**2.1 What a Rule Looks Like**

A simple makefile consists of “rules” with the following shape:

*target* … : *prerequisites* …

*recipe*

…

…

A *target* is usually the name of a file that is generated by a program; examples of targets are executable or object files. A target can also be the name of an action to carry out, such as ‘clean’ (see [Phony Targets](https://www.gnu.org/software/make/manual/make.html#Phony-Targets)).

A *prerequisite* is a file that is used as input to create the target. A target often depends on several files.

A *recipe* is an action that make carries out. A recipe may have more than one command, either on the same line or each on its own line. **Please note:** you need to put a tab character at the beginning of every recipe line! This is an obscurity that catches the unwary. If you prefer to prefix your recipes with a character other than tab, you can set the .RECIPEPREFIX variable to an alternate character (see [Special Variables](https://www.gnu.org/software/make/manual/make.html#Special-Variables)).

Usually a recipe is in a rule with prerequisites and serves to create a target file if any of the prerequisites change. However, the rule that specifies a recipe for the target need not have prerequisites. For example, the rule containing the delete command associated with the target ‘clean’ does not have prerequisites.

A *rule*, then, explains how and when to remake certain files which are the targets of the particular rule. make carries out the recipe on the prerequisites to create or update the target. A rule can also explain how and when to carry out an action. See [Writing Rules](https://www.gnu.org/software/make/manual/make.html#Rules).

A makefile may contain other text besides rules, but a simple makefile need only contain rules. Rules may look somewhat more complicated than shown in this template, but all fit the pattern more or less.