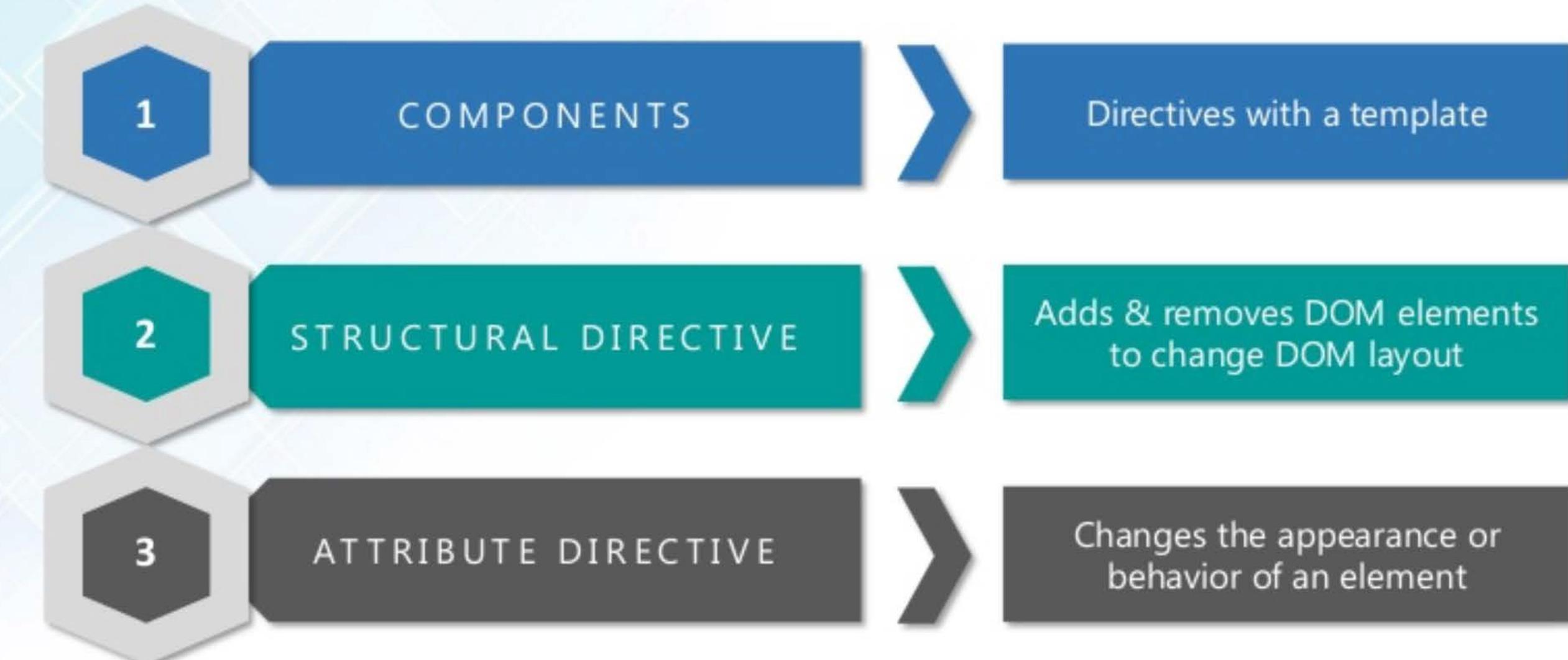
Template

Data Binding

Services

Directives

Changes the appearance or behavior of a DOM element



Building Blocks of Angular

edureka!

Module

Component

Metadata

Template

Data Binding

Services

Directives

2

STRUCTURAL DIRECTIVE

Adds & removes DOM elements
to change DOM layout

```
<ul>
  <li *ngFor = "let movie of movies">{{movie}}</li>
</ul>
```

Iterating over
the movies list

Building Blocks of Angular

edureka!

Module

Component

Metadata

Template

Data Binding

Services

Directives

3

ATTRIBUTE DIRECTIVE

Changes the appearance or behavior of an element

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

@Directive({
  selector: '[appBoldText]'
})
export class BoldTextDirective {

  constructor(private elementRef: ElementRef) {}

  @HostListener('mouseenter') onMouseEnter() {
    this.elementRef.nativeElement.style.fontWeight = 'bold';
  }

