Welcome to Introduction to Python.

Introduction to Python

What you will learn

At the core of the lesson

You will learn how to:

- Explain what Python is
- List reasons for using Python
- List examples of programs that can be used to write Python code
- Compare Python to shell scripting
- Recognize the purpose of AWS Lambda

aws re/start

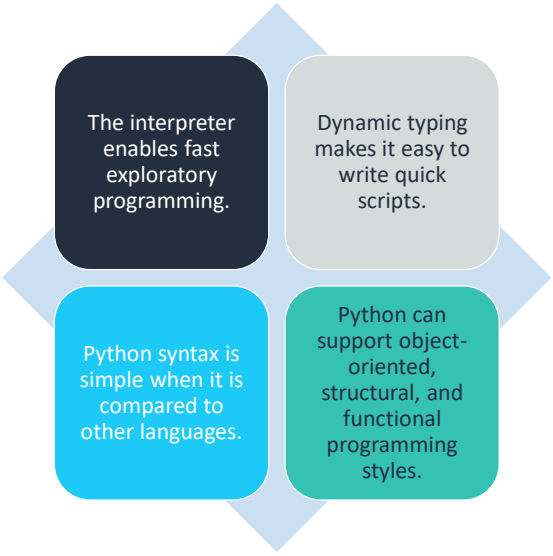
In this module, you will learn how to:

- Explain what Python is
- List reasons for using Python
- List examples of programs that can be used to write Python code
- Compare Python to shell scripting
- Introduction to AWS Lambda

What is Python?

Python is a free, easy-to-learn, general-purpose programming language.

Why Python?


The interpreter enables fast exploratory programming.
Dynamic typing makes it easy to write quick scripts.
Python syntax is simple when it is compared to other languages.
Python can support object-oriented, structural, and functional programming styles.

Why Python? continued

Another reason to use Python is that it works across platforms. It works on macOS, Linux, Microsoft Windows, and other platforms.

Though Bash scripts can be powerful, they mostly run on GNU/Linux.

Where can you write Python?

Python can be written in any text editor if you have the interpreter also installed. Many developers use special programs that are called *integrated development environments (IDEs)*. IDEs help with finding syntax and exception errors.

Some examples of programs that are used to write Python code include:

Python(x,y)	**AWS Cloud9**	**Microsoft Visual Studio Code**	**Eclipse**
Notepad++	**Spyder**	**PyCharm**	**Vim**

Integrated development environment

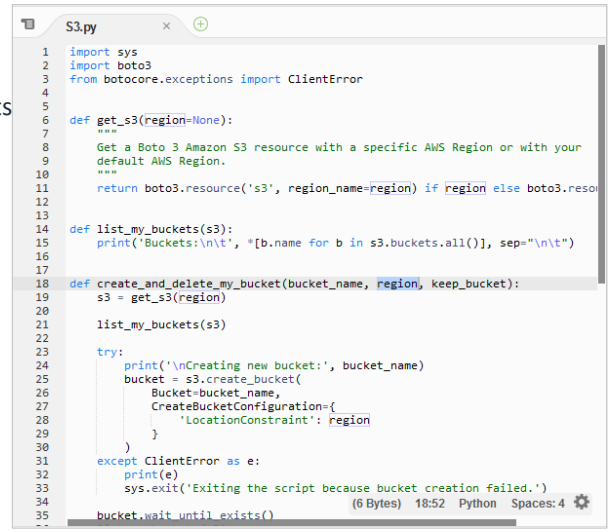
The following core capabilities are associated with IDEs:

- Syntax highlighting – Highlights keywords within the programming language
- Code completion – Similar to automatic completion for natural languages on modern cellular phones
- Debugging – Enables line-by-line inspection of the code while it is running with breakpoint capabilities
- Version control – Integrates popular version control systems, such as git and subversion

Using an IDE: Syntax highlighting

- Declaration of functions, classes, and imports have similar color.
- Comments are colored green.
- All syntax highlighting is customizable.

Note: The screen capture uses AWS Cloud9 as the IDE.

