Build Systems

Build Systems : Introduction

- A Build System is a piece of program that takes source code as input and produces a deployable software as output.
- The host machine contains the build system

Examples:

Make, Gradle, Maven, Ant e.t.c

E.g. of deployable software : An executable, a static library e.t.c.

Build Systems : Make

- The Make build system uses Makefiles.

- A Makefile is a text file with the name "Makefile" in a source directory, and it contains build targets and commands that tell Make how to build the current code base.

E.g. of deployable software : An executable, a static library e.t.c.

Makefiles : Introduction

- A simple makefile is made up of rules written in the following format :

```
target   :  prerequisite_1 prerequisite_2 prerequisite_n
        command_1
        command_2
        command_n
```

Note : A tab must be put at the beginning of every command line

Makefiles : Parts of a Rule

```
target   :  prerequisite_1 prerequisite_2 prerequisite_n
        command_1
        command_2
        command_n
```

Target :

The name of a file that is generated by a program; examples of targets are executable or object files. Can also be the name of the action to carry out.

Prerequisite :

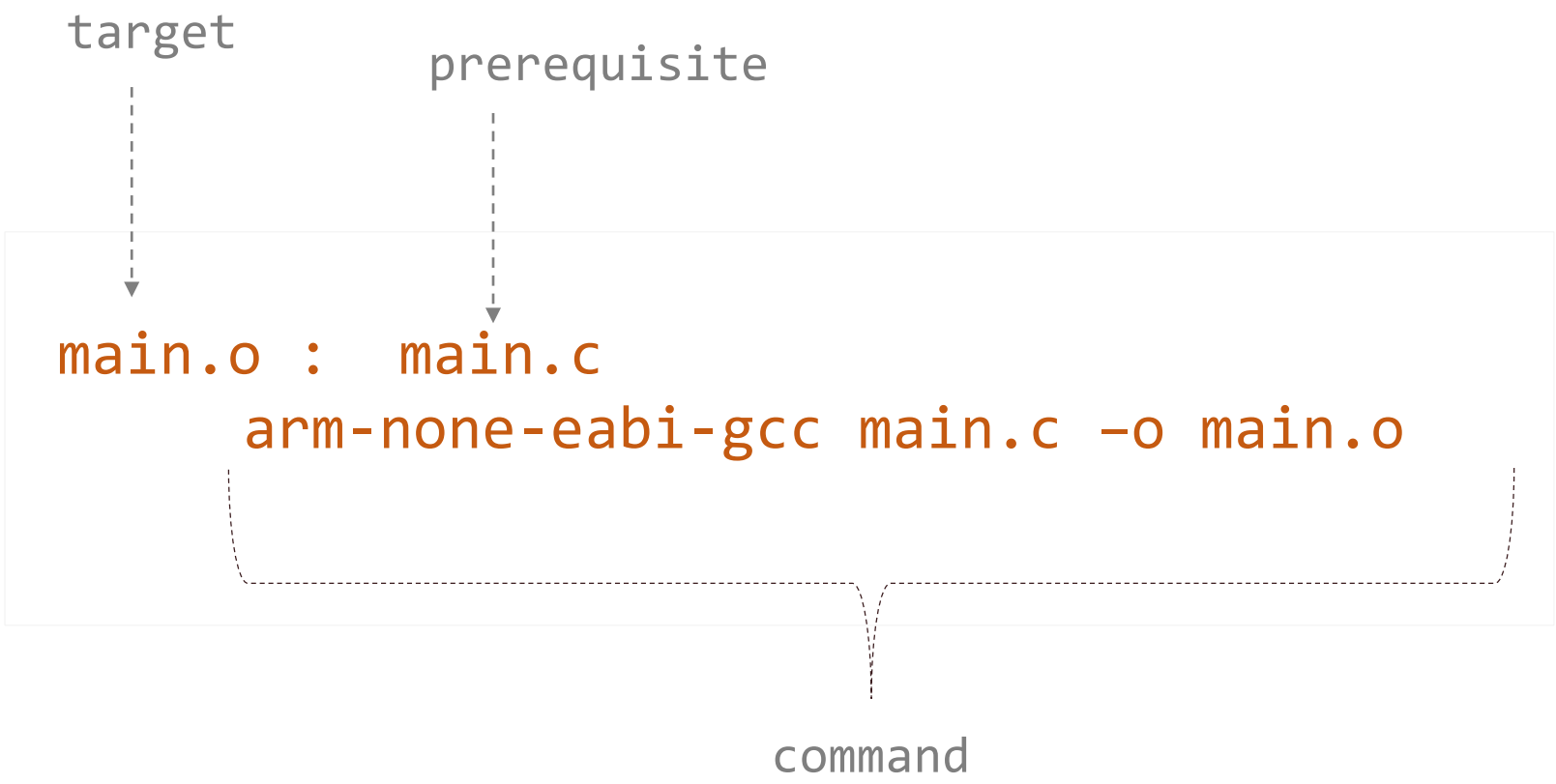
File used as input to create a target

Command :

