

环境配置:

Setup: node, npm. Jshint: report code's syntactic or structural problems

↳ { browser
browserify: load modules of client-side code
devel

Test { Jasmine: test framework
Sinon: mock HTTP request
Karma: test runner
PhantomJs: browser

Write code { Lo-dash
jQuery

Bootstrap:

Example App:

```
<body ng-app="myExampleApp">  
  <div ng-controller="ExampleController">  
    ...  
  </div>  
</body>
```

Bootstrap automatically

find element
↓
load modules
↓
create injector
↓
compile element

```
angular.module('myExampleApp', [])  
  .controller('ExampleController', function() {  
    ...  
  });
```

ngController Directive

find constructor, \$controller service
↓
attaching controller to the scope
↓
dependency injection
↓
should create a new (non-isolated) scope.

Modules and Injector:

modules and injector:

① angular.module('myModule', [])

② var injector = createInjector(['myModule'],

global angular object { window.angular initialized in loader.js
initializing just once.

module loader

module method { takes a module name and an array of module's dependencies.
returns a module object
register a module and get it.

module don't contain components, only methods for creating components, injector contains run the tasks of creating when loading modules.

injector service { createInjector method return injector object.
take an array of module names,
invoke queue: a collection of tasks, when injector loads a module, the injector runs the tasks from its invoke queue.

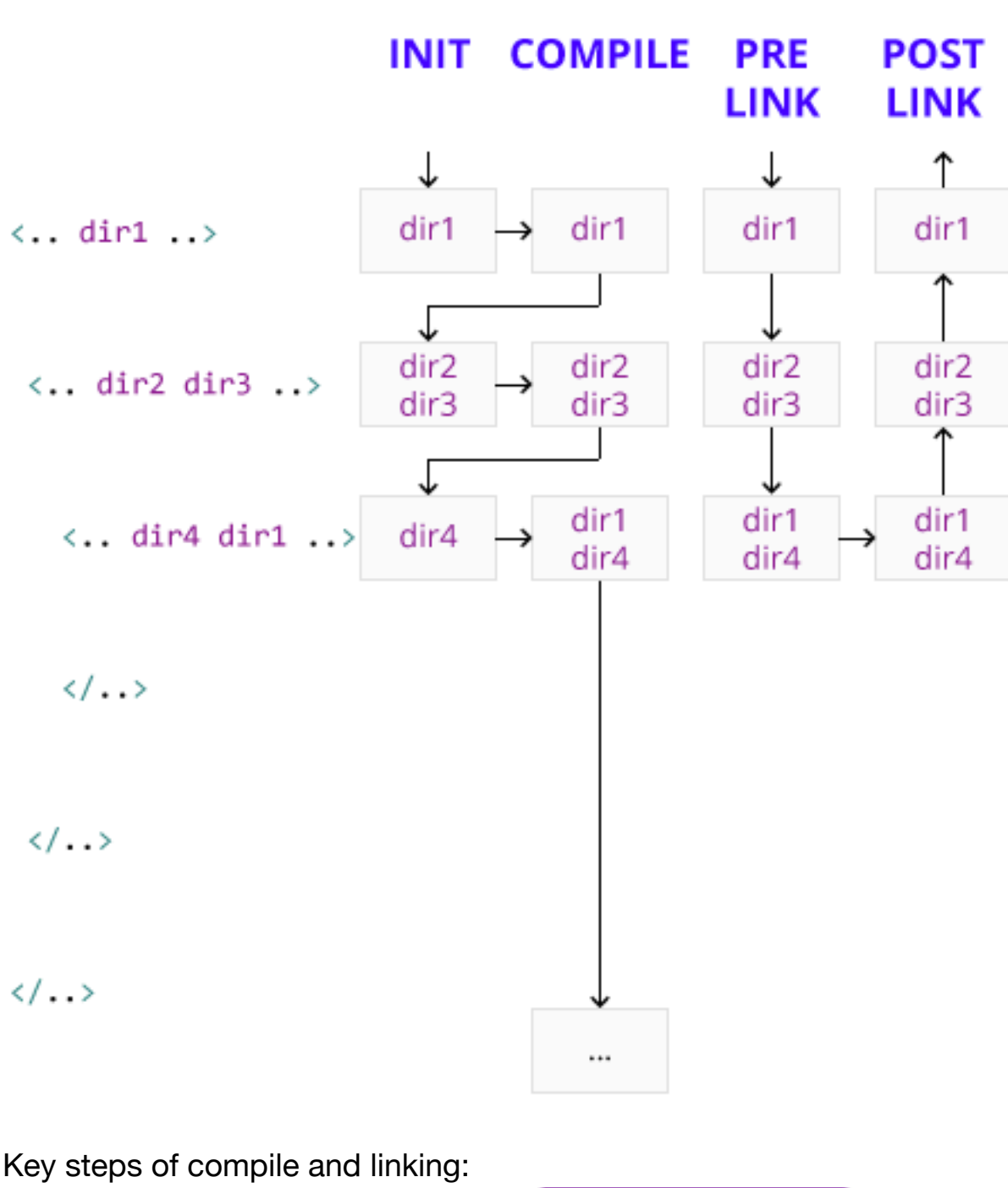
DI: dependency injection,

① give an injector a function → ② invoke function → ③ find the required dependencies.

injector.instantiate: uses use Object.create() prototype.

how to find? { ① explicitly, \$inject attribute.
② array style.

Compile and Linking:



Key steps of compile and linking:

