https://ajax.googleapis.com/ajax/libs/angularjs/1.8.3/angular.min.js

AngularJS

- It is one of the open-source, front-end, JavaScript-based frameworks widely used in creating single page applications on the client side.

- It has been developed by Google.

AngularJS

- It is capable of transferring the entire content directly from the server to the browser, loading all the pages at the same time.

- Once they are loaded, the page does not reload if a link is clicked upon; instead, the sections within the page are updated instantly.

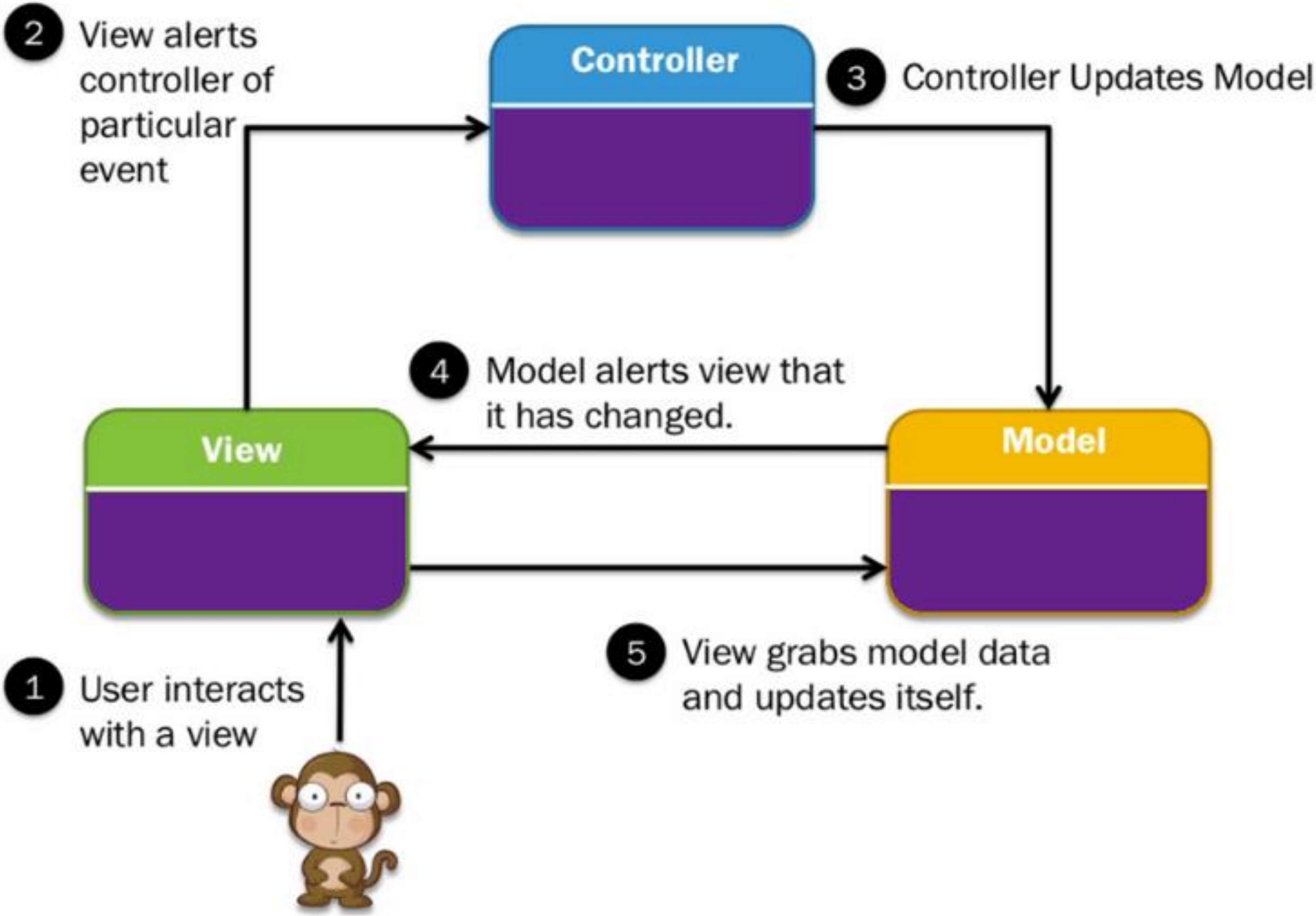
AngularJS

- AngularJS enables faster loading of pages as there is less load on the server.
- The two-way data binding feature saves the burden of writing numerous codes.
- Here, automatic integration of data between model and view components takes place. Further, AngularJS works on MVC architecture.

