aws re/start

Introduction to Python

Python Fundamentals

Name of presenter

Date

Introduction to Python

What you will learn

At the core of the lesson

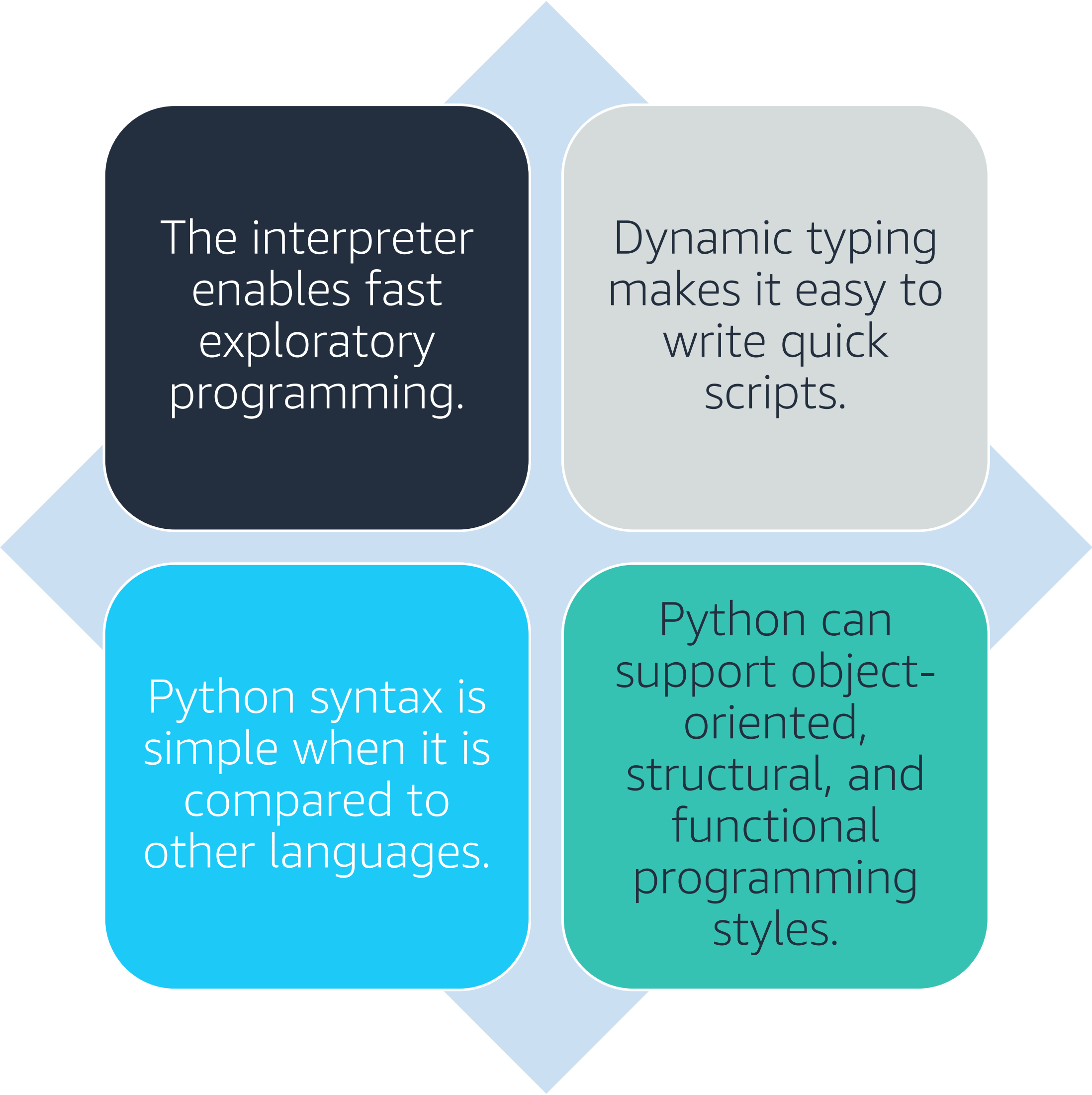
You will learn how to:

- 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

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.

