

Google IT Automation with Python Certificate

The Google IT Automation with Python Certificate is a hands-on, online program designed to help individuals with a background in IT Support advance to technical roles, like Systems Administrators. Developed by Google, the program helps IT professionals build in-demand skills, including Python, Git, and IT automation. Learners who complete the IT Automation with Python Certificate will:

- Learn how to program with Python and use Python to automate common system administration tasks
- Learn to use Git and GitHub to troubleshoot and debug complex problems
- Apply automation at scale by using configuration management and the Cloud
- Practice their technical skills with hands-on projects, including a capstone project where they will use their new knowledge to solve a real-world IT problem
- All learners must have a computer with reliable internet to access course content. Reliable internet access is necessary to complete hands-on course activities. Headphones are also a good option for a noisy environment.

SEMESTER PLAN:

Below includes a detailed guide for you to follow as you complete the course work. The goal is to keep you on track to finish your certificate within the allotted timeframe.

Expectations and Best Practices

Online learning allows for the flexibility of working at your own pace to meet a deadline, but it's helpful to establish a routine to stay on target and remember when work is due! Here's one that has worked well for students previously:

- **Monday/Tuesday:** Review all new material for the week and watch all assigned videos by end of day on Tuesday.
- **Rest of the week:** Complete work at your own pace. Aim to complete at least one task each day, no matter how large or small. With work, you may complete core coursework on Saturdays and Sundays. Reminder: To ensure you are actively participating, discussion posts may require you to post multiple times throughout the week.
- **Sunday:** Typically, weekly assignments are due at the end of the day on Sunday. Review that you have completed each assignment for that week.

Now that you have reviewed the coursework guidelines, please take a moment to review a general breakdown of each course, including estimates of how much time each may take based on the quantity and difficulty of the content. As you work through this plan, reach out to your professor with questions, concerns, and if you need clarifications.

COURSE 1: Crash Course on Python

This course is designed to teach the foundations needed to write simple programs in Python using the most common structures. No previous exposure to programming is needed. By the end of this course, learners will understand the benefits of programming in IT roles, be able to write simple programs using Python, learn how the building blocks of programming fit together, and combine all of this knowledge to solve a complex programming problem.

WEEK 1

MODULE 1: Hello Python (2 hours, 53 minutes total module time)

In this module, learners will study basics of programming languages and syntax, as well as automation using scripting. They will also be introduced to the Python programming language and some of the benefits it offers. Finally, they will discover some basic functions and keywords of the language, along with some arithmetic operations.

Module items:

- 3 discussion prompts
- 3 practice quizzes
- 1 graded quiz
- 5 readings
- 14 videos

MODULE 2: Basic Python Syntax (2 hours 59 minutes total module time)

Learners will explore different data types in Python, how to identify them, and how to convert between them. They will also learn how to use variables to assign data and how to reference variables. They will explore functions: how to define them, pass the parameters, and have them return information. They will explore the concepts of code reuse, code style, and refactoring complex code and learn to effectively use code comments. Finally, they'll learn to compare data using equality and logical operators and leverage these to build complex branching scripts using if statements.

Module items:

- 1 discussion prompt
- 3 practice quizzes
- 1 graded quiz
- 9 readings
- 14 videos

WEEK 2

MODULE 3: Loops (2 hours 40 minutes total module time)

Learners will explore the intricacies of loops in Python. They will learn how to use “while” loops to continuously execute code, as well as how to identify infinite loop errors and how to fix them. They will learn to use “for” loops to iterate over data and how to use the range() function with for loops. They will also explore common errors when using “for” loops and how to fix them.

Module items:

- 1 discussion prompt
- 3 practice quizzes
- 1 graded quiz
- 7 readings
- 13 videos

MODULE 4: Strings, Lists, and Dictionaries (4 hours 21 minutes total module time)

Learners will explore more advanced ways to manipulate strings using indexing, slicing, and advanced formatting. They will also explore the more advanced data types: lists, tuples, and dictionaries. They will learn to store, reference, and manipulate data in these structures, as well as combine them to store complex data structures.