AngularJS

- It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC).
- The **Model-View-Controller (MVC)** framework is an architectural pattern that separates an application into three main logical components Model, View, and Controller.
- Each architecture component is built to handle specific development aspect of an application. MVC separates the business logic and presentation layer from each other.

MVC Architecture



2 View alerts controller of particular event
Controller
3 Controller Updates Model
4 Model alerts view that it has changed.
View
Model
5 View grabs model data and updates itself.
1 User interacts with a view

Real life example

Every car consist of three main parts.

- View= User interface : (Gear lever, panels, steering wheel, brake, etc.)
- Controller- Mechanism (Engine)
- Model- Storage (Petrol or Diseal tank)

AngularJS

- AngularJS is a structural framework for dynamic web apps.
- It lets you extend HTML's syntax to express your application's components clearly.
- It eliminates much of the code you would otherwise have to write.

AngularJS

- It has an additional custom HTML attributes embedded into it.
- Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variable
- The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

AngularJS

- It can be added to an HTML page with a <script> tag.
- AngularJS extends HTML attributes with Directives, and binds data to HTML with Expressions.

AngularJS Expressions

- Expressions are used to bind application data to HTML.
- Expressions are written inside double curly braces such as in {{ expression}}.
- AngularJS expressions are pure JavaScript expressions and output the data where they are used.

AngularJS directives

AngularJS directives are used to extend HTML. They are special attributes starting with **ng**-prefix.

- **ng-app** − This directive defines and links an AngularJS application to HTML.
- **ng-init** − This directive initializes application data.
- **ng-model** − This directive binds the values of AngularJS application data to HTML input controls.
- **ng-bind** − This directive binds the AngularJS Application data to HTML tags.

- **ng-repeat** - Instantiate an element once per item from a collection.
- **ng-controller** - Specifies a JavaScript controller class that evaluates HTML expressions.
- **ng-options** - To create a dropdown list, based on an object or an array in AngularJS, you can use the ng-options directive. But you can use ng-repeat also.
- **ng-disabled -** Binds AngularJS application data to the disabled attribute of HTML elements.

AngularJS directives

ng-app –

- The ng-app directive starts an AngularJS Application.
- It defines the root element.
- You can only have one **ng-app** directive in your HTML document.
- It automatically initializes or bootstraps the application when the web page containing AngularJS Application is loaded.

AngularJS directives

- Typically ng-app directives should be placed at the root of an HTML document e.g. <html> or <body> tag, so that it can control the entire DOM hierarchy. However, you can place it in any DOM element.

```
<div ng-app = ""> ... </div>
```

AngularJS directives

ng-init –

The ng-init directive initializes an AngularJS Application data. It is used to assign values to the variables.

AngularJS directives

ng-model –

- The ng-model directive defines the model/variable to be used in AngularJS Application.
- It is used to bind the **data** in your **model** to the **view** presented to the user. The **ng-model** attribute is used for binding controls such as input, text area and selects in the **view** into the **model**

AngularJS modules

- AngularJS supports modular approach. Modules are used to separate logic such as services, controllers, application etc. from the code and maintain the code clean.
- We can define modules in separate js files and name them as per the module.js file.
- The module is a container for the application controllers.
- A module is created by using the AngularJS function angular.module

Angular js controllers

- The controller in AngularJS is a JavaScript function that maintains the application data and behavior using $scope object.

- You can attach properties and methods to the $scope object inside a controller function, which in turn will add/update the data and attach behaviours to HTML elements. The $scope object is a glue between the controller and HTML.

Scope

- As a part of the "MVC" architecture, the scope forms the "Model", and all variables defined in the scope can be accessed by the "View" as well as the "Controller".

