Frequently Asked Questions for Students

Hopefully these will alleviate some of the issues when starting assignments and setting up your development environment with Ed. I will preface this FAQ by saying most of these *can* be solved through a few online searches, please don't hesitate to search up your question, because there are some great online resources to help solve these issues.

Assignment Setup

Pre-required software to install on my machine?

Depending on your operating system, please see below:

Git: To take care of our code, we're going to be using `git`. This is a "version control" which pushes updates of your code to a remote server. The cool thing about this is that you can go backwards in versions if you ever pushed up something bad. It's definitely useful to learn, so I would recommend taking some time to get familiar with it, because it's useful for daily programming things, but very much helpful when you're working with more than one person.

Please note, you can install software to do this, but there are known problems on some of the programs so they don't work as well with Ed. I suggest getting familiar with the terminal since it will be useful later in life as well.

- MacOS
  - A terminal emulator program (terminal already installed)
  - Make sure in your terminal you can type the command `git`.
    - If you need to install git, a good tutorial is: https://www.atlassian.com/git/tutorials/install-git
- Linux
  - Any terminal emulator that has `git` installed.
    - To install git:
      - Ubuntu: `sudo apt install git`
      - Arch: `sudo pacman -Sy git`
      - Debian: `sudo apt install git`
      - Centos: `yum install git`
- Windows
  - https://gitforwindows.org/
  - Install the above, and use the program **"Git Bash"** (I recommend git bash, because it works well with the ssh keys and miscellaneous things)
    - Side note: if you use "Git CMD" it won't work with some of the commands below, "Git Bash" allows you to run commands similar to what mac/linux uses.
  - Here's a few videos:
    - https://www.youtube.com/watch?v=rWboGsc6CqI

    * https://www.youtube.com/watch?v=nbFwejIsHlY
    * https://www.youtube.com/watch?v=xueHs6fycTk

If you're looking for some nice tutorials on GIT, I would highly recommend:

- https://www.atlassian.com/git/tutorials
- https://learngitbranching.js.org/
- https://git-scm.com/docs/gittutorial
- https://try.github.io/

Editors: Since we're not using Ed workspaces, you're going to need to configure an editor on your own machine. This will let you write your programs on your computer.

There are many other editors available, here are a few that we can recommend:

Editors/IDEs come in various flavours of shapes and sizes, so there's definitely a lot of room for preference to play a part. Here's a list of them, take a look:

- PyCharm : https://www.jetbrains.com/pycharm/
  - Pretty cool python environment developed by JetBrains (They also do one for Go, Java, C, Web, etc.) [And you can sneakily ask for an education license for free by using your uni email!]
  - Available on most OS.
  - Pretty heavy on memory/gui things, but quite nice.
- VSCode : https://code.visualstudio.com
  - Nice editor developed with Microsoft's Visual Studio things, super heavy on the memory and processing, but it has a pretty nice tab completion. Supports a number of languages, but can be quite clunky.
  - Available on most OS.
- Atom : https://atom.io/
  - Great text editor that has plugins and can support a number of things. Really customisable to make it feel like what you want.
  - Quite heavy on the ram if you start ramping up on plugins so beware!
  - Available on most OS.
- Sublime Text : https://www.sublimetext.com/
  - Same as above, but not developed by GitHub.
- (hardcore mode, i.e. my favs) vim / emacs
  - vim or emacs are terminal based editors. They run out of your terminal, so there's very minimal "graphical interface", but they can get quite powerful. They definitely have shown their potential throughout the years.

- Great to learn for when you're doing things on servers/etc, but honestly it's quite a nice rabbit hole to go down, so if you do this... see you in a few months?
- Looks kinda cool and makes you feel awesome editing things in your terminal ;)

How do I set up my Ed to work with Git?

You will need to set up a private/public ssh key so that you can submit it to the edstem system.

Go to Account > Settings > SSH Keys and follow the steps to add a new key.

(Click Learn More about SSH Keys if you're stuck)

Quick Overview:

You can also follow these:

- https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys--2
- https://www.freecodecamp.org/news/the-ultimate-guide-to-ssh-setting-up-ssh-keys/

Open your terminal, and in terminal follow:

- OSX/Mac: **iTerm2** or **terminal**.
- Windows: **Git BASH** (Not GIT CMD!!)
- Linux: Any terminal emulator you have installed. (urxvt, terminator, xterm, etc.)

(IF YOU ARE USING GIT BASH ON WINDOWS, PLEASE RUN: `mkdir -p ~/.ssh/`, you need to create the folder first)

```
ssh-keygen -t rsa -b 4096
```

Generate your ssh-key by typing: `ssh-keygen -t rsa -b 4096` This is saying to create an RSA key (RSA is an algorithm for private/public keys) with 4096b. Follow the prompts to create the key.

By default, the file will be saved in /home/{your_user_here}/.ssh/id_rsa. (Note "your_user_here" will be your username on your computer.)