Action carried out by *make*

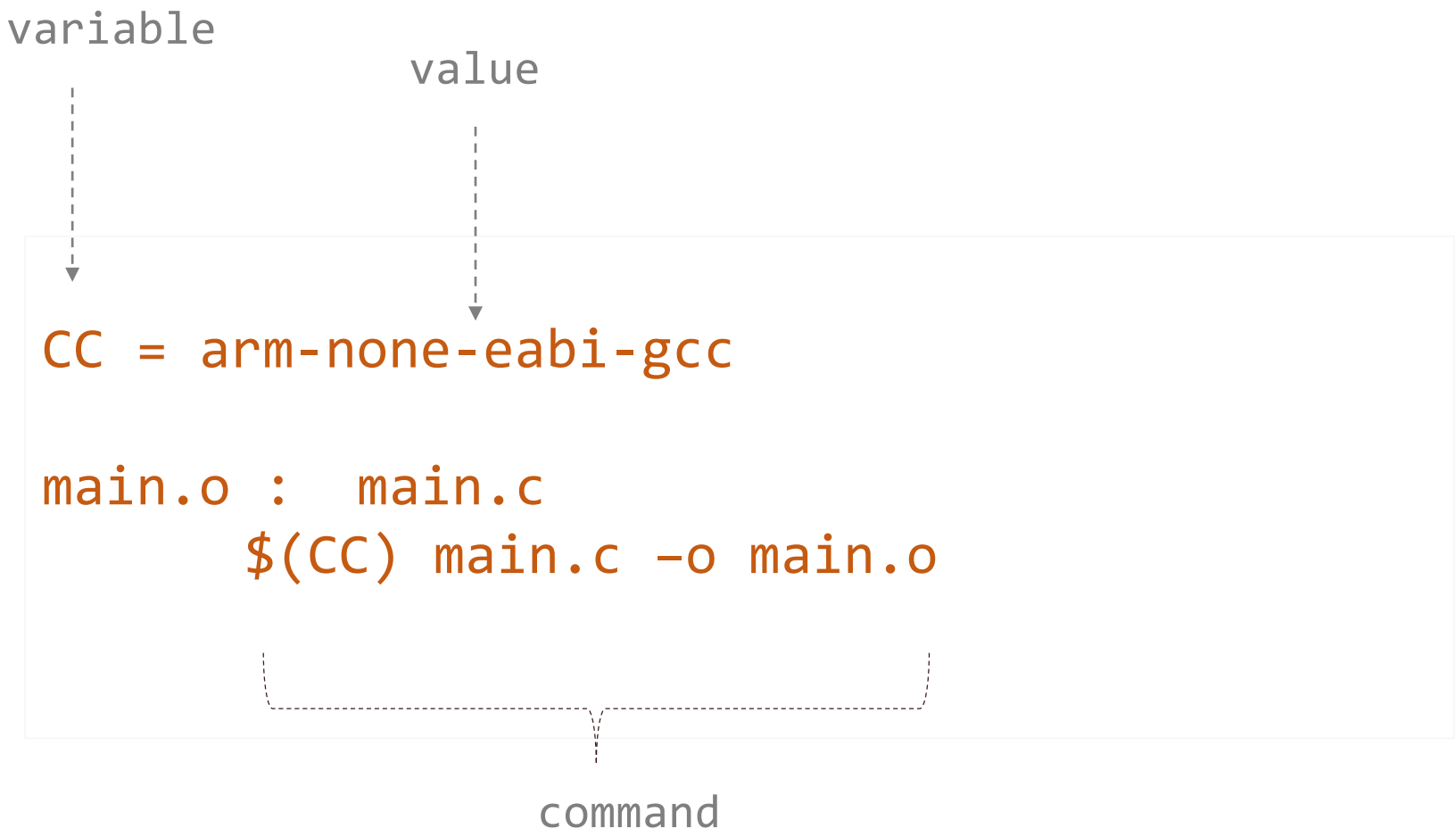
Note : A tab must be put at the beginning of every command line

Makefiles : Example of a Rule



Note : A tab must be put at the beginning of every command line

Makefiles : Using Variables



Makefiles : Special Variables

- Dependency can be replaced with `$^`
- Target can be replaced with `$@`

```
CC = arm-none-eabi-gcc

main.o :  main.c
	$(CC) $^ -o $@
```

command

Makefiles : How a Makefile is processed

- Make reads the Makefile in the current directory
- The first target is processed first. This is known as the ***default goal.***
- The other rules are processed because their targets appear as prerequisite of the ***goal.***
- If a rule is not depended on by the goal that rule is not processed unless it is explicitly called.