# Module Structure

A module is just a directory with stuff in it, and the magic comes from putting tha
expects to find it. Which is to say, arranging the contents like this:

- `my_module` — This outermost directory's name matches the name of the mod
  - `manifests/` — Contains all of the manifests in the module.
    - `init.pp` — Contains a class definition. **This class's name must ma
      name.**
    - `other_class.pp` — Contains a class named **`my_module::other_clas`**
    - `my_defined_type.pp` — Contains a defined type named **`my_module:`**
    - `implementation/` — This directory's name affects the class names b
      - `foo.pp` — Contains a class named **`my_module::implementation`**
      - `bar.pp` — Contains a class named **`my_module::implementation`**
  - `files/` — Contains static files, which managed nodes can download.
  - `lib/` — Contains plugins, like custom facts and custom resource types.
  - `templates/` — Contains templates, which can be referenced from the mo
  - `tests/` — Contains examples showing how to declare the module's class