

# Introduction to Components

---



**Deborah Kurata**

CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | [blogs.msmvps.com/deborahk/](https://blogs.msmvps.com/deborahk/)





# Module Overview



**What Is a Component?**

**Creating the Component Class**

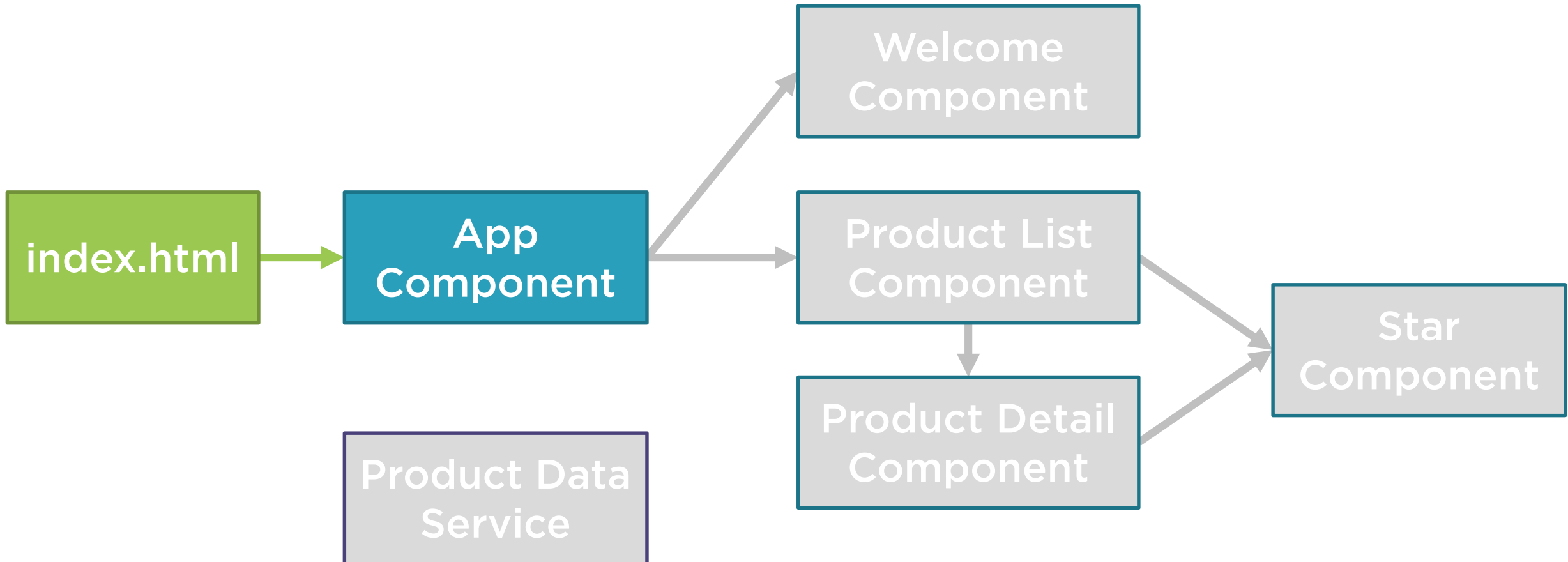
**Defining the Metadata with a Decorator**

**Importing What We Need**

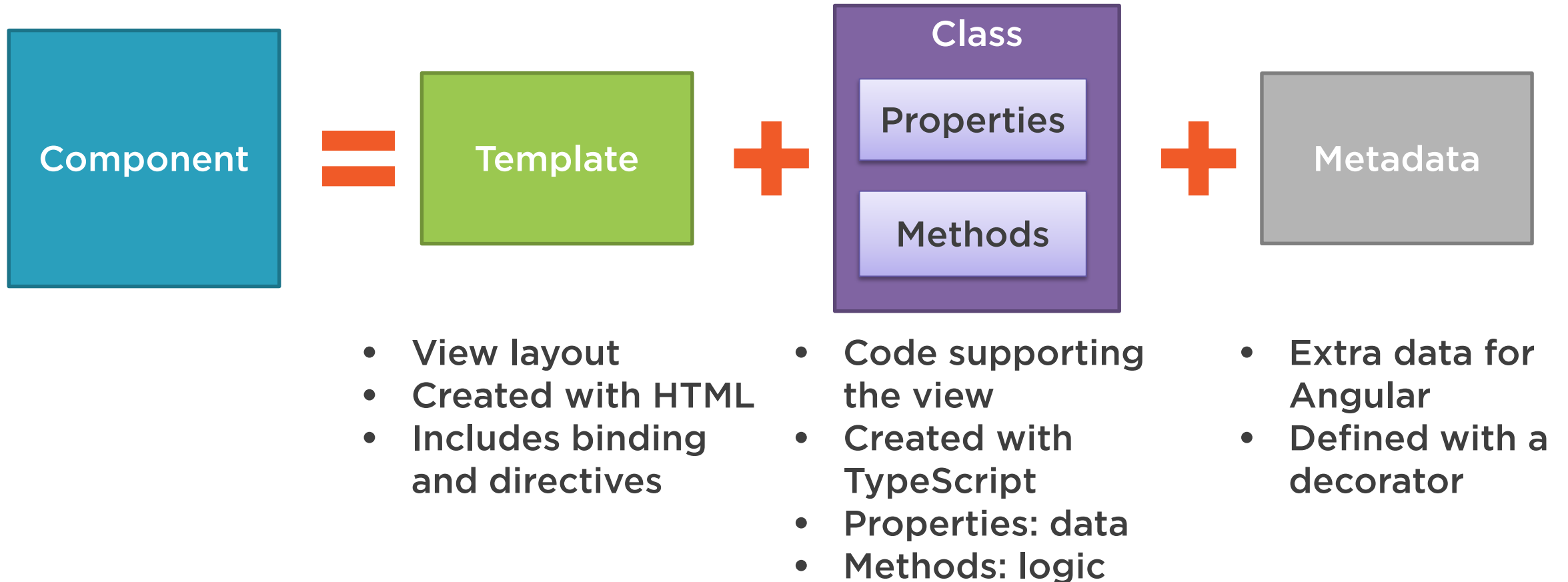
**Bootstrapping Our App Component**



# Application Architecture



# What Is a Component?



# Component

app.component.ts

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

Import

```
@Component({  
  selector: 'pm-root',  
  template: `  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
  `,  
})
```

Metadata &  
Template

```
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```

Class



# Creating the Component Class

app.component.ts

```
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```

class  
keyword

Class Name

export  
keyword

Component Name  
when used in code



# Creating the Component Class

app.component.ts

```
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```

Property  
Name

Data Type

Default  
Value

Methods





# Defining the Metadata

app.component.ts

```
@Component({  
  selector: 'pm-root',  
  template: `  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
  `,  
})  
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```



# Decorator

A function that adds **metadata** to a class, its members, or its method arguments.

Prefixed with an @.

Angular provides built-in decorators.

**@Component()**



# Defining the Metadata

app.component.ts

```
@Component({
  selector: 'pm-root',
  template: `
    <div><h1>{{pageTitle}}</h1>
      <div>My First Component</div>
    </div>
  `
})
export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```

Component  
decorator

Directive Name  
used in HTML

View Layout

Binding



# Importing What We Need



**Before we use an external function or class, we define where to find it**

**`import` statement**

**`import` allows us to use exported members from external ES modules**

**Import from a third-party library, our own ES modules, or from Angular**

# Angular Is Modular

@angular/  
core

@angular/  
animate

@angular/  
http

@angular/  
router

<https://www.npmjs.com/~angular>



# Importing What We Need

app.component.ts

```
@Component({  
  selector: 'pm-root',  
  template: `  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
  `,  
})  
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```