Module items

- 1 discussion prompt
- 3 practice quizzes
- 1 graded quiz
- 15 readings
- 16 videos

WEEK 3

MODULE 5: Object Oriented Programming (6 hours 4 minutes total module time) *Optional*

Learners will be introduced to the concept of object-oriented programming. They will learn how to build their own classes with unique attributes and methods. They will get a chance to write documentation for their classes and methods using docstrings. They will learn about object instances and object inheritance, as well as how to import and use Python modules to make use of powerful classes and methods. Learners will also be introduced to Jupyter notebooks, which they will use to write and execute more complex code.

Module items

- 1 discussion prompt
- 3 hands-on labs
- 1 practice quiz
- 12 readings
- 13 videos

MODULE 6: Final Project (5 hours 56 minutes total module time)

In the final module, learners will put everything they've learned so far into action! They will apply a problem-solving framework to tackle a challenging final project: implementing a script that generates a "word cloud" from some text. They will formulate a problem statement to understand the challenge,

conduct some research to determine what options are available, and then begin planning how they intend to solve the problem. Lastly, they'll write the code to implement their solution!

Module items

- 1 discussion prompt
- 2 hands-on labs
- 1 graded quiz
- 1 reading
- 11 videos

[COURSE 2: Using Python to Interact with the Operating System](#)

By the end of this course, learners will be able to manipulate files and processes on their computer's operating system. They will also have learned about regular expressions—a very powerful tool for processing text files—and they'll get practice using the Linux command line on a virtual machine. Learners will also write a program that processes a number of errors in an actual log file and then generates a summary file. They will start by exploring how to execute Python locally and organize and use code across different Python files. They will then learn how to read and write different types of files and use subprocesses and input streams. They will also explore Bash scripting and regular expressions—both very powerful tools for anyone working with systems. Next, learners will touch on automatic testing, which allows IT Specialists to automate how to check if their code is correct. To finish, they will put all this together by using the tools that they've acquired to process data and generate automatic reports.

WEEK 4

MODULE 1: Getting Your Python On (4 hours 23 minutes total module time)

Learners will learn about the different types of operating systems and how to get their Python code ready to interact with the operating system. They will learn about getting their environment set up and installing additional Python modules that will help them along the way. They will review interpreted versus compiled language and how they differ from one another. Learners will also explore the benefits of automation and learn common pitfalls so they can avoid them. Finally, they will learn about Qwiklabs, which will be used for graded assessments.

Module items

- 2 discussion prompts
- 3 practice quizzes
- 1 hands-on lab
- 5 readings
- 16 videos

MODULE 2: Managing Files with Python (4 hours 20 minutes total module time)

Learners will learn about reading and writing to files and the commands that will enable them to do this. They will learn the importance of managing files and how they can navigate through different directories. Learners will discover how to work with files and that there is a layer of abstraction between Python and the operating system. Finally, they will learn about CSV files and how to best utilize them.

Module items

- 1 discussion prompt
- 1 hands-on lab
- 2 practice quizzes
- 1 hands-on lab
- 3 readings
- 12 videos

WEEK 5

MODULE 3: Regular Expressions (3 hours 35 minutes total module time)

Learners will learn about what a regular expression is and why they would use one. They will explore the basics of regular expressions and give examples of wildcards, repetition qualifiers, escape characters, and more. Next, they will explore advanced regular expressions and repetition qualifiers. They will also tackle new exercises, like capturing groups and extracting PIDs using regexes. Finally, learners will receive a helpful guide for regular expressions.

Module items

- 2 discussion prompts
- 3 practice quizzes
- 1 hands-on lab
- 2 readings
- 14 videos

MODULE 4: Managing Data and Processes (3 hours 23 minutes total module time)

Learners will learn about reading and writing to data files based on an interaction with the user. Along the way, they will explore standard streams, environment variables, and command line arguments. Next, they will study Python subprocesses, including system commands and how they can be used. Learners will review how to obtain output from a system command and explore subprocess management, including how to check exit values and manipulate the normal versus error exit values. Finally, learners will explore what a log file is, how to process log files, how to filter log files using regular expressions, and how to understand the output captured from log files.

Module items

- 2 discussion prompts
- 3 practice quizzes
- 1 graded quiz
- 2 readings

- 12 videos

WEEK 6

MODULE 5: Testing in Python (5 hours 23 minute total module time)

Learners will explore how to create tests in Python. They will learn what testing is all about and understand the differences between manual and automated testing. Next, they will explore what unit tests are intended to do and how to write them. Then, they will learn about other test concepts, like black box and white box tests and how test-driven development can frame how they design and write their code. Finally, they will learn about errors and exceptions and how to combat them.