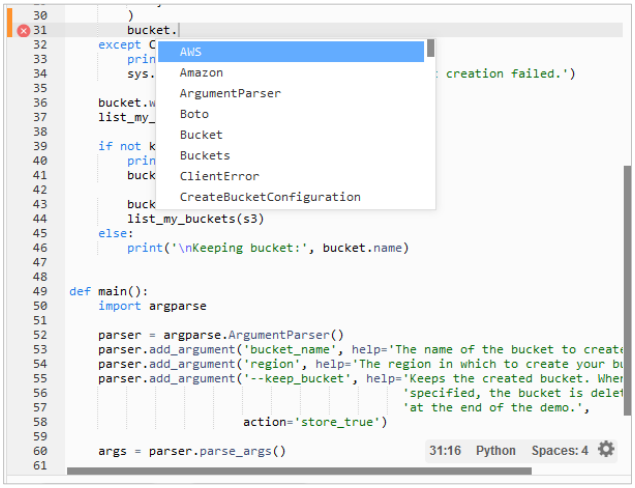


aws re/start

Using an IDE: Code completion

- Functions are portrayed differently in a selection.
- Functions include arguments that are available.

Note: The screen capture uses AWS Cloud9 as the IDE.

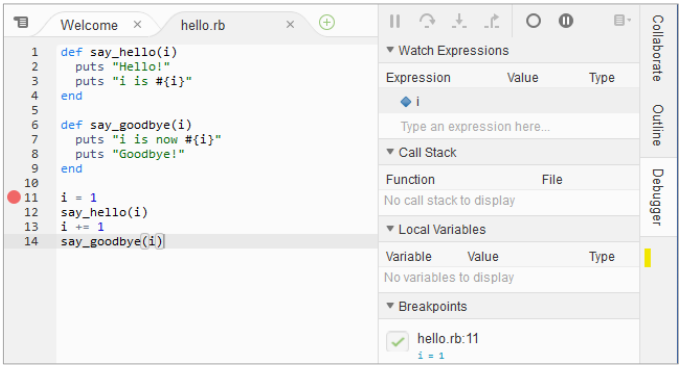


aws re/start

Using an IDE: Debugging

- Step through code line by line.
- Inspect variables as you step.
- Change values as you inspect.

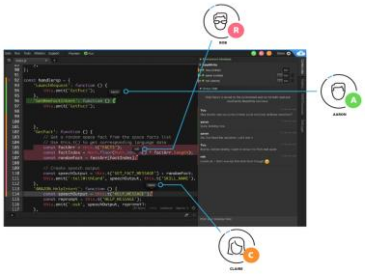
Note: The screen capture uses AWS Cloud9 as the IDE.



aws re/start

AWS Cloud9: Cloud-based IDE

Start projects quickly and code with only a browser.



Code together in real time.

Build serverless applications with ease.

aws re/start

AWS Cloud9 is a cloud-based IDE that lets you write, run, and debug your code with a browser. It combines the rich code editing features of an IDE with access to a full Linux server for running and storing code. Some of the editing features include code completion, hinting, and step-through debugging.

AWS Cloud9 provides a few key benefits:

- *Start projects quickly and code with only a web browser*. All that you need is an AWS account. You can log into the AWS Management Console, create an AWS Cloud9 environment, and start using it in your browser.
- *Code together in real time*. For example, you don't need to commit your changes to Git and then ask a colleague to look at the changes on their machine. AWS Cloud9 enables you to code together in real time through the AWS Cloud9 interface. AWS Cloud9 provides real-time feedback so you can see what your colleague is typing as they are typing it and give them feedback.
- *Build serverless applications with ease*. For example, you can edit and debug AWS Lambda functions locally. You don't need to upload your code to the Lambda console for testing and debugging.