NOTE: If you change the "Enter file in which to save the key", and you change the name from "id_rsa", it will save in the folder you are in, so be careful! If you want to change the name from "id_rsa" to "my_edkey" for example, copy the entire path in!

e.g. "/home/your_user_here/.ssh/my_edkey" (changing "your_user_here" with your username on your computer)

```
$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/{your_user_here}/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/{your_user_here}/.ssh/id_rsa
Your public key has been saved in /home/{your_user_here}/.ssh/id_rsa.pub
The key fingerprint is:
{SHA256:.....................................} {user}@{host}
The key's randomart image is:
+---[RSA 4096]----+
|.=O=o.           |
|+O+=o.           |
|ooO.+ .          |
| =.*oo +         |
|. ooo B S        |
|..   o X o       |
|.o   E=oo        |
|o   . +o .       |
|     . .o        |
+----[SHA256]-----+
```

You will notice two keys:

- `/home/{your_user_here}/.ssh/id_rsa` - your private key, keep that secret!
- `/home/{your_user_here}/.ssh/id_rsa.pub` - your public key, this you can share because they can't do much with that.

Now, click "New Key" on Ed, and paste the public key into the box. You want to paste in anything in that `.pub` file, so open it with notepad/textedit/any text editor to copy.

Or if you want to do it on terminal:

- Mac/Linux:
  - `cat ~/.ssh/id_rsa.pub`
- Windows:
  - `type ~/.ssh/id_rsa.pub`

(If you want to see more, click "Learn more about SSH Keys")

You should be good to go!

How do I work with Ed and my computer?

Open up your terminal:

1. Go to a folder that you want to put your assignment in, using the basic commands like:

- `ls` - list the folder you are in now

- `cd <foldername>` - change to a folder, for example: `cd Downloads` will change to the Downloads folder from my current folder.
  - note the folder should be accessible from the current folder
  - If you need some help: https://www.digitalocean.com/community/tutorials/how-to-use-cd-pwd-and-ls-to-explore-the-file-system-on-a-linux-server

2. Clone the git assignment
   - In your terminal type: `git clone <url_here>` - get the URL from your assignment, and replace `<url_here>` (remove the angle brackets `<>`) with your url for the assignment.
     - If you get issues, it may be you didn't set up your ssh keys properly, take a look at the FAQs below.
   - Move to that folder using `cd`
3. Do some work
   - Open up the folder in your editor, and start making changes to the files.
   - Write your code, test it, edit some files.
4. Add your changes to git
   - Move to the folder using `cd`.
   - Run: `git commit -a -m "I made some changes"` - this adds the message "I made some changes", try change it to something useful!
     - There are alternatives to this, please check: https://www.atlassian.com/git/tutorials/saving-changes
5. Submit it up to ed, typing: `git push -u origin master`
   - IF it's the first time you're doing this command, then `git push -u origin master`
   - If you did the above, then the next time you can just use `git push` - pushes it up to the Ed git server.
   - Note; you need to be in your assignment folder.

Questions:

Permission Denied (publickey)

If you're getting an error like:

```
___ permission denied (publickey)
fatal: Could not read from remote repository

Please make sure you have the correct access rights and the repository exists
```

Then it looks like it's not getting your key, so how do we fix this?

Run the command: (changing the `id_rsa` to whatever you named your key)

```
eval `ssh-agent`
ssh-add ~/.ssh/id_rsa
```

And then try again.

When using SSH keys: No such file or directory

Did you point it to the correct file? Does that file exist?

It should point to `~/.ssh/id_rsa` by default, but if you changed the names then the key could (A) be a different name in the same folder, or (B) be in a different location, such as the folder you were in when you ran that command.

If you can't find your key, you can just re-make a new one and add it to Ed. You won't lose anything from adding a new key.

`git` is not a recognised command

This is saying you don't have git installed. If you installed it and still cannot access it, then it looks like it isn't in your `PATH` variable.

- https://stackoverflow.com/questions/26620312/installing-git-in-path-with-github-client-for-windows

fatal: not a git repository (or any of the parent directories)

This means you're not in the correct folder.

Example, after I `git clone`, it will create a folder. I need to be in that folder.

```
git clone https://git.edstem.org:challenge/1337/myassignment
```

Then after doing work:

```
cd myassignment

git commit -a -m "Updated a function"

git push
```

Temporary failure in name resolution:

- Are you using workspaces/terminal on Ed?
  - Please don't ;) It doesn't allow external connections, and it's not how the assignments are set up.
  - You need to clone it on your actual computer.
- This error is from not being able to access the internet, do you have internet connection / using a proper terminal?

Do I need to create a new key if I already used Ed/Git in another subject (e.g. COMP2017)?

No, you can use the same Ed key. :)

I'm using sourcetree/Git GUI, and it's giving me errors

SourceTree or GitGUI unfortunately don't have good integration with public/private keys. They don't like reading secret files, so unfortunately you're going to need to use the terminal / Git Bash program to do this.

