COMMAND LINE INTERFACE

CLI

Until now, you most likely use computers through a GUI. GUI stands for **G**raphic **U**ser **I**nterface, and it refers to the visual layer over the computer's actual core.

Starting today, we will be switching out the GUI for a faster version: the **CLI**!

OPENING TERMINAL

- Mac Users:
  - Open the “Terminal” app (Applications > Utilities > Terminal)
- Windows Users:
  - Install and open Git Bash (https://gitforwindows.org/)

COMMANDS

ls

List

The `ls` command is shows us what is in the current directory. Typing:

```
ls
```

Will give us a list of all the files and folders wherever we are

cd

Change Directory

The `cd` command is how we move between folders. Typing:

```
cd Desktop/
```

Will move us into the desktop folder

pwd

Print Working Directory

The `pwd` command tells us where we are. Typing:

```
pwd
```

Will return something like:

/Users/jackie/Desktop

RELATIVE VS ABSOLUTE PATHS

Relative paths are based off of where you currently are, while absolute paths are the exact address regardless of your current working directory.

For instance when I open my terminal I am in my user's directory and I can cd into my desktop using:

```
cd Desktop/
```

This is relative to where I currently am. But if I wanted to use an absolute path, I would need to specify exactly where it is:

```
cd Users/jackie/Desktop/
```

NAVIGATING TIPS

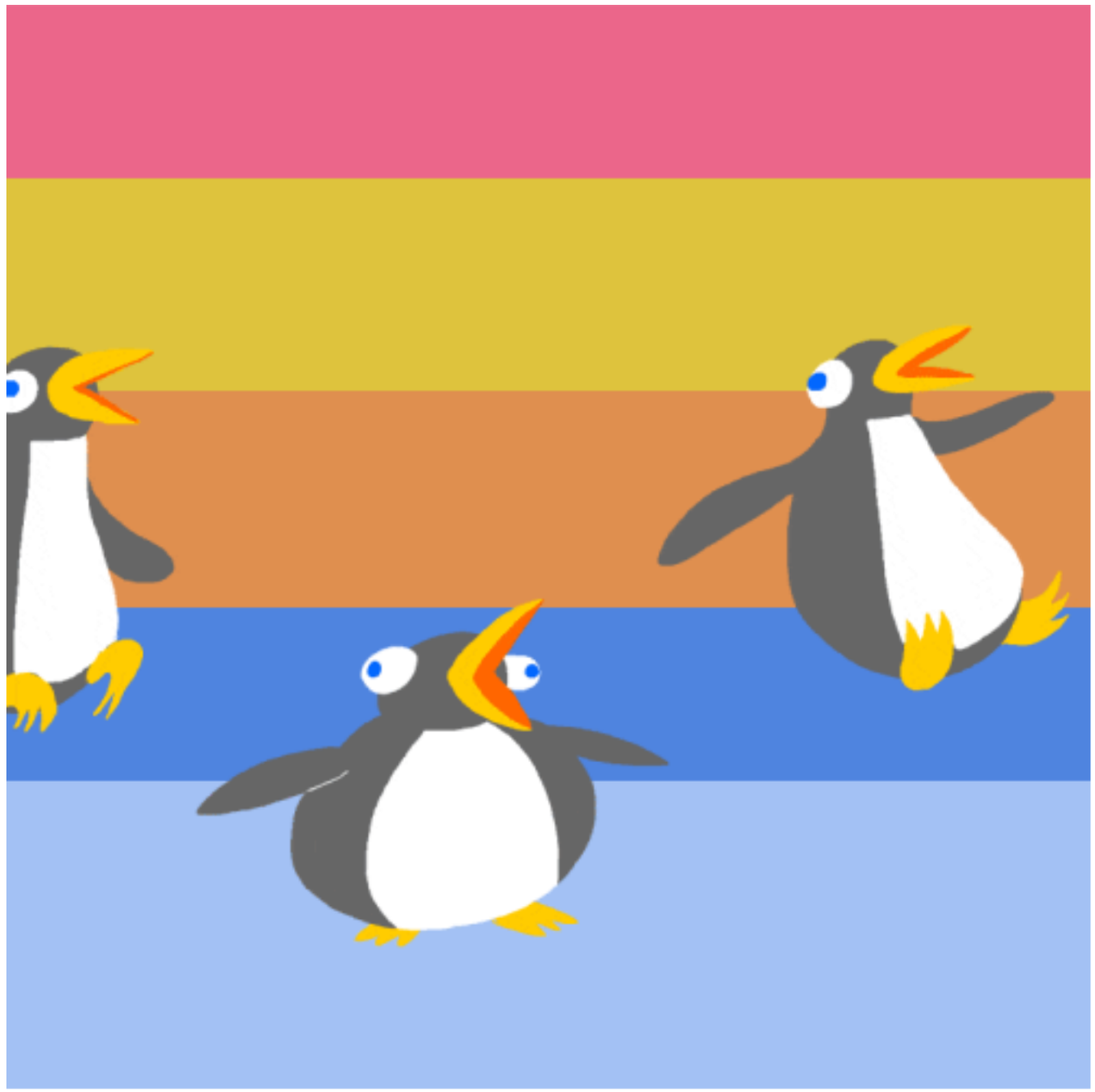
- When you first open terminal you will always be in your root directory (`~/`)
- You can always navigate back to the root with `cd ~/`
- One directory above can be referred to with `../` for instance typing `cd ../` will move you up a directory
- You can refer to the current directory with `./`
- You don’t need to type out the entire name of things! If you start and then press `tab` it will fill in the blanks for you!

CODEALONG

Lets try navigating around our computer

REVIEW

- `ls`
- `cd`
- `pwd`
- `Relative path`
- `Absolute path`
- `~/`
- `../`
- `./`



mkdir

Make Directory

The `mkdir` command creates a new directory in the current location. Typing:

```
mkdir code
```

Will create a new folder named “code”

touch

Create File

The `touch` command creates a new file in the given location. Typing:

```
touch index.js
```

Will create a new javascript file named “index”

mv

Move File or Directory

The `mv` command will move or rename a specified file or directory. Typing:

```
mv index.js code/index.js
```

Will move the file named index.js in the current directory to the code directory

rm

Remove

The `rm` command removes a given file or directory. Typing:

```
rm index.js
```

Will delete the index.js file

rm -rf

Remove (for directories that have stuff in them)

You can add options to the end of a command by using a -. For instance `rm -rf` adds two options:

r: says to remove the children

f: says to force the removal

WARNING

Using rm will **PERMANENTLY** remove the file or directory. There is no going back!

Wildcard

Using * works as a wildcard. It will match anything that resembles what you have typed. Typing:

```
cd Desk*
```

Will open up your desktop

CODEALONG

Lets create the directories we will use in this class!

REVIEW

- `ls`
- `cd`
- `pwd`
- Relative Path
- Absolute Path
- `~/`
- `../`
- `./`
- `mkdir`
- `touch`
- `mv`
- `rm`
- `rm -rf`
- `code`
- `*`

YOUR TURN

Practice using the CLI