---
What is dependency injection and what are the benefits of it?
---
Dependency injection is a powerful design pattern that allows separating the concerns of different components in an application and provides a way to inject the dependent component into the client component.

Consider the below code:

```
myApp.controller('myController', function ($scope, $http, $location)
  {
    //logic
  });
```
Here, a controller is declared with its dependencies.

.$http, $location are all services which are injected into the controller as a dependent entity. All of them have some i ndependent specific task associated with it. MyController does not need to create their instance, but it can directly us e them.

dependency इंजेक्शन एक शक्तिशाली डिजाइन पैटर्न है जो किसी एप्लिकेशन में विभिन्न components की चिंताओं को अलग करने की अनुमति देता है और क्लाइंट component में dependent component को इंजेक्ट करने का एक तरीका प्रदान करता है।

नीचे दिए गए कोड पर विचार करें:

```
myApp.controller('myController', function ($scope, $http, $location)
    {
        //logic
    });
```
यहां, इसकी dependencies के साथ एक controller घोषित किया गया है।

.$http, $location वे सभी सेवाएँ हैं जिन्हें एक dependent entity के रूप में controller में inject किया जाता है । उन सभी के साथ कोई न कोई independent specific कार्य जुड़ा हुआ है। MyController को अपना instance बनाने की आवश्यकता नहीं है, लेकिन यह सीधे उनका उपयोग कर सकता है।
==========

---
What do you understand by Dependency Injection in AngularJS?
---
Dependency Injection (also called DI) is one of the best features of AngularJS. It is a software design pattern where objects are passed as dependencies rather than hard coding them within the component. It is useful for removing hard-coded dependencies and making dependencies co nfigurable. To retrieve the required elements of the application that need to be configured when the module is loaded, the "config" operation uses Dependency Injection. It allows separating th e concerns of different components in an application and provides a way to inject the depende nt component into the client component. By using Dependency Injection, we can make compon ents maintainable, reusable, and testable.

A simple case of dependency injection in AngularJS is shown below:

```
myApp.controller('myController', function ($scope, $http, $location)
    {
        //logic
    });
```
Here, a controller is declared with its dependencies.

AngularJS provides the following core components which can be injected into each other as de pendencies:

Value
Factory
Service
Provider
Constant

डपिेंडेंसी इंजेक्शन (जसिे DI भी कहा जाता है) AngularJS की सबसे अच्छी वशिेषताओं में से एक है। यह एक सॉ फ्टवेयर डजिाइन पैटर्न है जहां वस्तुओं को components के भीतर हार्ड कोडगि के बजाय dependencies के रू प में pass कयिा जाता है। यह हार्ड-कोडेड dependencies को हटाने और dependencies को configuration य ोग्य बनाने के लएि उपयोगी है। एप्लकिेशन के आवश्यक तत्वों को पुनः प्राप्त करने के लएि जन्हिें मॉड्यूल लो ड होने पर कॉन्फ़गिर करने की आवश्यकता होती है, "कॉन्फ़गिरेशन" ऑपरेशन डपिेंडेंसी इंजेक्शन का उपयोग करत ा है। यह एक एप्लकिेशन में वभिन्नि components की चतिाओं को अलग करने की अनुमतदिेता है और क्लाइंट component में dependent component को इंजेक्ट करने का एक तरीका प्रदान करता है। dependency inj ection का उपयोग करके, हम components को maintainable, reusable, and testable योग्य बना सकते हैं।

AngularJS में dependency injection का एक साधारण example नीचे दखिाया गया है:

```
myApp.controller('myController', function ($scope, $http, $location)
    {
        //logic
    });
```

यहां, इसकी dependency के साथ एक controller घोषति कयिा गया है।

AngularJS नम्निलखिति मुख्य component प्रदान करता है जन्हिें controller के रूप में एक दूसरे में इंजेक्ट कयिा जा सकता है:

Value
factory
service
Provider
Constant

==========

---

What is the difference between compile and link?

---

Compile can be considered as a service which traverses the HTML, find for all the directives a nd returns a link function.
कंपाइल को एक ऐसी सेवा के रूप में माना जा सकता है जो एचटीएमएल को traverse करती है, सभी directive को ढूंढती है और एक लकि फ़ंक्शन लौटाती है।

Link, on the other hand, combines the model with a view. Any change in model reflects the ch ange in view and any change in view reflects in the model. It is executed once the template h as been cloned.

दूसरी ओर, लकि मॉडल को एक view के साथ जोड़ता है। मॉडल में कोई भी परविर्तन view में परविर्तन को दर् शाता है और view में कोई भी परविर्तन मॉडल में reflect होता है। टेम्प्लेट के क्लोन होने के बाद इसे execute d कयिा जाता है।

==========
---
What do you know about injector?
---
An injector is referred to as a service locator. It is used to receive object instances as defined by the providers, invoke methods, instantiate types, and load modules. Each Angular application consists of a single injector which helps to look upon an instance(Instance variables are nothing but called properties of the object) by its name.

इंजेक्टर को सर्वसि लोकेटर कहा जाता है। इसका उपयोग प्रदाताओं द्वारा परभिाषति ऑब्जेक्ट इंस्टेंस प्राप्त करने के लिए कयिा जाता है, वधियिों को आमंत्रति करता है, तत्काल प्रकार, और लोड मॉड्यूल लोड करता है। प्रत्येक कोणीय अनुप्रयोग में एक एकल इंजेक्टर होता है जो एक उदाहरण को उसके नाम से देखने में मदद करता है।