PYTHON CRASH COURSE

A BEGINNER'S GUIDE TO MASTER THE BASICS OF PYTHON AND DATA SCIENCE. LEARN CODING WITH THIS MACHINE LEARNING TOOL. DISCOVER THE ENDLESS POSSIBILITIES OF COMPUTERS AND CODES.

ERIC WALL

PYTHON CRASH COURSE

A BEGINNER'S GUIDE TO MASTER THE BASICS OF PYTHON AND DATA SCIENCE. LEARN CODING WITH THIS MACHINE LEARNING TOOL. DISCOVER THE ENDLESS POSSIBILITIES OF COMPUTERS AND CODES.

ERIC WALL

PYTHON CRASH COURSE

A Beginner's Guide to Master the Basics of Python and Data Science. Learn Coding with This Machine Learning Tool. Discover the Endless Possibilities of Computers and Codes.

ERIC WALL

Table Of Contents


DATA BREACH
CYBER ATTACK
SYSTEM PROTECTION
SECURUTY BREACH
SYSTEM
SAFETY

Introduction

Learning any programming language, including Python, opens the door to the computer world. Using programming, you can do almost everything a computer can do—giving you plenty of room to be creative. If you think of a need—say, to count the word frequency in a Harry Potter novel—you can program it yourself. If you have a good idea, such as a website for mutual learning, you can open up your computer and start writing. Once you learn to program, you'll find that software is mostly about brains and time, and everything else is extremely cheap. There are many rewards to be gained for writing programs. It could be an economic return, such as a high salary or starting a publicly traded internet company. It can also be a reputational reward, such as making programming software that many people love or overcoming problems that plague the programming community. As hackers and painters put it, programmers are as much creators as painters. Endless opportunities for creativity are one of the great attractions of programming.

Programming is the basic form of human-machine interaction. People use programs to operate machines. Starting with the industrial revolution of the 18th century, people gradually moved away from the mode of production of handicrafts and towards the production of machines. Machines were first used in the cotton industry, and the quality of the yarn produced at the beginning was inferior to that produced by hand. However, the machines can work around the clock, tirelessly, and in prodigious quantities. By the end of the 18th century, most of the world's cotton had become machine-made. Today, machines are commonplace in our lives. Artificial intelligence is also penetrating more and more into production and life. Workers use machines to make phones and other devices, doctors operate with machines to perform minimally invasive surgeries, and traders use machines to trade high-frequency stocks. To put it cruelly, the ability to deploy and possess machines will replace bloodlines and education as the yardstick of class distinctions in the future. **This** is why programming education is becoming more and more important.

Changes in the world of machines are revolutionizing the way that the world works. Repetitive work is dead, and the need for programmers is growing. A lot of people are teaching themselves to program to keep up with the latest trends. Fortunately, programming is getting easier. From assembly language to C to Python, programming languages are becoming more accessible.


NSIS

CHAPTER 1:

Getting Started

Before using Python, you first need to install and run it on your computer. Once you do that, you will be able to write your first program. Python is known as a programming platform that cuts across multiple platforms. You can use it on Linux, macOS, Windows, Java, and .NET machines freely and as an open-source resource. Most of the Linux and Mac machines come preinstalled with outdated versions. That is the main reason why you will need to install the latest, current version.

Download Python to your PC

Depending on your operating system, you might need to download the latest version of Python to your system. For PC's running on Windows OS, you can download the Windows Python interpreter from Python's website free of charge. Depending on the version of your operating system, ensure to download the appropriate version of Python that works for it. PCs running on Linux and Mac OS X already have Python pre-installed so you don't have to make any further installing asides a code editor. However, since a good many OS X and Linus systems still run the Python 2.x series, you might want to install the newer version of Python. To do this, you have to visit Python's website and download a 3.x series version appropriate for your operating system.

Install a Python interpreter

The next step is to install a Python interpreter. Do this without altering any changes to the program. From this point, Python can be integrated into your

PC's Command Prompt by activating the last option on the list of available modules.

1. Test the installation by opening the command prompt on Windows and Terminal on Linux or Mac. Enter the word "Python," and click **Enter** . The package would load and display the version of Python you have installed. From here, the Python interpreter command prompt will load. Enter ‘print("Hello, World!")’ and click enter. The text "Hello World!" should appear under the Python command line.

2. Feel free to experiment with the interpreter. Try out the functionality of codes in the interpreter before adding it to your program. Doing this can aid you in learning the workings of the commands in Python and in being able to write a throw-away program.

3. Learn the basics of how variables and objects are run on Python.

4. You can learn to use Python in carrying out simple calculator functions that will help familiarize you with the syntax of the program, as well as how strings and numbers are managed.

5. To open the interpreter, visit your command prompt center or Terminal. Enter Python into the prompt and click **Enter** to be directed to the Python command prompt. To execute the interpreter, you would have to manually navigate your way to the Python directory if Python isn't already integrated to your Command Prompt.

Use Python to carry out basic arithmetic problems such as addition, subtraction, multiplication, exponent, etc. This is done using operators in Python, for instance, to calculate the values of 7^2, 3+7, and 5^7. To calculate the following, you enter them into the prompt as follows:

```
>>> 3 + 7 10

>>> 7 ** 2 # 7 squared

49

>>> 5 ** 7 # 5 to the power of 7

78125
```

Note that the interpreter in Python programming does not execute the # sign.

An easier way to run Python is by using Thonny IDE. This is because it is bundled with the latest version of Python. This is an advantage since you will not need to install it separately. To achieve all that, you can follow the simple steps below:

- First, you will need to download Thonny IDE.
- Then run the installer to install it on your computer.
- Click on **File** option, then new. Save the file as a **.py** extension, for instance, morning.py or file.py. You are allowed to use any name for the file, as long as it ends with **.py** . Write the Python code on the file before saving it.

- To run the file, click on **RUN** ; then run the current script. Alternatively, click on **F5** to run it.

There is also an alternative to install Python separately; it does not involve installing and running Python on the computer. You will need to follow the listed steps below:

- Look for the latest version of Python and download it.
- The next step is to run the installer file in order to install Python.
- When installing, look for **Add Python to environment variables** . This will ensure that Python is added to the environment variables and that you will enable you to run Python from any computer destination and part. You have the advantage to choose the path to install Python.
- When you complete the process of installing, you can now run Python.

There is also an alternative and immediate mode to run Python. When Python is installed, you will type Python on the command line; the interpreter will be in immediate mode. You can type a Python code, and when you press **Enter** , you will get the output. For instance, when you type

1+1 and then press **Enter** , you will get the output, 2. You can use it as a calculator and you can quit the process. For that type ‘quit,’ then press **Enter** .

The second way to do it is by running Python on the Integrated Development Environment. You can use any editing software to write the Python script file. All you need to do is to save it under the extension **.py** . This is considered a lot easier when you use an IDE. The IDE is a feature that has distinctive and useful features like file explorers, code hinting, syntax checking, and highlighting that a programmer can use for application development.

You need to remember that when you install Python, there is an IDE labeled **IDLE** that will also be installed. That is what you will use to run Python on the computer and it is considered the best IDE for beginners. You will have an interactive Shell when IDLE is opened. This is the point where you can have a new file and ensure that you save it as a **.py** file.

Who can use Python Programming?

There is a big challenge out there in choosing a programming language that you can use for your coding businesses. The bigger question is which language are you supposed to learn? Python is a program that is easy to use, and there are known companies that use it. This is one of the reasons why you should adapt to its uses. This is also the reason why worldwide developers have taken advantage of it.

Google: Since the beginning of Python, Google has been its supporter. They chose Python because it was easy to maintain and deploy, and is faster in delivery. The first web-crawling spider used for Google was in Java 1.0. It was difficult to use and maintain, and they had to do it again on Python. Python is one of the main programming languages that Google uses. The others include Java, C++, and Go which are used for production. Python is an important part of Google. They have been using it for many years, and it remains a system that is evolving and grows. Many engineers that work for Google prefer using Python. They keep seeking engineers with Python skills.

Facebook: Production engineers that work for Facebook have positive comments about Python. This has made Python among the top three programming languages after C++ and Hack. Facebook adopted Python because it is easy to use. With over 5000 services on Facebook, this is definitely the best programming language. The engineers do not need to maintain or write much code, and this allows them to focus on live improvements. This is one of the reasons why Facebook infrastructure scales efficiently. Python is used for infrastructure management, network switch setup, and imaging.

Instagram: From 2016, engineers working for Instagram declared that they were running the biggest Django web framework that was entirely written in Python. The engineers stated that they like Python because of its simple use and how practical it is. That is why the engineers have invested their resources and time in using Python in all their trades. In recent times, Instagram has moved their codes from Python 2.7 to Python 3.

Spotify: Spotify is a music-streaming platform that uses Python as its programing language for back-end services and data analysis. The reason why Spotify decided to use Python is that they like the way it works in writing and in coding. Spotify will use its analytics to offer its users recommendations and suggestions. For the interpretation, Spotify uses Luigi which collaborates with Hadoop. The source will handle the libraries that work together. It will consolidate all the error logs and helps in troubleshooting and redeployment.

Quora: Before implementing their idea, Quora decided to use Python programming for its question-and-answer platform. The Quora founders decided to go for Python because it was easy to read and write it. For great performance, they implemented C++. Python is still considered because of the frameworks it has, like Pylons and Django.

Netflix: Netflix uses Python programming language to help in the analysis of data from their servers. They also use that in coding and other Python applications. It uses Python in the Central Alert Gateway, and also for tracking any security history and changes.

Dropbox: This cloud storage system uses Python for the desktop client. Their programs are coded in Python. They use different libraries for

Windows and Mac. And the reason is it is not preinstalled on any Mac or Windows and the Python version differs.

Reddit: Python programming language was used to implement Reddit. They chose Python because it has different versions of code libraries, and it was flexible to develop.

What Can You Do With Python Programming?

There are numerous applications for Python programming, like machine learning, data science, and web development. In addition, other several projects can use Python skills.

With Python programming, you are able to automate the boring stuff. This is the best approach for beginners. It helps with spreadsheet updates and renaming files. When you get to master Python basics, then this is the best point to start with. With the information, you will be able to create dictionaries and objects, work on files, and do web scraping.

Python will help you stay on top of the Bitcoin prices. Bitcoin and cryptocurrencycurrency have become a popular investment. This is because of its price fluctuation. To know the right move concerning Bitcoin, you will need to be aware of their prices. With Python, it is possible to create a price notification for Bitcoin. This is the best way to start working with Python for cryptocurrencycurrency.

When your intention and plan is to create a calculator, the Python is the best programming language. You will be able to build back-end and front-end services, which are the best when it comes to deployment. It is important to create applications that users can easily use. If your interest is in UX/UI design, then Python has a graphical user interface that is easy to work with.

Python is the best programming language to use when mining data from Twitter. With the influx of technology and the internet, it is easy to get data and information easily. Data analytics is a very important concept. It involves what people are talking about and their behavioral patterns. To get all the answers, Twitter is the best place to start with when your interest is in data analysis. There is a data-mining project on Twitter, and that is when your Python skills will come handy.