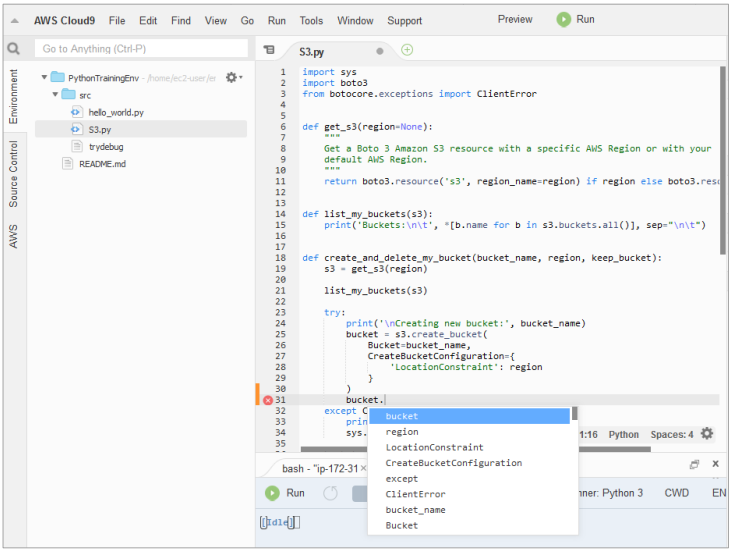
For more information about AWS Cloud9, see the product page: AWS Cloud9

AWS Cloud9 development, runtime, and version control

Development environment	Runtime environment	Version control system
		
Your computer	Amazon Elastic Compute Cloud (Amazon EC2) + AWS Cloud9 environment	AWS CodeCommit repository
+	**OR**	**OR**
		
AWS Cloud9	Your server + AWS Cloud9 environment	Other remote repository type

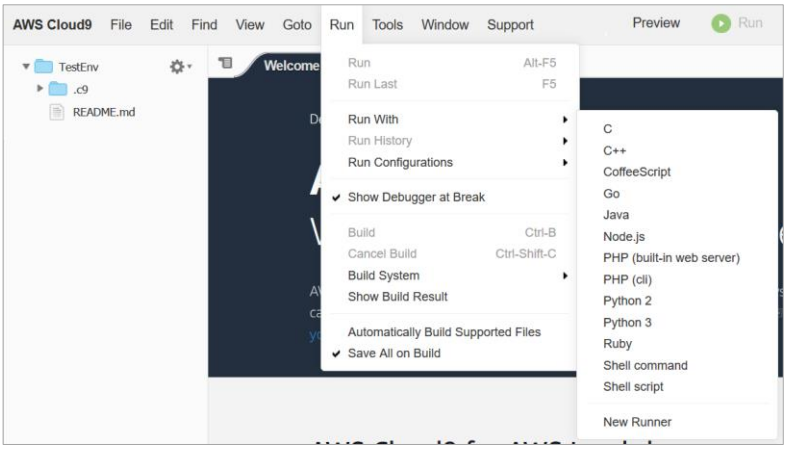
AWS Cloud9 features

- Live syntax checking
- Automatic indentation
- Automatic completion
- Code folding
- Split panes
- Version control integration
- Multiple cursors and selections



aws re/start

AWS Cloud9: One IDE, many languages



aws re/start

With AWS Cloud9, you can use many runtime environments that enable you to build and run your code.

Running Python applications

After you finish writing an application in Python, you can:

- Run the application from the command line: `$python myPythonapp.py`
- Run it from inside the IDE

However, what if you must run your code or application? For example:

1. Make it available to users around the world
2. Handle millions of users

AWS Lambda

- Upload your code to AWS Lambda.
- Set up your code to trigger from an event, such as a user who is visiting your webpage.
- Lambda runs your code only when it is triggered, and it uses only the compute resources that are needed.
- You pay only for the compute time that you use.
- Multiple languages are supported.
- AWS Cloud9 is included in the Lambda interface, so you can share code with developers.

aws re/start

AWS Lambda offers several benefits:

- You have no servers to manage. You pay only for the compute time that you use, with continuous scaling. With sub-second metering, you do not pay for compute time when your code is not running. AWS Lambda works well for variable and intermittent workloads.
- You can run code for virtually any application or backend service, with almost no administration—including server and operating system maintenance.
- You can set up your code to automatically trigger from other AWS services, or you can call it directly from any web or mobile application.

Lambda supports various different programming languages, including Go, Node.js, Java, C#, and Python.

AWS Lambda example

Users capture an image for their property listing.

The mobile app uploads the new image to Amazon Simple Storage Service (Amazon S3).

A Lambda function is triggered and calls Amazon Rekognition.

