BIT

Programming Principles

with **JavaScript**

BIT

Day overview

- **Getting to Know the Command Line**
- Code Versioning

BIT

Getting to Know the Command Line

The Ultimate Seat of Power on Your Computer

BIT

What Is the Command Line?

The command line is the ultimate seat of power on your computer.

Using the command line, you can perform amazing feats of wizardry and speed, taming your computer and getting it to do precisely what you want.

BIT

What Is the Command Line?

Unfortunately, the price of this power is **complexity**: nobody ever said that ruling your computer would be easy.

BIT

What Is the Command Line?

The command line is, at its heart, simply a place where you type commands to the computer.

The computer is your obedient servant, and will attempt to carry out any command that it understands.

B I T

What Is the Command Line?

In order to give it commands, we must first start learning the language of the computer.

BIT

What Is the Command Line?

NOTE

The command line, as with all power, has its risks. You have the capability to instruct the computer to do anything it has the capability of doing. If you instruct the computer to erase all of your data, it will cheerfully proceed to do so.

BIT

What Is the Command Line?

Do not run a command just to see what it does.

Make sure you understand what the command is supposed to do first, especially if the command involves changing or removing files.

BIT

Finding the Command Line

Most people don't use the command line on a regular basis, so it can be a bit difficult to find the first time.

The Windows operating system doesn't even have a proper command line built in — to execute these commands, you will have to install one.

BIT

Finding the Command Line

- macOS
  - The Mac command line is a program called Terminal
- Linux
  - depends on whether you are using the Gnome or KDE window manager - Konsole or Terminal
- Windows
  - you will have to install your own command line program. Windows comes with a command line, but it is non-standard and more difficult to use.
  - We will be using **GitBash** (alternative **Babun**)

BIT

Command Syntax

All commands have three parts: the utility, the flags, and the arguments. The utility always comes first. The other two parts have different rules, depending on which command you are using: you may not have to use any flags or arguments at all.

B I T

Command Syntax

Here is a sample command that you might type into a command line

```
ls -l ~/Desktop
```

BIT

Command Syntax

```
ls -l ~/Desktop
```

ls is a utility. Utilities are also sometimes known as commands all on their own, because they indicate the general idea of what you want.

Most of the time, you can simply run a utility all by itself, without any flags or arguments. Most commands only have one utility.

BIT

Command Syntax

```
ls -l ~/Desktop
```

-l is a flag that alters how the utility operates. The utility will usually work perfectly well with the defaults, but sometimes, you want to modify how it works slightly.

Flags always start with either one or two dashes (-), and they usually come between the utility and the arguments.

BIT

Command Syntax

```
ls -l ~/Desktop
```

~/Desktop is an argument to the utility. Arguments are used when the utility needs to know exactly what you want for a certain action.

The number of arguments used generally depends on the utility: some don't need any arguments, some require exactly one argument, some require lots of arguments, and some are flexible in the number of arguments they can take.

BIT

```
ls -l ~/Desktop
```

This command uses the **ls** utility, which is used to list the contents of directories. We use the **-l** flag to indicate to the utility that we want more information than it usually provides, and so it should show us the directory contents in a long format (-l is short for "long").

Last, the utility wants to know, "But which directory should I list the contents of?" Using the argument, we reply, "Show me the contents of my Desktop."

BIT

Executing Command

In all cases, to submit a command to the computer, press enter.

BIT

Basic Utilities

Here is a list of basic utilities that you will use on a regular basis.

Anything in capital letters that starts with a dollar sign, like `$THIS`, is an argument to the utility.

You should replace `$THIS` with the actual argument you want to give the computer.

BIT

Basic Utilities

ls	list all files and directories
ls -ah or ll	list long files and directories
mkdir	make a directory
touch	create a file
cd $directory	change to named directory
cd	change to home-directory
cd ~	change to home-directory
cd ..	change to parent directory

B I T

Basic Utilities

By typing `ls` into the command line and pressing enter the computer will reply with a list of names.

These names are the names of files and folders in the directory you are currently in.

Whenever you open up a new command line, you start in your home directory, which is the directory that generally contains all of your files.

BIT

Basic Utilities

`cp $FILE1 $LOCATION`	**c**o**p**y file1 to other location
`mv $FILE1 $LOCATION`	**m**o**v**e or rename file1 to file2
`rm $FILE`	**rm**ove a file
`rmdir $DIR`	**rm**ove a **d**irectory
`cat $FILE`	display a file
`pwd`	**p**rint **w**orking **d**irectory.
`whoami`	list users currently logged in

B I T

Neat Tricks

Computer programmers are lazy. Because they are lazy, they invented some techniques to do more with less work.

Here are some of those techniques

B I T

Neat Tricks

Whenever you need to type out a location in an argument (for example, in the `cd` command), you don't have to type out the whole thing: the first few letters will do.

Once you've typed three or four letters, press the **tab key**, and the command line will fill in the rest for you!

For example, if you are in your home directory, and you type cd Desk and then press the tab key, the command line will automatically complete the command to read cd Desktop! You can also use this if you find yourself mistyping folder names: tab autocompletion will always fill it in correctly.

BIT

Shortcuts

The command line has a few shortcuts built in.

For example, to see your previously typed command, just press the up button. You can do this to submit the same command multiple times, or to edit a command that you didn't type in quite right.

Another shortcut: you can use `~` (tilde) to refer to your home directory: `cd ~` will take you back home.

BIT

Practical