You will have the ability to create a microblog with a flask. In recent times, most people have a blog. But again, it is not a bad idea to have your

personal hub online. With Instagram and Twitter, microblogging has become a popular concept. With Python skills, you will be able to create your own microblog. When you are into web development, you do not need to be worried about knowing Flask. You can learn about it online and then move to Django, which helps build web applications on a large scale.

With Python skills, it is possible to build a blockchain. The main reason for the development of blockchain was financial technology, even though it is spreading to other industries. As of now, blockchains can be used for any type of transaction, like medical records or real estate. When you get to build one for yourself, you will understand it better. You need to remember that, blockchain is not just for the individuals who are interested in cryptocurrency. When you build one, you will have a creative way for technology implementation to your own interest.

You can bottle your Twitter feed with your Python skills, and this will help in web applications. You can create a simple web application that can help in navigation on your Twitter feed. You will not be using Flask, but rather Bottle, which is a low-dependency approach that is easy and quick to implement.

There are PyGames that are easy to play using Python skills. You can use the skills to code several games and puzzles. With the Pygame library, it is easier to create your own games and developing them. It is an open and free source with computer graphics and sound libraries, and these help in adding up interactive functions in the application. There are different games that can be used for library creation.

With Python, it is possible to create something in relation to storytelling. Since the language is easy to use, it creates a better environment for development and interaction.


PHP
C#
PYTHON
JAVA
SQL

CHAPTER 2:

Variables and Simple Data Types

Variable is the term used to refer to the location of a memory. Also known as an **identifier** , and used for holding value.

In Python, we don't need to specify the variable type, because Python is an inferior language and is smart enough to get the type of variable.

Variable names can be a group of letters and digits but must start with a letter or an underscore.

Using lowercase letters for the name of the variable is recommended. ‘Rahul’ and ‘rahul’ are both two different variables.

Identifier Naming

Variables - example of identifiers. To identify the terms used in the program, an identifier is used. The rules for naming an identifier are set out below.

The variable must have an alphabet or underscore (_) as its first character.

The name of the identifier shall not contain any white space or any special character (@,#,%,^, &, *).

The name of the identifier must not be similar to any keyword set in the language.

Identifier names are case-sensitive. For example, ‘my name’ and ‘MyName’ are not identical.

Declaring Variable and Assigning Values

Python does not bind us to declare a variable before we use it in the application. It enables us to create a variable in the time required.

In Python, we do not need to declare explicitly the variables. A variable is declared automatically when we assign any value to the variable.

The operator equal (=) is used to assign value to a variable.

Object References

When we declare a variable it's necessary to understand how the Python interpreter works. The process of treating variables differs slightly from many other programming languages.

Python is a highly object-oriented programming language. Therefore, each data item belongs to a specific class type.

Object Identity

In Python, each object created uniquely identifies with Python. Python ensures no two objects will have the same identifier. The built-in Id() function is used to identify the identifier of the object.

We declared a few valid variable names such as name, name, etc. in the above example. However, it is not recommended because it can create confusion when we try to read code. To make code more readable, the name of the variable should be descriptive.

Multiple Assignment
In a single statement, which is also called a multiple assignment, Python allows us to assign a value to multiple assignments.

Python Data Types

Variables can hold values, with each value having a data type. Python is a dynamically typed language. Therefore, when declaring it, we don't need to define the type of variable. Implicitly the interpreter binds the value to their type.

a = 5

The variable holds the value of the integer, 5, and we have not defined its type. Python interpreter will interpret variables as an integer automatically.

Python allows us to check what type of variable the program uses. Python gives us the type() function, which returns the type of the passed variable.

Standard data types

A variable can hold many values.

Python provides different standard data types on each of them which define the storage method. Below you will find the data types defined in Python.

Numbers

Sequence Type

Boolean

Set

Dictionary

Numbers

Number numbers store numeric values.

Python creates a number object when a variable is assigned a number.

String

In the case of string handling, the operator (+) is used to combine two strings, as the "hello"+" Python operation returns "hello Python."

The example below illustrates a Python string.

Tuple

In many ways, a **tuple** is similar to a list. Like lists, tuples also include the collection of items of various data types. Tuple items are separated by a comma (**,**) and bound in parentheses (**()**).

A tuple is a read-only data structure because we can not change the size and value of the tuple items.

Dictionary

A dictionary is an unordered set of a pair of items with a key-value. It is like an associative array or a hash table where a specific value is stored on

each key. A key can hold any type of primitive data, whereas a value is an arbitrary Python object.

The dictionary items are separated by the comma (**,**) and enclosed within curly brackets(**{}**).

Boolean

A boolean type provides two incorporated values: True and False.

SET

A Python set is an unordered data-type collection. It is an iterable, is mutable (can change after it has been created), and has unique elements. In a set, the order of the elements is undefined; the changed sequence of the element may be returned. The set is created using a built-in function set(), or a sequence of elements passed through curly brackets and separated by a comma. It can contain different values.

Python Keywords

Assert

This keyword is used in Python as a debugging tool. The code is checked for correctness. It raises an *assertion error* if any error has been found in the code, and it also prints an error message.

Class

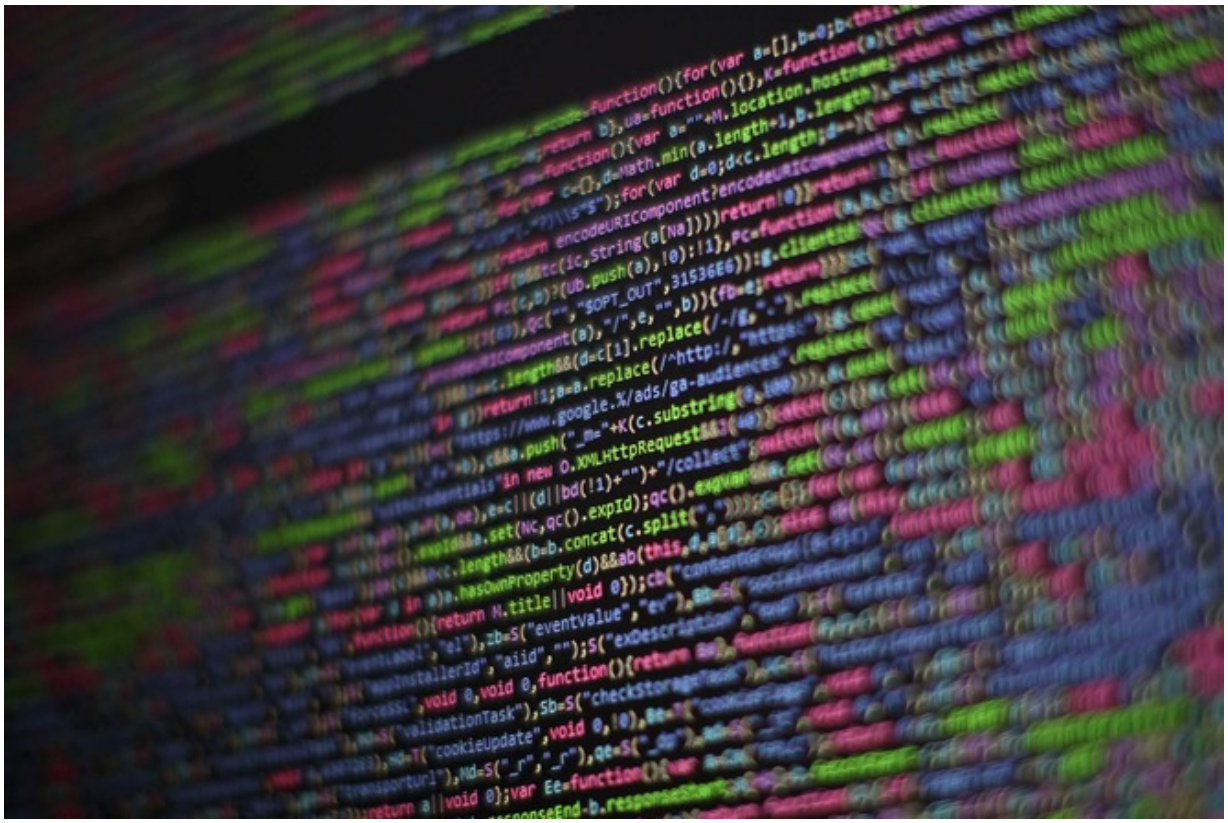
In Python, it is used to represent the class. The class is an object's blueprint. It is the variable collection and the method.

Break

It is used to terminate the execution of the loop and is used to transfer control to the end of the loop.

Elif

Uses this keyword to check multiple conditions. If the previous condition is false then check until you find the true condition.



CHAPTER 3:

Functions in Python

A function is a collection of codes that are grouped together to create a specific function or task. This section will teach you how to create and utilize functions to their fullest extent. When you use something like "print" or "append," you are using a function which is a group of code that does a certain task.

Creating a Method

The syntax to create a method is as follows:

def *functionName(parameters):*

    # code to execute

    # return value (optional)

def: 'Def' is a keyword that identifies a function. All functions start with this keyword.

Function Name: It is the function name, can be named any legal name of your choosing.

Parameters: Parameters are variables you can give the function to work with. When you print you give it a parameter of the word you want it to print.

Code: The code in the function is executed when the function is called.

Return Statement: The return statement is optional. It causes the function to return the value you choose for it to return.

Example 1:

```
x = 1
def value_increase(num):
        num += 1
        return num
print(value_increase(x))
```

In this example, we create a method called “value_increase” and have it take one parameter, num. The function will then take the value of num and add one to it. It then returns the value of num. Since this function has a return type, it returns a value. I call this method directly inside a print statement. The code will return ‘2’. This is because since it has a return type, it will print the value of what it returns.

Example 2:

```
x = 1
y = 5
def average (num1, num2):

        print("average is ", ((num1+num2) / 2))
average(x, y)
```

In the example above, I create another method that does not return anything. This is good for if you want the method to perform a specific function that does not actually set any value outside of it. The example above has a function that takes two numbers and prints the average of them within the function. Note that if we tried to print the function, it would return an error, because it does not return a value which can be used outside of itself. Also, note that a function always needs to be called below where it was written, or else the code will not find it.

Example 3:

```
x = 1
y = 5
def sum():
```

```
        print("average is ", ((x+y) / 2))
sum()
```

In this example, we create a function, sum, that takes no parameters at all. When a function has no parameters, you must still have brackets, except you put nothing within them. Although this function has no parameters, we can still access the variables above it to use. A function can access code outside of it if it is accessible. We will go more in-depth into this concept when we talk about classes; for now, just know that a function can access code if it is called somewhere in the same file.

Anonymous Functions

Anonymous functions are functions that are not declared in the standard way, using the ‘def’ keyword. Instead of the ‘def’ keyword, anonymous functions use the ‘lambda’ keyword. These are used to create small one-line functions. Unlike functions, anonymous functions cannot access variables outside of their own namespace or global namespace. These functions can take any number of arguments and return exactly one value in an expression form.

