setup

exercise 0 .

Before we begin, let's:

1. [Set up your environment](#_ow7anncwcgsr),
2. [Download a dummy exercise](#_roextmq4n35f), and
3. [Submit its solution](#_a3s7zbqgtqpa), just to make sure everything is working.
4. Finally, please make sure to read the [re-submission and extensions](#_et6r4zrp716e) guidelines.

##### An Apology

During this setup, you might have to run commands or copy-paste configurations that you don't fully understand.

As a student, I hated this — so I intended to start the course with a thorough explanation of these commands and configurations. However, this made the first lesson a bit too... dry, so I decided to postpone it until **Lesson 4: Workflow**.

Rest assured, you will understand everything we're doing here soon enough; until then, just follow the instructions (or, better yet — google for what you don't understand!).

# the environment

In this course, we'll use the **Linux** operating system — it's simple, beautiful, and powerful, and used in most production-grade servers.

Assuming you're not already using it, the first thing we're going to do it set it up. Luckily, it's free — but still requires some installation and configuration.

To start with, you have three options:

1. [**Use a Virtual Machine**](#_asaml7ne0dux)  
   Install a **virtual machine** and run the **course image**, so that you know for certain that everything is configured properly.  
   This is the **recommended option**; its only drawback is that virtual machines run slower than native operating systems, so if your computer is old, it can be... painful.
2. [**Use Linux**](#_g4avcyfk697a)  
   Install Linux as your **primary operating system**, and configure it to use Python, Git, etc. manually.  
   This option is only recommended if you're already familiar with Linux.
3. **Use Something Else**  
   Use Mac, Windows, or anything else throughout the course — most of the tools we use are cross-platform, so most of the exercises should be perfectly doable on the platform of your choice.  
   This option is not officially supported by the course staff, so if you run into any trouble with your environment, you'll have to figure it out on your own.

## use a virtual machine .

1. [Download VirtualBox](https://www.virtualbox.org/wiki/Downloads) and install it.
2. Download the [course image](https://storage.googleapis.com/advanced-system-design/advanced-system-design.ova).
3. In VirtualBox, go to **File** and select **Import Appliance...**.
4. Select the course image you've just downloaded (**advanced-system-design.ova**), click **Continue**, and then click **Import**.
5. After the course image is imported, select the **Advanced System Design** virtual machine and click **Start**.
6. The user is **user**, the password is **1234**.

## 

## use linux

1. Run the following command to install our basic dependencies:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **sudo** apt install build-essential curl git llvm libbz2-dev libffi-dev liblzma-dev libncurses5-dev libncursesw5-dev libreadline-dev libsqlite3-dev libssl-dev make python-openssl tk-dev wget xz-utils zlib1g-dev | |  |
|  |

1. Install **pyenv** to manage python versions:  
   1. Run the following command to download **pyenv**:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **git** clone https://github.com/pyenv/pyenv.git **$HOME**/.pyenv | |  |
|  |

Add the following lines to your profile (e.g. **.bashrc**):

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **export** **PYENV\_ROOT**="**$HOME**/.pyenv" **export** **PATH**="**$PYENV\_ROOT**/bin:**$PATH**" **if** command -v pyenv >/dev/null 2>&1 **then**  **eval** "$(pyenv init -)" **fi** | |  |
|  |

* 1. Restart your shell for the changes to take effect, and then run the following command to install Python 3.8 and make it the default:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **pyenv** install 3.8.0  $ **pyenv** global 3.8.0 | |  |
|  |

* 1. Test your setup:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **which** python $HOME/.pyenv/shims/python $ **python** --version 3.8.0 | |  |
|  |

1. Configure your python environment:
   1. Run the following command to upgrade **pip**:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **pip** install --upgrade pip | |  |
|  |

* 1. Run the following command to install **virtualenv**:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **pip** install virtualenv | |  |
|  |

* 1. Add the following lines to your profile (e.g. **.bashrc**):

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **export** **PYTHONDONTWRITEBYTECODE**=1  **export** **PYTHONIOENCODING**="utf8" | |  |
|  |

* 1. If you'd like to add history and autocompletion to your Python interpreter:  
     1. Write the following lines in **~/.pythonrc**:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **import** atexit  **import** pathlib  **import** readline  **import** rlcompleter  **def** **enable\_history**():  path = pathlib.Path.home() / '.pythonrc'  **if** path.exists():  readline.read\_history\_file(path)  atexit.register(readline.write\_history\_file, path)  **def** **enable\_completion**():  readline.parse\_and\_bind('tab: complete')  enable\_history()  enable\_completion() | |  |
|  |

* + 1. Add the following lines to your profile (e.g. **.bashrc**):