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)

Microsoft Visual Studio Code

Eclipse

Notepad++

Spyder

PyCharm

Vim

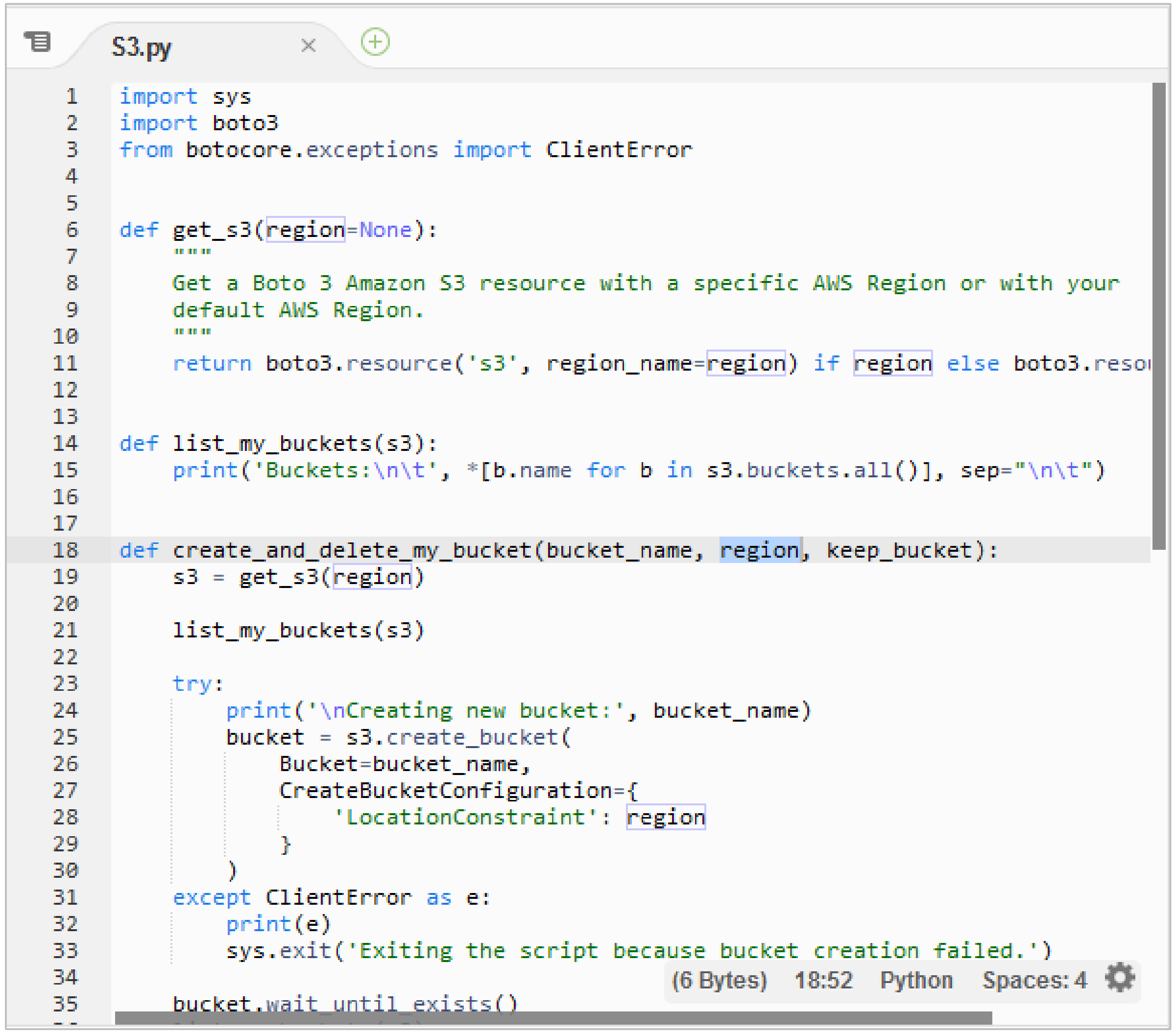
Integrated development environment

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

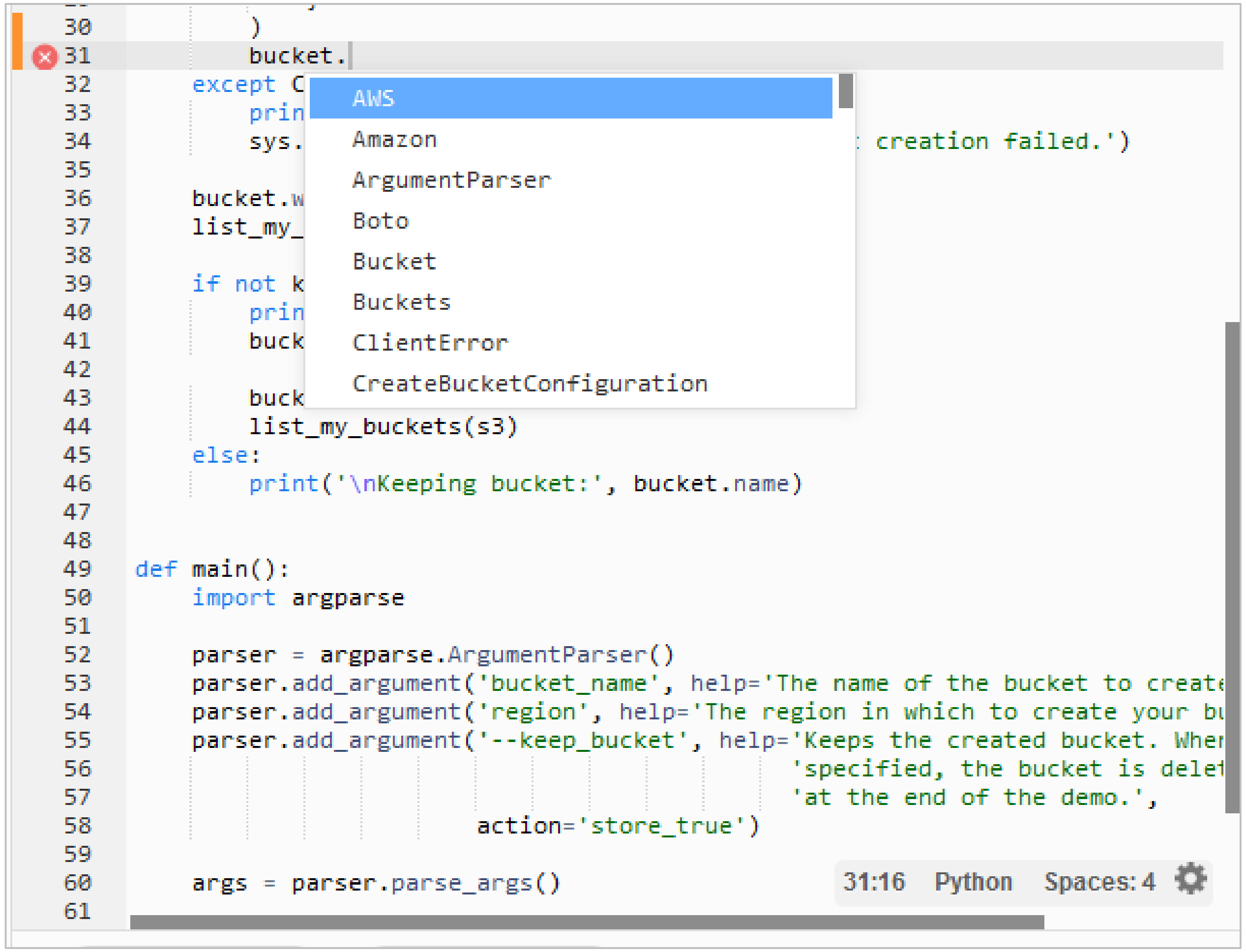
- Declaration of functions, classes, and imports have similar color.
- Comments are colored green.
- All syntax highlighting is customizable.



```
import sys
import boto3
from botocore.exceptions import ClientError


def get_s3(region=None):
    """
    Get a Boto 3 Amazon S3 resource with a specific AWS Region or with your
    default AWS Region.
    """
    return boto3.resource('s3', region_name=region) if region else boto3.resou


def list_my_buckets(s3):
    print('Buckets:\n\t', *[b.name for b in s3.buckets.all()], sep="\n\t")


def create_and_delete_my_bucket(bucket_name, region, keep_bucket):
    s3 = get_s3(region)

    list_my_buckets(s3)

    try:
        print('\nCreating new bucket:', bucket_name)
        bucket = s3.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
                'LocationConstraint': region
            }
        )
    except ClientError as e:
        print(e)
        sys.exit('Exiting the script because bucket creation failed.')

    bucket.wait_until_exists()
```

