**Directives-**

Directives are classes that add additional behavior to elements in your Angular applications. With Angular's built-in directives, you can manage forms, lists, styles, and what users see.

Or

When template is rendered on the browser, DOM elements / DOM tree gets changed according to the instruction given by Directives. They allow you to manipulate the DOM.

Component

Structural

Attributes

Custom

**Component-**

By using this, we can design the web pages in reusable manner and component approach.

**Structural**

It modifies the document object model that is DOM, it means add or remove attributes.

* [NgIf](https://angular.io/guide/built-in-directives#ngIf)—conditionally creates or disposes of subviews from the template.
* [NgFor](https://angular.io/guide/built-in-directives#ngFor)—repeat a node for each item in a list.
* [NgSwitch](https://angular.io/guide/built-in-directives#ngSwitch)—a set of directives that switch among alternative views.

**Attributes-**

Attribute directives listen to and modify the behavior of other HTML elements, attributes, properties, and components.

Many NgModules such as the [RouterModule](https://angular.io/guide/router" \o "Routing and Navigation) and the [FormsModule](https://angular.io/guide/forms" \o "Forms) define their own attribute directives. The most common attribute directives are as follows:

[NgClass](https://angular.io/guide/built-in-directives#ngClass)—adds and removes a set of CSS classes.

[NgStyle](https://angular.io/guide/built-in-directives#ngstyle)—adds and removes a set of HTML styles.

[NgModel](https://angular.io/guide/built-in-directives#ngModel)—adds two-way data binding to an HTML form element.

When template is rendered on browser then tree gets changed according to the instruction by directives, they allow you to manipulate the term.

What is DOM- Document object Model.

Whenever you load any html page, then it will check the syntax error, closing tag, etc. then it will create the DOM.

When the application is being loaded on browser at that time browser will create the DOM.

Structural- it is responsible for dom manipulation. It manipulates the dom by adding, removing and replacing the elements in dom.

Example-

<p>children works!</p>

<div \*ngFor="let course of courses">{{course}}</div>

Ts file

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

@Component({

  selector: 'app-children',

  templateUrl: './children.component.html',

  styleUrls: ['./children.component.scss']

})

export class ChildrenComponent implements OnInit {

  constructor() { }

  courses :any=['Core Java','Advanced Java','Spring','Frameworks']

  ngOnInit(): void {

  }

}