Example

```
sum = lambda num1, num2: num1 + num2
print(sum(1,2))
print(sum(5,10))
```

The variable sum gets the value of an anonymous function. It does this by using the lambda keyword, followed by the parameters you want it to take, separated by commas. After that, you use a colon (**:**) to separate the parameters from the expression. The value of sum then becomes the value of the expression, which in this case, is the value of *num1* , and *num2* combined together. Now when you print sum, it works exactly like a method. However, by using lambda the code becomes a lot more compact. This is again, another case of the programmer’s preference, and whether s/he would prefer to use an anonymous function over a normal one since there is no difference in the way they work, other than the structure.

Assignment

Create a more advanced calculator. Have three values. The first tells you which kind of operation to do (for example, if a == 1, add the values; if a == 2, subtract them). The next two values will be used to manipulate. Make the calculator support addition, subtraction, division, and multiplication, and have each feature in its own method.

```
action = 5  # variable that states which operation to do
x = 4.5  # first number to manipulate
y = 3  # second number to manipulate
# function to add variables
def addition (num1, num2): # takes two arguments
    return num1 + num2 # returns the sum of two numbers
# function to subtract variables
def subtraction (num1, num2): # takes two arguments
    return num1 - num2 # returns the difference of two numbers
# function to multiply variables
def multiply (num1, num2): # takes two arguments
    return num1 * num2 # returns the product of two arguments
# function to divide variables
def divide (num1, num2): # takes two arguments
    return num1 / num2 # returns the division of two numbers
if action is 1: # if the value of action is 1
    print("The sum is ", addition(x, y)) # calls the addition function
elif action is 2: # if the above fails, check if value of action is 2
    print("the difference is ", subtraction(x, y)) # calls the subtraction function
elif action is 3: # check for value 3
    print("the product is ", multiply(x, y)) # calls the multiply function
```

elif action *is* 4: # check for a value of 4

print("the answer is ", divide(x, y)) # calls the divide function

else*:*

print("invalid action, please use a action number between 1 and 4")

At this point, the code above should be pretty self-explanatory. You should attempt to spice things up a little. Maybe work with a list, to play with a few hundred values instead of two. Add some more functions and see how complex you can make this calculator. You have all the knowledge you need to make the most awesome calculator ever now.

CHAPTER 4:

Tuple in Python

A tuple is like a list, but we cannot change elements in a tuple.

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

tuple_mine = (21, 12, 31) print (tuple_mine) tuple_mine = (31, "Green", 4.7) print (tuple_mine)

Access to Python Tuple Elements

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

tuple_mine = ['t', 'r', 'o', 'g', 'r', 'a', 'm']

Press (tuple_mine [1]) #output: 'r' print (tuple_mine [3]) #output: 'g'

Negative Indexing

Just like lists, tuples can be indexed negatively.

Like lists, -1 refers to the last element in the list and -2 refers to the penultimate element.

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

tuple_mine = ['t', 'r', 'o', 'g', 'r', 'a', 'm']

print (tuple_mine [-2]) # The output is 'a'

Disc

The division operator, the colon, is used to access several elements in a tuple.

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

tuple_mine = ['t', 'r', 'o', 'g', 'r', 'a', 'm'] print (tuple_mine [2: 5]) #Production: 'o ',' g ',' r ',' un '

Press (tuple_mine [: - 4]) # 'Gram'

Important

Tuple elements are immutable in the sense that they cannot be changed. However, we can combine ingredients in a tuple with a concatenation operator (+). We can also repeat elements in a tuple with the (*****) operator, just like lists.

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

print ((7, 45, 13) + (17, 25, 76)) print (("Different",) * 4)

Important

Since we cannot change elements in a tuple, we cannot remove items. However, the entire tuple can be removed with the keyword ‘of.’

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

t_mine = ['t', 'k', 'q', 'v', 'y', 'c', 'd'] of the t_mine

Tuple Methods Available In Python

There are only two methods available for working with Python tuples. District)

When called, there are object numbers equal to *y* .

Index (y)

When called, it indexes the index of the first element equal to y.

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

t_mine = ['t', 'k', 'q', 'v', 'y', 'c', 'd'] print (t_mine.count ('t')) print (t_mine.index ('l '))

Tupelo Membership Test

The keyword for checking the specified item is in a tuple.

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

t_mine = ['t', 'k', 'q', 'v', 'y', 'c', 'd']

print ('a' t_mine) # Output: True print ('k' in t_mine) # Output: false

Integrated Python Functions With Tuple

Method	*Description*	*Method*	*Description*
enumerate()	Returns an enumerated object. Contains the index and the value of all tuple elements as pairs.	Tuple ()	Convert an iterable to a tuple.
sorted ()	Take the articles in the tuple and return a new ordered list (do not order the tuple yourself)	Max ()	Returns the largest element in the tuple.

all()	Returns True if all elements of the tuple are true (or if the tuple is empty).	Total()	Returns the sum of all elements in the tuple.
len ()	Returns the length (the number of elements) in the tuple.	Minimum ()	Return the smallest element in the tuple.
		any()	Returns True if an element of the tuple is true. If the tuple is empty, False is returned.

String in Python

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

string_mine = 'Bunter' print (string_mine) string_mine = "Hallo" print (string_mine) string_mine = " 'Hallo' " print (string_mine) string_mine = "" I feel born a programmer "" "print (string_mine))

Access Items In A String

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

str = 'colorful' print ('str =', str)

print ('str [1] =', str [1]) # Output of the second print element ('str [-2] =', str [-2]) # Output the penultimate element

print ('str [2: 4] =', str [2: 4]) # Output from the third to the fifth element

Eliminate or Change In Python

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

the string_mine

String Operations

Multiple operations can be performed on a string, making it a common data type in Python.

Repeat chaining with the (**+**) operator; repeat with the (*****) operator

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

string1 = 'Welcome' string2 = 'Reprint' ('string1 + string2 =', string1 + string2)

print ('string1 * 3 =', string1 * 3)

exercise

Given string_a = "I'm awake" and string_b = "encoding in Python in pajamas."

Iteration of Strings

The control statement is used to continuously scan an entire scan until the specified number is reached before the scan ends.

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

tracker = 0 for the letter in "Welcome Again": if (letter == 'I'): tracker + = 1

Print (tracker, "letters found")

String Membership Test

The keyword is used to check if there is a second string.

Example

't' in "triumph" # Returns True

Integrated Python functions for working with strings

These include the enumerating() and len() functions. The len() function returns the length of the string.

Format Strings In Python

Escape sequences

Single and double quotes

Example

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

print ('You said: "Do we need a new team?"') # Escape with single quotes

double quotes escape

print ("They said:" We need a new team ")

Escape Sequences in Python

Escape sequences allow us to format our output to improve clarity for human users. A program will continue to run correctly without using escape sequences, but the production will create a lot of confusion for social users. Writing and displaying the edition in the expected version is part of good programming practice. The following are commonly used escape sequences.

Method	*Description*	*Method*	*Description*
\ n	ASCII line feed	\ b	ASCII backspace
\ "	Double quote	\\	Overturned pole
\ f	ASCII formatting	\a	ASCII Bell
\ New line	Backslash and ignored line breaks	\ '	Simple quotation marks
\ r	Back to ASCII transport	\ t	ASCII horizontal map
\ v	Vertical ASCII card	\ 000	Character With octal value 000
\ xHH	character With Hexadecimal value HH		

Examples

Start IDLE.

Go to the **File** menu and click **New Window** .

Enter the following:

print ("D: \\ lessons \\ programming")

print ("Print \ n in two lines")

Overview

Integers, floating-point numbers, and complex numbers are supported in Python. There are integers, moving and complex classes that can be used to convert different types of numerical data.

The presence or absence of a decimal point separates integers and floating-point numbers. For example, 4 is an integer, while 4.0 is a floating-point number.

Programmers often have to convert decimal numbers to octal, hexadecimal, and binary forms.

We can represent binary, hexadecimal, and octal systems in Python by merely prefixing the particular name. Sometimes called a constraint, type conversion allows us to change one type of number to another.

Integrated functions like int() enable us to convert data types directly. The same features can be used to convert strings.

We create a list in Python by putting **elements** in square brackets, separated by commas. In programming and especially in Python, the first time is always indexed with zero. For a list of five elements, we access index4 from index0. If you do not access items in a file in this way, an index error is generated.


PYTHON
10100
01101
11011
01001

CHAPTER 5:

Conditional Statements in Python and Control Flow Statements

These are going to be an important part of the code that we work with because they are going to ensure that your system is able to respond to the kind of input that the user provides to you. It is hard to predict how a user is going to work with the system. However, you can set up some of the conditions that you would like to look at and work from there to come up with the way your program is going to work.

As we can imagine here, it is pretty much impossible for a programmer to create something and guess ahead of time what answers or input the user is going to provide to the program. And the programmer can't be there watching each use work with the program either, which means that they need to work with the conditional statements. When these are set up in the right manner, it will ensure that the program is going to run properly, and will respond to any information that the user is providing to you.

There are a lot of different types of programs that are going to respond well to the conditional statements that we will talk about in this guide. These are actually pretty simple to work with, and we will take a look at some of the examples of how you can code with these conditional statements as well.

As we go through this section, we are going to take a look at the three main types of conditional statements: the if statement, the if else statement, and the elif statement. Let's take a look at how each of these statements works and when we would be able to use these conditional statements.

The If Statement

As we mentioned, there are three types of conditional statements that we can take a look at. The first one that we need to explore a bit is the if statement. Out of the three that we will spend some time on, the if statement is the most basic. These are not going to be used as much as some of the other options because they often leave a bit to be desired. However, they are a good springboard for helping us to learn what these conditional statements are about and how we can work with them.

With the if statement, the program is set up so that it can only proceed forward if the user provides us with an input that works with the conditions that we set ahead of time. If the input that we get from the user doesn't match with our conditions, then the program will just stop and nothing is going to happen.

As we can see already, there are going to be some issues with this because we don't want the program to just stop with the answer. It should still provide us with some of the basis that we need.

With this in mind, we can work with a simple code that shows us how the if statement is going to work. This is shown below:

```
age = int(input("Enter your age:"))

if (age <=18):

print("You are not eligible for voting, try next election!")

print("Program ends")
```

There are a few things that will show up with this code. If you have a user go to the program and state that their age is under 18 years, then the program will display the message that is listed there. The user can read this message and end the program right there.

But, things can go wrong if the user puts in that their age is above 18 years. This is true for the user, but because it doesn't meet the conditions that you coded in. Thus, the program will see it as false. Like the code is written right now, nothing is going to happen because it isn't set up to handle this. The user will just see a blank screen any time they enter an age that is over 18 years.

