

**edureka!**



**Angular 4 Tutorial**

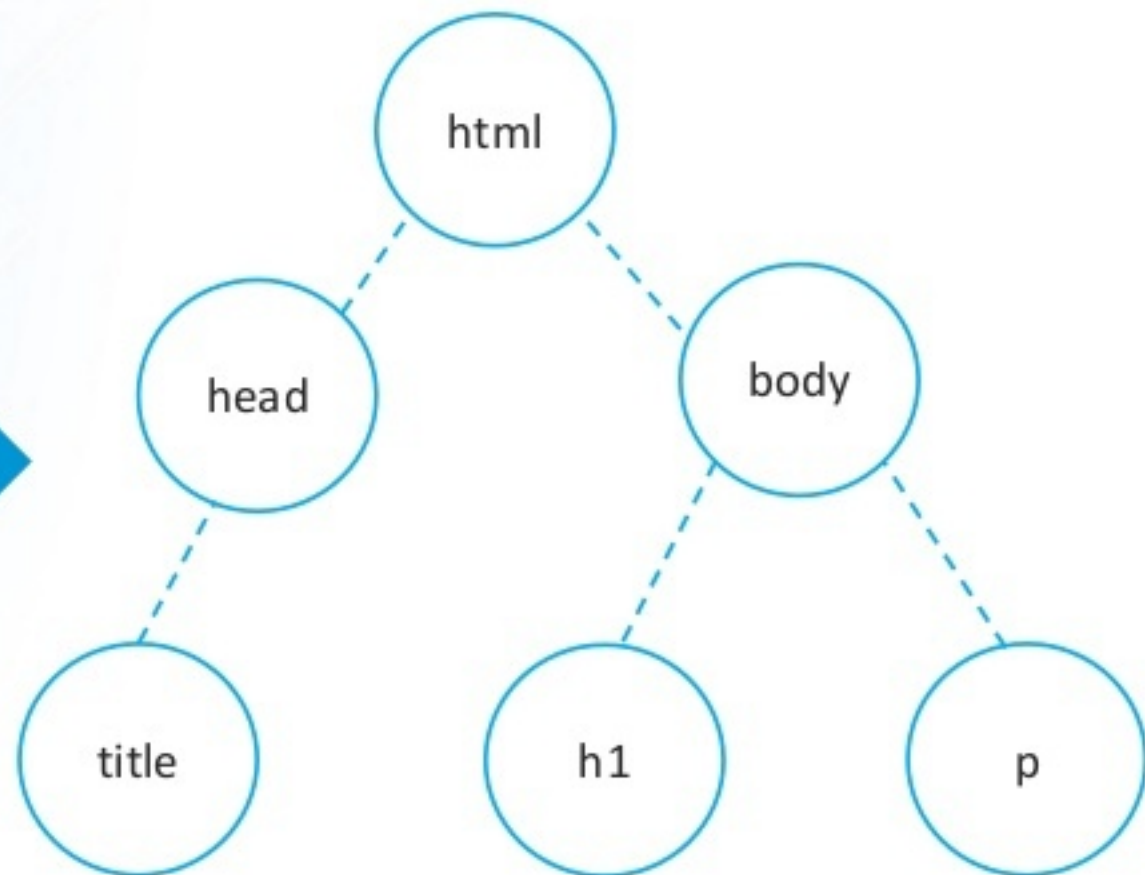
- ❖ Webpage and DOM
- ❖ DOM Manipulation
- ❖ JavaScript and jQuery
- ❖ Why Angular?
- ❖ What is SPA?
- ❖ Angular Introduction
- ❖ Angular Features
- ❖ Angular Installation
- ❖ Basic Building Blocks of Angular
- ❖ Angular Architecture



