EE 551
Programming in Python
Fall 2019

Lecture 2

09/05/2019

Assistant Prof. Sergul Aydore

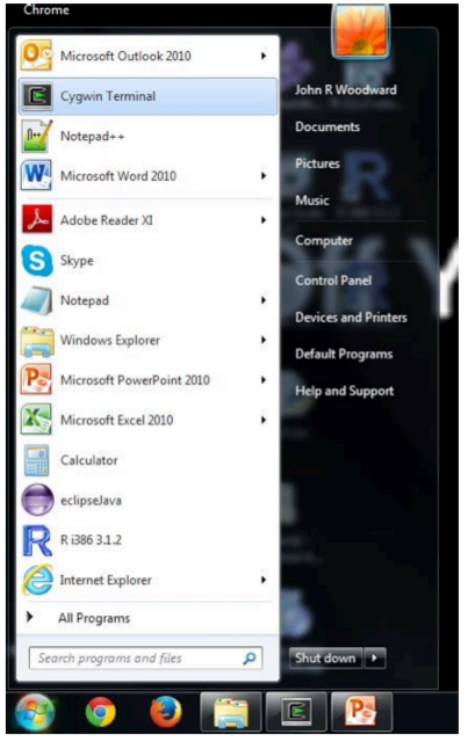
Department of Electrical and Computer Engineering

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

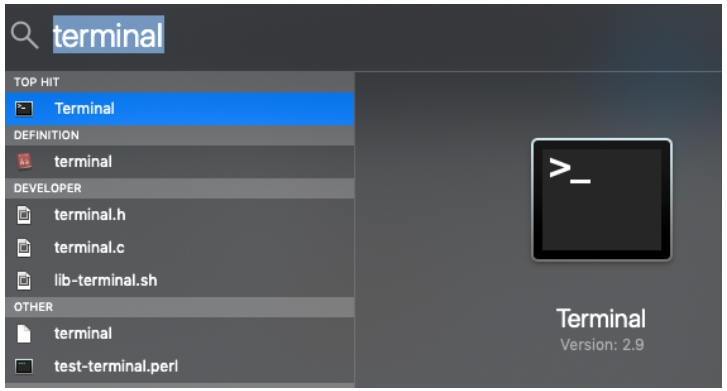
Start Your Command Line Prompt

- Windows Users: Open your Cygwin Console by clicking

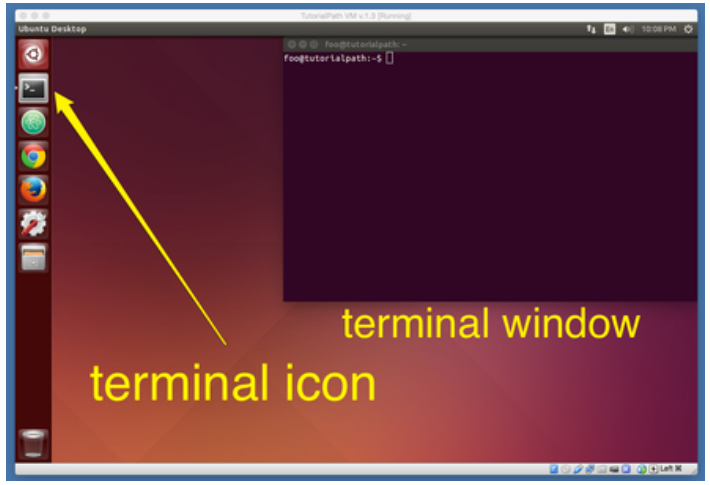
`Start > All Programs > Cygwin > Cygwin Bash Shell`



- Mac Users: Type `terminal` in Your spotlight search bar



- Linux Users: Press `CTRL+ALT+T`



- You now should see a window that is designed to “prompt” you to action

Directory Operations

- It would be nice to see the current directory. Type `pwd` and hit `enter`
- So, what is in? Type `ls` to list the files
- Change the directory: `cd ~/Desktop`
- Check if it was really changed: `pwd`
- Create a directory: `mkdir practice`
- In your newly created directory, create a directory called `test`
- Now, clean our mess by deleting the practice directory:
  - `cd ..`
  - `rm –r practice`
- Need help? Type `man <command>`, for example, `man rm`

Creating, Manipulating and Listing Files

- Create a file: `touch my_file`
- Edit the file: `vim my_file`
  - Hit `i` to start editing
  - Hit `ESC` to quit editing mode
  - Press `SHIFT+:+w` to save
- Edit more on command line: `echo "here we add more stuff" >> my_file`
- Dump the contents of the file to the screen `cat my_file`
- Create a hidden file `echo "*.txt" > .gitignore`
- Can you view this with `ls` command?