Amazon Rekognition retrieves the image from Amazon S3 and returns labels for the detected property.

You will learn more about AWS Lambda later in the course.

aws re/start

This example illustrates the use of Lambda for an image recognition application, and demonstrates how it works.

First, the users capture an image for their property by using an app on their mobile phone. The mobile app then uploads the new image to Amazon Simple Storage Service (Amazon S3). Adding this image to Amazon S3 triggers a Lambda function and calls Amazon Rekognition. Amazon Rekognition can identify objects, people, text, scenes, and activities. It provides highly accurate facial analysis and recognition. Lambda retrieves the image from Amazon S3 and returns labels for the property and its amenities.

This is just one example of a Lambda use case. With Lambda, you can run code for virtually any application or backend service. Other Lambda use cases include:

- Automated backups
- Processing objects that are uploaded to Amazon S3
- Event-driven log analysis
- Event-driven transformations
- Internet of Things (IoT)
- Operating serverless websites

Other tools: Shell scripting

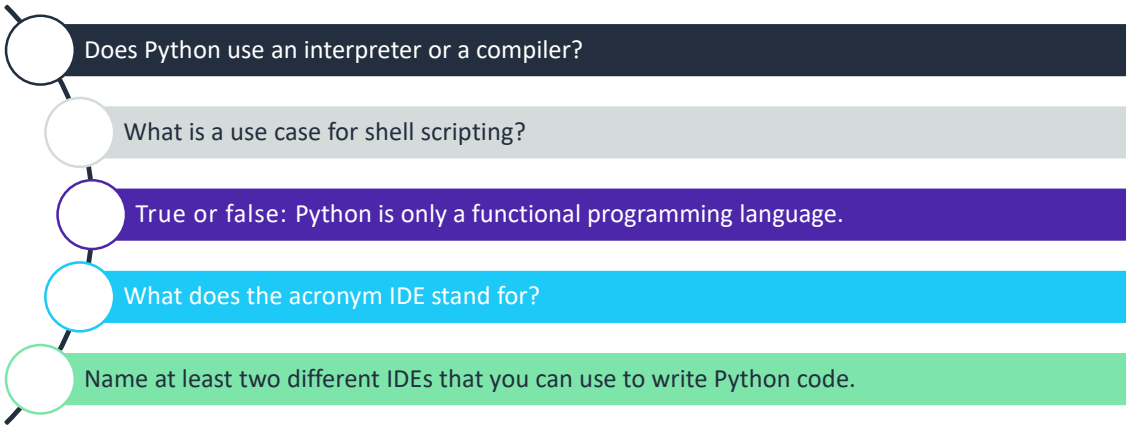
Shell scripting commands are run directly from the command line of an operating system. They are available on any machine and on any operating system without the need to install new software.

Different environments require different syntax or types of shell scripting, such as Bash and Zshell.

Shell scripting versus Python

- Shell scripting can be a powerful tool for system administration and command-line work, but it can be challenging when you want to use more complicated data structures.
- For example: Python can perform some actions—such as creating an HTTP server—in a single line. However, it could require many lines of code to do the same action in Bash.
- Python has many external libraries and resources. It is a complete programming language.

Checkpoint questions



aws re/start

Answers:

1. Python uses an interpreter
2. Some examples of uses for shell scripts are:
    - Alerting a system administrator about low disk space
    - Moving log files into an archive directory
3. False
4. Integrated Development Environment
5. Eclipse, PyCharm, Spyder, Notepad++, others

Key takeaways

- Python is a free, general-purpose programming language. It has a simpler syntax compared to other programming languages.
- Python can be written in any text editor if the interpreter is also installed.
- An integrated development environment (IDE) provides several advantages for writing code.
- Shell scripting is a powerful tool for administration. When administration becomes complicated, Python is generally preferred for such tasks.

aws re/start

Some key takeaways from this lesson include:

- Python is a free, general-purpose programming language. It has a simpler syntax compared to other programming languages.
- Python can be written in any text editor if the interpreter is also installed.
- An integrated development environment (IDE) provides several advantages for writing code.
- Shell scripting is a powerful tool for administration. When administration becomes complicated, Python is generally preferred for such tasks.