# 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>
```

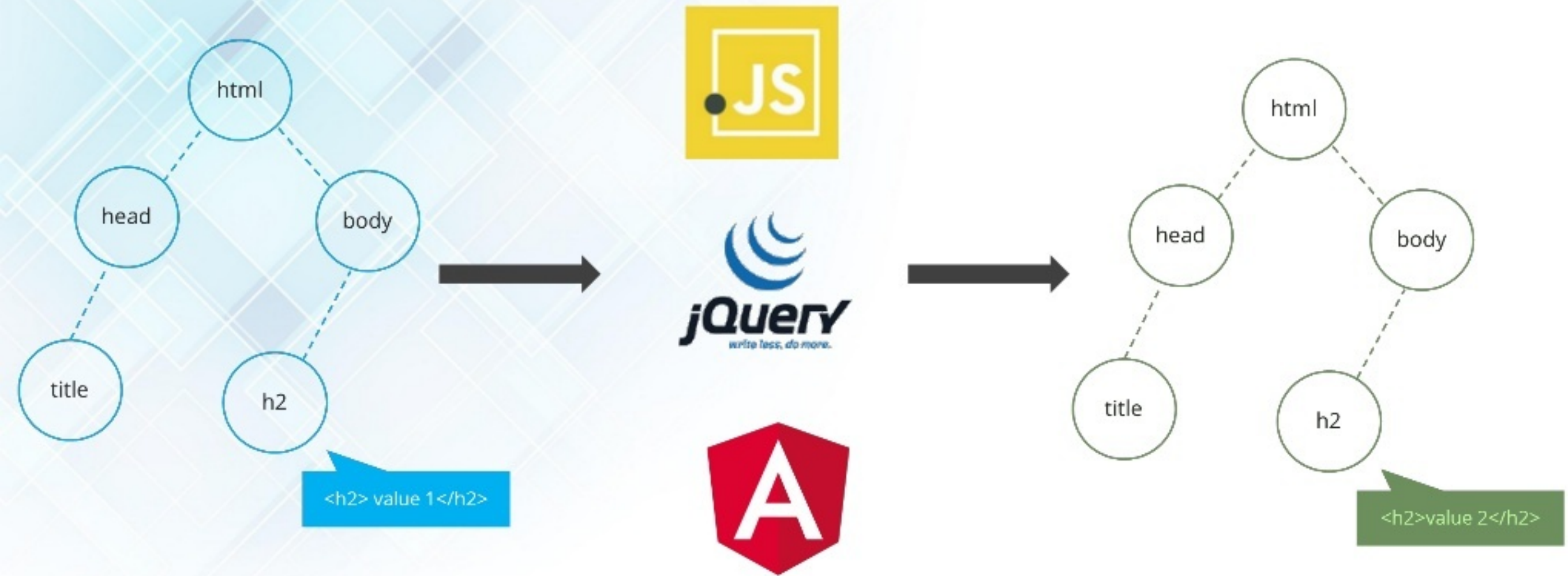
HTML Markup



DOM Tree of the HTML document

# DOM Manipulation





JavaScript



- 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?



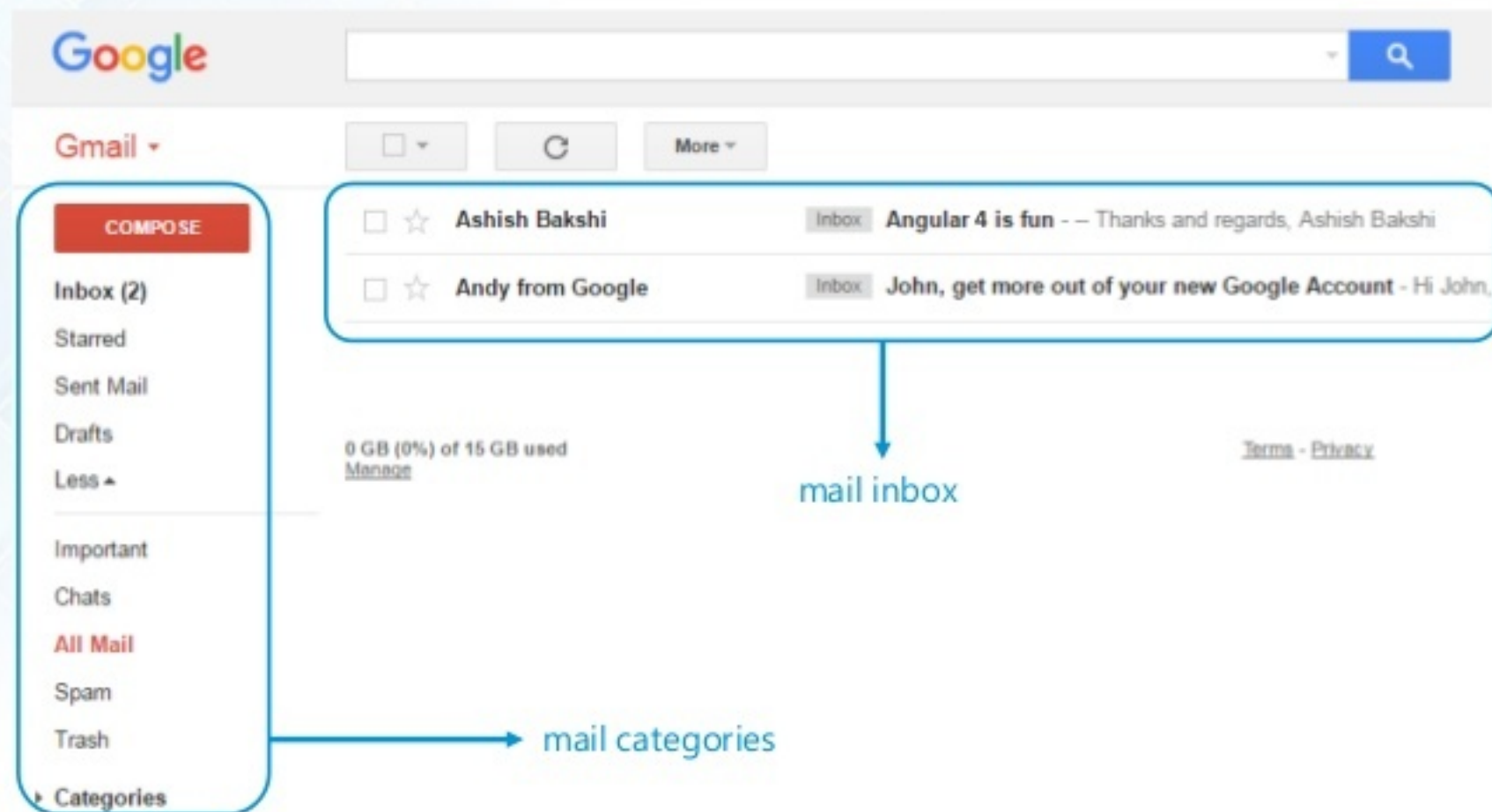
	jQuery	Angular
DOM Manipulation	✓	✓
RESTful API	✗	✓
Animation Support	✓	✓
Deep Linking Routing	✗	✓
Form Validation	✗	✓
2 Way Data Binding	✗	✓
AJAX/JSONP	✓	✓

# 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 Life Cycle



## Single Page Application Life Cycle

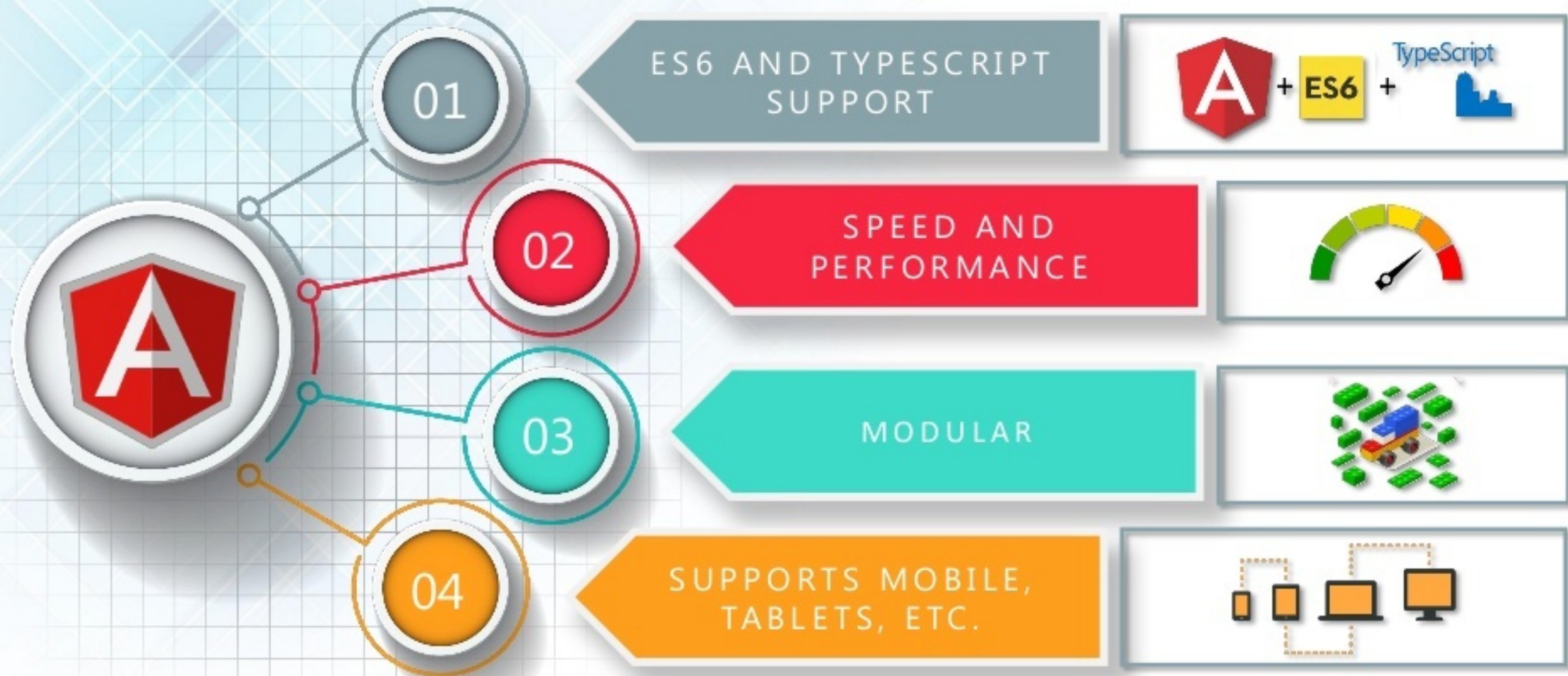


# Angular Introduction



# Angular Features





# Angular Installation

# Building Blocks of Angular

Module

Component

Metadata

Template

Data Binding

Services

Directives

# Building Blocks of Angular

**Module**

Component

Metadata

Template

Data Binding

Services

Directives

Module is a class with  
@NgModule metadata

Every Angular app has at  
least one root module

Encapsulation of different  
similar functionalities

*Similar  
Functionalities*

Components

Directives



Pipes

*export*

Single Module



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

Module

Component

Metadata

Template

Data Binding

Services

Directives



Nav Bar

News Feed



Module

**Component**

Metadata

Template

Data Binding

Services

Directives

Sports  
Component



Module

**Component**

Metadata

Template

Data Binding

Services

Directives

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

Importing Component Decorator