The If Else Statement

Now that we have had some time to take a look at the simple if statement, it is time for us to move on to the if else statement. The if statement is a good way to get a bit of practice in coding, but there are not going to be all that many times when we are programming and need to work with this kind of statement.

You want to make sure that when your user works with the program, no matter what input they use, something shows up on the screen.

If you use the if statement, like in the example above, and the user puts in an answer (above 18 years), using the code that we had from before, the screen will come back blank. This is definitely not something that we want to see, so we need to move on to the if else statement to see what we can do regardless of what information the user puts into the program.

The if else statement is going to provide us with an output, and will ensure that we provide these outputs to the user, regardless of the age or other information we provide to the program. So, with the example above, if the user comes in and says that they are 40 years old, then the code will still be able to respond to it.

There are a few options that you can use with this one, but with the idea of the voting option that we talked about with the if statement.

With this option, you are adding in the else statement, which will cover every age that doesn't fall under 18. This way, if the user does list that as their age, something will still appear on the screen for them. This can provide you with more freedom when working on your code and you can even add in a few more layers to this. If you want to divide it so that you get four or five age groups and each one gets a different response, you simply need to add in more if statements to make that happen. The else statement is at the end to catch everything else.

For example, you can take the code above and ask the user what their favorite color is. You could then have if statements to cover some of the basic colors, such as red, blue, green, yellow, orange, purple, and black.

If the user puts in one of those colors, then the corresponding statement will show up on the screen. The else statement will be added in to help catch any other colors that the person may try to use, such as pink or white.

The Elif Statements

The third type of conditional statement that we can work within this process is known as the elif statement. These are going to help us add another level to what we did with the other step. However, they are still going to make sure that the codes we write are as easy as possible.

We can create as many of these elif statements as possible in the code, as long as we make sure to add in the else statement at the end. The else statement ensures that we can handle any of the other answers that the user puts in, even the ones that we may not have thought of ahead of time.

When working with the elif statement, it is going to be similar to giving the user a menu to pick from. You can choose how many of these elif statements you would like to have present in the menu, similar to what is found in a lot of games. Then, the user can pick and choose which one they would like to work with. You can then have a certain action happen or a certain statement show up on the program to meet your needs.

Another thing to notice with the elif statement is that you can add in as many different options as your code needs. It is possible to make a small menu that just has two or three items in it, or it is possible to expand this out, to as many of these as you need, to make the code work properly.

The fewer options that you work within this one though, the easier your code writing will be, so keep that in mind when determining how many options are actually needed.

Now that we know a little bit about elif statements and how they work, let's dive in and take a look at a good example of one that you can write out. Open up your compiler and type in the following code:

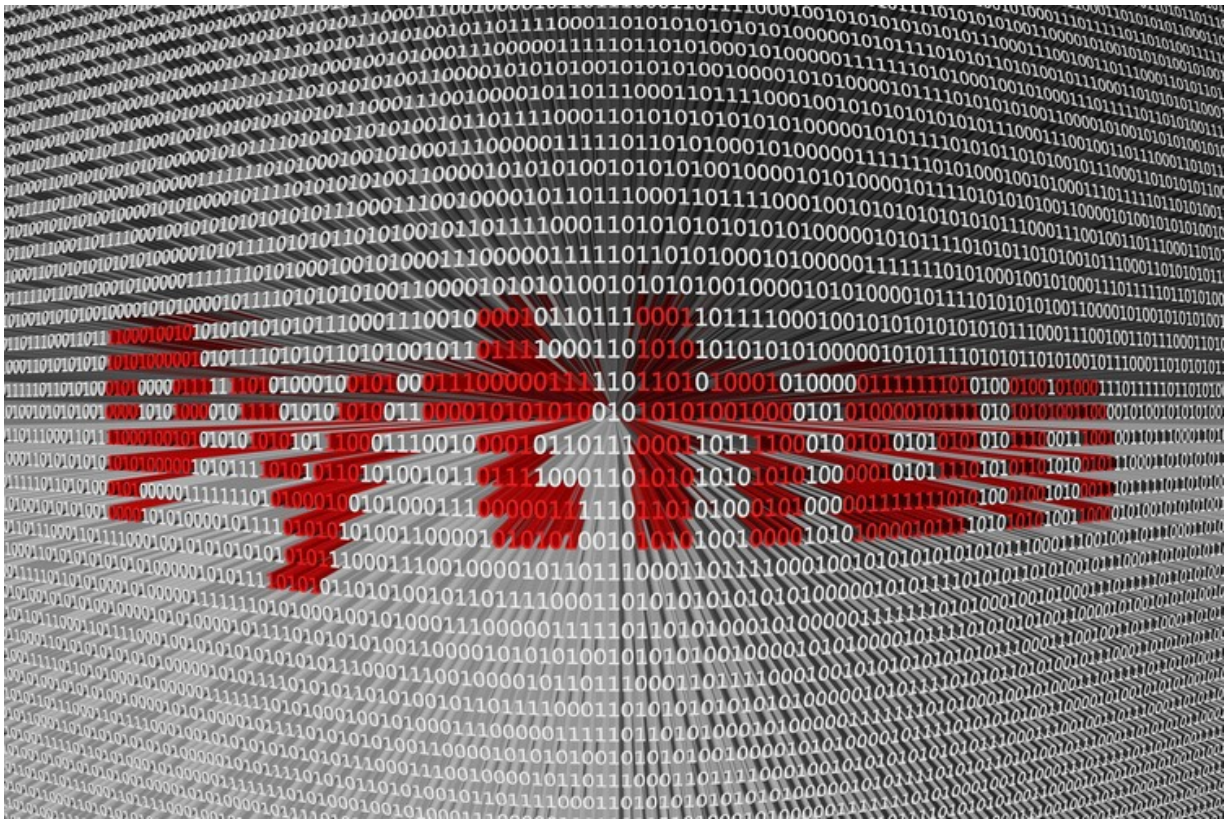
```
Print("Let's enjoy a Pizza! Ok, let's go inside Pizzahut!")

print("Waiter, Please select Pizza of your choice from the menu")

pizzachoice = int(input("Please enter your choice of Pizza:"))

if pizzachoice == 1:

print('I want to enjoy a pizza napoletana')

elif pizzachoice == 2:

print('I want to enjoy a pizza rustica')
```

```
elif pizzachoice == 3:
print('I want to enjoy a pizza capricciosa')
else:
print("Sorry, I do not want any of the listed pizza's, please bring a Coca Cola for me.")
```

With this option, the user can choose the type of pizza they want to enjoy, but you can use the same syntax for anything you need in your code. If the user enters the number 2 in the code, they are going to get a pizza rustica. If they don't like any of the options, then they are telling the program that they just want to have something to drink, in this case, a Coca Cola.

Control Flow

The control flow in a program highlights the order of steps of the program execution. In a Python program, control flow is carried out by function calls, conditional statements, and loops. Here, we will deal with if statements, while loops, and for-loops.



CHAPTER 6:

Classes

A **class** is a code of template that is used in creating various kinds of objects or entirely a major blueprint that is used in creating objects in the Python programming language. Classes consist of member variables with several kinds of methods involved and any other operations that may be involved in the program. A class is constructed by including the keyword ‘class’ just before the desired class name.

Classes in the Python world normally bear a certain function, called init function, that normally executes when a particular class is being initiated and values are being assigned to key-value object properties. This function is usually called automatically every time the class is used to create an object in a particular class. Another term, called method, defines the functions that belong to specific objects in the program. When defining particular methods, you usually provide the first argument to the method using a self keyword. Methods are normally able to access the class attributes in the program. This is what makes it possible to define an object. We could see that the objects are central to Python. To build an object of a given class is very simple:

object1 = Class1 (arg1)

We will see later in this book that machine learning models are based on classes.

mysample = LinearRegression ()

A class allows you to create a specific object that fits your needs. So, this object will have characteristics and methods. For example, a **boolean** is an object of the bool class. It has the following properties and methods:

BOOL1 = True

bool1.denominator

bool1.conjugate ()

How to Define A Class?

We use the keyword ‘class.’ We use the class constructor **__init__** (expect double underscore) in Python. Then, we define all the attributes in the constructor by providing default values. Suppose we want to build an image object in which we can modify characteristics:

myImage class:

def __init __ (self, resolution = 300, source = "./", size = 500):

self.resolution = resolution

self.source = source

self.size = size

If we want to create an object from this class, we can use the below:

image_1 = MyImage (source = "./ docs / image1.jpg")

We will have :

In []: image_1.size

Out []: 500

So, we see a first approach to simplify the creation of multiple objects. What interests us now is to associate methods with this class. For that, we just need to create functions whose first argument is ‘self’ and which use the attributes of our class.

myImage class:

def __init __ (self, resolution = 300, source = "./", size = 500):

self.resolution = resolution

self.source = source

self.taille size =

def poster_caract (self):

print ("Resolution:", self.resolution)

print ("Size:", self.taille)

print ("Source:", self.source)

def enlarge_image (self, factor):

self.size * = factor

We can then have:

In []: image_1.agrandir_image (2)

In []: print (image.taille)

1000

> We can see that the zoom_image method of the MyImage class has directly changed my object. Classes and objects include notions of inheritance and other standard functions.

Python is a good tool for data processing. You should know how to work with files in Python. Some of the file formats which can be used in Python include comma-separated values (CSV), HTML, text, and JavaScript Object Notation (JSON).

Before you can write to or read from a file, you should first open it by using the open() function. The method works by creating a file object, which can then be used for calling other methods that are supported.

Opening and Closing Files

For a file to be read, written to, or even modified, first it must be opened. This is done using the Python in-built function, open(). When invoked, the function will create file objects that can be used for calling support methods associated with it. Here is the method's syntax:

file objectName = open(file_name [,access_mode][,buffering])

The *file_name* is a string representing the name of the file to be opened. For the *access_mode* , the file can be opened to read, write, or append. The default mode for opening a file is *read (r).* If the value for buffering is set to 0, then no buffering will be done. If it is set to 1, line buffering will be done after accessing the file. If you specify another integer greater than 1 for buffering, then the buffering will be done at the size that you have

specified. The integer is normally taken at buffer size. If it is set to a negative integer, the default buffering size for the system is used.

w The file is opened for writing only. If the file exists, it is overwritten. If the file does not exist, a new one is created.

a The file is opened for appending.

a+ The file is opened for both reading and appending.

The file object is related to these attributes:

file.closed It will return *true* if the file is closed, and *false* otherwise.

file.mode This returns the mode in which the file is opened.

file.name This will return the file name.

Here is an example:

```
#!/usr/bin/Python3

# Opening the file

p = open("friends.txt", "wb")

print ("The file name is: ", p.name)

print ("File closed? ", p.closed)

print ("The mode of the file? ", p.mode)

# close the file

p.close()
```

You should have a file named *friends.txt* in the directory you have saved the file with the above script. On running the code, I get the following output:

```
The file name is:  friends.txt
File closed?  False
The mode of the file?  wb
```

We have the name of the file, which was obtained by calling the name attribute. The ‘false’ in the result tells us that the file is not closed. Also, it is clear that the file has been opened in binary format for writing.

