# Creating and using component in Angular 4

An Angular app can be viewed as combination of components. Thus it's important to understand components in Angular.

This guide will instruct you how to create a component by using provided components, how to modify one.

## What is Component

First we need to understand what a component is. Components control views. They also coordinate other components and services to bring functionality to your app.

Components consists of 3 elements, each is defined in a separate file for better organizing. - Templates (HTML), - Stylesheets (CSS) and - Component Class (Javascript/Type Script code).

Components will be nested in other components, or imported in Modules.

## Defining a component

To define a component, we define 3 files as mentioned above. With Angular CLI, we can generate new component using the following command: `ng generate component [ComponentName]` For example, to generate `Foo` component we use

```
ng generate component Foo
```

Angular CLI will create `foo` directory which contains 4 files

```
foo.component.css       # Stylesheet
foo.component.html      # Template
foo.component.ts        # Component class
foo.component.spec.ts   # Spec file used for testing
```

We will ignore the spec file for now and focus on 3 other files. Now we will explain the meaning of each file.

### Template

A template is a form of HTML that tells Angular how to render the component. Angular has its own template syntax.

In short, a template looks like regular HTML, except for a few differences. Almost all HTML syntax is a valid template syntax.

Inside template of a component, you can insert another component using its `selector` as mentioned in Component code section below.

### Component code

A component is just a class with @Component decorator. Inside this decorator, we define the metadata of the component.

Metadata tells Angular how to process a class.

Here are some important properties of the metadata of a component.

- `selector`: this defines the selector for this component. It tells Angular to create and insert this component to where it find this selector in a parent component. For example if the selector is `sample-comp`, then you put `<sample-component></sample-component>` in an HTML code where you want to insert this component to.
- `templateUrl`: the address of this component's HTML template.
- `styleUrls`: the address of the stylesheet file for this component. In this file you apply style for elements in the component by using CSS rule.
- `providers`: the array of services that the component requires.

As stated before, a component controls view. It does that using "HTML-alike" language call template.

You can think of component code is the controller of the component. Component code is written in Javascript or language like Typescript that compiles to Javascript.

## Using a template

For example we have a component `FooComponent` defined as below:

- Template: `foo.component.html`

```
<h2>This is Foo component</h2>
<p>Lorem ipsum dolor sit amet, paulo dolorum efficiendi ea eos, nihil labores
interesset at vis. Aeque ponderum vel ut. Quo dicat complectitur ad, mea
facilisis maluisset ex. In legendos corrumpit omittantur quo. Mei vero lorem
choro no, brute perfecto nam in.</p>
```

- Style: `foo.component.css`

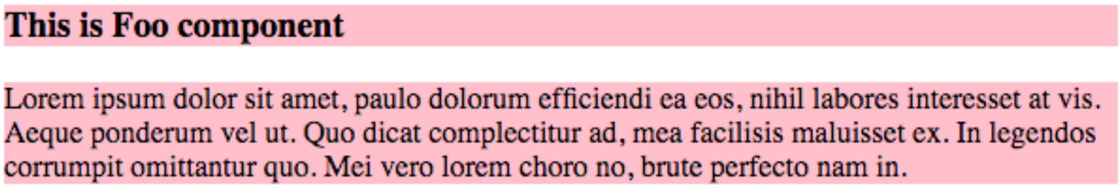
```
* {
    background-color: pink;
}
```

- Component class: `foo.component.ts`

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

@Component({
  selector: 'app-foo',
  templateUrl: './foo.component.html',
  styleUrls: ['./foo.component.css']
})
export class FooComponent implements OnInit {
  constructor() { }
  ngOnInit() {
  }
}
```

Here's what Foo component looks like.

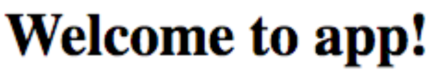


This is Foo component

Lorem ipsum dolor sit amet, paulo dolorum efficiendi ea eos, nihil labores interesset at vis. Aeque ponderum vel ut. Quo dicat complectitur ad, mea facilisis maluisset ex. In legendos corrumpit omittantur quo. Mei vero lorem choro no, brute perfecto nam in.

*FooComponent.png*

We will insert this component to `app.component.html` file. `app.component.html` is the template file for the default component of Angular app. By default it looks like this.



Welcome to app!

*DefaultComponent.png*

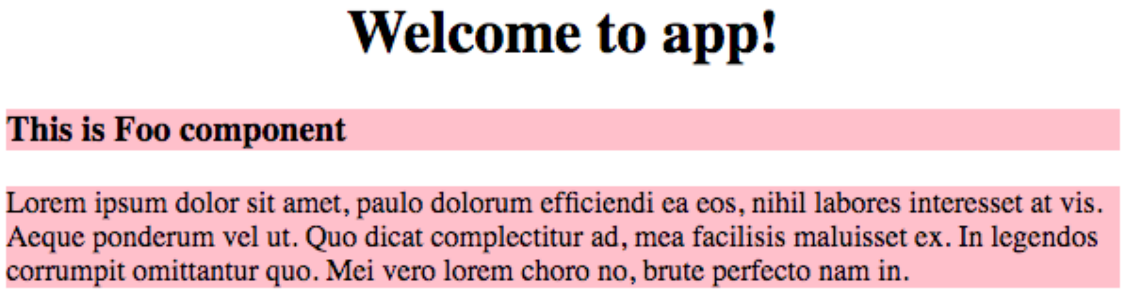
The default content of `app.component.html` file

```
<!--The content below is only a placeholder and can be replaced.-->
<div style="text-align:center">
  <h1>
    Welcome to {{title}}!
  </h1>
</div>
```

We will insert the following line at the end of `app.component.html` with this line

```
<app-foo></app-foo>
```

Notice the appearance changes to



Welcome to app!

This is Foo component

Lorem ipsum dolor sit amet, paulo dolorum efficiendi ea eos, nihil labores interesset at vis. Aeque ponderum vel ut. Quo dicat complectitur ad, mea facilisis maluisset ex. In legendos corrumpit omittantur quo. Mei vero lorem choro no, brute perfecto nam in.

*Default+Foo.png*

The whole content of the `Foo` component is inserted to default component by inserting its `Foo`’s selector `app-foo`.