```
@Component({
```

Decorator

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

Meta Data

```
})
```

```
export class ExampleComponent implements OnInit {
```

Exporting Component Class

```
  constructor() { }
```

```
  ngOnInit() {  
  }
```

```
}
```



- Module
- Component
- Metadata**
- Template
- Data Binding
- Services
- Directives

Metadata describes how to process the class

Decorator is used to attach metadata

Example:



*MyClass*



```
@Component({  
  .....  
})
```

*Decorator*



```
Component  
{ }
```



*AppClass*



```
@NgModule({  
  .....  
})
```

*Decorator*



```
Module  
{ }
```

Module

Component

**Metadata**

Template

Data Binding

Services

Directives

```
@Component({
```

```
  selector: 'app-example',
```

```
  templateUrl: './example.component.html',
```

```
  styleUrls: ['./example.component.css'],
```

```
  providers: [ExampleService]
```

```
})
```

Decorator that specifies how to process an Angular Class

Creates an instance of the component

HTML template for the component

CSS Styling

Provides Service for the Component

Module

Component

Metadata

**Template**

Data Binding

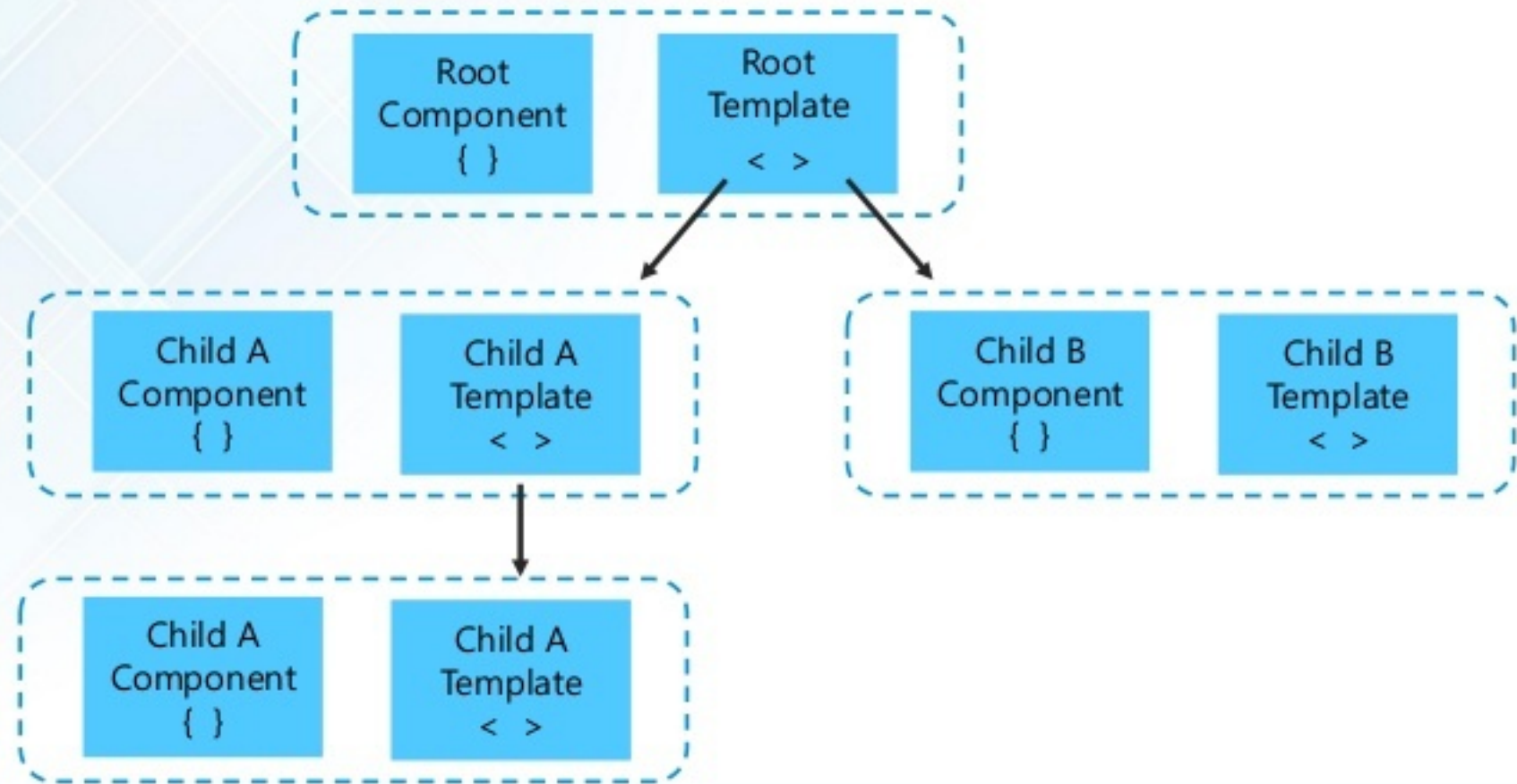
Services

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.





