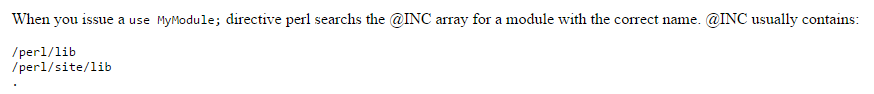
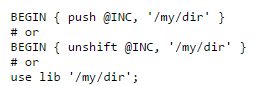
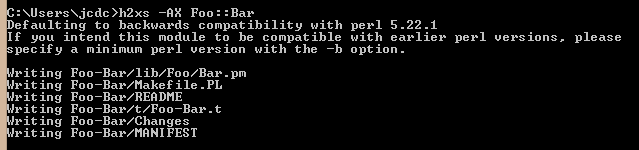
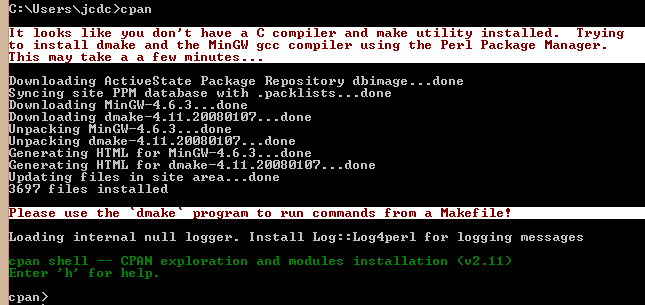
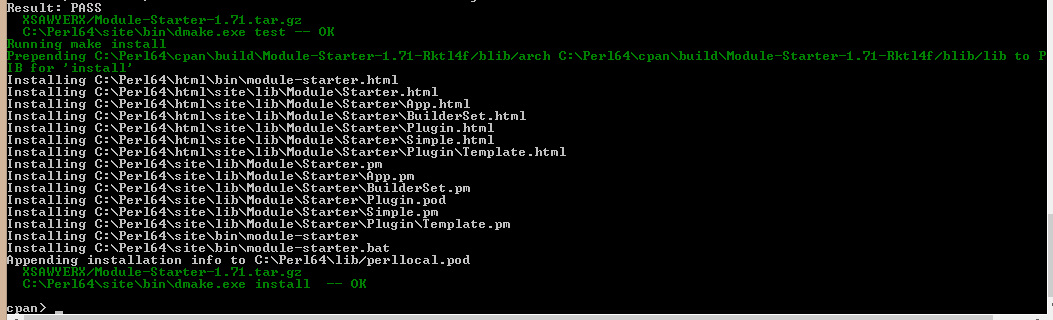
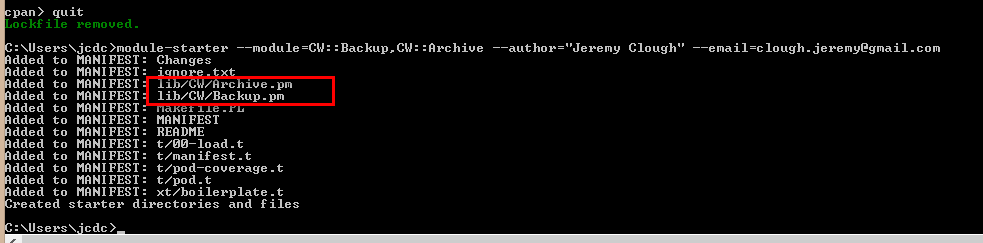
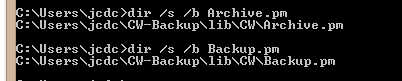
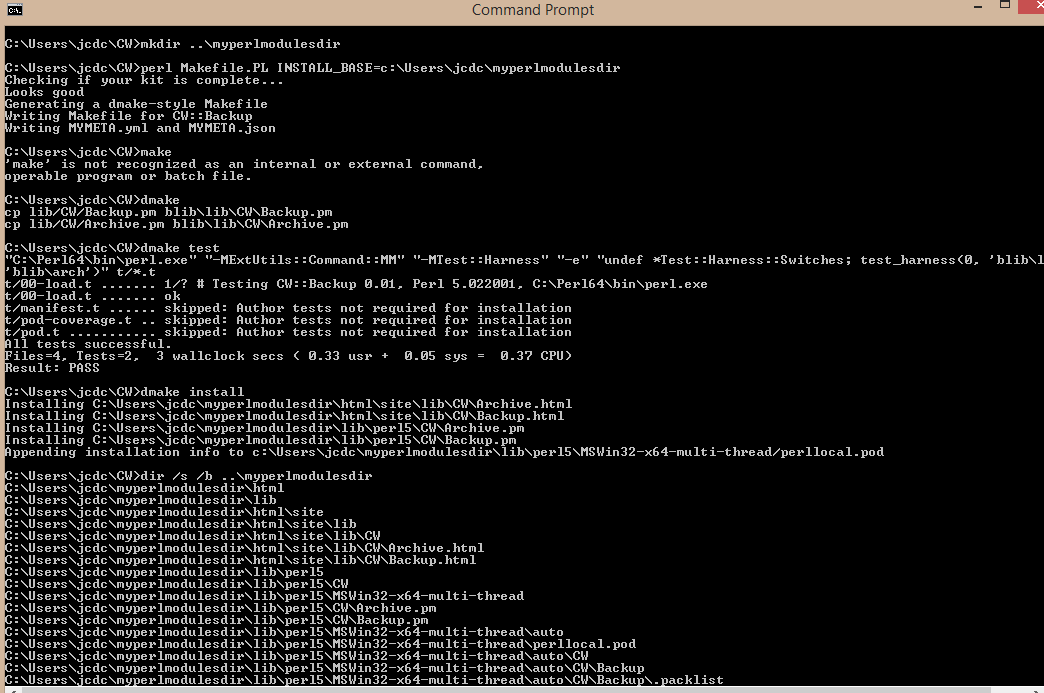
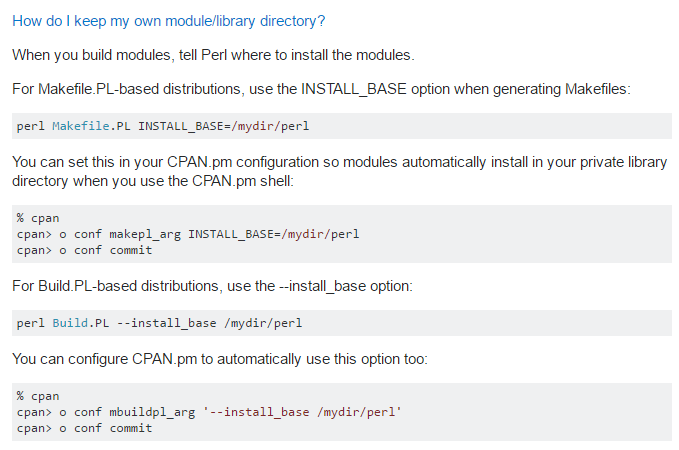
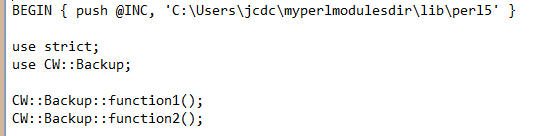
**Perl: Using Perl Modules**  
Simple module example and explanation:  
<http://www.perlmonks.org/?node=Simple%20Module%20Tutorial>