Module items

- 1 discussion prompt
- 2 practice quizzes
- 3 hands-on labs
- 4 readings
- 14 videos

MODULE 6: Bash Scripting (3 hours 29 minutes total module time)

Learners will be exposed to what the Linux OS has to offer and learn about Bash scripting. They will discover basic Linux commands and explore the many processes Linux has to offer, including a key concept called redirection. They will then learn about creating Bash scripts using variables and globs. Finally, learners will explore advanced Bash concepts and develop an understanding of when to use Bash versus Python.

Module items

- 3 practice quizzes
- 1 graded quiz
- 3 readings
- 13 videos

MODULE 7: Final Project (1 hour 52 minute total module time)

Learners will put everything they've learned so far into action! They will apply their scripting knowledge to tackle a challenging final project: writing a script that scans for a specific error in the log files. They will create a problem statement to understand the challenge, conduct some research to see what options are available, and then begin planning how they intend to solve the problem. Lastly, learners will write the code to implement their solution.

Module items

- 1 discussion prompt
- 1 hands-on lab

- 6 videos

[COURSE 3: Introduction to Git and GitHub](#)

In this course, learners will learn how to keep track of the different versions of their code and configuration files using a popular version control system (VCS) called Git. They will also go through how to set up an account with a service called GitHub so that they can create their very own remote repositories to store their code and configuration. Throughout this course, they will learn about Git's core functionality so they can understand how and why it's used in organizations. They will explore both basic and more advanced features, like branches and merging. Learners will gain an understanding of why having a working knowledge of a VCS like Git can be a lifesaver in emergency situations or when debugging. Then they will explore how to use a VCS to work with others through remote repositories, like the ones provided by GitHub. By the end of this course, learners will be able to store their code's history in Git and collaborate with others in GitHub, where they will also start creating their own portfolio!

Week 7

MODULE 1: Introduction to Version Control (4 hours 25 minutes total module time)

In this module, learners will be introduced to the concept of version control, which will help them manage and roll back their code. They will learn how to differentiate between files and the tools at their disposal to make this happen. Next, they will be introduced to Git and how they can leverage that platform to improve their coding abilities. Once they've got a grasp on what Git is, learners will install it and start using it to create and clone code repositories. Finally, they will explore Git in order to get more familiar with the different tools and commands it has to offer.

Module items

- 2 discussion prompts
- 4 practice quizzes
- 1 hands-on lab
- 6 readings
- 16 videos

MODULE 2: Using Git Locally (3 hours 44 minutes total module time)

Learners will explore advanced Git interactions by skipping the staging area for small code changes and understand how Git uses the HEAD alias to represent checked-out snapshots. Next, they will explore how to move and remove files and get a guide to help them in their Git explorations! Then, they will learn how to undo changes before committing and how to amend commits once they're submitted. Finally, learners will be asked to identify errors in commits that were submitted a while back. In the final section of this module, they will explore the concept of branching and merging. They will learn what a branch is, how to create one, and how they work in harmony with their code. Once they're comfortable with branching, learners will explore merging, how it works with branched data, and how to deal with merge conflicts.

Module items

- 2 discussion prompts
- 4 practice quizzes
- 1 hands-on lab
- 3 readings
- 14 videos

WEEK 8

MODULE 3: Working with Remotes (3 hours 43 minutes total module time)

Learners will be introduced to GitHub and learn how it works with Git. They will create new repositories and clone those repositories onto their computer. Next, they will learn about remote repositories, how they can work with them, and how they can host them. They will get familiar with commands like “modify,” “stage,” and “commit,” which will be used for local changes, as well as the “fetch” command, which can pull any changes from remote repositories. Their final lesson will focus on learning about conflicts. This will allow learners to explore the concepts of pull-merge-push workflows, pushing remote branches, and rebasing their changes.