To close the file, you should call the *close()* method. This method should be called for closing a file. It first flushes the unwritten information; then it closes the file. Once the file has been closed, no further writing can be done. If the reference object for the file is assigned to some other object, Python will automatically close the file. Whenever you need to close a file, call the *close()* method. The method syntax is as follows:

fileObject.close();

Example

```
#!/usr/bin/Python3

# Opening the file

p = open("friends.txt", "wb")

print ("The file name is :", p.name)

# Close the opened file

p.close()

print("File closed? ", p.closed)
```

Run the code in the directory with the file *friends.txt.* The following result will be printed:

```
The file name is : friends.txt
File closed?  True
```

In our previous example, we got ‘false’ when we called the *p.closed* property. This meant that the file was not closed. The reason is that we had not called the *close()* method. In the above example, we have called the *close()* method on our file object. This closed the file. Hence we get ‘true’ after calling the *p.closed* property. This means that the file has been closed.

Reading from Files

The *read()* method helps us read a string from a file. The file must be opened before reading. Other than text data, Python files may also have binary data. The syntax is:

file_name.read([count]);

The parameter to the function is the number of bytes that you need to read from the file. The method usually begins to read from the beginning of the file. If you don’t specify a value for the count, then the method will read from the file as much as it can. Most probably, the method will read until the file ends. Here is an example:

```
#!/usr/bin/Python3

# Opening the file

p = open("friends.txt", "r+")

txt = p.read(10)

print ("The method read the string : ", txt)

# Closing the opened file

p.close()
```

The code returns the following after execution:

```
The method read the string :  nicholas j
```

Note that we had instructed the method to read only 10 bytes from the file, and that is why not all the file contents were read.

File Positions

You may use the *tell()* method to tell the current position in a file. This tells where the next read or write will start from the next time you attempt to do so on the file.

If you need to change this position, you can use the *seek(offset[, from])* method. The argument is that *offset* specifies the number of bytes that should be moved. The argument *from* specifies the reference position from which bytes should be moved.

If the value of *from* is 0, then the reference position is the starting point of the file. If you set it to 1, then the current position will be used as a reference position. If it is set to 2, the file’s end will be used as a reference position.

Example

```
#!/usr/bin/Python3
```

```
# Openig the file
p = open("friends.txt", "r+")
str = p.read(10)
print ("The function read the string : ", str)
# Checking the current position
pos = p.tell()
print ("The current position for the file is : ", pos)
# Reposition the pointer to the beginning
pos = p.seek(0, 0)
str = p.read(10)
print ("The read String again is : ", str)
# Close the opened file
p.close()
```

Run the code from the directory you have stored the file *names.txt* in. In my case, it returns the following:

```
The function read the string :  nicholas j
The current position for the file is :  10
The read String again is :  nicholas j
```

Notice that in both cases, the same string was read. We first read the first 10 bytes of the file. This moved the position in the file to 10, as shown in the output. We then called the *seek()* function to reset the position of the file to the beginning. When we issue the read command, it again reads from the beginning. Hence we get the same output.

Renaming Files

The *rename()* method helps in renaming file names, and it takes two arguments. The method takes two arguments: the first one is the current name of the file, and the second one is the new name to be given to the file. Note that this method is provided by a Python module, **os** . For you to use

the function, you must first import the module. The method has the following syntax:

os.rename(current_filename, new_filename)

Example

#!/usr/bin/Python3

import os

Rename the file from friends.txt to colleagues.txt

os.rename("friends.txt", "colleagues.txt")

We began by importing the **os** module via the ‘import’ keyword. It is after that we have called the *rename()* method. Note the syntax used for calling the method. We began by the module name (i.e. os) then the method name (i.e. rename()).

Two arguments were passed to the method. The first one is the name of the file we need to rename, which is *friendss.txt* . We have then defined the new name we need to give to the file, that is, *colleagues.txt* . That is how files should be renamed in Python.

Deleting Files

The *remove()* method can be used for the deletion of a file. The method is called, and the name of the file to be deleted or removed is passed as the argument.

Again, this method is provided in the **os** module. Hence, you must first import the module before using the method. The syntax for the method is as follows:

os.remove(file_name)

Example:

#!/usr/bin/Python3

import os

Deleting the file named colleagues.txt

os.remove("colleagues.txt")

In the above case, we first imported the *os* module into the script. We have then called the *remove()* method from this module. Again, we used the same syntax to call the method, as we did in our previous example. The name of the file to be deleted is *colleagues.txt* . Hence this has been passed as the argument to the function.


class credit_card
function clean_no ($cc_no)
// Remove non-numeric characters
return ereg_replace
function identify ($cc_no)
$cc_no = credit_card::clean_no
// Get card type based on prefix
return 'Visa';
return 'Mastercard';
return 'American Express';

CHAPTER 7:

Testing Your Code

In Python programming language, several technique methods are used to test a particular kind of code. Let us venture into the actual methods that are normally involved.

Automated versus Manual Testing

Exploratory testing is a type of manual testing that is committed without a specific plan, where a developer is just trying to explore the actual application. In the bid to complete certain amounts of manual tests, a developer is obligated to draft a list of all the available features that a certain application contains, the various kinds of inputs that are normally accepted, and the possible expected results. With this, the developer is expected to go back to his/her list whenever changes are made to a particular set of programs—an activity that is much tiresome and unsatisfying. It is at this point where automated testing comes in.

Automated testing is the execution of the set program of code by a script in place of a developer. Python language comes in handy as a set of tools and libraries that aids a particular developer in enabling automated testing in its application.

Unit tests versus Integration Tests

A unit test is a smaller test that checks whether a single component in a particular program works the right way to make it functional. An integration test is a type of testing that ensures all the components that are involved in a particular program work well with each other. Both unit tests and integration tests can be written in a specific program.

Test runners are tools that pick up the source code directory of a particular program (which contains unit tests and various settings), get to execute them, and eventually output the results to the log files/console.

Let us look at some of the common test runners that are used by programmers.

Pytest

This particular test runner supports the execution of various unit cases. A good example of the great characteristics used in a Pytest test runner.

Unittest

Unittest is available in the Python standard library since the 2.1 version of Python, and can probably be seen in commercial Python applications and various open-source projects. It contains both a test framework and a test runner. To add it, a few requirements are called for when using unittest during testing of your particular program. You are required to put your tests into classes as methods, and you are also expected to use special assertion methods in the particular unittest; for instance, using test case class in place of the built-in the assert statement.

For this particular operation to work, one is required to import unittest from the particular standard library; then create a class, called test sum, that gets to inherit from the test case class; then change the test functions into methods, by including ‘self’ as part of the first argument; then change the assertions to use the ‘self asset equal’ method on the particular test case class; and, eventually change the entry point of the command line to call a unittest main.

Nose or Nose 2

If you are a novice in using test runners, it is highly recommended that you commence with Nose 2 instead of the Nose when running various kinds of your programs. Nose and Nose 2 are compatible with multiple unittest frameworks and can be used in place of the unittest test runner for its functions okay. To commence with the Nose 2 test runner, install the Nose 2 test runner from the Python Package Index (pypl) and execute it from the command line. The test runner will try to discover all the test scripts named ‘test py’ and the various test cases that are inherited from the unittest.

Python Data Types

Each incentive in Python has a data type. Since everything is an item in Python programming, information types are classes, and factors are occasion (object) of these classes.

There are different information types in Python. A portion of the significant kinds of records beneath.

Python Numbers

Whole numbers, skimming point numbers, and complex numbers fall under Python numbers classification. They are characterized as **int, float,** and **complex** class in Python.

We can utilize the **type()** capacity to realize which class, variable, or worth has a place. With the **instance()** ability we can check if an item has an area with a specific type.

Script by

```
1. a = 5
2. Print (a, "is of type", type (a))
3. a = 2.0
4. Print (a, "is of type", type (a))
5. a = 1+2j
6. Print (a, "is complex number?" isinstance(1+2j,complex))
```

Whole numbers can be of any length; they are just constrained by the memory accessible. A gliding point number is exact up to 15 decimal spots. Whole number and drifting focuses are isolated by decimal focuses. **1** is a whole number, **1.0** is a skimming point number. Complex numbers are written in the structure, **x + yj** , where **x** is the genuine part and **y** is the non-existent part. Here are a few models.

```
1. >>> a = 1234567890123456789
2. >>> a
3. 1234567890123456789
4. >>> b = 0.1234567890123456789
5. >>> b
6. 0.12345678901234568
7. >>> c = 1+2j
8. >>> c
```

```
9.          (1+2j)
```

Notice that the float variable *b* got shortened.

Python List

A rundown is an arranged succession of things. It is one of the most utilized datatypes in Python and is entirely adaptable. Every one of the items in a rundown shouldn't be of a similar sort.

Proclaiming a rundown is entirely straight forward. Items isolated by commas are encased inside sections [ ].

```
1.          >>> a = [1, 2.2, 'Python']
```

We can utilize the cutting administrator [ ] to extricate a thing or a scope of things from a rundown. The file begins structure 0 in Python.

Script by

```
1. a = [5,10,15,20,25,30,35,40]
2. # a[2] = 15
3. print("a[2] = ", a[2])
4. # a[0:3] = [5, 10, 15]
5. print("a[0:3] = ", a[0:3])
6. # a[5:] = [30, 35, 40]
7.
   print("a[5:] = ", a[5:])
```

Records are changeable, which means, estimation of components of a rundown can be modified.

```
1.          >>> a = [1,2,3]
2.          >>> a[2]=4
3.          >>> a
4.          [1, 2, 4]
```

Python Tuple

A tuple is an arranged grouping of things, just like a list. The just contrast is that tuples are unchanging. Tuples once made can't be changed.

Tuples are used to ensure information and are generally quicker than list as they cannot change powerfully.
It characterizes inside enclosures () where commas isolate things.

1. **>>> t = (5,'program', 1+3j)**

We can utilize the cutting administrator [] to remove things; however, we can't change its worth.

```
t = (5,'program', 1+3j)
# t[1] = 'program'
print("t[1] = ", t[1])
# t[0:3] = (5, 'program', (1+3j))
print("t[0:3] = ", t[0:3])

# Generates error

# Tuples are immutable
t[0] = 10
```

Python Strings

The string is a succession of Unicode characters. We can utilize single statements or twofold statements to speak to strings. Multi-line strings can indicate utilizing triple statements, ''' or """.

