**Using Python to Interact with the Operating System**

## **Welcome to the Course**

### Welcome to the course!

In this course, you’ll learn how to use Python to perform system administration tasks and interact with a computer’s operating system. You’re about to embark on an exciting journey to learn one of the most-in-demand job skills in IT today!

### Course prerequisites

This course builds upon the contents taught in the introduction to Python course:

* The basic Python syntax (if, for, while, defining functions, classes, and methods)
* How to use the most common data structures (strings, lists, tuples, and dictionaries)
* How to import and use additional Python modules

This course also requires some familiarity with some basic operating system concepts:

* Files, directories, and file systems
* Processes
* Log files

To get the most out of this course, we strongly recommend that you **install software in your machine**, or have your system administrator install it.

### How to pass the class

The course certificate gives you a way to prove your new programming skills to employers. To qualify for the certificate, you have to enroll in the program, pay the fee, and pass the graded assessments. If you don’t want to pay, you can still audit the course for free. This lets you view all videos and submit practice quizzes as you learn. One thing to remember though, this option doesn’t let you submit assessments, earn a grade, or receive the certificate.

### How deadlines work

When you enroll in the course, the system automatically sets a deadline for when you need to complete each section. Heads up: These deadlines are there to help you organize your time, but you can take the course at your own pace. If you "miss" a deadline, you can just reset it to a new date. There’s no time limit in which you have to finish the course, and you can earn the certificate whenever you finish.

### Qwiklabs

For some of our exercises, you'll be using an application called Qwiklabs. Qwiklabs lets you interact with a computer running an operating system that might not be the one running on your machine. The Qwiklabs scenarios will allow you to solve some real-world problems, putting your knowledge to work through active learning exercises.

### Getting and giving help

Here are a few ways you can give and get help:

1. **Discussion forums:** You can share information and ideas with your fellow learners in the discussion forums. These are also great places to find answers to questions you may have. If you're stuck on a concept, are struggling to solve a practice exercise, or you just want more information on a subject, the discussion forums are there to help you move forward.
2. **Coursera learner support:** Use the [Learner Help Center](https://learner.coursera.help/hc/en-us) to find information on specific technical issues. These include error messages, difficulty submitting assignments, or problems with video playback. If you can’t find an answer in the documentation, you can also report your problem to the Coursera support team by clicking on the Contact Us! link available at the bottom of help center articles.
3. **Qwiklabs support:** Please use the [Qwiklabs support request](https://qwiklab.zendesk.com/hc/en-us/requests/new) form to report any issues with accessing or using Qwiklabs. A member of the Qwiklabs team will work with you to help resolve the problem.
4. **Course content issues:** You can also flag problems in course materials by rating them. When you rate course materials, the instructor will see your ratings and feedback; other learners won’t. To rate course materials:

* Open the course material you want to rate. You can only rate videos, readings, and quizzes.
* If the content was interesting or helped you learn, click the thumbs-up icon.
* If the content was unhelpful or confusing, click the thumbs-down icon.

### Finding out more information

Throughout this course, we teach you how to do a range of things with Python, Bash, and other tools. While we’ll provide a lot of information through videos and supplemental readings, sometimes, you may need to look things up on your own, now and throughout your career. Things change fast in IT, so it’s critical to do your own research to stay up-to-date on what’s new. We recommend you use your favorite search engine to find more information about concepts we cover in this course — it’s great practice for the real world!

On top of search results, here are some good programming resources available online:

* [Automate the Boring Stuff with Python](https://automatetheboringstuff.com/): This book (available online and in print) includes a lot of practical programming exercises for beginners. You can refer to this content to read more about some of the things that we'll be discussing, and get inspired with more ideas of things that can be automated.
* [Hitchhiker’s Guide to Python](https://docs.python-guide.org/): This site (available online and in print) also covers a lot of what we can do with Python. Again, you can use this resource to learn more about the subjects we cover (and the ones we had to omit for time constraints).
* The [official language reference](https://docs.python.org/3/reference/index.html): Once you know what Python tool you'll be using to do a certain task, this technical reference of all Python language components can be a great

## **Pointers for Getting Your Environment Setup**

### Learning more about operating systems

We’ve talked briefly about what an operating system is and what we'll need to know about operating systems for this course. If you want to learn some additional operating system concepts, check out the videos on this subject in the [Technical Support Fundamentals course](https://www.coursera.org/lecture/technical-support-fundamentals/module-introduction-I3n9l). If you want to dive deeper onto how to manage Windows and Linux, check out the [Operating Systems and You: Becoming a Power User](https://www.coursera.org/learn/os-power-user) course.