Module items

- 2 discussion prompts
- 4 practice quizzes
- 1 hands-on lab
- 3 readings
- 13 videos

MODULE 4: Collaboration (3 hours 36 minutes total module time)

In this module, learners will continue to explore the collaboration tools available in Git. They will learn about the tools that are available to help improve the quality of their code and to better track their code. This includes an overview of pull requests and how the typical workflow of a pull request looks like on GitHub. Next, they will explore how they can squash changes in their code. They will also receive a helpful guide on fork and pull requests. Then, learners will learn what code reviews are and what the code review workflow is like. They will learn about how to use code reviews on GitHub. The final lesson of this module will focus on managing projects. Learners will learn best practices for managing projects and how to manage collaboration within those projects. They will explore different ways of tracking issues and the concept of continuous integration with their projects.

Module items

- 1 discussion prompt
- 4 practice quizzes
- 1 hands-on lab
- 3 readings

COURSE 4: Troubleshooting and Debugging Techniques

In this course, learners will explore the tools to quickly identify and solve real-world problems that they might come across in their IT role. They will explore a bunch of different approaches for tackling the most common pitfalls of their code and IT infrastructure. They will also learn strategies for approaching almost any technical problem and then see how those apply to solving different real-world scenarios.

Week 9

MODULE 1: Troubleshooting Concepts (4 hours 4 minutes total module time)

Learners will be introduced to the fundamentals of troubleshooting and learn different strategies to tackle problems that they might encounter. They will learn about the concept of debugging and how debugging is one of the core principles of troubleshooting. Learners will be introduced to some tools that will help them in the debugging process, like tcpdump, ps, top, itrace, and more. Next, they will explore how to “understand the problem.” After that, they will explore the different approaches when troubleshooting reproducing errors versus intermittent errors. Finally, learners will learn about “binary searching a problem.” They will explore the different types of searches, including binary and linear searches. Then, learners will learn about the concept of bisecting and how it can be used in their troubleshooting approach, and finish up with finding invalid data in a CSV file.

Module items

- 3 discussion prompt
- 4 practice quizzes
- 1 hands-on lab
- 3 readings
- 14 videos

MODULE 2: Slowness (3 hours 58 minutes total module time)

In this module, learners discover what factors can cause a machine or program to act slowly. They will learn ways to address slowness by identifying the bottleneck that might be causing the slowness. They will also learn about tools used to identify which resources are being exhausted, including iotop, iftop, and activity monitor in MacOS. Next, learners will examine how computers use resources and learn the differences between CPU, RAM, and Cache in order to help them find the possible causes for slowness in their machines or scripts. After that, learners will learn how to write efficient code, then explore profilers to help them identify where their code is spending most of its time. Next, they will explore data structures and understand which ones are right for them to use. These data structures include lists, tuples, dictionaries, sets, and expensive loops. Then, they will learn about complex slowness problems and how utilizing concurrency and adding a caching service can improve the execution of their code. Finally, learners will discover how using threads can make the execution of their code much quicker.

Module items

- 2 discussion prompt
- 4 practice quiz

- 3 readings
- 1 hands-on lab
- 15 videos

Week 10

MODULE 3: Crashing Programs (3 hours 52 minutes total module time)

Learners will discover strategies to answer the age old question, “Why has my program crashed?” They will learn how to troubleshoot system crashes and application crashes, what tools can be used to help identify the cause of the crash, and what log files to look at in order to find what might have gone wrong. Next, they will investigate why code crashes and what they can do to prevent that from happening. Then, learners will explore what happens when an unhandled error occurs and throws an exception. They will learn about the printf debugging technique, which will help them identify these errors and exceptions. Finally, they will explore the concept of handling crashes and incidents at a much larger scale. Learners will delve into a scenario where a large e-commerce site will throw an error 20% of the time. Once that issue has been fixed, they will understand the importance of communication and documentation during these incidents and how writing a postmortem can prevent issues from happening again.

Module items

- 2 discussion prompts
- 4 practice quizzes
- 1 hands-on lab
- 2 readings
- 14 videos

MODULE 4: Managing Resources (3 hours 56 minutes total module time)

Learners will learn how to manage their applications. They will explore some common issues that may cause applications to crash. They will also learn what memory leaks are and how to troubleshoot and prevent them. Then, learners will find out how to manage disk space. They will explore some scenarios where this happens and learn how to identify what process or application is taking up all their disk space. Then, they will learn what network saturation is, how it can be caused, and some useful tools and techniques to solve a network saturation problem. After that, they will shift from managing applications to managing their time. They will get tips on how to prioritize tasks, estimate how long a particular task will take before finishing, and communicate expectations when dealing with important tasks. The final lesson will delve into how to deal with complex problems by breaking them down into small, digestible chunks while keeping a clear goal. They will learn that using proactive approaches, like continuous integration, can help them with future issues that might come up. They will also explore how to plan for future resource usage by making good use of monitoring.

