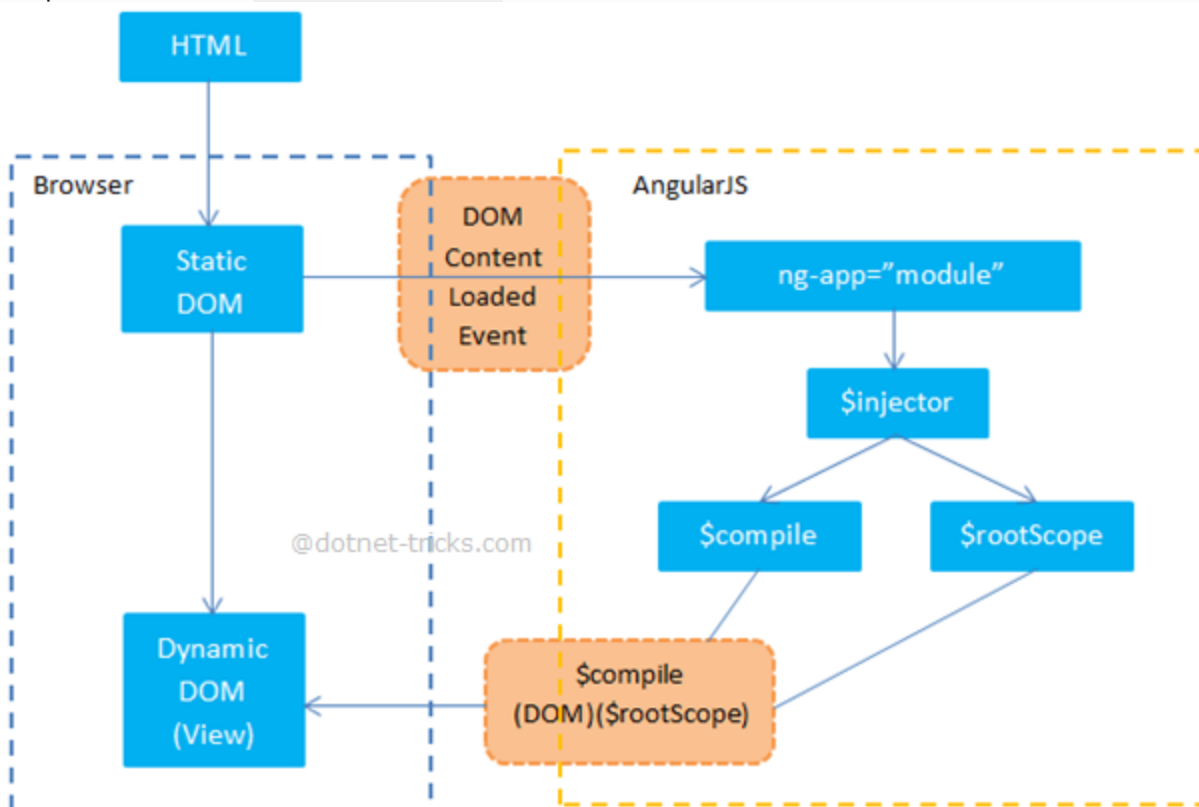


Angular initializes automatically upon `DOMContentLoaded` event or when the `angular.js` script is downloaded to the browser and the `document.readyState` is set to `complete`. At this point AngularJS looks for the `ng-app` directive which is the root of angular app compilation and tells about AngularJS part within DOM. When the `ng-app` directive is found then Angular will:

1. Load the module associated with the directive.
2. Create the application injector.
3. Compile the DOM starting from the `ng-app` root element.

This process is called `auto-bootstrapping`.



```

1. <html>
2. <body ng-app="myApp">
3. <div ng-controller="Ctrl">
4.   Hello !
5. </div>
6.
7. <script src="lib/angular.js"></script>
8. <script>
9.   var app = angular.module('myApp', []);
10.  app.controller('Ctrl', function ($scope) {
11.    $scope.msg = 'World';
12.  });
13. </script>
14. </body>
15. </html>

```

Manual Bootstrap Process

You can manually initialize your angular app by using `angular.bootstrap()` function. This function takes the modules as parameters and should be called within `angular.element(document).ready()` function. The `angular.element(document).ready()` function is fired when the DOM is ready for manipulation.

```
1. <html>
2. <body>
3.   <div ng-controller="Ctrl">
4.     Hello !
5.   </div>
6.   <script src="lib/angular.js"></script>
7.
8.   <script>
9.     var app = angular.module('myApp', []);
10.    app.controller('Ctrl', function ($scope) {
11.      $scope.msg = 'World';
12.    });
13.
14.    //manual bootstrap process
15.    angular.element(document).ready(function () {
16.      angular.bootstrap(document, ['myApp']);
17.    });
18.   </script>
19. </body>
20. </html>
```

Note

1. You should not use the `ng-app` directive when manually bootstrapping your app.
2. You should not mix up the automatic and manual way of bootstrapping your app.
3. Define modules, controller, services etc. before manually bootstrapping your app as defined in above example.