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **export** PYTHONSTARTUP="**$HOME**/.pythonrc" | |  |
| ç |

# the exercise .

1. Sign up to [GitHub](http://github.com/) (choose the Free Plan), or sign in if you already have an account.
2. Configure your workstation (virtual machine or native Linux) to work with GitHub seamlessly (otherwise, you'd have to provide your GitHub username and password every time you clone/push/pull a repository):  
   1. Click on your avatar at the top left corner, and select **settings**.
   2. Go to **SSH and GPG keys** and click **New SSH key**.
   3. On your workstation, run the following command to get your public key, and copy it.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **cat** ~/.ssh/id\_rsa.pub | |  |
| ç |

If this returns an error ("no such file or directory"), run the following command to generate your keys, and then try again.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **ssh-keygen** | |  |
| ç |

* 1. Paste the public key in the **Key** textbox (don't worry about the **Title**), and click **Add SSH key**.

1. Register to the [course's website](http://advanced-system-design.com/register/).
2. You should then get an email inviting you to the course's GitHub organisation.
3. Find the **exercise-0** repository, click on **Clone or download**, make sure the title is **Select with SSH** (otherwise, click **Use SSH**), copy the URL, and paste it into the following command:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **git** clone git@github.com:advanced-system-design/exercise-0.git | |  |
| ç |

1. This should create an **exercise-0/** directory on your workstation. Enter it, and run the following command to install the exercise's dependencies, virtual environment and requirements:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **cd** exercise-0/ $ ./scripts/install.sh | |  |
| ç |

1. Run the following commands to activate the virtual environment and run the tests:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **source** .env/bin/activate $ **pytest** tests/ | |  |
| ç |

**This should fail**.  
This is because the exercise is not yet complete; if you submit it like this, you won't get the optimal grade. In the next section, we'll complete the exercise and submit it.

# the submission

1. To complete the exercise, edit **hello.py** and implement the **hello()** function to print **Hello, world!**.
2. Run the following command to run the tests:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **pytest** tests/ | |  |
| ç |

If the tests pass, it's time to commit all your changes and upload your submission!

1. If this is the first time you're using git, run the following commands to configure your name and email address (be sure to change the values to yours):

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **git** config --global user.name "Dan Gittik"  $ **git** config --global user.email "dan.gittik@gmail.com" | |  |
| ç |

1. Run the following commands to commit all your changes:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **git** add .  $ **git** commit -m 'Submitting exercise 0.' | |  |
| ç |

1. Finally, to submit the exercise, you'd have to upload your changes to GitHub. However, we don't want everyone submitting their changes to the same place, so first you'd have to create your own repository, and upload your changes there.  
   1. Go to the [course's GitHub page](https://github.com/advanced-system-design), and click **Create a new repository**.
   2. Make sure the **Owner** is **advanced-system-design**, and call your repository **exercise-0-*<your identity number>***.
   3. Make sure your repository is private, and click **Create repository**.
   4. Go back to your project, and run the following commands to re-configure it, so that instead of pointing to the **exercise-0** repository, it points to yours:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **git** remote remove origin  $ **git** remote add origin \ git@github.com:advanced-system-design/exercise-0-123456789.git | |  |
| ç |

1. Finally, upload your changes:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **git** push origin master | |  |
| ç |

1. If at some point you realize you've made a mistake, and would like to re-submit the exercise, simply commit your changes and upload them again:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| $ **git** add .  $ **git** commit -m 'Submitting exercise 0 again.' $ **git** push origin master | |  |
| ç |

You can see your latest submission in your repository's page on GitHub; its URL should be something like <https://github.com/advanced-system-design/exercise-0-123456789>.

Approximately one week after the exercise is published, I'll run a script that will pull all the submissions, run a bunch of tests on them, and generate the grades, which you will be able to see on the [course's website](http://advanced-system-design.com/exercises/) (assuming you're logged in).

# re-submission and extensions

I don't really believe in grades, but seeing as this is a university, I have to grade you; so let's make it as painless as possible.

1. If you are unhappy with your grade, just contact the course staff and ask to re-submit your exercise.  
   You can re-submit any exercise at any time, even if you decide to change something in the first exercise on the last day of the course.
2. If you are unable to submit an exercise on time, just contact the course staff and ask for an extension.  
   Please note that some exercises are easier, and some are harder, but in general, they're spaced out and organized so that you have plenty of time to submit them all. If you constantly ask for extensions, we'll grant them — but you might end up with an unreasonable amount of work by the end of the semester.

The only thing I ask of you is that you don't cheat. I'm not very interested in policing, but plagiarizing other students' work is not cool. While it's OK to discuss the exercises, and even the solutions, with your fellow students — I'd like everyone to submit their own code.