Renaming, Copying and Deleting

- Create a file called test: `echo "this goes to the test file" > test`
- Oh, you want to change the file name to text.txt: `mv test test.txt`
- You want to make a copy of test.txt and name it test2.txt: `cp test.txt test2.txt`
- Remove all files that ends with .txt: `rm *.txt`

You want to learn more?

- Check out this: https://www.learnenough.com/command-line-tutorial/basics

Git and Github Setup

- You need a GitHub Account, make sure you have one!
- Check if you have git installed: `git –version`
- Install if you don’t have it yet:
  - Windows: http://git-scm.com/download/win
  - MacOS: http://git-scm.com/download/mac
  - Linux: `sudo apt install git-all`
- Set your username and email address
  - `git config --global user.name "John Doe”`
  - `git config --global user.email johndoe@example.com`

Create a local directory

- Open up a terminal and move where you want to create your project, for example: `cd ~/Desktop`
- Create a new repository (repo for short): `mkdir myproject`
- Move into your project: `cd myproject`
- To initialize a git repository, type: `git init`
- Add a new file to the repo: `touch README.md`
- Type `git status` on command line

An interlude: The staging environment, the commit, and you

- A **commit** is a record of what files you have changed since the last time you made a commit.
- Commits make up the essence of your project and allow you to go back to the state of a project at any point.
- So, how do you tell git which files to put into a commit? This is where the **staging environment** come in.
- To add a file to a commit, you first need to add it to the staging environment. To do this, you can use the **git add <filename>** command

Add a file to staging Environment, Commit and Create a branch

- Add a file to the staging environment: `git add README.md`
- Run `git status` to see that git added the file to staging environment
- Create your first commit: `git commit -m "First commit"`
- Create a new branch to experiment with your project without making changes in the main project: `git checkout -b my_dev`
- Confirm that you are on the new branch: `git branch`
- If not, you can change your branch to my_dev: `git checkout my_dev`
- Modify the `README.md` file

Create a new repository on GitHub

- To create a new repo on GitHub, log in and go to the GitHub home page. You should see a green '+ New repository' button
- After clicking the button, GitHub will ask you to name your repo and provide a brief description
- GitHub will ask if you want to create a new repo from scratch or if you want to add a repo you have created locally. In this case, since we've already created a new repo locally, we want to push that onto GitHub so follow the **'....or push an existing repository from the command line'** section:
  - `git remote add origin https://github.com/sergulaydore/mynewrepository.git`
  - YOUR LINK WILL BE DIFFERENT!
  - `git push origin master`

Push a branch to github

- To push changes onto a new branch on GitHub, run `git push origin yourbranchname`
- Refresh your github page to see the new branch
- Create a pull request (PR) to merge your changes to the master branch
- If you are the sole owner, you can accept the PR by clicking green merge pull request
- Get changes to your local: `git pull origin master`
- You can view all the history by typing `git log`

That’s all for now about git and github.

- For more:
  - https://blog.udacity.com/2015/06/a-beginners-git-github-tutorial.html
  - https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

Let's dive into Python now!

- Clone the repository for the course if you have not done yet: `git clone git@github.com:sergulaydore/EE-551-Fall-2019.git`
- If you cloned it already, go into the folder and type `git pull`
- Create a virtual environment for the course if you have not done yet `mkvirtualenv EE551Fall2019`
- Activate the virtual environment: `workon EE551Fall2019`
- To install jupyter notebook, type
  - `pip3 install ipython`
  - `pip3 install jupyter`
- Activate jupyter by typing: `jupyter notebook`
- Start running Lecture2.ipynb by clicking