1. >>> s = "This is a string"
2. >>> s = **'''a multiline**

Like rundown and tuple, cutting administrator [ ] can be utilized with string. Strings are permanent.
Script by

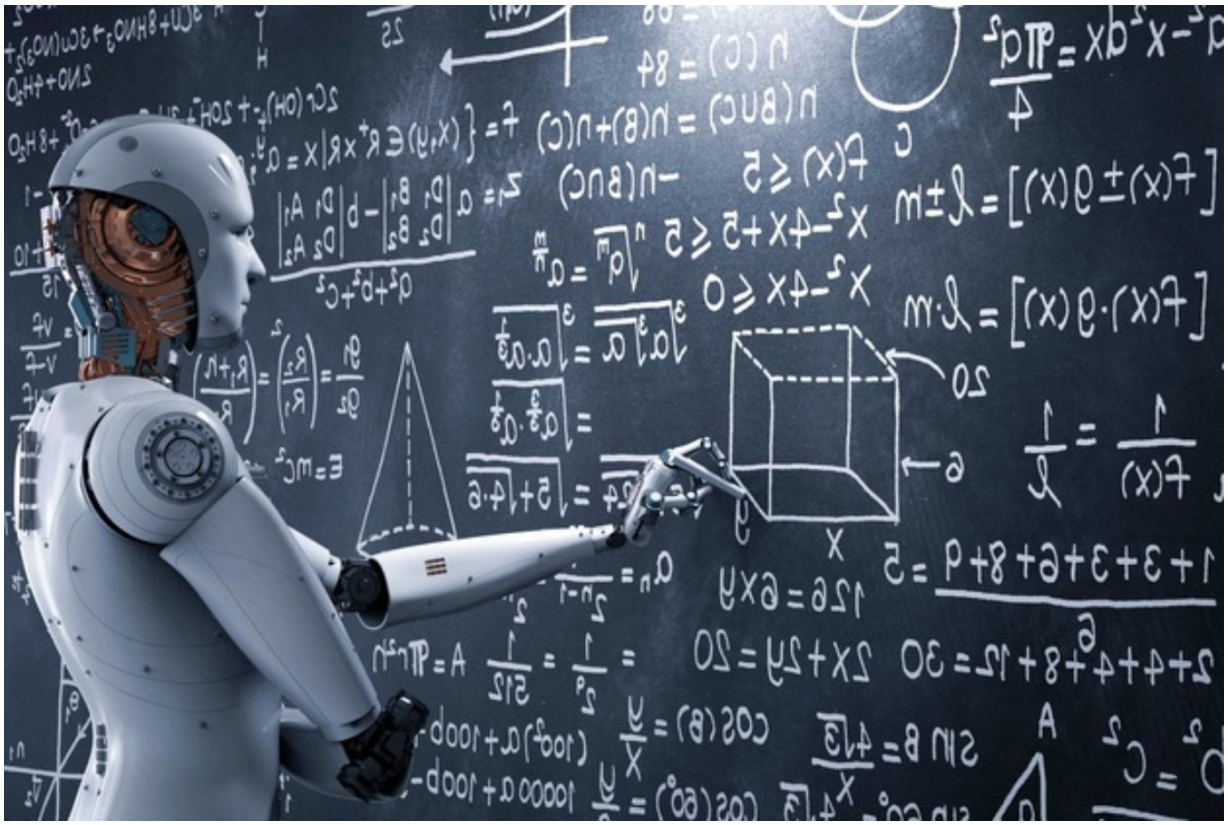
```
s = 'Hello world!'
# s[4] = 'o'
print("s[4] = ", s[4])
# s[6:11] = 'world'
```

```
print("s[6:11] = ", s[6:11])
# Generates error
# Strings are immutable in Python
s[5] ='d'
```

We can perform set tasks like association and crossing points on two sets. Sets have exceptional qualities. They wipe out copies.

```
>>> a = { 1 , 2 , 2 , 3 , 3 , 3 }
>>> a
{ 1 , 2 , 3 }
```

Since, sets are unordered assortment, ordering has no importance. Consequently, the cutting administrator [] doesn't work.



CHAPTER 8:

Data Science with Python and Machine Learning

There are several capacities in which your knowledge of data science will come in handy in whichever organization you are employed. Given that many businesses are increasingly implementing machine learning models in their operations and the great future prospects for adoption and integration, learning data analysis in Python will help you improve your prospects for future employment in the workforce. Below are some reasons why knowledge of data analysis in Python will make you more marketable.

- Room for Growth

The growth potential for the data analytics market is very high. At the moment, giant tech companies have made significant progress towards implementing data analysis. However, many small and medium-sized businesses are still lagging. If there is anything we can learn from recent world history, it is that where technological advancements are concerned, changes can take place in a very short time.

We can expect that in the next decade or two, the majority of the small and medium-sized businesses will have figured out a way to implement data at their behest into making important business decisions. Therefore, data science is definitely an area you want to be acquainted with. Currently, some of the top roles in data science involve big data, where you can seek employment as a data scientist, data engineer, or data analyst.

- Remuneration Packages

Immense growth potential is often followed by high remuneration packages. Data science is one of the fields where experts are earning some of the top wages in the world. This is because they are basically in charge of all the decisions that make companies as successful as they are. All decisions that are made in organizations that have embraced data have their foundation in the hands of a data expert.

- Career Specialization

Gone are the days when you would specialize in one area and focus on it all your life. The level of technology that we use today has made it important for most professionals to learn other skills to stay competitive in their fields. As we consume and use more data for decision-making in the future, it will be imperative for most professionals to have at least a basic knowledge of data analysis and how to use different tools.

Apart from that, when you are good at handling data, there is no limit to the work you can do. This prepares you for a versatile future in employment where your skills can be implemented in so many industries. This further increases your marketability in the job market because your skills will be in demand in almost all industries.

In the current job market, data analysts are highly sought after in the following capacities:

Marketing Analysts

A marketing analyst is an important part of many marketing teams, especially those that have an active digital marketing department. Each time customers are online, they make important decisions concerning their purchase processes. All such decisions are based on data and it is important that the company understands this data and what to do with it.

The role of a marketing analyst varies depending on the organization’s needs. However, the consensus is that such analysts will be involved in mapping the customer demographics, determining how long marketing campaigns should last, and any other initiatives that can help the company move forward.

Sales Analysts

All businesses exist to make a profit. Therefore, sales are one of their ultimate goals. The more sales you make, the better chances you have at realizing significant profits. Sales analysts usually study the sales strategies for their businesses and based on data available to them, can advise the relevant departmental heads on how to proceed.

Doing business is no longer just about making sales. It is also about optimizing sales such that the company can earn more without spending more in the process. Customers need value from a product to commit to a purchase and stay loyal to the brand. Your role is to find out how to offer customers the desired value while advancing the goals and objectives of the company and at the same time, to ensure you meet the sales targets.

Operational Analysts

Different activities take place in the business on a daily basis. An operations analyst must study the data available from these operations and recommend the best possible method of optimizing processes and workflow in the business. The result here is improved business processes that should eventually be felt at the close of the financial year, when the accounting books look amazing.

Security Analysts

A security analyst is tasked with protecting the ecosystem wherein they are employed. As most businesses are moving their operations to cloud platforms, hackers and nifty competitors try to find ways of breaching these networks to gain an unfair advantage. A security analyst must use data to find loopholes in the security system and protect the business from vulnerability. They will also monitor trends in the security environment to highlight and flag possible threats that the business might be exposed to, and recommend stealthy methods of shoring the security systems.

Financial Analysts

Financial analysts are highly sought after in the financial world at the moment. From hedge funds to banks, everyone needs the best. Money markets require astute decision-makers because one wrong move could cost the company a fortune.

Businesses are constantly looking for ways to make more profit while spending less on overheads. One of your roles as a financial analyst will be

advising other department heads on how they can scale down on their spending while achieving the desired results.

In investment markets, a skill that is necessary for you to make a name for yourself is the ability to identify trends, ride them, and exit from at the right time. Predicting trends helps you advise the company and customers on the best way to invest their wealth and how to maximize returns on their investments in different portfolios.

Unique Environment

Most job descriptions are monotonous. You get bored doing the same thing every day until you move to a different department or leave the company. This is not the case in data handling. You will probably encounter different types of data all the time. This means that you have to keep rising to the challenge and implement better ways of providing solutions to problems presented to you. If you are the kind of person who enjoys challenges, this is something you would enjoy. From strategies and planning to coming up with new ideas, there is always something new for you to work on and help you get closer to your objectives.

Pandas and Others

The next library in Python that you want to work with, to make machine learning and data science do what you would like is Pandas. Pandas stand for the Python Data Analysis Library, which helps us to do a lot of the work that is needed in the Python world. This is an open-sourced tool that helps us with some of the data structures that are needed to do data analysis. You can use this library to add in the right tools and data structures to make sure your data analysis is complete. Many industries like to work with this one to help out with some different processes like finance, statistics, engineering, and social science.

The Pandas library is adaptable, which makes it great for getting a ton of work done in less time. It can also help you work with any kind of data that you can bring in, no matter what kind of source you are getting that information from, thereby making it a lot easier to work with. This library comes with many different features that you can enjoy and some of the best ones include:

1. You can use the Pandas library to help reshape the structures of your data.

2. You can use the Pandas library to label series as well as tabular data, to help us see an automatic alignment.

3. You can use the Pandas library to help with heterogeneous indexing of the information. It is also useful when it comes to systematic labeling of the data as well.

4. You can use this library because it can hold onto the capabilities of identifying and then fixing any of the data that is missing.

5. This library provides us with the ability to load and then save data from more than one format.

6. You can easily take some of the data structures that come out of Python and NumPy and convert them into the objects that you need to Pandas objects.

Setting Up Your Virtual Environments For Data Science

In machine learning with Python, you will have to build predictive models from time to time. How do you go about this and deliver the best outcome? Success with predictive modeling depends on how much time you can invest in the initial stages. From generating the hypothesis to brainstorming sessions with your team, you must invest a lot of resources in quality. Why is this an important prerequisite for predictive modeling?

First, you need to ensure you have sufficient resources to handle the task ahead. This is not just limited to financial resources, but includes time and more importantly the necessary computing equipment and software. Remember that machine and deep learning frameworks are resource-intensive, and without matching your needs to the resource allocation, you will probably fail.

Second, you must be critical about the process so that you eliminate any bias before you begin. Biased data points will affect the output when you eventually build and deploy the data model.

Finally, creating time to oversee all the processes necessary at the beginning is important because it saves you a lot of time that you would otherwise have to rush through, as the model nears completion. If possible, you should

try to complete your projects ahead of time and use the additional time to evaluate, test, and ensure it runs properly.

Generally, there are four important stages in predictive data modeling. Descriptive data analysis and data treatment should take up at least 80% of your time allocation. Data modeling and performance evaluation and estimation should be apportioned accordingly in the remaining time.

- Descriptive Data Analysis

Data exploration is often a time-consuming process. There are so many advanced machine learning tools that you can use to help you claw back on time. In this stage, most of the work you do will involve identifying the input and output target features, looking for columns missing data values, and identifying numeric and categorical distinctions between input data.

- Treating Data

Data treatment is one of the most important stages in building predictive models. This is important in that without the right data, your model will hardly deliver the output you need. Most databases contain data that needs serious cleaning before they can be used in predictive models.

