Computer Science Lecture -1,Class-XII

SPLIT-UP SYLLABUS

SUB: COMPUTERSCIENCE(083)

CLASS- XII(NEW SYLLABUS)

(SESSION 2019-20)

DISTRIBUTIONOF MARKS

UNIT	UNITNAME	MARKS
1	Programming and Computational Thinking-2	30
2	Computer Network	15
3	Data Management-2	15
4	Society, Law and Ethics-2	10
5	Practical	30
	TOTAL	**100**

MONTH-WISE DISTRIBUTION

Month	Topics to be covered	Th.	Pr.
April	**Unit1:Programming and Computational Thinking-2** • Revision of the basics of Python • Functions: scope, parameter passing, mutable/immutable properties of data objects, pass arrays to functions, return values, functions using libraries: mathematical, and string functions.	25	15
May-June	• File handling: open and close a file, read, write, and append to a file, standard input, output, and error streams, relative and absolute paths. • Using Python libraries: create and import Python libraries **MONTHLY TEST-1 (20-28 June)- Covering Syllabus up to 9th May-2019**	15	10

July	• Recursion: simple algorithms with recursion: factorial ,Fibonacci numbers; recursion on arrays: binary search • Idea of efficiency: performance defined as inversely proportional to the wall clock time, count the number of operations apiece of code is performing, and measure the time taken by a program. Example: take two different programs for the same problem, and understand how the efficient one takes less time. • Data visualization using Pyplot: line chart, pie chart and bar chart. **MONTHLY TEST-2 (26-31 July)- Covering Syllabus up to July-2019**	30	20
August	• Data-structures: lists, stacks, queues. **Unit 2: Computer Network(CN)** • Structure of a network: Types of networks: local area and wide area (web and internet),new technologies such as cloud and IoT, public vs. private cloud, wired and wireless networks; concept to client and server. • Network device such as a NIC, switch, hub, router, and access point. • Network stack: amplitude and frequency modulation, collision in wireless networks, error checking, and the notion of a MAC address, main idea of routing. IP addresses :(v4 and v6), routing table, router, DNS, and web URLs, TCP: basic idea of retransmission, and rate modulation when there is congestion (analogy to a road network), Protocols:2G,3G,4G, Wi-Fi. What makes a protocol have a higher bandwidth? • Basic network tools: trace route, ping, ipconfig, nslookup, who is, speed-test. • Application layer: HTTP(basic idea),working of email, secure communication:	10 15	10 10

	Encryption and certificates(HTTPS),network applications: remote desktop, Remote login, HTTP, FTP, SCP, SSH, POP/IMAP, SMTP, VoIP, NFC. **MONTHLY TEST-3 (27-31August)- Covering Syllabus up to August-2019**		
September	**Unit3:Data Management(DM-2)** • Write a minimal Django based web application that parses a GET and POST request, and writes the fields to a file-flat file and CSV file. • Interface Python with an SQL database SQL commands: aggregation functions–having, group by, order by. **MONTHLY TEST-4 (27 Sep-03 Oct)- Covering Syllabus up to September-2019**	18	12
October	**UNIT4:Society, Law and Ethics(SLE-2)** • Intellectual property rights, plagiarism, digital rights management, and licensing (Creative Commons, GPL and Apache), open source, open data, privacy. • Privacy laws, fraud; cyber-crime-phishing, illegal downloads, child pornography, scams; cyber forensics, ITAct,2000. • Technology and society: understanding of societal issues and cultural changes induced by technology. • E-waste management: proper disposal of used electronic gadgets.	12	6
November	• Identity theft, unique ids, and biometrics. • Gender and disability issues while teaching and using computers. • **Revision, Project Work Submission**	7	3
Dec	**Pre-Board Examination-1 (02-10 Dec-2019)**		
Jan	**Pre-Board Examination-2 (16-23 Jan-2020)** **Mock Practical Exam (Last week of January-2020)**		
Feb	**AISSCE Practical Examination -2020** **Pre-BoardExamination-3**		
Mar	**AISSCE-2020**		

Unit1:Programming and Computational Thinking-2

- An ordered set of instructions to be executed by a computer to carry out a specific task is called a **program**, and the language used to specify this set of instructions to the computer is called a **programming language**.

