**Module Level Dependency Injection**

Angular Module: The module is a container for the different parts of an application.

Modular based programming will help in 2 ways

1. Parallel development
2. For ex:we have below modules to create

* Login
* Registration
* Products
* Checkout

4 teams are working on 4 different functionalities like login, registration etc.,

**Creating a module :-**

Syntax: angular. module()

This will take 2 parameters, Module name, and array



Module name is attached with ng-app in our html page

**Header Module**

(function(){

Angular.module(“header”,[]);

})();

We are developing individual pieces and making it available to other applications

Add header to the blogger



And refer that in the application



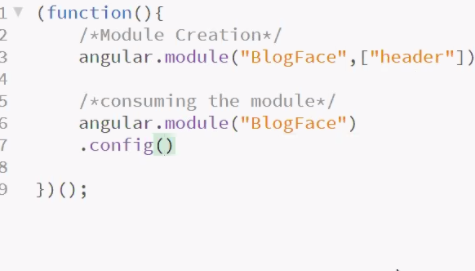
Note: First dependent has to be created first, so we are iirst

Each and every module we can create a configuration including header in our application

**Config Function:**

Whenever we are creating a module, automatically a config function will be created internally

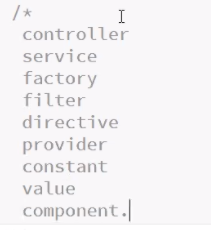
But if we want to see we can create a config function as well by extending the already default config function



Call back function which will trigger the function

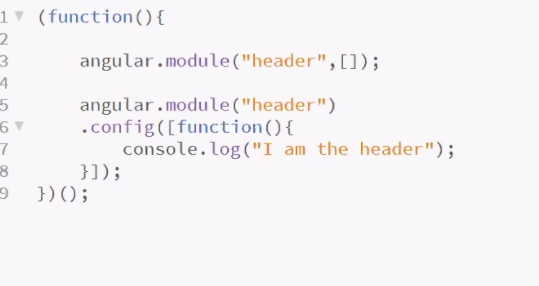


The moment we are declaring a module, these features can do



Header.js

We will extend the config functionality



Interw qsns

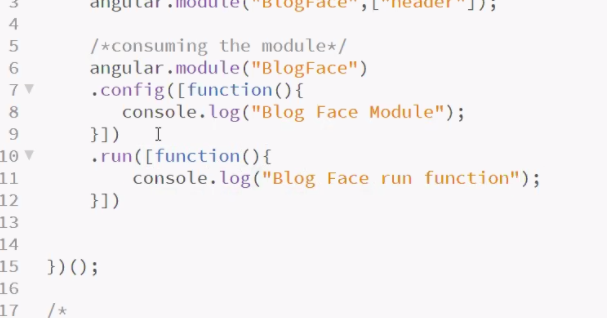
Whenever create any app in angular first function will create is config

Angular.module(“”).config(callback function);

Run will be the last function.

Angular.module(“header”).run([function(){

}



Order of execution of blocks

