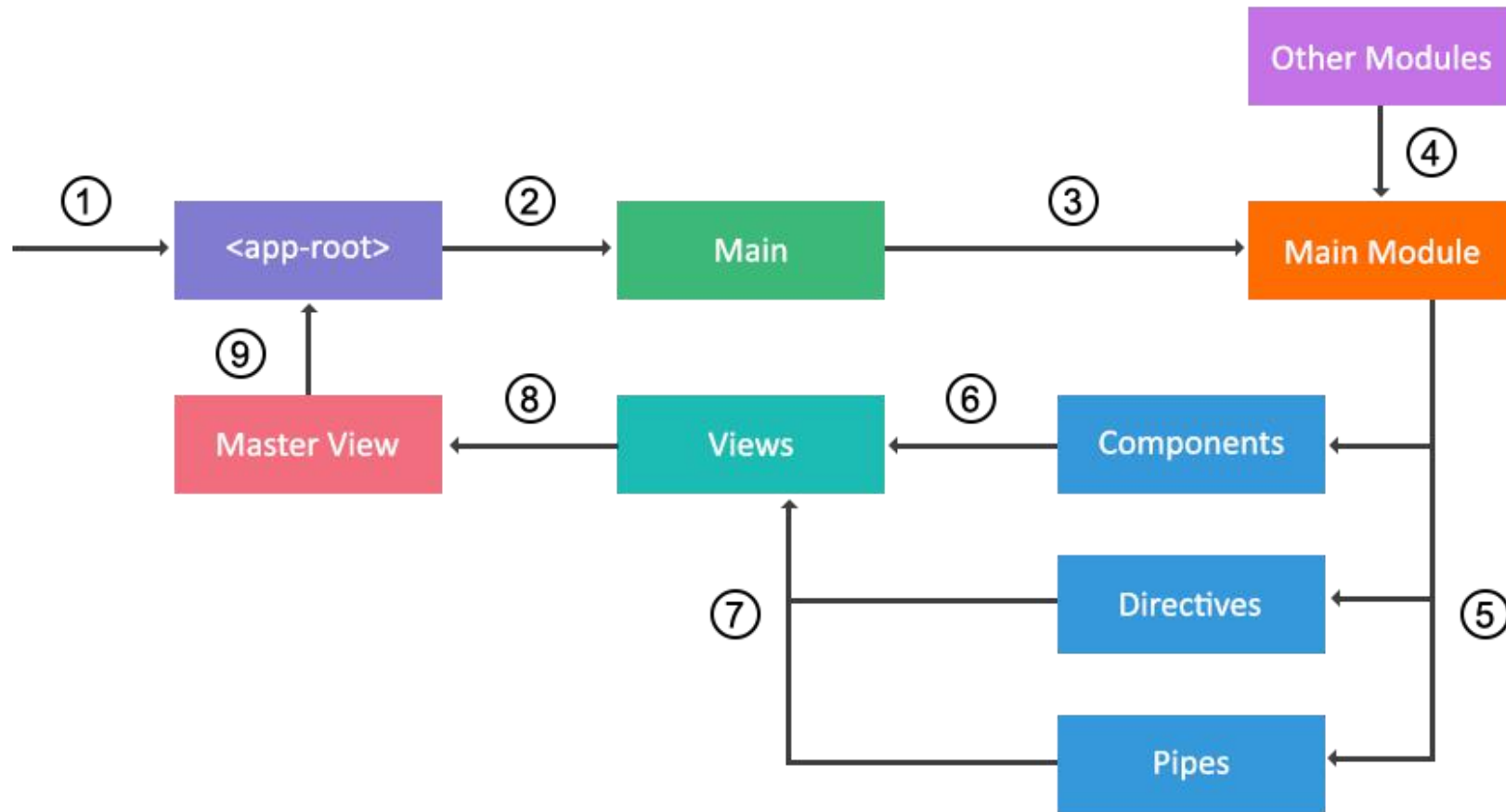


Getting Started with Angular

Angular Initialization Process



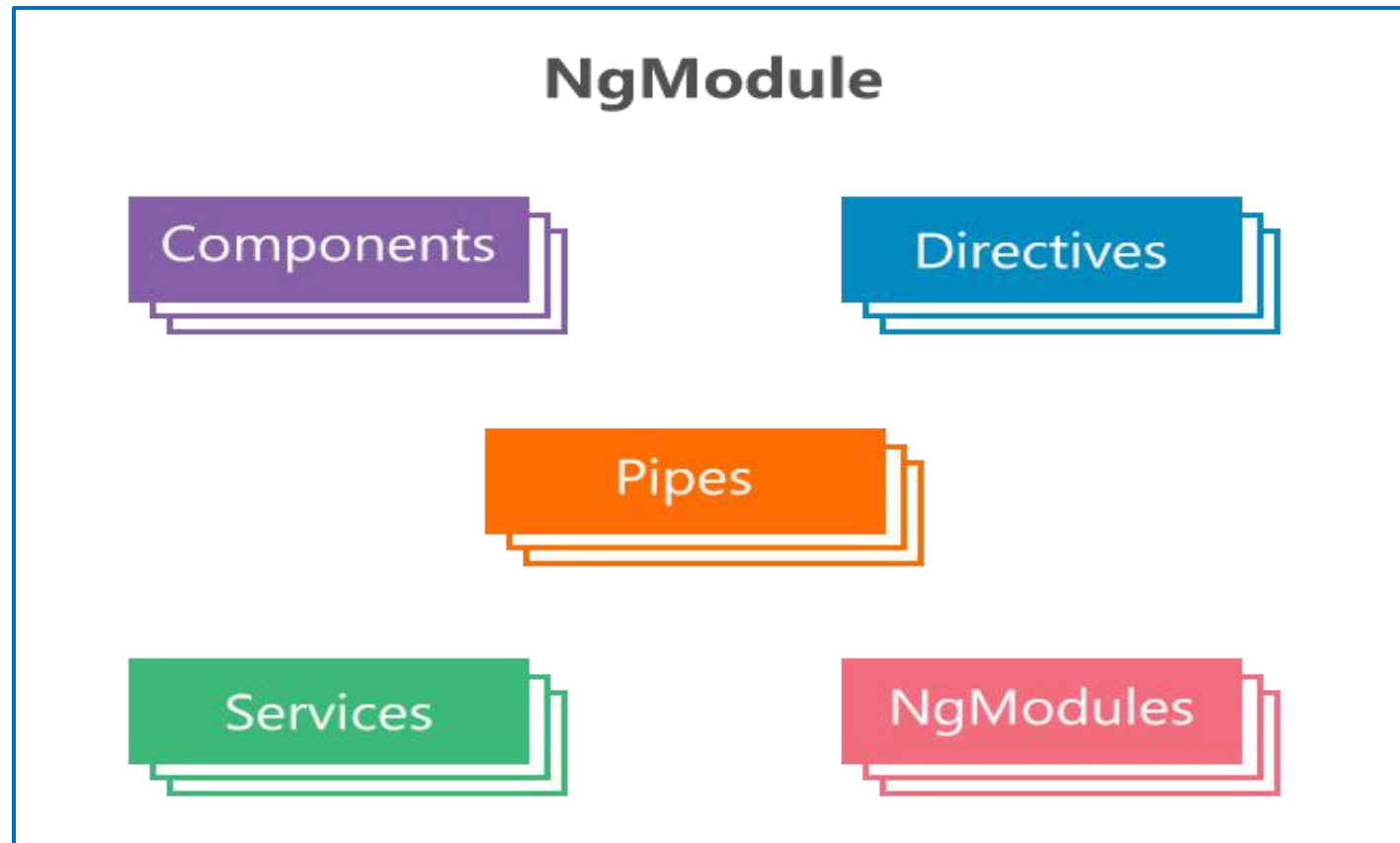
Angular Building Blocks

- Modules
- Components
- Templates
- Metadata
- Data binding
- Directives
- Pipes
- Routing
- Forms
- Services
- Dependency injection

Modules

- A module organize an application into unified blocks of functionality
- An Angular module is a class with an *@NgModule* decorator
- Accepts a single metadata object whose properties describe the module
- Each Angular app must have at least one module, known as root module