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

DOM

{{ value }}

COMPONENT

PROPERTY BINDING

02

DOM

[property] = "value"

COMPONENT

EVENT BINDING

03

DOM

(event) = "event handler"

COMPONENT

2 WAY DATA BINDING

04

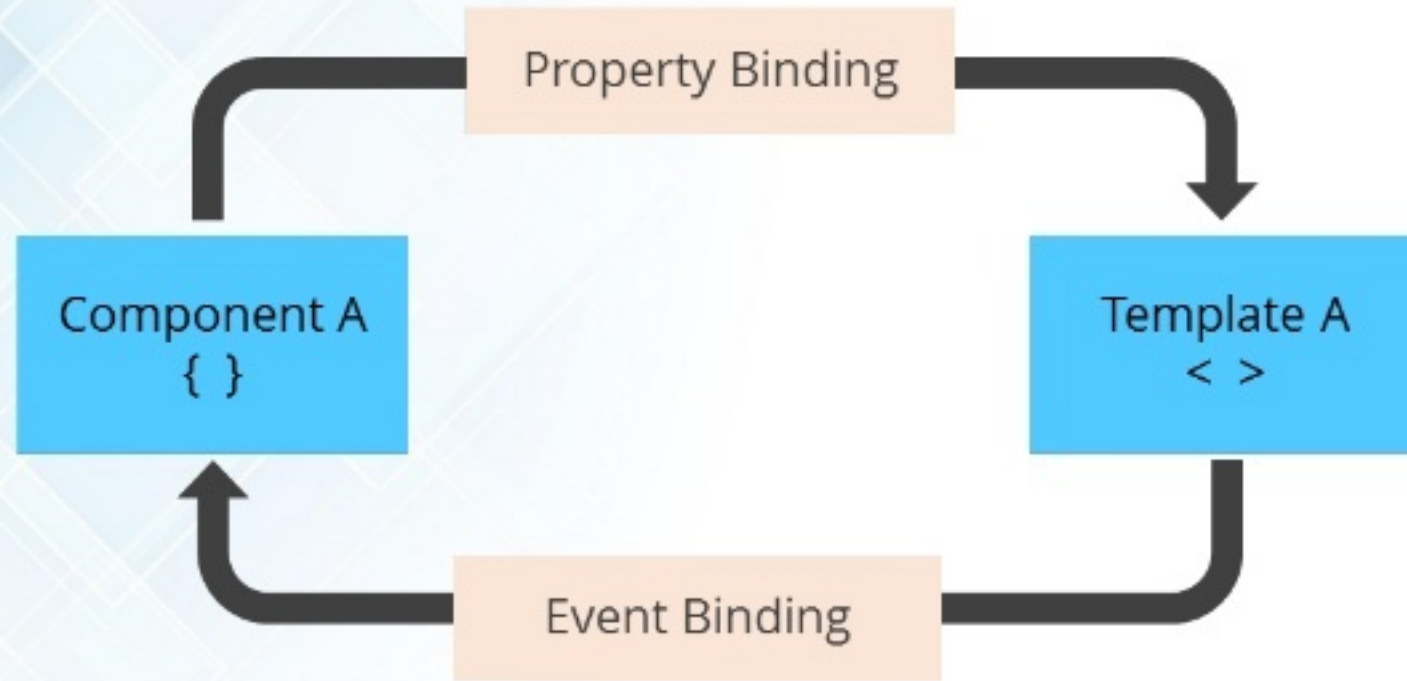
DOM

[( ngModel )]

COMPONENT



- Module
- Component
- Metadata
- Template
- Data Binding**
- Services
- Directives



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

Module

Component

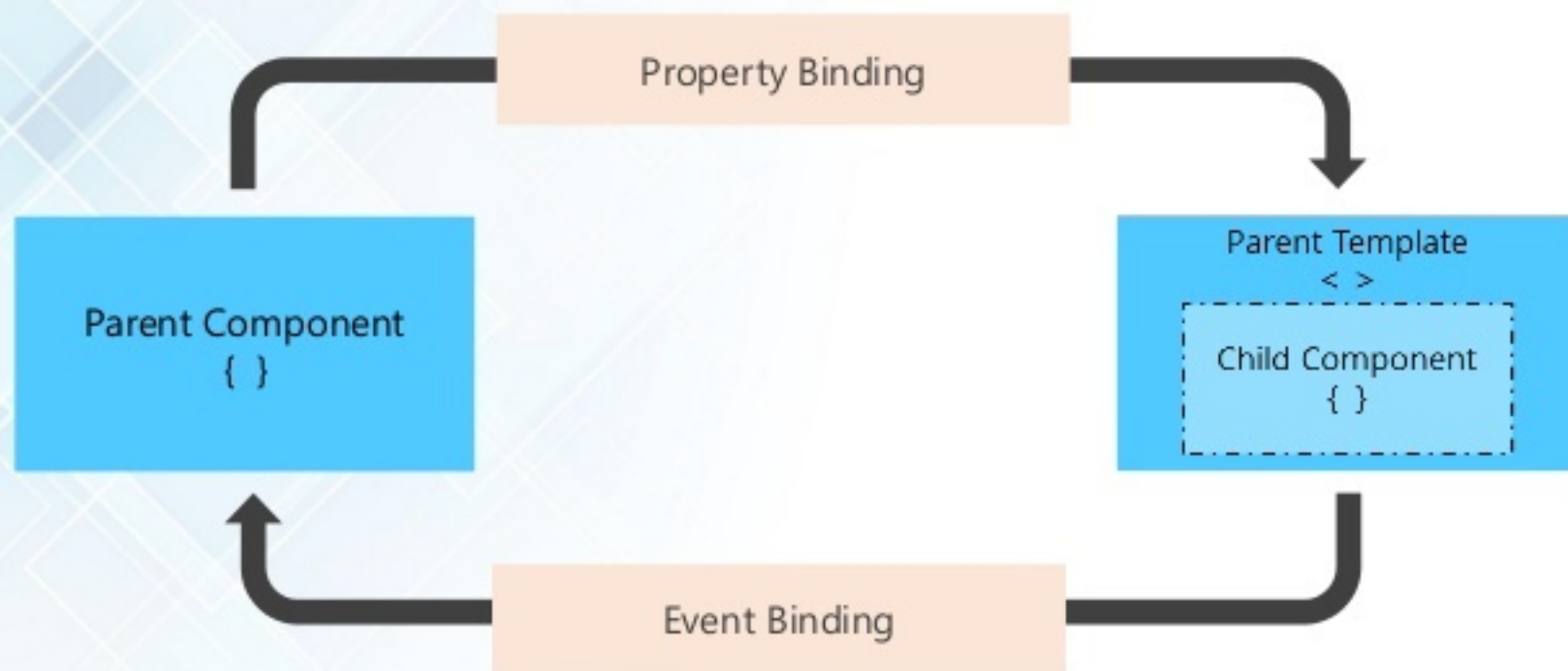
Metadata

Template

**Data Binding**

Services

Directives

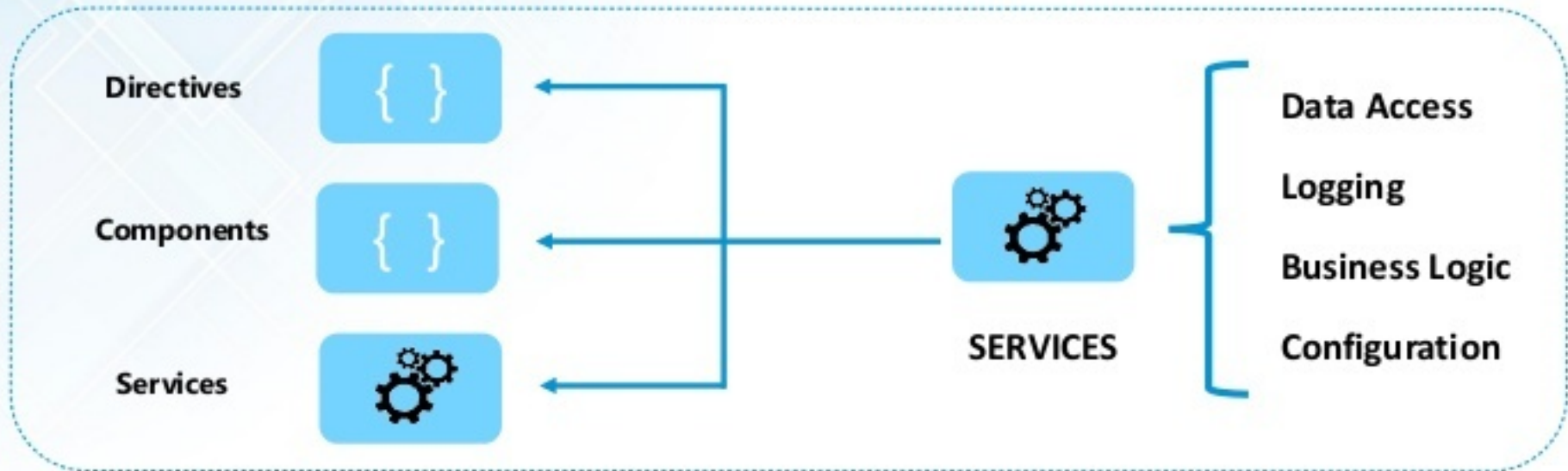


Data binding is also important for communication between parent and child components.

