Getting Started

By: Kendra Wannamaker

Overview

- Command Line Navigation (Basic Bash)
- Development Environments
- Group Organization

Basic Bash - Moving Around

- Where am I?
  - print working directory
  - >pwd
- What is here?
  - list files
  - >ls
- I want to move…
  - change directory
  - >cd <fileName>
  - Home: >cd ~
  - Back: >cd ../

Basic Bash - File Manipulation

- Create a File
  - ><textEditor> <newFileName>.<fileExtension>
  - >gedit hello.txt
- Move a file
  - >mv <filePath>/<file> <newFilePath>/<MaybeNewFileName>
  - >mv hello.txt ../myFile/hi.txt
- Delete a file
  - >rm <fileName>
  - >rm hello.txt
- Copy a file
  - >cp <fileName> <fileName>
  - >cp hello.txt ../myFile/hi.txt

Basic Bash - My Favourites

- Make a directory
  - >mkdir <folderName>
  - >mkdir myFolder
- Delete a directory
  - >rmdir <folderName>
  - >rm <folderName>
    - Recursive -r
    - Force -f
- How do I…
  - >man
  - ><command> --help
- Where is…
  - >grep -<flags> “<stringYouAreLookingFor>” <file>
  - >gep -rin “Hello” *

Dev Environments - Basics

- Examples: Gedit, Notepad++, Notepad, Wordpress
- Pros
  - Simple
  - Syntax Highlighting
  - Light Weight
- Cons
  - Simple

Dev Environments - Advanced Text Editors

- Examples: Vim, Emacs
- Pros
  - Very Fast
  - Very Customizable
  - Cool
- Cons
  - Difficult to learn all the key short cuts
  - Probably require googling to learn how to do anything
  - Very Customizable

(vim adventure game)

Dev Environments - IDEs

- Examples: Eclipse, Intellij, Visual Studios
- Pros
  - Debugging
  - Open Declaration/Call Hierarchy
  - Lots of extra tools
- Cons
  - Very heavy
  - Complicated

DownloadING

When installing new software on your computer:

- Know if your computer is 32-bit or 64-bit
- Double check you install the right version of software, does it need to work with other software?
- Look are all the checkboxes, Are you setting Bing to be your default search engine?
- If you run into trouble just google it
  - Try to be as specific as possible, if you get an error message/code search that.

Groups - Forming a Group

- Make sure you schedules a line
  - at least 3 hours a week
- Match with people that value the class equally
  - Is your groupmate not willing to finish the project this weekend?
  - Is your groupmate asking you to give up a weekend?

Groups - Agreements

- Agree on what happens when someone drops the ball
  - Coffee/Donuts?
  - What happens if they do it again
- Agree on communication methods
  - Facebook? Email? Slack?
  - How are you sharing code? (Git Workshop Plug)
- Agree on syntax
  - camelCasing? Underline_naming_convention?
  - Where do brackets go?
  - Spaces or tabs?

Groups

To Be a Good Group Mate:

- Comment your code (If you write code others can't understand, you will end up writing more code)
- Follow through on commitments
- If you are stuck ask for help

To Be a Good Group:

- Meet up right when the assignment is given.
- How long will this take? (Leave time for things to go wrong)
- Divide the project into subtasks and allow group members to work at it when they have time

QED