Windows, permission denied (publickey) even though I followed the steps

The "git bash" program provides a command-line shell that you can use to easily interact in a "bash" terminal. This provides nice commands that operate similar to what you would use on linux or osx. (I'm assuming you're talking about this: https://gitforwindows.org/)

For now, the "git bash" application that shows a command-line like terminal gives you the ability to type in the commands that we mentioned in the post above. It provides the functionality that will help you follow through the git tutorials. If you've created your key, then you just need to make sure it's in the correct place for the "git bash" application to use it.

1. Open Git Bash that you just installed (Start->All Programs->Git->Git Bash)
2. Make a directory `mkdir -p ~/.ssh`
3. Type in the following: `ssh-keygen -t rsa` (when prompted, enter password, key name can stay the same)
4. Open file `your_home_directory/.ssh/id_rsa.pub` with your favorite text editor (or atom) and put it into the SSH field for Ed, under your account.
   - Or run: `type ~/.ssh/id_rsa.pub`
5. Be sure that you don't copy any whitespace while copying public key's content (id_rsa.pub) Note, for windows: your_home_directory is either C:\Users\your_username (on Windows Vista / 7 / 8 / 10), or C:\Documents and Settings\your_username (on Windows XP)
6. try the `git clone` command again on the "git bash" shell application.

fatal not a git repository

This is saying you're not in the folder for the assignment. You need to be in the folder for the assignment to be able to use the git commands correctly.

Take a look at using the `ls` and `cd` commands to navigate into the folder for the assignment. Once you `cd` into the assignment folder, then you should be solving this problem.

Please tell me who you are

- This is just asking you to fill out your name and email, it's so you know who is doing work.
- It tells you which commands to run:

To change it just for this assignment:

- run: `git config user.name "Person"` (change person to your actual name)
- run: `git config user.email "abcd1234@uni.sydney.edu.au"` - change to your unikey email.

To change it for your entire computer:

- `git config --global user.name "Person"` (change person to your actual name, and notice the `--global`)
- `git config --global user.email "abcd1234@uni.sydney.edu.au"` - change to your unikey email.

The authenticity of host git.edstem.org (XXX.XXX.XXX.XXX) can't be established

This is just your computer saying "I have no idea who git.edstem.org is, do you trust them?". The answer in this case, is most likely yes :)

This happens when you're connecting to a new website your computer doesn't know.

Code

Test Files

We are using python's `unittest` library to check how your code is working, we will supply you with test cases so you can get an idea of how we are testing your code. HOWEVER, please note that the tests provided are only a baseline, they will not test ALL the functionality we are testing, and we won't include the exact tests in your local tests. Use these as an example to try and motivate you to create more tests.

No Module Named XXXX

When your python code complains about no module being found, there are a few possible causes:

init.py

In some cases, you might need an "init" file to be defined, this is just an empty file that sits in the test directory, but it allows the tests to be used by Python. Sometimes that will be required to run tests, but this should be provided. (If it isn't just create a blank file inside the `tests/` folder named `__init__.py` and try again).

Are you running the tests from the correct folder?

You should be running your tests from the folder you are in. For example, if we have a `tree` assignment with the following structure:

```
-node.py
-README.md
```

```
-tests
|   |_test_simple_functions.py
-tree.py
```

You need to run your tests from the folder that contains `node.py` and `tree.py`.

You run the tests like (replacing the name of the file with the test file you are given):

```
python -m unittest tests/test_simple_functions.py
```

(OR on mac, you might need to specifically use the `python3`):

```
python3 -m unittest tests/test_simple_functions.py
```

Failed assertion error, expected type

Most of the times, this means you failed the test case. If it's saying that it's expecting a certain type, then you need to be returning that type.

For example, if we expect an empty list `[]`, but you give a `None`, then it will fail because they are not the same.

Other Questions

Can we create new functions?

Yes, you're free to create new functions and call them in your code.

AS LONG AS YOU DO NOT MODIFY THE FUNCTION SIGNATURE OF THE GIVEN FUNCTIONS!

If we give you `def hello_world(a, b, c)`, you CANNOT change it to be `def hello_world(a, b, c, d)` - this will break the tests.

function does not exist? or NameError: "name 'function_name' is not defined"

Since we are using "classes" in the assignment, you need to refer to the class method, for example:

```
class ExampleClass:
    def __init__(self, value, connected_object):
        # Let's make a value (an int)
        # and a pointer to another object with
        # the same class - connected_object
        self.connected_object = connected_object
        self.value = value

    def class_function(self):
        return self.value
...
```

```
# Now let's print it, it should be 10 here
print(object_a.value)

# Now let's print the connected object.
# Which should be object_a's value, which is also 10
print(object_b.get_connected_value())
```

If you are in class A and your attribute has class A type, then you can treat it just like calling an object with that class. So for example, in the Node class we have self.parent, where that is a node. We can call self.parent.children to get the children of that Node's parent.

Make sure you have a **self** infront of the function name.