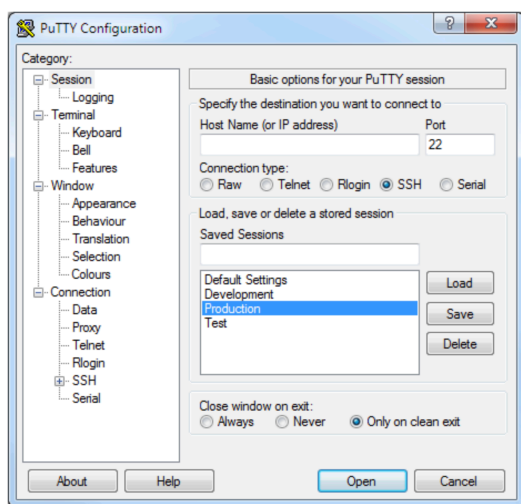


### Connecting to the remote machine via SSH

In this class, you will be working on a Linux machine that we've setup just for you. In computational biology, it is common to work on machines that you connect to remotely via secure shell (SSH). These machines generally have a lot of memory (RAM) and many cores—way more than your laptop. Genome assemblers generally require a lot of memory, and short read aligners run best with many cores.

Each team of two in this lab will be provided with a single account to share. Your GSI will provide you with the hostname, username, and password that you'll need to connect.

**Windows.** You will need to download PuTTY from the following URL, then run the executable to get started: <https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe>



(1) You should see a screen like this. Copy the hostname your GSI provided into the “Host Name” field. Make sure Port 22 is selected and the connection type is “SSH” then press “Open” to connect.

(2) You will be presented with a black terminal screen asking for your username and password. Use the information provided by the GSI. You should see a new prompt from the remote machine.

**Apple.** macOS is based on UNIX and comes with an SSH client. Press Command+Space to bring up Spotlight, type “Terminal” and press enter. From the command line prompt, type the following and press enter:

```
ssh your_username_here@your_hostname_here
```

Enter the password that you've been given, press enter, and you should see a new prompt from the remote machine.

**Once you're connected,** change your password by typing `passwd` at the command prompt. Make it something both you and your lab partner can remember!

## Connecting to iPython on the remote machine

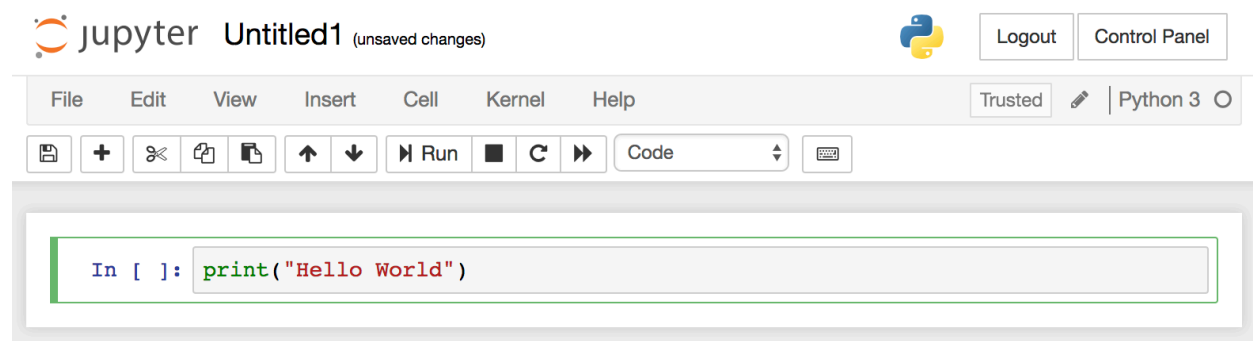
It is possible to run iPython on your laptop. You might also consider installing Jupyter, a Python development environment. Both are available here: <https://ipython.org/>

For this class, we have setup an instance of iPython on the remote machine for you to use. This allows you to keep your files in one place, access them remotely, and run your code using more memory and CPU cores. To get started, point your web browser to the iPython URL provided by your GSI. You should see a simple landing page prompting you for a username and password. Your username and password are the same that you used to connect to the machine via SSH.

Once connected, you'll see something like this, listing the contents of your home directory:



**Create a test notebook.** Click “New” and select “Python 3.” You’ll get a new notebook. Give it the old “Hello World” test and press “Run” to make sure everything’s working, then go to File→Close and Halt to end the session.



**Test the terminal function.** Rather than connecting to the remote machine via SSH using PuTTY or the macOS terminal, you can achieve the same effect using your web browser. Back on the main iPython page, choose “New” and select “Terminal.” You should see a black-and-white terminal akin to the one you got when you logged in via SSH. Isn’t that convenient?

### **Check out these Git and Markdown tutorials**

Markdown is a really handy way to format text documents like a README or your computational notebook. It's how GitHub likes you to store your documentation. It's really easy.

Spend a few minutes clicking through this tutorial: <https://www.markdowntutorial.com>

Git has become the standard version control system over the past decade. Taking the time to really learn it will be invaluable for you going forward, if you're thinking about a career that requires any kind of programming.

Here's a one-pager for creating new Git repositories and linking them to GitHub:  
[http://kbroman.org/github\\_tutorial/pages/init.html](http://kbroman.org/github_tutorial/pages/init.html)

Completing the first four lessons in the "Main" tab, and the first seven lessons in the "Remote" tab on this page will be /really/ helpful: <https://learngitbranching.js.org/> That said, they might make a little more sense once you've been using Git for a little while.

### Create a Git repo and commit a Markdown file

First, you and your lab partner should both create GitHub accounts if you don't already have them. It's a pretty simple sign-up process at <http://github.com>.

Once you've done that, one of you should create an empty GitHub repository.

Next, get to a terminal on the remote machine via the iPython web interface or SSH. Create a new directory and initialize it as a git repo:

```
mkdir hello_world
cd hello_world
git init
```

Now, use iPython's file editor to create a Markdown file in your new Git repository. From the main iPython web page, browse to your newly-created "hello\_world" folder. Then, click "New" and select "Text File." You should be presented with an empty text entry field. Select "Language" and choose "Markdown." Rename your file to "test.md" by clicking the filename at the top of the page and entering the new file name.

Enter some text that makes use of some of Markdown's formatting, and save by selecting File→Save. Here's an example:



Once your file is saved, tab back over to your web-based or SSH terminal connected to the remote machine. Now, add your file to the git repo and save the changes as a new commit:

```
git add test.md
git commit -a
```

You'll be asked to enter a commit message. These are generally very short descriptions of the changes that you've made. They should be concise, but verbose enough that you can tell pretty easily what the commit is about.

Finally, push your commits to GitHub with the following:

```
git remote add origin git@github.com:your_username_here/your_repo_name_here
git push -u origin master
```