Modules



NgModule Metadata Main Properties

- **imports** – Specify other dependent modules whose classes are required by the component templates declared in the module
- **declarations** – Specify the components, directives, and pipes that belong to the module
- **bootstrap** – Specify the main app view i.e root component. Only the **root module** can have this bootstrap property
- **exports** – A subset of declarations that will be visible and usable in the other modules. A root module doesn't have export option.
- **providers** – Specify the services, accessible across the app

Built-In Modules

- Angular has built-In library modules starting with the @angular as prefix

@angular/core

@angular/router

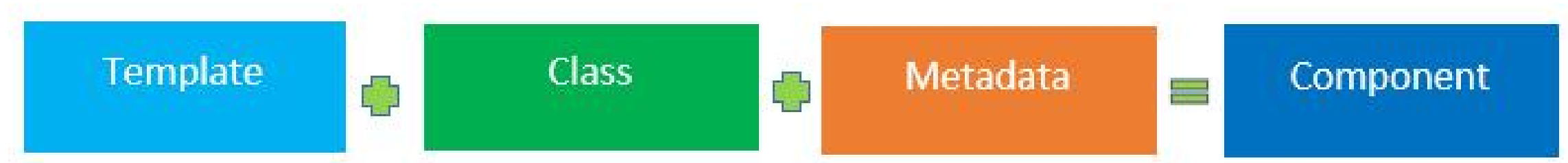
@angular/forms

@angular/http

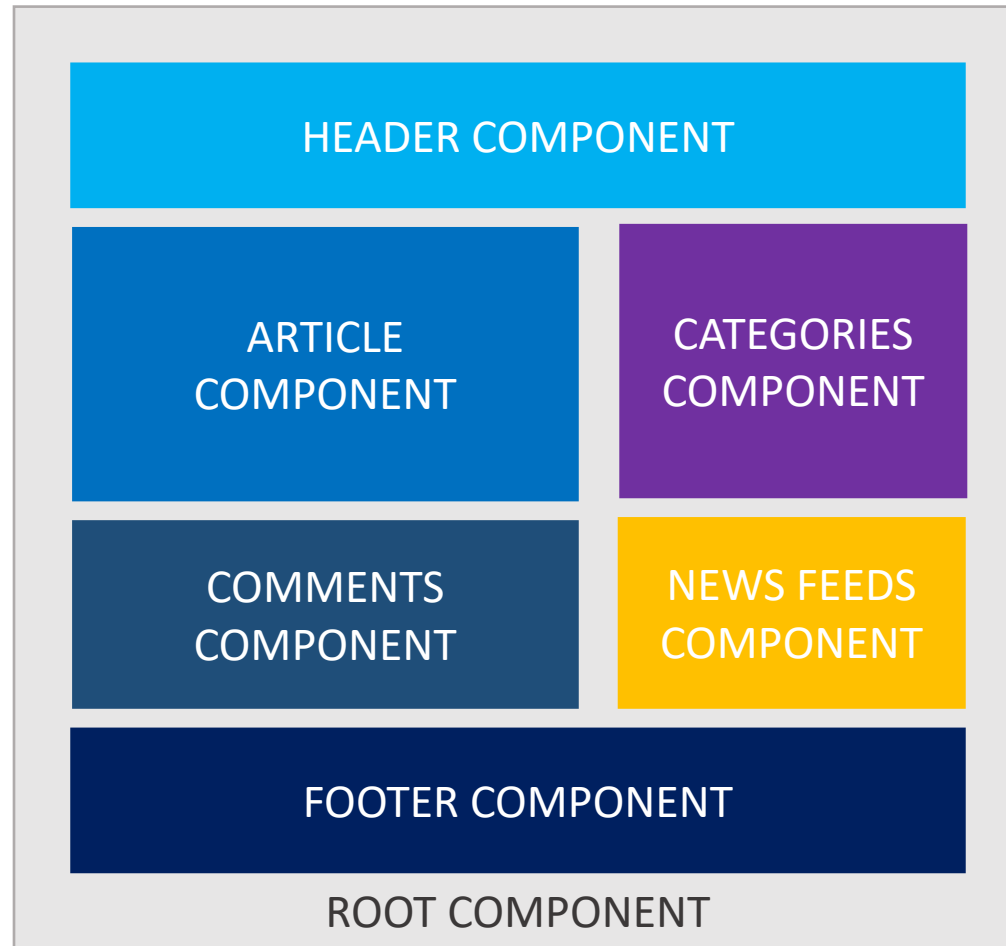
- Built-In library & third part modules can be installed using npm manager
- Built-In modules, components, services, directives etc. can be imported by using built-In library modules