Perl pragmas “strict”, “warnings”, and “diagnostics”:  
<http://www.perlmonks.org/?node=use%20strict%20warnings%20and%20diagnostics%20or%20die>  
*"If you intend on building production-level code, then "use strict" and "use warnings" should be disregarded only if you know exactly why you are doing it"*  
**@INC**

Adding your own directory to the @INC path:  
  
  
**Note**: use Foo::Bar means search @INC for a \*subdir\* called "Foo" and a \*module\* called "Bar.pm".  
  
**Note:** After use’ing a module, when we say &Foo::Bar::some\_func we are referring to the \*package name\* not the (dir::)file name that we used in the use. This allows you to have many package names in one use'd file.

**Example: Create module stub files (http://www.perlmonks.org/?node\_id=158999)**  
h2xs -AX Foo::Bar  
  
**Note:** To create a multi-module package, you can either use h2xs and manually add further modules, or you can use Module::Starter (see later).  
  
**How to automatically install Module::Starter using the cpan command**(see <http://www.thegeekstuff.com/2008/09/how-to-install-perl-modules-manually-and-using-cpan-command/>)  
Then we install the module:  
  
**Note:** This installs **a lot** of stuff, so I probably wouldn’t want to do this on a ‘production’ Perl installation. Fortunately, we’re just installing this module in order to create module stub files.  
After quite a while:  


**Using Module::Starter to generate module stub files**module-starter --module=CW::Backup,CW::Archive --author="Jeremy Clough" [--email=clough.jeremy@gmail.com](mailto:--email=clough.jeremy@gmail.com)  
(see <http://search.cpan.org/~xsawyerx/Module-Starter-1.71/lib/Module/Starter.pm>)  
  
Which creates the following files:  
  
**Note:** I’ve renamed this “CW-Backup”directory to just “CW”, as it contains two modules: Backup.pm and Archive.pm.  
  
**Installing my new package to a non-standard installation directory**From <http://stackoverflow.com/questions/540640/how-can-i-install-a-cpan-module-into-a-local-directory>:  


**Using my new package (now installed in a non-standard directory)** **Note:** This does work, because I installed my new package into “C:\Users\jcdc\myperlmodulesdir”.  
  
**Using Module::Starter with the templates from book “Perl Best Practices” (untested by Me)**After installing Module::Starter::PBP:  
perl -MModule::Starter::PBP=setup  
  
*JeremyC 3/1/2017***END**