Module items

- 1 discussion prompt
- 4 practice quizzes

- 3 readings
- 1 hands-on lab *16 videos

[COURSE 5: Configuration Management and the Cloud](#)

In this course, learners will discover how to apply automation to manage fleets of computers. They will understand how to automate the process for deploying new computers, keeping those machines updated, managing large-scale changes, and more. They will learn how to manage both physical machines running in offices and virtual machines running in the Cloud. Learners will start by studying an automation technique called configuration management, which lets IT Specialists manage the configuration of computers at scale. Specifically, they will learn how to use Puppet, the current industry standard for configuration management. They will explore some simple examples and then learn to apply the same concepts to more complex cases. Later on, learners will expand their automation skills by understanding how to use the cloud to help scale their infrastructure. They will discover some best practices for handling hundreds of virtual machines running in the cloud and troubleshooting them when things don't go according to plan.

Week 11

MODULE 1: Automating with Configuration Management 3 hours 39 minutes total module time
Learners will be introduced to the concept of automation at scale and how it can successfully be achieved. They will learn what it means to work at scale and how automation is needed to scale effectively. Next, they will be introduced to configuration management. They will learn the differences between unmanaged and managed configuration management. Then, learners will explore infrastructure as code and learn about the benefits it brings, like making a fleet of nodes more reliable and repeatable. Next, learners will be introduced to Puppet. They will learn how to apply basic configuration management and how Puppet agents and masters interact with each other. They will also learn about Puppet resources and classes. They will learn how resources are basic units for modeling configurations and how classes are a collection of resources used to achieve a single goal. The final lesson will introduce learners to the building blocks of domain-specific language. They will learn what Puppet facts are and how they can use a program called Facter to analyze, store, and gather this information. They will discover the driving principles of configuration management. They will also learn about declarative, procedural, and idempotent principals and how they differ from one another.

Module items

- 2 discussion prompts
- 4 practice quizzes
- 4 readings
- 1 hands-on lab
- 11 video

MODULE 2: Deploying Puppet 3 hours 19 minutes total module time
Learners will learn how to deploy Puppet on their local machine. Once they've completed that task, they will start creating and applying Puppet rules, managing resource relationships, and organizing their Puppet modules, which are a collection of manifests and associated data. Next, they will learn about Puppet nodes and node

definitions and how they're used to apply rules to their fleet. Then, they will learn about the Puppet certificate infrastructure, which explores the logic behind how the server can trust that a client is really who it claims to be. This topic will introduce the concepts of public key infrastructure and secure socket layer, which can ensure that clients can be trusted. Once they understand these concepts, they will get to see a Puppet deployment in action! The final lesson will center on updating, modifying, and testing manifests that they've deployed to their fleet. They will explore Puppet parser validate commands that will allow them to check the syntax to ensure its correct. Next, they will explore the difference between production and testing environments and how they can safely roll out changes to the testing environment to catch any errors. Learners will also learn about development environments and how they can siphon part of their fleet to an early-adopters or canary track to roll out changes, modification, or updates to that subset of machines.

Module items

- 1 discussion prompt
- 4 practice quizzes
- 3 readings
- 1 hands-on lab
- 10 videos

Week 12

MODULE 3: Automation in the Cloud (3 hours 31 minutes total module time)

Learners will learn about the different types of cloud services, like SaaS, PaaS, and IaaS. Next, they will learn about scaling in the cloud and how to modify the capacity of your service. They will explore horizontal and vertical scaling, which then ties into automatic versus manual scaling. Then, they will explore how much control they have over their system—depending on whether they choose SaaS, PaaS, or IaaS—and assess which one is right for their business. Learners will discover the options available to migrate their business to the cloud using strategies like lift and shift.