- Data Modeling

In data modeling, you will choose the right algorithm models to help you find the right solution to your problems. Random forest methods are common in many model-building scenarios. However, you should not limit your options to this. Remember that the ideal option will depend on the type of data you are working on, and the problem for which you need a solution.

- Performance Evaluation And Estimation

There are many ways you can evaluate the performance of your models. At your level, it is wise to use the simplest method, because it is easy to understand and can be implemented in many projects that you will come across. Apportion your data into a 70:30 ratio for training and validation, respectively. After training, you will build the model on the 70%-data set. Use the validation data set to determine how well the model will perform against different types of data.

Numpy Arrays

This is an array of structures similar to what we see in C language structure. We also see that these arrays are homogenous, which means that they are only going to be able to contain data that is the same type. So instead of creating an array that is an integer or a floating number, we can go through and create one of these arrays with other homogenous structures as well.

The best way to see how this is going to work is to take a look at an example. Let's say that we are going through some work, and we want to be able to create our own NumPy array with elements of the following structure:

```
struct
{
char name[10];
float marks;
int grade level;
}
```

What this means is that each element that is going to show up in the array that we are trying to make is going to be of the above type of structure. This one is called a structured NumPy array, in case you need to use this later on.


INITIALIZATION
SD 009999 00992 99003
SD 11 0009 002
ARD 23 00000 00000 99022
SD 229905 00000 856
SD 00000 77772 999201
ARD 000 82990000 9910 33
333 000 550 223
992 000 216 000
002 000 821 363
244 111 993 831
000 328 002 781
000 002 555 781
000 920 209 999
090 011 882 002
002 221 227 992
000 345 662 993
000 932 729 333
001 001 522 883
/// fixing /// password XXXX XXX XXXX XXXXX
/// done /// code process
/// password denied

CHAPTER 9:

Object-Oriented Programming

Python is an object-oriented programming language. In fact, most modern languages are. But exactly what does this mean? We've spoken in vague terms of objects and classes, but we haven't really established quite what this means in certain terms.

An object is an instance of a class. Most things you'll deal with in Python are objects. Earlier, when we worked with file input and output, we created instances of a file class. Every instance has built-in methods that it can access, which are derived from the class definition itself. So, what exactly is a class?

A class is a way of defining objects. This sounds terribly vague, but let's look at it this way. You likely have or have had a pet, right? Let's say there's a dog, and his name is Roscoe.

Well, Roscoe is an animal. Animals have broad, generally defined characteristics, but they're all animals, as is Roscoe. Get comfy with Roscoe, because we're going to be talking about him a lot while discussing the relations between classes and the relations between classes and objects.

We've established that Roscoe is most certainly an animal. He fits the definition of an animal. In this manner, Roscoe is a specific instance of the animal class. If you were writing a simulation of life, and you had people and animals, you would define Roscoe as an instance of animal, just as you declared variable file1 as an instance of a file, or you declared tongue twister as an instance of a string.

Now, we need to talk about how we define a class and an object within Python.

Create a new file to work with. I'm calling mine *pursuitOfRoscoe.py.*

Within this file, we're going to start right out the bat by defining a class.

To declare a class, follow the following template:

class name(parent)

Let's just make our animal class. Every class which isn't deriving from another class has an 'object' as its parent, so let's put that.

class Animal(object):

We're on our way to defining Roscoe, now. We need a way to define an animal. Let's think about what most animals have. Most animals have legs, that's a start. Animals also have Latin names. Let's work with those two. If your class stores data, you generally need to have an initializer function within your class. It's not a necessity, but it is a very common practice.

Perfect. Since Roscoe's a dog, he'll have four legs, and his species is *Canis Lupus Familiaris* .

With that in mind, we now have a definition for animal classes that can be used amongst many animals, not just Roscoe. That is the entire idea behind classes: creating reusable data structures for any given object so that the code is more readable, easy to understand, cleaner, and portable, among other buzzword adjectives that are surprisingly very, very true.

How do we declare an instance of this class now? Like anything else!

We can go in and change these variables too. 'Canis lupus' is so formal, and Roscoe's our buddy, so let's change that to 'Roscoe.'

There we go. Much better.

Hopefully, this makes the distinction between classes and objects much clearer.

Roscoe is a dog and an animal. Thus, he takes from the common concept of being an animal. Since he's an instance of an animal, he automatically receives the traits that all animals have. How cool is that?

Let's go a bit further, and incorporate some functions. What's something that every animal does? Sleep. Every single animal sleeps, aside from Ozzy Osborne.

Let’s give ‘animals’ a function so that it is defined that they can sleep.

Below our initializer, create a new function called ‘sleep’ that takes the arguments of self and hours. Then print out a line of text that says the animal’s name and how long it’s sleeping for. My code ended up looking a bit like this, and hopefully, yours will as well.

Save this and run it. If all goes well, it should print out “Roscoe is sleeping for 4 hours!.”

More on Object-Oriented Programming and Classes

There are four primary concepts within object-oriented programming that we need to discuss in-depth. These are inheritance, polymorphism, abstraction, and encapsulation. Python provides for all of these, and very well at that.

Inheritance is the notion of deriving a class and things from within that class into another child class. There’s a very simple way to explain this concept. Classes can break down into other more specific classes. For example, Roscoe is an animal. But he’s also a dog. A dog is a type of animal. Shouldn’t Roscoe be a dog and not an animal? Isn’t he both? How do we handle this?

Think of it this way: every dog is an animal, but not every animal is a dog. So we can break down the animal class even further. The way that we derive one class from another is by inheritance. Here’s how we’d declare a Dog class that extends the animal class. All dogs have four legs (for the most part), so we can declare that ahead of time and manually change it if a dog ever doesn’t have four legs.

class Dog(Animal):

def __init__(self, name):

self.name = name

self.legs = 4

The way that this works is that the Dog class is an extension of the Animal class. The Dog class receives all the functions and variables of the dog class, so we don’t have to redefine them.

This also means that if we were to erase our first line and re-declare Roscoe more accurately as a dog, we could still declare it to sleep. Observe:

roscoe = Dog(“Roscoe”)

roscoe.sleep(4)

It should go without a hitch. However, the cool thing about child classes is that you can also give them their own functions that their parent can’t use. For example, most animals don’t bark - dogs do. Let’s create a bark function in our dog class for practice’s sake.

def bark(self):

“%s says: Bark!” % self.name

Now let’s try to declare bark via Roscoe.

roscoe.bark()

It should print out exactly what we entered. To illustrate further, create an instance of the parent class Animal. Let’s call it “lion”:

lion = Animal(4, “panthera leo”)

Try to call the method bark by way of Lion.

lion.bark()

There should be an error. Why is this? Well, it’s because - as we said - every dog is an animal, but not every animal is a dog. The bark() function was defined in the Dog class but not in the Animal class, so instances of the Animal class can’t access this method at all.

The next concept of object-oriented programming is called **polymorphism** . This means that something has the property of being able to perform the same task as something else but differently. There are two ways of achieving this: **function overloading** (performing a similar function/method but with different parameters) and **function overriding** (rewriting a function of a parent class so that it works better for your own class).

To illustrate this, let’s go back to our bark method. Under our bark method, we’re going to create another bark method, declared like this:

def bark(number):

print “%s just barked, %d times! How cute.” % (self.name, number)

Now, we have two different forms of the bark function. If you declare ‘roscoe.bark()’, you’re going to see, ‘Roscoe says: Bark!’

However, if you declare ‘roscoe.bark(3)’, then you’ll see ‘Roscoe just barked, 3 times! How cute.’

This is the basic idea of function overloading and polymorphism in essence: giving multiple ways to do a similar thing.

This program is already adorable, but we can make it even more adorable while also learning more about Python coding and string manipulation. Go back to your bark(number) method, and change it so it looks like this:

barkString = “Bark! “ * number

print “%s just barked, %d times. How cute. %s) (self.name, number, barkString)

Now save it and run it. You can repeat a string multiple times by simply using the multiplication sign and inputting how many times to multiply!

The next major concept of object-oriented programming languages is **abstraction** .

This is the idea of hiding internal details and functionality: to be more forward and safer for both the programmer and end-user.

Python shows this by having a very abstract interface compared to other languages, and providing a large amount of functionality for you so you never have to get down to the nitty-gritty of what your computer is doing behind the scenes.

The last major concept of object-oriented languages is called encapsulation, wherein code and data are wrapped together into a single unit.

The primary way that we can display this is by the notion of having a class not only in Python but everywhere. Using a class automatically wraps important data and functions together in one easily accessible and usable place.

Other data have something called access control, by which you can dictate which classes can and can’t access the data that you’re putting in your class. Class data in Python is public by default.

All in all, object-oriented programming isn’t very tough to grasp, but it’s full of concepts that stand for much bigger and larger things, and these are the concepts that can be difficult to understand and implement in the end.

CHAPTER 10:

Web Applications

A web application runs on a remote server as a software application. Most of the time, web browsers are for web applications, like the internet.

Some of the applications are used for intranets, schools, firms, and organizations. They are not the same as other applications since you do not need to install them. Some of the common web applications include Flickr, Wikipedia, Facebook, and Mibbit. They are popular since most of the operating systems are on the web browser and programmers can change them with ease.

Several benefits come with using web applications:

- They do not need to be installed since they run inside a browser.
- They do not require a lot of space for storage, only a display of data.
- It helps in dealing with compatibility problems; all that is needed is a browser.
- Much of the data used is remotely stored. Hence, there is the ease of cooperation and communication.
- A web application helps in mail and communication.

Apart from the listed benefits of web applications, there are also drawbacks.

Most of the known web applications will seem to look different as compared to the regular programs. The reason is that they run inside a browser. The user experience might be different and not liked by many.

To be able to follow standards, web applications need to be coded, and any small changes will prevent the web application to be used in any browser.

There is a need to have a connection between the web application and the server for it to run smoothly. For the connection to happen, you will need bandwidth. When the connection is not adequate, you may experience data loss or the application will be usable.

Most of the web applications depend on the server that hosts them. When it is off, the web application is not usable, but the traditional applications will still work.

The overall control of the web application is with the mother company. They have the power to create a new version when they feel like it.

When the data is remotely stored, exporting it to be used by other applications will be hard.

Web applications enable the company to track all the activities of the users, hence privacy issues.

At this point, you need to know how a web application works. Most of the web applications are coded in a language that is browser supported, like HTML or JavaScript. And the main reason is that the languages depend on the browser to execute their programs. You should know that some of these applications are dynamic, and they will require server-side processing. Others are considered static and will not need any processing from the server.

When you have a web application, you will need a webserver to manage all the requests that the client has. The server will help in performing all the tasks and store data and information. The application server includes ASP, PHP, and JSP. A normal web application has a specific flow:

> The user will trigger a request using the internet that goes to the webserver. This can be done through the web browser or user interface on the application.
>
> The web server will then forward that request to the specific web application server.
>
> The requested task will be performed by the web application server. This includes querying the database or data processing that will

generate the required results.

