

VS Code

f1 → search settings

Ctrl (~) → bring up terminal

Bash

Pwd -- print working directory

Ls -- list current content

Cd/.. c/revature

Mkdir -- make directory

Rm -- remove files

Rm -r -- remove files and folders

Touch -- make empty file (essay.txt)

Cp oldfile.txt newfile.txt (copy)

Cp -r (copy everything in directory)

Mv file.txt /movehere (move) → use to rename (mv file.txt newfile.txt)

Git

Git clone <github.com/x/y/z>

Git pull

```
git config --global user.name "User Name"
```

```
git config --global user.email "email"
```

```
git config --global color.ui true
```

Git init → initialize git in current repository (adds hidden .git folder)

```
git remote add origin \ https://github.com/user/repo.git
```

Git push origin master

Git status → see if files are up to date

Git add <file>

Git add -A → adds everything in current directory

Git add . → add all file changes in current directory

Git add p → this will ask y' or no' for what changes to commit

Git commit → without -m it uses vim for commit message

Git commit <filename> -m 'first commit' // add and commit

Git commit -am 'message' → add and commit staged file

-- amend 'new message' → changes most recent

Git rebase -i (combine commits)

Git checkout -b branchname → create local branch

Git checkout branch_1 //switch to this branch

Git push -u origin branch_2 //upstream push to backup

Git branch -d branch_name → delete local branch → -D deletes even if hasn't merged

Git branch -a (view all branches) --merged or --no-merged

Git branch (just local) -r (for remote)

Git rebase origin/master

Git merge branch_2 → merge to master

Git reset head → head^ -- filename → head~3,5 → git reset (commit id) -- filename

Visual Studio

The project file specifies everything that's needed to restore dependencies and build the program. Output type (.exe) target framework .net core

Solution (container for related projects)

- Project (Unit of compilation and deploy)

- namespaces (naming container for classes)

- class (.cs file)

- properties / methods

- variables (in methods)

Two Projects

- console app

- application has main method (starting point)

- Class Library

- library has no main / cannot run

Dotnet --version

Dotnet new console --name HelloWorld

Dotnet run HelloWorld

alter helloworld

Dotnet build -- compile changes

Example

Mkdir animals

Cd animals

Dotnet new console --name animals.UI

Dotnet new sln

Dotnet sln add Animals.UI

Dotnet add reference /Animals.Library

Dotnet new classlib -- name Animals.Library

Code program.cs

<https://docs.microsoft.com/en-us/dotnet/csharp/>
<https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/index>