Using an IDE: Code completion

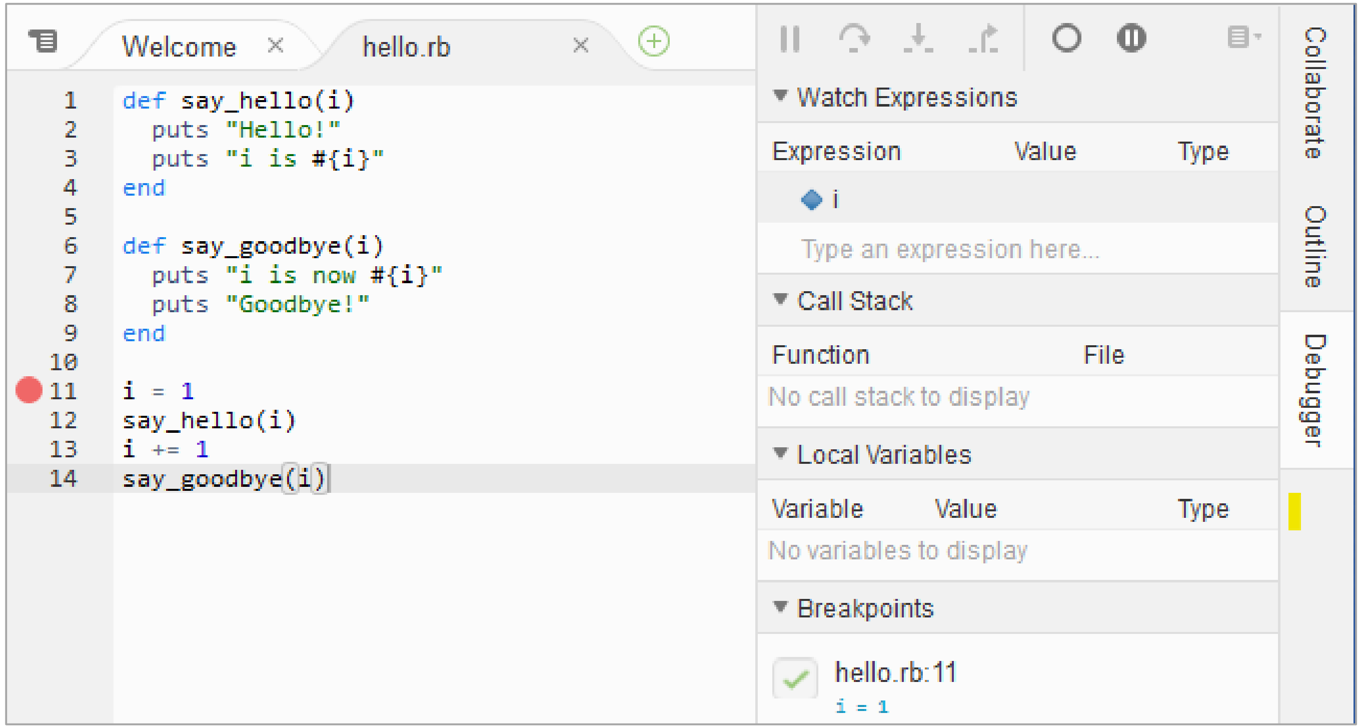
- Functions are portrayed differently in a selection.
- Functions include arguments that are available.



```
            )
        bucket.
    except C
        prin
        sys.                                        creation failed.')

    bucket.w
    list_my_

    if not k
        prin
        buck

        buck
        list_my_buckets(s3)
    else:
        print('\nKeeping bucket:', bucket.name)


def main():
    import argparse

    parser = argparse.ArgumentParser()
    parser.add_argument('bucket_name', help='The name of the bucket to create
    parser.add_argument('region', help='The region in which to create your bu
    parser.add_argument('--keep_bucket', help='Keeps the created bucket. Wher
                                                        'specified, the bucket is delet
                                                        'at the end of the demo.',
                                    action='store_true')

    args = parser.parse_args()
```

