Standard structure

app/

----- controllers/

---------- mainController.js

---------- otherController.js

----- directives/

---------- mainDirective.js

---------- otherDirective.js

----- services/

---------- userService.js

---------- itemService.js

----- js/

---------- bootstrap.js

---------- jquery.js

----- app.js

views/

----- mainView.html

----- otherView.html

A better structure and foundation

app/

----- shared/ // acts as reusable components or partials of our site

---------- sidebar/

--------------- sidebarDirective.js

--------------- sidebarView.html

---------- article/

--------------- articleDirective.js

--------------- articleView.html

----- components/ // each component is treated as a mini Angular app

---------- home/

--------------- homeController.js

--------------- homeService.js

--------------- homeView.html

---------- blog/

--------------- blogController.js

--------------- blogService.js

--------------- blogView.html

----- app.module.js

----- app.routes.js

assets/

----- img/ // Images and icons for your app

----- css/ // All styles and style related files (SCSS or LESS files)

----- js/ // JavaScript files written for your app that are not for angular

----- libs/ // Third-party libraries such as jQuery, Moment, Underscore, etc.

index.html

[Benefits of the Modularized Approach](https://scotch.io/tutorials/angularjs-best-practices-directory-structure#toc-benefits-of-the-modularized-approach)

Code Maintainability

Follow the approach above will logically compartmentalize your apps and you will easily be able to locate and edit code.

Scalable

Your code will be much easier to scale. Adding new directives and pages will not add bloat to existing folders. Onboarding new developers should also be much easier once the structure is explained. Additionally, with this approach, you will be able to drop features in and out of your app with relative ease so testing new functionality or removing it should be a breeze.

Debugging

Debugging your code will be much easier with this modularized approach to app development. It will be easier to find the offending pieces of code and fix them.

Testing

Writing test scripts and testing modernized apps is a whole lot easier then non-modularized ones.