Related: [[Dev Guide: Best Practices]], [[Dev Guide: Anti-Patterns]]
Editors: put 2 spaces at the end of a question to create a `<br>`

1. **Why does it think the jQuery plugin is missing?**
   Remember: load jQuery *before* AngularJS if you are using jQuery plugins!

2. **How do I access the DOM from a controller?**
   DO NOT perform DOM selection/traversal from the controller. The HTML hasn't rendered yet. Look up 'directives'.

3. **Why does angular say my controllers/directives/etc are missing?**
   Calling `angular.module('myApp', [])` will ALWAYS create a new module (and wipe out your existing one). Instead, make sure to call `angular.module('myApp')` with only 1 parameter to refer to an already created module.

4. **How to render unescaped data?**
   $sce.trustAsHtml

5. **How can I watch when an array/object/$resource - result is modified? *
   *'scope.$watch' has a third parameter to monitor changes by value (and not by reference).

6. **How can I serialize form data for submitting?**
   Don't. Don't try to collect input values manually. Just attach `ng-model="data.myField"` onto every form input and take a look at `$scope.data` when you finally need it.

7. **Always have a '.' in your ng-models.**
   Misko best practice.

8. **How should I access the scope from services?**
   `$rootScope` is equivalent to the `ng-app` tag and can be injected into your bootstrap or services to add functions and values accessible on all scopes.
   NOTE: avoid doing this - it's the equivalent to defining global variables

9. **What is the difference between `module().factory()` and `module().service()`?**
   Checkout this discussion thread.

10. **How do I prevent** Flash Of Unstyled Content **(FOUC) (and curly braces)?**
    Leverage `ng-bind` instead of `{{...}}` and `ng-cloak` in a few places

11. **Why isn't `<a ng-click="go({{myVal}})">` working?**
    The only `ng-*` attributes that require `{{...}}` is `ng-src` and `ng-href` because the final result must be a string, *not* an expression. All others will work without.

12. **Nested Routes / Views?**
    Maybe…

13. **Can I specify templates or partials inline?**
    Yes! You can always do `<script id="some/partial.html" type="text/ng-template">` and angular will use it instead!

14. **How do I use a port in my `ngResource` url?**
    Escape it: `$resource('example.com\\:8080')`

15. **Why do plugins that trigger the `change` event not seem to work?**
    Angular watches the input event, not the 'change' event.

16. **Don't use jQuery to toggle crap. Just use a lot of variable flags inline** `<a ng-click="flags.open=!flags.open">...<div ng-class="{active:flags.open}">`

17. **How can I view the scope from the DOM inspector?**
    Google Chrome: install the Batarang extension, inspect a DOM element, *then* type `$scope` in the console
    Firefox / Firebug: inspect a DOM element, *then* type `angular.element($0).scope()` (or `$($0).scope()`) in the console
    IE 10+: Using the F12 tool, inspect a DOM element, *then* type `angular.element($0).scope()` (or `$($0).scope()`) in the console.

18. **Do you have any good directive samples / libraries?**
    AngularUI is an awesome collection of AngularJS tools (and even BETTER example code).

19. **Internet Explorer!?!**
    For IE v8.0 or earlier you may want to read this and use this.

20. **Do I have to use the # for the router?**
    `$locationProvider`

21. You should try using the AngularUI Passthru Directive (uiJq) before trying to roll your own jQuery plugin wrapper directive.

22. **Why is my `$scope.$watch()` firing recursively?**
    If you change `newVal` inside your `$scope.$watch(newVal, oldVal)` it could fire again (recursively?). After a `$watch()` is runs, the `$scope` is re-evaluated and relevant watchers will be re-triggered.

23. **When should I use `$scope.$apply()`?**
    You should **ONLY** use `$scope.$apply` in non-angular events/callbacks. It *usually* doesn't belong anywhere else.

24. **With `html5mode` enabled, how do I get the default `<a href>` behavior back?**
    If you want a link to cause a full page refresh then add `target="_self"` to your `<a>` tag

25. **How do I `.preventDefault()` or `.stopPropagation()`?**
    All `ng-click` and related bindings inject a `$event` object that you can call things like `.preventDefault()` or even pass the object to your methods

26. **AngularJS isn't working in my Chrome extension!**
    You want to use `ng-csp`

27. **How do I cachebust $http and html partials?**

```
myAppModule.config(function($routeProvider, $provide) {
  $provide.decorator('$http', function($delegate){
    var get = $delegate.get;
    $delegate.get = function(url, config){
      url += (url.indexOf('?') !== -1) ? '?' : '&';
      url += 'v=' + cacheBustVersion;
      return get(url, config);
    };
    return $delegate;
  });
});
```

Testing

1. **Rejecting / Resolving a `$q.defer()` doesn't go through**
   You must add a `$scope.$apply()` for these to process

2. **Jasmine `spyOn()` is not executing the spy'd function**
   Not necessarily an AngularJS question, but you need to append `.andCallThrough()`

3. **How do I test async code?**
   Focus on creating mocks (fake objects / functions) that let you flush the async stuff synchronously. `$timeout` has a `.flush()` method specifically for this reason, just remember to add a `$scope.$apply()` too. If that still doesn't work, look up Jasmine's `runs()` and `waits()` functions.

4. **How does `module()` and `inject()` work?**
   You can call `module()` as many times as you want, and spin up as many unrelated modules as you want at time **BEFORE** your very first `inject()` is called. All subsequent calls to `module()` afterwards will either behave incorrectly or fail. If you load modules that contain assets with the same name, the last module to have loaded for that specific test will take precedence. This is a useful and simplistic way to load mock assets.

5. **How can I mock or modify services?**
   Call `module()` and pass `$provide` to a callback. Use `$provide.decorator()` as an easy way to manipulate or replace services before they get injected into other services! Alternatively, you could just create `myApp.mocks.myService` modules that contain mock services with the identical names to their originals and load it on-demand. The last module you load will take precedence.