# Importing What We Need

app.component.ts

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

```
@Component({  
  selector: 'pm-root',  
  template: `  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
  `,  
})  
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```

import keyword

Angular library  
module name

Member name



# Completed Component

app.component.ts

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

@Component({
  selector: 'pm-root',
  template: `
    <div><h1>{{pageTitle}}</h1>
      <div>My First Component</div>
    </div>
  `
})
export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```





Demo



## Creating the App Component



# Bootstrapping Our App Component



**Host the application**

**Defining the Angular module**



# Single Page Application (SPA)



`index.html` contains the main page for the application

This is often the only Web page of the application

Hence an Angular application is often called a Single Page Application (SPA)

# Hosting the Application

index.html

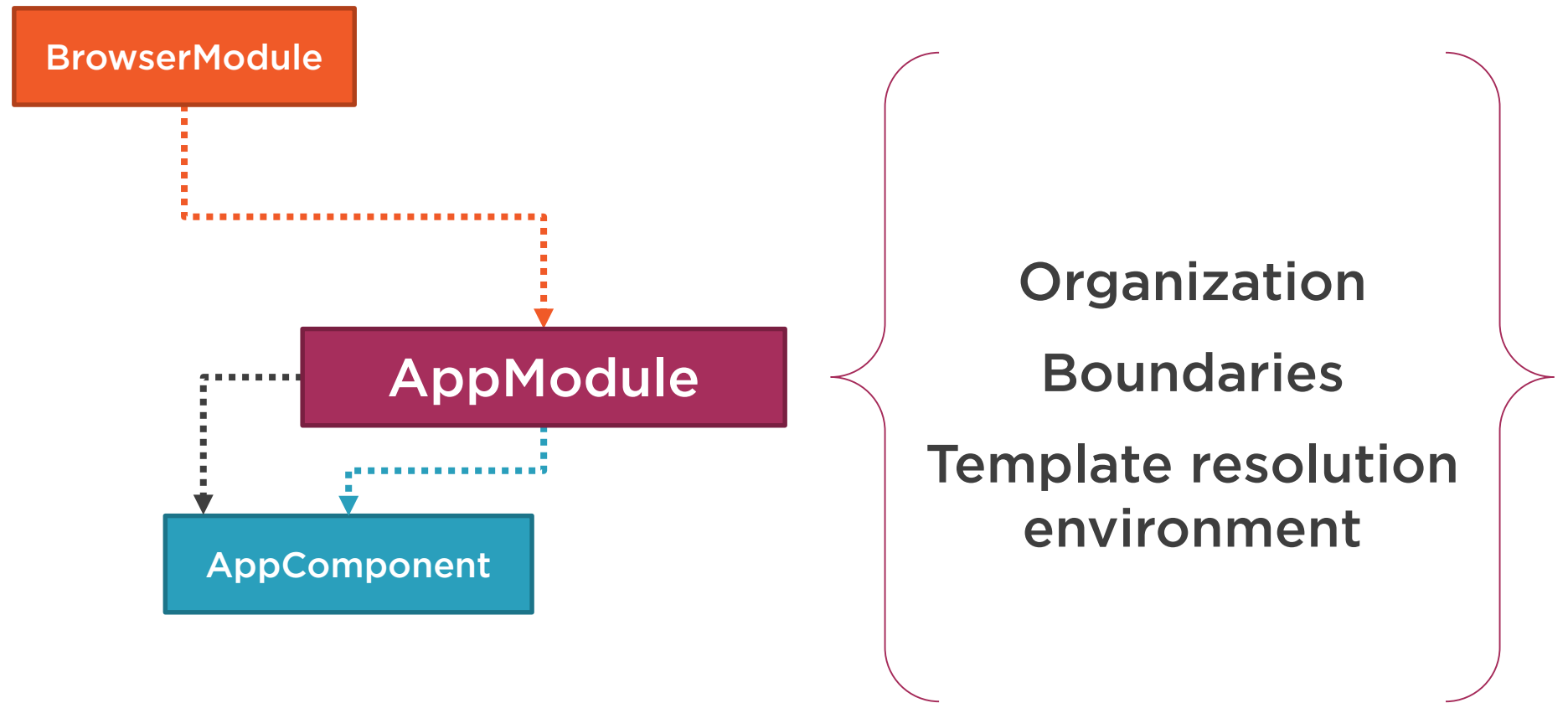
```
<body>
  <pm-root></pm-root>
</body>
```

app.component.ts

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

@Component({
  selector: 'pm-root',
  template: `
    <div><h1>{{pageTitle}}</h1>
      <div>My First Component</div>
    </div>
  `
})
export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```





- ..... Imports
- ..... Exports
- ..... Declarations
- ..... Providers
- ..... Bootstrap



# Defining the Angular Module

## app.module.ts

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

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



Demo



## Bootstrapping Our App Component



# Component Checklist



\_\_\_\_\_



\_\_\_\_\_



\_\_\_\_\_

Class -> Code

Decorator -> Metadata

Import what we need





# Component Checklist: Class



## Clear name

- Use PascalCasing
- Append "Component" to the name

## export keyword

## Data in properties

- Appropriate data type
- Appropriate default value
- camelCase with first letter lowercase

## Logic in methods

- camelCase with first letter lowercase

# Component Checklist: Metadata



## Component decorator

- Prefix with @; Suffix with ()

## selector: Component name in HTML

- Prefix for clarity

## template: View's HTML

- Correct HTML syntax

# Component Checklist: Import



**Defines where to find the members that this component needs**

`import keyword`

**Member name**

- Correct spelling/casing

**Module path**

- Enclose in quotes
- Correct spelling/casing

# Something's Wrong! Checklist



**F12 is your friend**

**Recheck your code**

- HTML
  - Close tags
  - Angular directives are case sensitive
- TypeScript
  - Close braces
  - TypeScript is case sensitive

# Something's Wrong! Checklist (cont.)



## Check my blog for a solution

- <http://blogs.msmvps.com/deborahk/angular-2-getting-started-problem-solver/>

## Post to the course discussion

- <https://app.pluralsight.com/library/courses/angular-2-getting-started/discussion>



# Summary



What Is a Component?

Creating the Component Class

Defining the Metadata with a Decorator

Importing What We Need

Hosting Our App Component



# Application Architecture