In this chapter

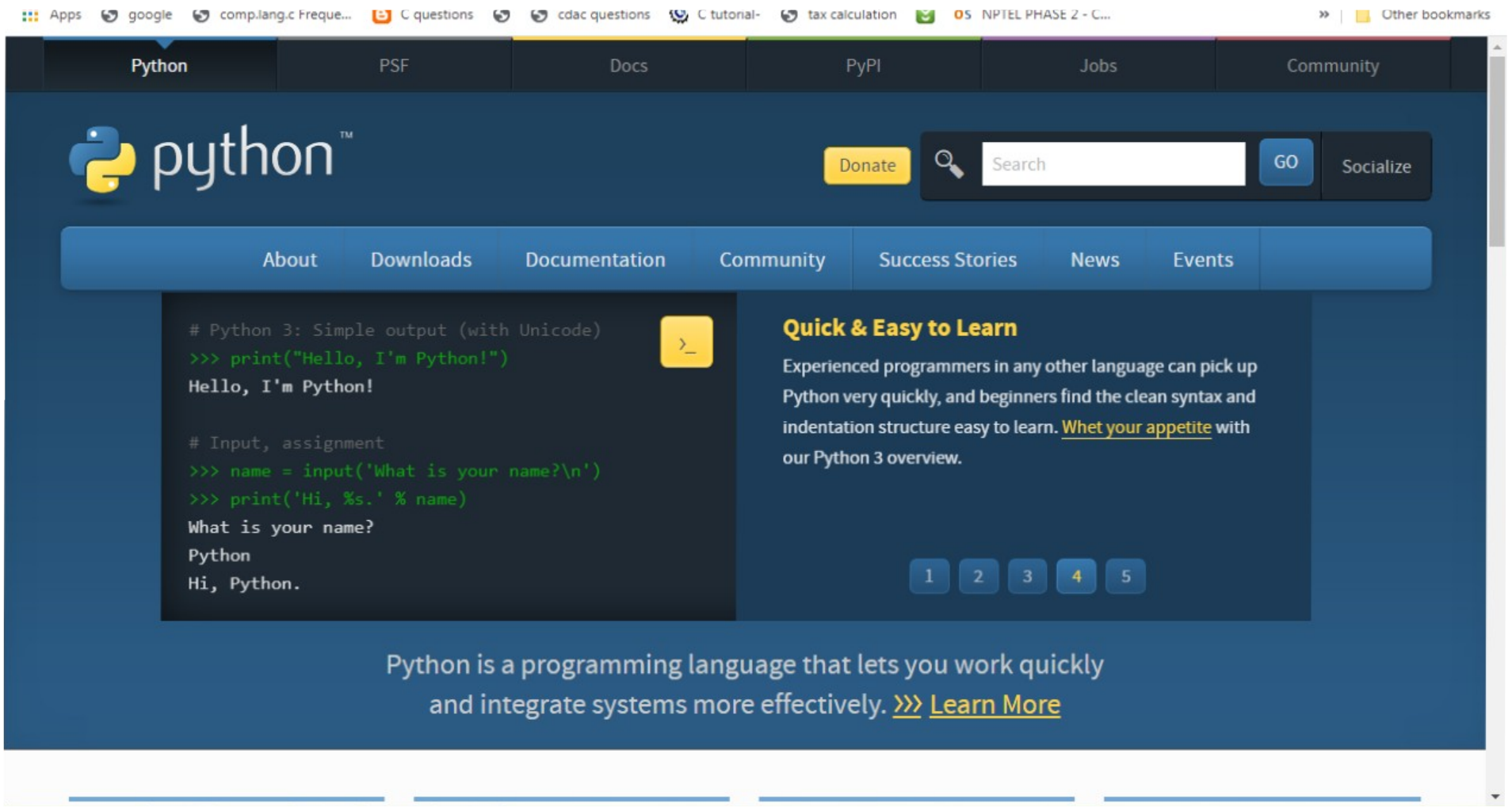
- Introduction to Python
- Python Keywords
- Identifiers
- Comments
- Data Types
- Operators
- Expressions
- Statement
- Input and Output
- Type Conversion
- Debugging

Features of Python

- Python is a high level language. It is a free and open source language.
- It is an interpreted language, as Python programs are executed by an interpreter.
- Python programs are easy to understand as they have a clearly defined syntax and relatively simple structure.
- Python is case-sensitive. For example, NUMBER and number are not same in Python.

Working with Python

- To write and run (execute) a Python program, we need to have a Python interpreter installed on our computer or we can use any online Python interpreter.
- We can download python from ***https://www.python.org/***


Apps
google
comp.lang.c Freque...
C questions
cdac questions
C tutorial-
tax calculation
OS NPTEL PHASE 2 - C...
Other bookmarks
Python
PSF
Docs
PyPI
Jobs
Community
python™
Donate
Search
GO
Socialize
About
Downloads
Documentation
Community
Success Stories
News
Events
# Python 3: Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!
# Input, assignment
>>> name = input('What is your name?\n')
>>> print('Hi, %s.' % name)
What is your name?
Python
Hi, Python.
Quick & Easy to Learn
Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. Whet your appetite with our Python 3 overview.
1
2
3
4
5
Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

