



# Making Makefiles

By Terry and Dylan

```
git clone git@github.com:edev/MakefileDemo.git
```



# Motivation

- Ease of flag organization
    - `g++ -g *.cpp -Wall`
  - Quick control of compiling
  - Repeatability
  - Ease of Development
- 
- Just type “make” to compile your programs



# Version One

A list of **build instructions** that make the compiler work in a specific way.

- Target
- Dependencies
- Recipe

For instance:

```
all: *.cpp *.h
    g++ -Wall -o cookiemonster *.cpp
```



## Version Two

```
all: *.cpp *.h
    g++ -Wall -o cookiemonster *.cpp

clean:
    rm cookiemonster
```

```
git clone git@github.com:edev/MakefileDemo.git
```



# Running a Makefile

```
make
```

```
ls
```

```
./cookiemonster
```

```
make clean
```

```
ls
```

```
git clone git@github.com:edev/MakefileDemo.git
```



# Version Three

```
all: *.cpp *.h
    g++ -Wall -o cookiemonster *.cpp

test: testThreeTimes

testThreeTimes: all
    ./cookiemonster < testThreeTimes.input

clean:
    rm cookiemonster
```

## Makefile

```
1 all: dir latex dvi ps mv
2
3 dir:
4     rm -rf build
5     mkdir build
6
7 latex: *.tex
8     latex -output-directory build main.tex
9
10 dvi: build/main.dvi
11     cd build && dvips main.dvi
12
13 ps: build/main.ps
14     cd build && ps2pdf main.ps
15
16 mv:
17     mv build/main.pdf Notes.pdf
```



# Questions?

(No, we don't have cookies. Sorry!)

Terry: [tet2@pdx.edu](mailto:tet2@pdx.edu)

Dylan: [lauf@pdx.edu](mailto:lauf@pdx.edu)