Then, learners will learn how to manage instances in the cloud by creating one. They will explore the different options available, like regions, machine types, and more. Then they will learn about customizing and templating virtual machines to enable them to deploy them at scale. The final lesson will explore the concept of automating cloud deployments. Learners will learn about load balancers, which ensure that each node receives a balanced number of requests, as well as autoscaling, which shuts nodes on and off as needed. Next, learners will learn about orchestration and how orchestration and automation differ. Finally, they will explore infrastructure as code (IaC), or machine readable configuration files that automate configuration management.

Module items:

- 1 discussion prompt
- 4 practice quizzes
- 1 hands-on lab
- 2 readings
- 13 videos

MODULE 4: Managing Cloud Instances at Scale (3 hours 51 minutes total module time)

Learners will learn all about storing data in the cloud. They will study the different types of storage available, like block storage and object storage, and how they differ. They will explore load balancing further and dive into some load balancing techniques, like round-robin DNS and sticky sessions. Next, they will dive into change management, including the different ways to test changes and how to push them. They will explore different testing methods, like unit tests and integration tests. Learners will also cover continuous integration, the use of continuous deployment, and how to apply A/B testing. Next, they will look at some errors they might encounter along the way, like quotas or limits, and how best to avoid or prepare for these. In the next lesson, learners will get an understanding of monitoring and altering and review some systems that offer it. They will then explore the concept of SLAs and how to set achievable ones. Next, they will look at basic monitoring in GCP and create altering policies, set up conditions, and choose aggregators to manage the data. The last lesson will explore troubleshooting and debugging our systems. Learners will learn how to troubleshoot and debug remotely, understand techniques for how to identify where the failure is coming from, and determine how to recover from a failure when it strikes.

Module items

- 1 discussion prompt
- 4 practice quizzes
- 1 hands-on labs
- 3 readings
- 14 videos

[COURSE 6: Automating Real-World Tasks with Python](#)

In the final course, learners will tie together the concepts that they've learned up until now. They will tackle real-world scenarios in Qwiklabs that will challenge them to use multiple skills at once. First, they will take a closer look at how to use external Python modules to extend their code's capabilities and spend some time learning how to use documentation to learn a new module. For example, they will use the Python Image Library (PIL) to create and modify images. They will view some simple examples of how to perform common tasks in the course material, but it will be up to them to explore the module documentation to figure out how to solve specific problems.

Next, learners will discover how to communicate with the world outside of your code! They will use data serialization to turn in-memory objects into messages that can be sent to other programs. Their program will send messages across the network to Application Programming Interfaces (APIs) offered by other programs. For those times when their code needs to talk to a person instead of a program, they will also learn to send email messages. At the end of this course, learners will be able to take a description of a problem and use their skills to create a solution—just like they would on the job. In the final capstone project, they will be given a description of what their customer needs, and it will be up to them to create a program to do it!

Week 13

MODULE 1: Manipulating Images (3 hours 20 minutes total module time)

In this module, learners will put to practice some of their newly acquired Python skills. They will make use of interesting Python libraries, understand how to read their APIs, and learn how to use the Python Imaging Library (PIL). By the end of this module, learners will build a program that automatically resizes and rotates a bunch of images and converts them from one image format to another.

Module items

- 8 readings
- 1 hands-on lab
- 3 discussion prompts

MODULE 2: Interacting with Web Services (3 hours 20 minutes total module time)

Learners will be introduced to several different tools that can be really useful in today's IT world. First, they will learn how they can use different text formats to store data in text files, retrieve it, and even transmit it over the internet. Later on, they will explore how they can get their code to interact with services running on different computers using a module called Python Requests.

Module items

- 10 readings
- 1 hands-on lab
- 1 discussion prompt

Week 14

MODULE 3: Automatic Output Generation (3 hours 10 minutes total module time)

Learners will explore a different aspect of automation: automating the generation of nicely-formatted output from scripts, like sending emails. By the end of this module, learners will be able to send an email message with an attachment from Python. They will even learn how to generate PDF files to attach to those emails.

Module items

- 8 readings
- 1 hands-on lab
- 2 discussion prompts

MODULE 4: Putting It All Together (2 hours 51 minutes total module time)

For the final project in this course, learners will use the techniques and concepts they've learned to build a solution to a complex IT task.

Module items

- 3 readings
- 1 hands-on lab

- 1 discussion prompt