Using an IDE: Debugging

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



```
def say_hello(i)
  puts "Hello!"
  puts "i is #{i}"
end

def say_goodbye(i)
  puts "i is now #{i}"
  puts "Goodbye!"
end

i = 1
say_hello(i)
i += 1
say_goodbye(i)
```

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

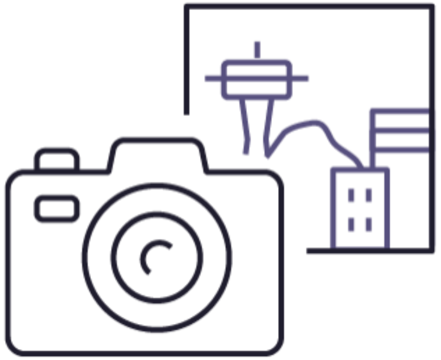
However, what if you must run your code or application? For example:

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

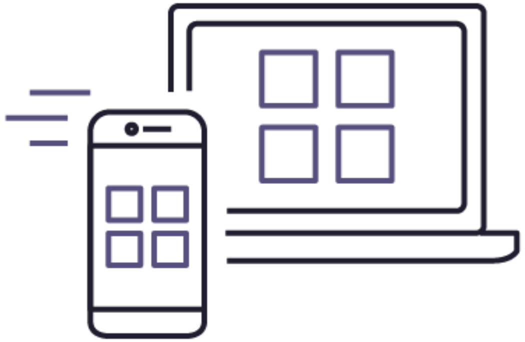
AWS Lambda

- 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.

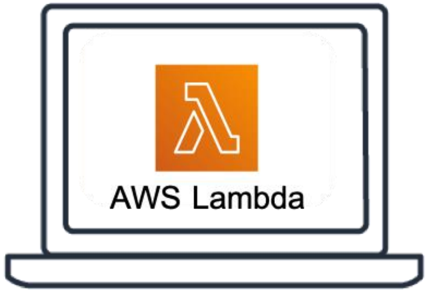
AWS Lambda example



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.

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.

BASH
THE BOURNE-AGAIN SHELL

zsh

Bash logo – Copyright Free Software Foundation, licensed under the Free Art License. Designed by ProspectOne. Available at the Official GNU Bash Logo GitHub page.

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.

Bash logo – Copyright Free Software Foundation, licensed under the Free Art License. Designed by ProspectOne. Available at the Official GNU Bash Logo GitHub page.

Checkpoint questions

- Does Python use an interpreter or a compiler?
- What is a use case for shell scripting?
- True or false: Python is only a functional programming language.
- What does the acronym IDE stand for?
- Name at least two different IDEs that you can use to write Python code.

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.