The results will be sent to the web server by the web application server. This will be in regard to the data processed or the required information.

The client will get a response from the webserver. They will get the information that they have requested, and it will appear on the user's display.

There are several examples of web applications such as shopping carts, word processors, online forms, file conversions, and scanning, online forms, and email programs like Yahoo and Gmail.

How to Work with Django

Django is used to create web applications. It is specifically meant to create a web application that connects to a database. You can also deal with user management, good security, and internationalization. Some of the common web applications include Disqus, Pinterest, and Instagram. You can use Django as standalone libraries even though it will require extra work. That is the reason why it is not advisable to use it as a standalone.

Django is a combination of different components that work by responding to user requests.

The first step is passing the request-or-response system. The main work is to receive and return web responses. Django will accept all the requests of the URLs and return all the HTML information to the web browser. The page can be in plain text or something better.

The web requests will enter the Django application through the URLs. The only entry point for any Django application is the URL. Developers have the control of the available URLs. When you access the URL, Django will enable the viewing.

All your requests will be processed by the views. Django views are considered to be codes generated from Python when the URL is accessed. Views are something simple like returning a text to the user. The text can be made complex. It can be form processing, credit card processing, and database querying. When the view has completed processing, a web response is sent to the user.

When web response is returned, the user can access the URL on the browser they will access the response. This could be an HTML web page that shows a combination of images and text, and they are created using the templating system from Django.

With Django information, there is flexibility to have more applications. You can use that you create a simple blog, mobile applications, or a desktop. Django framework is powered by sites like Instagram and Pinterest.

User Accounts

A user account is on the network server that is used to store the username of the computer, password, and any relevant information. When you have the user account, it will allow you or not to connect with other computers or networks. With a network with multiple users, you will need user accounts. A good example of a user account is your email account.

There are different types of user accounts, regardless of the operating system that you are using. You will be able to trace, authenticate, and monitor all the services. When you install an operating system, it creates user accounts to have access after the installation. After the installation, you will have four user accounts: a system account, a super user account, a regular account, and a guest user account.

System account : These are accounts that are used to access resources in the system. The operating system will use these accounts to know if a service is allowed to access the resources or not. When they are installed, they create relevant accounts. After installation, the account will be able to access the needed information. If you are a network or system administrator, you will not need to have any information about the accounts.

Super user account : This account is privileged in the operating system. When one is using Windows, the account is referred to as the administrator account. When using Linux, the account is the root account, and the operating system will help the user to complete different tasks. Tasks are like starting services, creating and deleting new user accounts, installing new software, and changing system files.

Regular user account : This account does not have many privileges and cannot make changes in the system properties and files. They only operate

on tasks that they are authorized like running applications, creating files, and customizing variables.

The Guest user account : This is the account that has less privilege: you will not be able to change anything in the system. The account is known to perform temporary tasks like playing games, watching movies, or browsing the internet. Using Windows, this account will be created after installation. In Linux, you will need to create the account manually after installation.

The next step is to know how to create a user account. When you have multiple users using the same computer, you will need to have new user accounts for each user. When using Windows, you can create several accounts. Each of the user accounts has its own settings. It will allow you to control the files separately, and when each user logs in, it will be like their own computer.

The first step in creating a user account is to click on **Start** on the **Control Panel** then click on **Add Or Remove User Accounts** . Click on **Create A New Account** and choose the account type. You will enter the account name and then select the account type that you wish to create. The administrator has the privilege to create and change accounts and installing programs. The difference is a standard user cannot perform such tasks. The last step will be to click on the **Create Account** button and close the **Control Panel** .

How to Style and Deploy an Application

There are different deployment options that need to be considered. When an application is developed in the application builder, it is created in the workplace. All workplaces have IDs and names; all you need to do is to create an application in the development and then deploy it in production.

During deployment, you will decide where you want the existing ID to be in the workplace, the existing HTTP server, or in new ones. The deployment options include the following considerations.

You will first create an application that is expressed by end-users. The best way to deploy an application is by creating an Application Express for end-users. Then, send the URL and login details to the users. It will work when the user population is tolerant and small.

You will need to use the same schema and workplace. You need to export and then import the application, and then install that under a different application ID. This strategy will work when there are fewer changes to any known objects.

Use the same schema and a different workplace, export all, and then import the applications into another workplace. It will prevent any production and modification by developers.

Use a different schema and workspace. Export and then import the application into a separate workplace, and install it in a separate schema.

Use a different database for all variations. Export, then import to another oracle application. Then install it to a different database and schema.

To deploy an application, in the configuration manager console, click on **Software Library** . Go to **Application Management** and then choose **Application** or **Application Group** .

Choose from an application or application group in the deploy list, and click **Deploy** .


1001101001
PYTHON

CHAPTER 11:

Tips and Tricks to Get the Most Out of Python

Print Things Out

When there are times when you are not able to figure out what is wrong with the code you worked on, then you can take some time to print out things to see what is going to work. If you are unsure about which values are ever attached to one of your variables, then print it out. if you are uncertain about anything that is going on in the code that you are writing, then it is time to print it out. Then, when you are actually going through and running the program, you are able to look at your screen and see how the values change or what has been a bit different, thanks to that list.

You may find that it is helpful to print out some strings before you go through and print out the variable. This is a good way to make sure that your print statements are going to help you learn how to make the program work for your needs.

Work with the Code You Know Will Behave

When you are in doubt about things, you may want to take some time to practice codes that are in existence already, ones that you know are going to work and about which you don't have to worry if they are going to do what you would like or not. This is going to help you to figure out some of the syntax and the structure that has to come with these codes, and then you can work from there to get the best results.

Read All of the Error Messages

This one is sometimes hard as a beginner because you may assume that you will have no idea what the error messages mean in the first place, so why should you take the time to learn them and actually read them? But, actually

taking your time to read through them and learn how they work and what they are telling you, can make it easier to fix some of your code when things are not working well.

You will quickly find that a lot of your error messages are accurate and will actually be pretty descriptive for you. The language runtime may have tried to execute the program, but it could have run into a problem. This would mean that you skipped a step, you had a typo, or something else is missing from your code.

There may be times when you do not understand the message that comes up with an error, but it does try to tell you what went wrong with the code. At a minimum, there will be information on the line number the error is on, and you can head to that part of the code and look for where the bug might be located.

Run the Code Often

Another thing that you need to consider doing when it is time to work with the Python code is to run it on a regular basis. There is nothing worse, especially when you are a beginner, than sitting down and writing code for hours. You may feel accomplished, but when you try to run that code, you are going to be in for a surprise. There may be a ton of errors that you need to go through and fix, and then you will feel really frustrated with yourself in the process.

Instead of having hours of code and errors that you need to sort through, it is best if you try to run the code as often as possible. If you are able to go through it and run it every few minutes or after you finish each little part that you are working with, then this is going to be the best way. It will ensure that you are catching most of the mistakes and issues early on in the work, and will make it so much easier for you to really figure out what is going on in the code and fix the errors.

It is a lot easier to find the errors and the mistakes in a small amount of code than it is to try and do this in a large block of code that you worked on for hours. It may seem like a waste of time and that it is slowing you down to run the code so often. But in the long run, it is going to end up saving you a ton of time and energy, and it can ensure that you are able to find the solutions that you are looking for. It makes it so much easier to be able to

fix any of the errors that show up because you know exactly where they are going to fall along the way as well.

Take a Break When Needed

Sometimes, working on code can be a breeze. Everything fits in just as you need it to and you can spend hours working on a part, getting it to match the way that you would like it to. At other times, things are not going to run as smoothly for you and will not work out as well. When the latter starts to happen, it is time to take a break.

Often, we work on the code for far too long. We want to get it done or figure something out with it, so we will spend hours and hours working on it. Then, we get frustrated. One thing or another on the code may not be working out the way that we want, and we can't seem to get it fixed and working well again. And no matter how hard we try and how much work we put into it, we end up just making the whole situation worse in the process. While it is hard to just let go of the code that we are writing and take a break, sometimes this is just what we need. When we are tired and have been staring at something for a long time, it is just going to be worse if we continue to stare at it and try out too many things. When we take a break and learn how to walk away from what we are doing on the computer, it is going to become so much easier.

You may have to force yourself to take this break a bit. When we are frustrated, it is usually the time when we want to dig our heels into it a bit more, and we don't want to leave all of the work that we are doing at the time. But even an hour break or more can be nice to allow us to get away and rethink things. Then, when we come back, we can find the mistake or the error or the problem that we couldn't before, and we can get it taken care of.

Ask for Help

As a beginner, there may be some times when you will need help with your code. It is hard for some people to ask for help because they want to be able to do it all on their own. But learning how to code takes time and patience, and then there are going to be times when we just need some help because we aren't sure what steps to take next.

When you are asking for help, there are a few steps that you should consider taking in order to make it easier. This helps the other person know

what you have tried and he doesn't waste time going through all of that again. Some of the things that you can consider knowing and explaining to the person who is going to help you out with your code will include:

1. Explain what you want the program to do for you and where the error is occurring.
2. Show the other person the code that is sending you an error so they can see it for themselves.
3. Show the other person the stack trace, all of it, including the message you got stating the error.
4. Explain everything that you have already tried on the error. This helps the other person have a good idea of what you have tried and what they should try to get the code to work.

You may find that the process of thinking through these items in your mind. Getting them written down on paper can help make a brand-new solution obvious. If that doesn't work, try to find someone who may be able to look through the code and tell you what is going wrong, and what you can do to fix the error.

There are a lot of parts that need to come together when you are working as a programmer for the first time. Being able to get all of these to work with one another, and ensuring that you learn how to fix some of the problems that come up in your own code can be very important when you are a beginner trying to figure all of this out. Make sure to check out some of the different steps and tips that you can do when working in Python, to make sure that you are coding correctly and that your program or application is going to work for your needs.


FINAL
THOUGHTS

Conclusion

Thank you for making it to the last chapter of the book *Python Crash Course* . I hope that you found it informative and helpful. Every measure was taken into consideration to ensure that all the chapters give you detailed and easy-to-understand information. I intentionally used simple language throughout the book to make sure that you get empowered after reading it. The book has deliberately avoided sophisticated theories and stuck to simple explanations that you can use at your convenience when studying.

The moment you understand programming basics using Python language, it becomes easier to learn advanced concepts such as artificial intelligence and machine learning. Artificial intelligence is important in everyday life. This book has taken you through many concepts of Python Programming such as lists, classes, loops, objects, variables, methods, and many more. There is no one specific thing that you can do to learn object-oriented programming overnight. However, if you follow the right steps with commitment and dedication, you will get the results you desire.

Make it your routine to combine several practical sessions to improve your Python programming skills. If you are working with an experienced programmer, follow all the instructions provided to you, and ask questions when you do not understand.

The next step is to stop reading and start applying the lessons you have learned in real life. Do whatever you have identified as necessary to improve applications of programming in real life. You will realize that the majority of those who seem to have it all together lack the basic Python programming skills.