  @HostListener('mouseleave') onMouseLeave() {
    this.elementRef.nativeElement.style.fontWeight = null;
  }
}
```

Importing Directive, ElementRef & HostListener

Directive Metadata

Injecting ElementRef to access the DOM elements

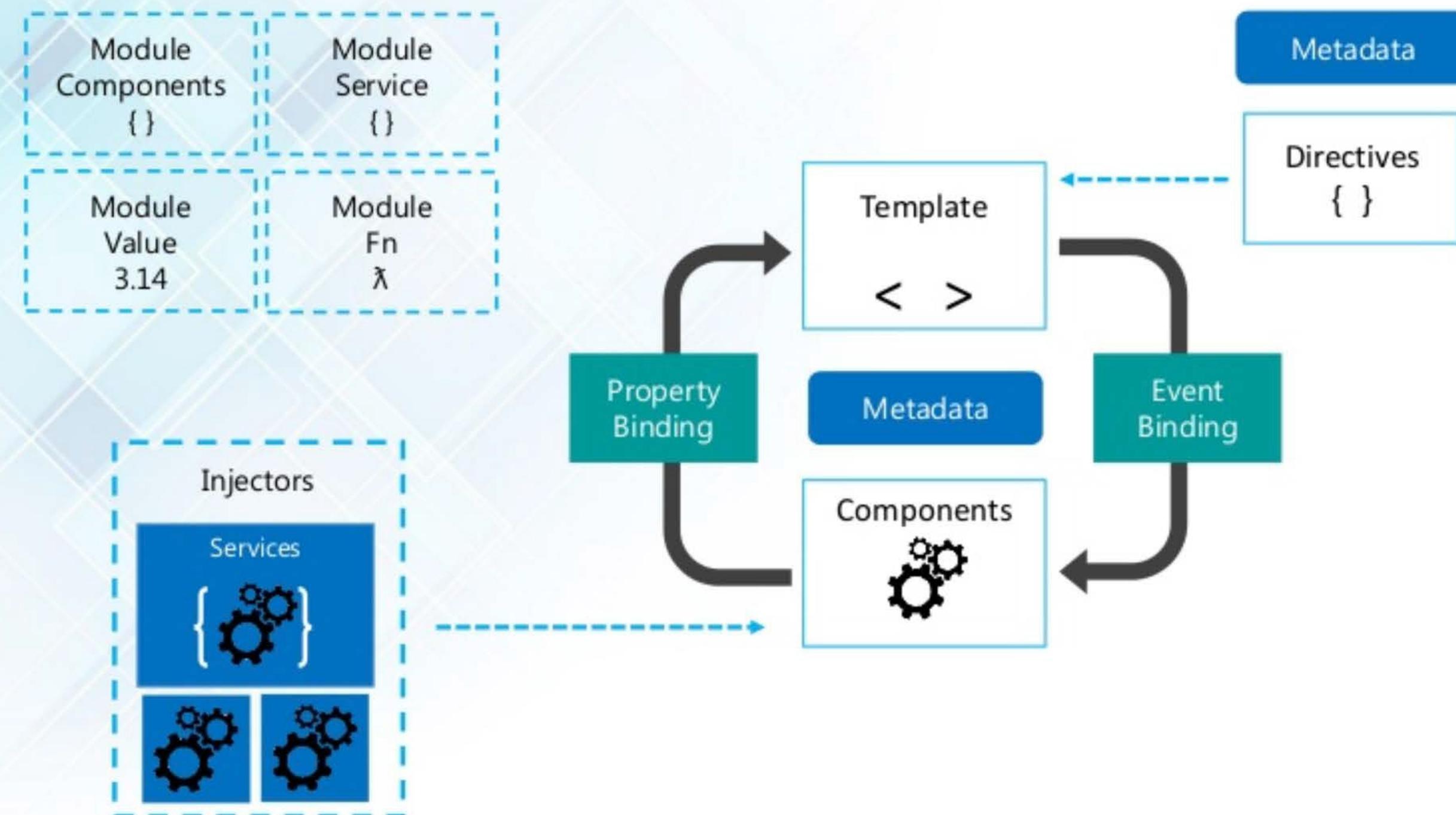
Bold the text on cursor hover

Un-bold the text

Angular Architecture

Angular Architecture

edureka!



References

- Edureka! Angular Tutorial
 - <https://www.slideshare.net/EdurekaIN/angular-4-tutorial-whats-new-in-angular-4-angular-training-edureka>
- Tutorial: Tour of Heroes
 - <https://angular.io/tutorial>
- An Overview of Angular 4
 - <https://www.slideshare.net/Cynoteck/an-overview-of-angular-4>
- Lecture Notes (Angular)
 - <http://oak.cs.ucla.edu/classes/cs144/notes/pdf/angular.pdf>