Component

- A type of directives with template, styles and logic for user interaction
- Exported as a custom HTML tag like as:
 - `<my-component></my-component>`
- Initialized by Angular Dependency Injection engine



Angular Components Page View



Component Example

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

@Component({
  selector: 'my-component',
  template: `<h3>Interpolation</h3>
    <p>Name : {{name}}</p>
    <p><input type="text" value="{{name}}" /></p>`,
  styles: []
})
export class MyComponent {
  name: string = 'Shailendra';
  constructor() { }
}
```

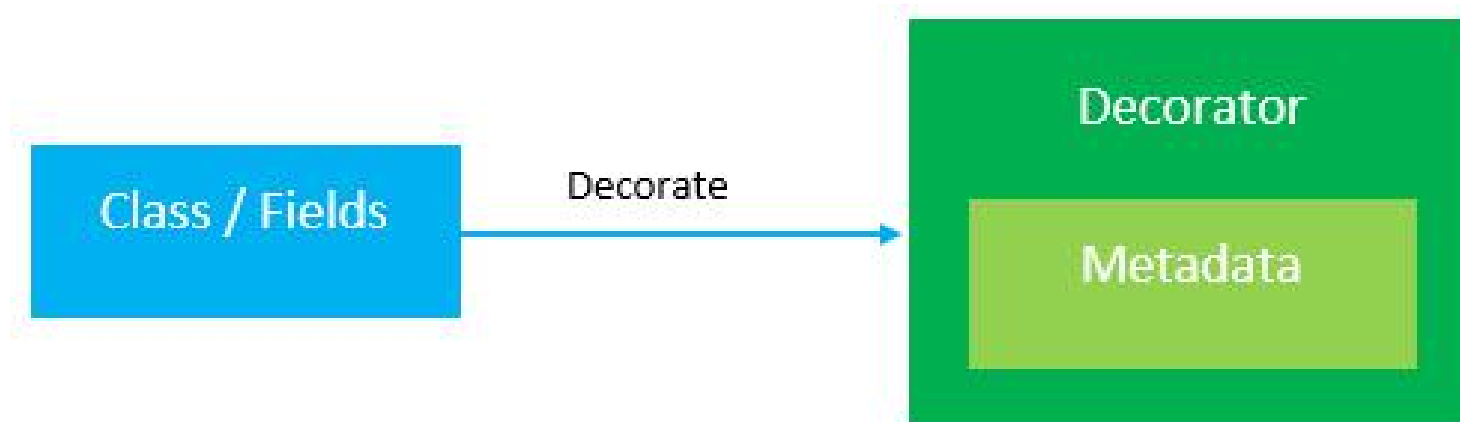
Template

- Define the view of a component
- Contains Html markup and angular directives, attributes etc.
- Describe how a component is rendered on the page

```
<h3>Interpolation</h3>
<p>Name : {{name}}</p>
<p>
  <input type="text" value="{{name}}" />
</p>
```

Decorators

- A function that adds metadata to a class, class members
- These are prefix with @ symbol
- Angular has built-In decorators like - @Component, @NgModule, @Directive, @Pipe etc.



Types of Decorators

- Class decorators
 - @NgModule – Used for defining a module
 - @Component – Used for defining a component
 - @Directive – Used for defining a directive
 - @Injectable – Used for injecting dependencies
 - @Pipe – Used for defining a pipe
- Class field decorators
 - @Input – Used for receiving data (input) from parent to child component
 - @Output – Used for passing data (events) from child to parent component

Metadata

- Tells Angular how to process a class
- Decorators are used to attach metadata to a class