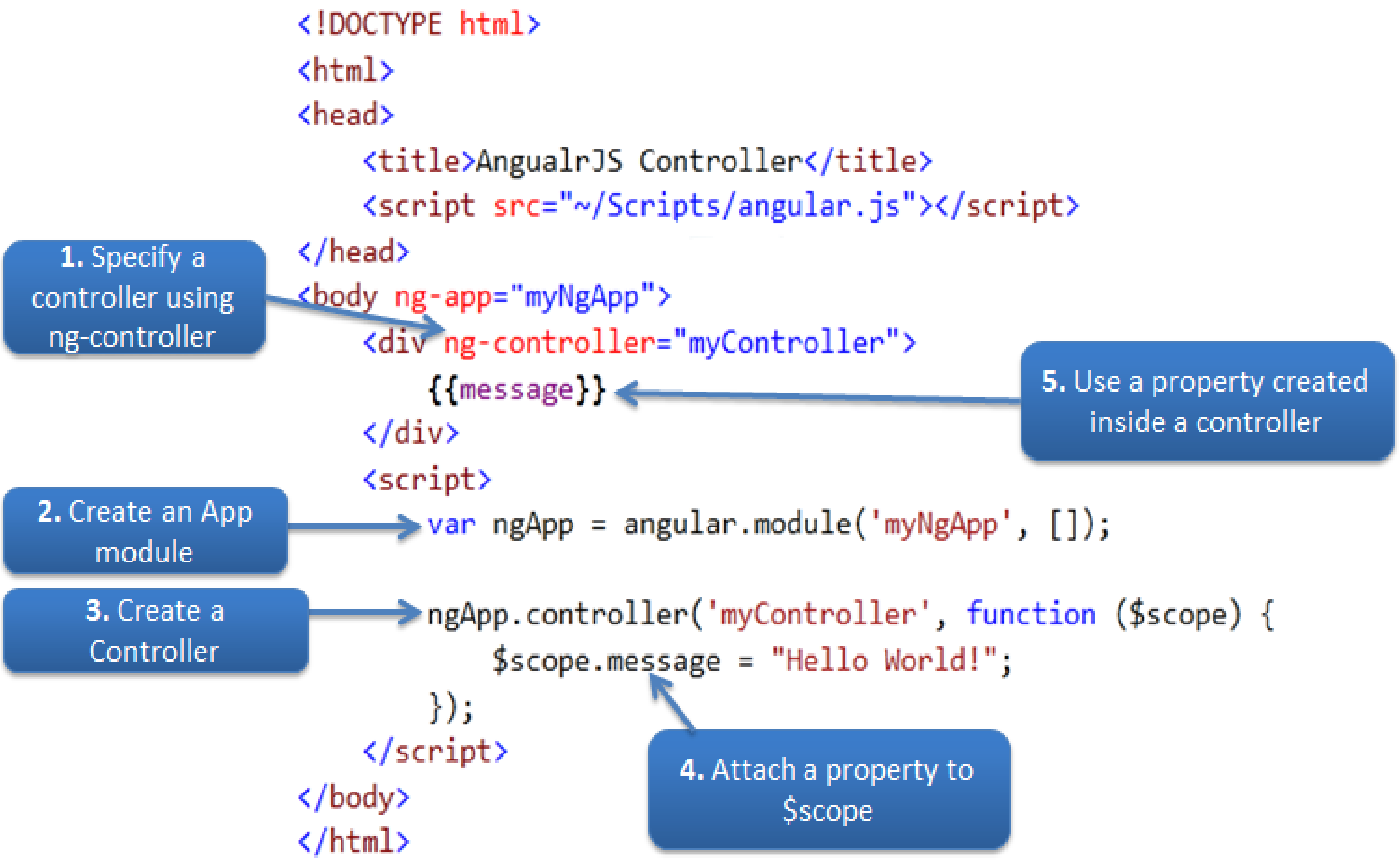
- The scope behaves as a glue and binds the "View" and the "Controller".

Scope

- Scope contains model data. In controllers, model data is accessed via $scope object.

- The $scope service in Angular detects changes to the model section and modifies HTML expressions in the view via a controller.

- Likewise, any alterations to the view are reflected in the model.



```
<!DOCTYPE html>
<html>
<head>
    <title>AngualrJS Controller</title>
    <script src="~/Scripts/angular.js"></script>
</head>
<body ng-app="myNgApp">
    <div ng-controller="myController">
        {{message}}
    </div>
    <script>
        var ngApp = angular.module('myNgApp', []);

        ngApp.controller('myController', function ($scope) {
            $scope.message = "Hello World!";
        });
    </script>
</body>
</html>
```

1. Specify a controller using ng-controller

2. Create an App module

3. Create a Controller

4. Attach a property to $scope

5. Use a property created inside a controller