- Module
- Component
- Metadata
- Template
- Data Binding
- Services**
- Directives

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

Example:





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



Module

Component

Metadata

Template

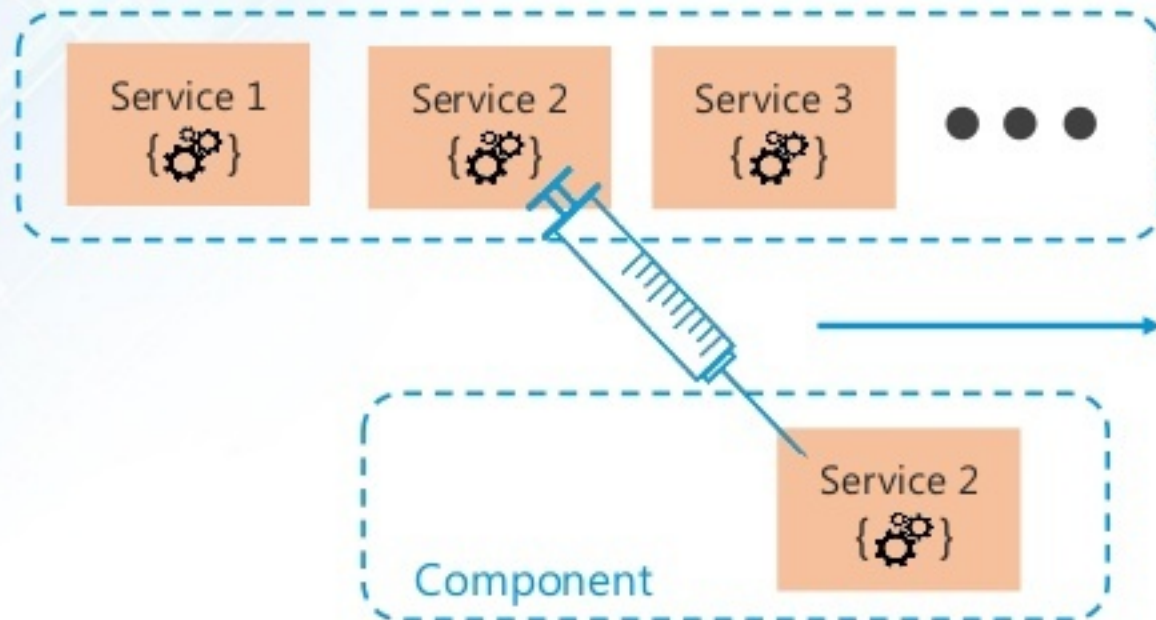
Data Binding

**Services**

Directives

Creates a new instance of class along with its required dependencies

Used to provide services to a component



Module

Component

Metadata

Template

Data Binding

**Services**

Directives

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

Importing Service Class

