Puppet Module Cheat Sheet

- Modules are directories with a predictable structure.
- Puppet can automatically load manifests, files, and plugins from modules in its modulepath.
- Use `puppet master --configprint modulepath` to see where Puppet expects to find modules on your system.

Example Module: /etc/puppetlabs/puppet/modules/apache

manifests

This directory holds the module's Puppet code.
- Each .pp file should contain one and only one class or defined type.
- Filenames and class/defined type names are related; see the examples below.
- Within a module, the special `$module_name` variable always contains the module's name.

apache/manifests/init.pp

```
class apache {
  ...
}
```

Init.pp is special; it should contain a class (or define) with the same name as the module.

apache/manifests/vhost.pp

```
define apache::vhost ($port, $docroot) {
  ...
}
```

Other classes (and defines) should be named **modulename::filename** (without the .pp extension).

apache/manifests/config/ssl.pp

```
class apache::config::ssl {
  ...
}
```

Subdirectories add intermediate namespaces.

lib

This directory holds Ruby plugins, which can add features to Puppet and Facter.

apache/lib/puppet/type/apache_setting.rb

A custom type.

apache/lib/puppet/parser/functions/htpasswd.rb

A custom function.

apache/lib/facter/apache_confdir.rb

A custom fact.

files

Nodes can download any files in this directory from Puppet's built-in file server.
- Use the `source` attribute to download file contents from the server.
- Use `puppet:///` URIs to specify which file to fetch.
- Files in this directory are served at **puppet:///modules/modulename/filename**.

apache/files/httpd.conf

To fetch this file:

```
file {'/etc/apache2/httpd.conf':
  ensure => file,
  source => 'puppet:///modules/apache/httpd.conf',
}
```

apache/files/extra/ssl

Puppet's file server can navigate any subdirectories:

```
file {'/etc/apache2/httpd-ssl.conf':
  ensure => file,
  source => 'puppet:///modules/apache/extra/ssl',
}
```

templates

This directory holds ERB templates.
- Use the `template` function to create a string by rendering a template.
- Use the `content` attribute to fill file contents with a string.
- Template files are referenced as **modulename/filename.erb**.

apache/templates/vhost.erb

To use this template:

```
file {'/etc/apache2/sites-enabled/wordpress.conf':
  ensure  => file,
  content => template('apache/vhost.erb'),
}
```