If you want to discover more about the history of Unix, you can read all the details on the Unix [Wikipedia page](https://en.wikipedia.org/wiki/History_of_Unix).

### Installing Python and additional modules

If you don't have Python installed yet, we recommend that you visit the [official Python website](http://www.python.org/) and download the installer that corresponds to your operating system.

There’s a bunch of guides out there for installing Python and they all follow a similar process to the one we described in the videos. This [guide from Real Python](https://realpython.com/installing-python/) includes instructions on how to install python on a range of different operating systems and distributions.

Once you have Python installed on your operating system, it's a good idea to familiarize yourself with pip and the associated tools. You can find more info about these [here](https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/).

### Using package management systems

Package management systems help you better manage the software installed on your machine. These management systems vary a lot from operating system to operating system. So, you need to pick the one that works for the OS you’re using. Check out these guides for help with this:

* [Installing Python 3 on Windows 10 with Chocolatey](https://www.digitalocean.com/community/tutorials/how-to-install-python-3-and-set-up-a-local-programming-environment-on-windows-10)
* [Installing Python 3 on MacOS with Homebrew](http://www.pyladies.com/blog/Get-Your-Mac-Ready-for-Python-Programming/)
* [Package management basics on Linux](https://www.digitalocean.com/community/tutorials/package-management-basics-apt-yum-dnf-pkg)

### Other information

* [Python in the Microsoft Store for Windows 10](https://devblogs.microsoft.com/python/python-in-the-windows-10-may-2019-update/)

## **Setting up Your Environment**

After you’ve installed Python and checked that it works, the next step to set up your developer environment is to choose your main code editor.

These are some of the common editors for Python, available for all platforms:

* [Atom](https://atom.io/)
* [Eclipse](http://www.eclipse.org/)
* [PyCharm](https://www.jetbrains.com/pycharm/)
* [Sublime Text](http://www.sublimetext.com/)
* [Visual Studio Code](https://code.visualstudio.com/)

You can read more about these editors, and others, in these overview comparatives:

* [Python IDEs and Code Editors (Guide)](https://realpython.com/python-ides-code-editors-guide/#pycharm)
* [Best Python IDEs and Code Editors](https://www.softwaretestinghelp.com/python-ide-code-editors/)
* [Top 5 Python IDEs for Data Science](https://www.datacamp.com/community/tutorials/data-science-python-ide)

We encourage you to try out these editors and pick your favorite. Then, install it on your computer and experiment with writing and executing Python scripts locally.

## **How to Log in to Qwiklabs**

In the following assessments, you’ll be using Qwiklabs for hands-on learning. Qwiklabs provisions resources backed by Google Cloud that will be used to perform the tasks in the assessments. By using Qwiklabs, you won't have to purchase or install software yourself, and you can use the Linux operating system as if it was installed on your local machine.

**Important details:**

* You will have 90 minutes to complete each lab.
* You'll experience a delay as the labs load, as well as for the working instances of Linux VMs. So, please wait a couple of minutes.
* Make sure to access labs directly through Coursera and not in the Qwiklabs catalog. If you access labs through the Qwiklabs catalog, you will not receive a grade. (As you know, a passing grade is required to complete the course.)
* You'll connect to a new VM for each lab with temporary credentials created for you; these will last only for the duration of the lab.
* The grade is calculated when the lab is complete, so be sure to hit "End Lab" when you're done. Note: after you end the lab, you won't be able to access your previous work.
* To get familiar with entering labs, find the links below for the **operating system of the machine you are currently using** for a visualization of the key steps. Note that while video resources linked below do not have a voiceover or any audio, all important details will still be housed in each lab’s set of instructions on the Qwiklabs platform.
* If you receive the error "**Sorry, your quota has been exceeded for the lab**", please [submit a request](https://qwiklab.zendesk.com/hc/en-us/requests/new) or reach out to the Qwiklabs support team directly via chat support on qwiklabs.com.

**Demo videos for accessing labs:**

* [For Windows users](https://www.youtube.com/watch?v=Al1opDxb3ok)
* [For Mac users](https://www.youtube.com/watch?v=76VlwjMYIxg)
* [For Linux users](https://www.youtube.com/watch?v=YtrO8nW0ugM)
* [For Chrome OS users](https://youtu.be/HklttPmGGKc)