```
@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) {}
```

Injecting Service into the Component

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

Retrieving data

Module

Component

Metadata

Template

Data Binding

Services

**Directives**

Changes the appearance or behavior of a DOM element

1

COMPONENTS

Directives with a template

2

STRUCTURAL DIRECTIVE

Adds & removes DOM elements to change DOM layout

3

ATTRIBUTE DIRECTIVE

Changes the appearance or behavior of an element



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





## ATTRIBUTE DIRECTIVE



Changes the appearance or behavior of an element

Module

Component

Metadata

Template

Data Binding

Services

**Directives**

```
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

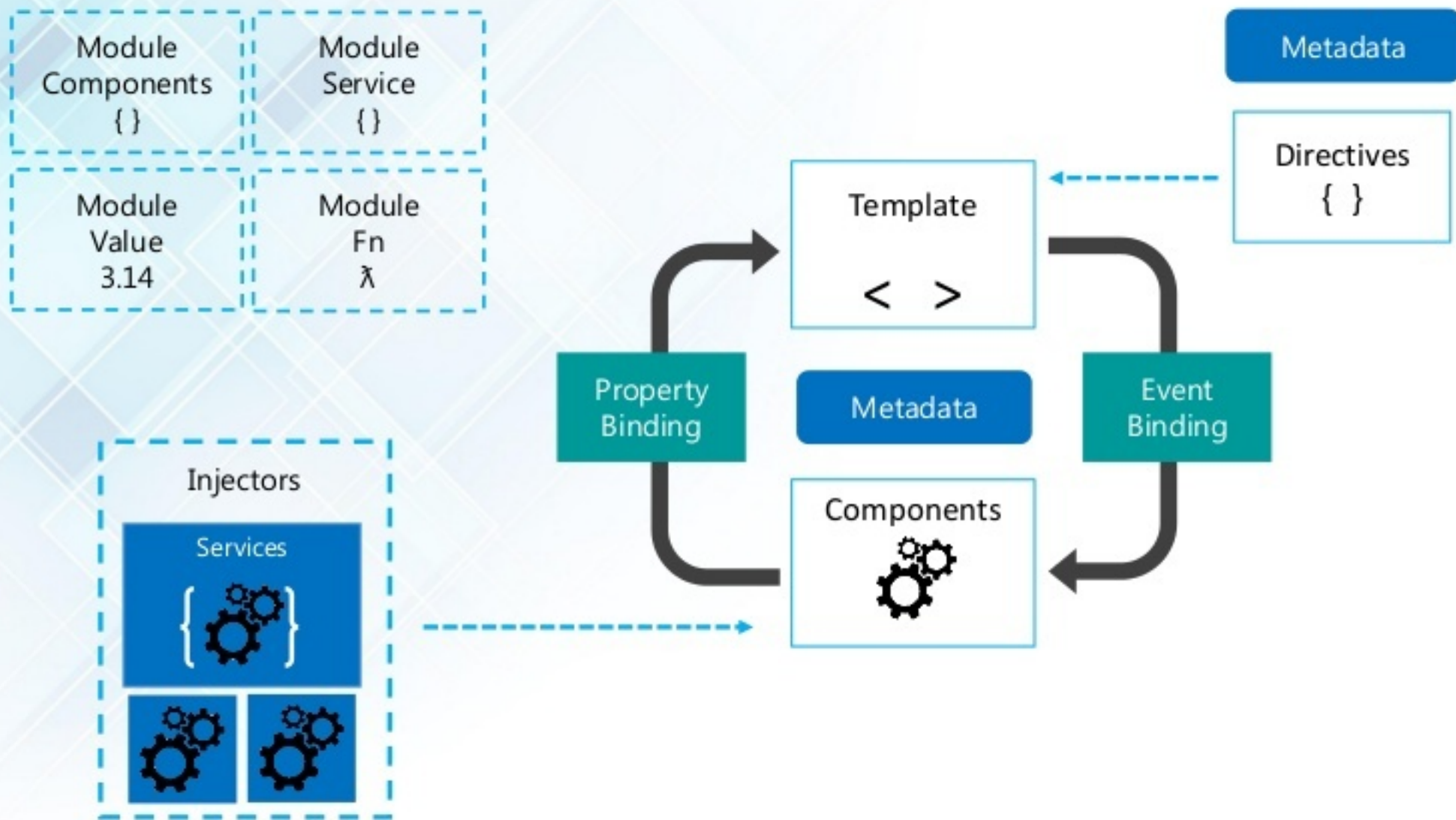
Directive Metadata

Injecting ElementRef to access the DOM elements

Bold the text on cursor hover

Un-bold the text

# Angular Architecture





**edureka!**

**Thank You**

For more information please visit our website  
[www.edureka.co](http://www.edureka.co)