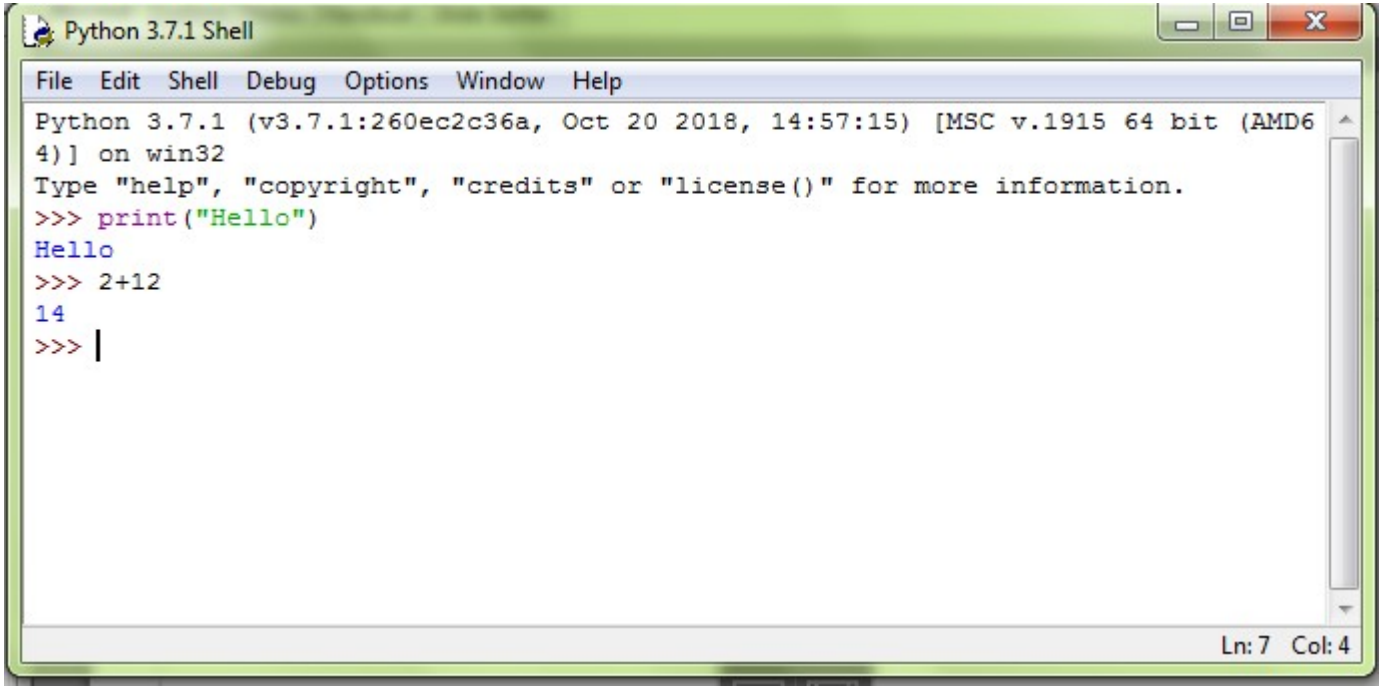
Execution Modes

There are two ways to use the Python interpreter:

- a) Interactive mode
- b) Script mode

(A) Interactive Mode

- To work in the interactive mode, we can simply type a Python statement on the >>> prompt directly. As soon as we press enter, the interpreter executes the statement and displays the result(s), as shown in Figure



(B) Script Mode

- In the script mode, we can write a Python program in a file, save it and then use the interpreter to execute it. Python scripts are saved as files where file name has extension ".py".

Python Keywords

- Keywords are reserved words. Each keyword has a specific meaning to the Python interpreter, and we can use a keyword in our program only for the purpose for which it has been defined. As Python is case sensitive, keywords must be written exactly as given in Table

Keywords in Python				
False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Identifiers

In programming languages, identifiers are names used to identify a variable, function, or other entities in a program. The rules for naming an identifier in Python are as follows:

- The name should begin with an uppercase or a
- lowercase alphabet or an underscore sign (_). This may be followed by any combination of characters a–z, A–Z, 0–9 or underscore (_). Thus, an identifier cannot start with a digit.
- It can be of any length. (However, it is preferred to keep it short and meaningful).
- It should not be a keyword or reserved word given in We cannot use special symbols like !, @, #, $, %, etc., in identifiers

Variables

A variable in a program is uniquely identified by a name (identifier). Variable in Python refers to an object — an item or element that is stored in the memory. Value of a variable can be a string (e.g., ‘b’, ‘Global Citizen’), numeric (e.g., 345) or any combination of alphanumeric characters (CD67).

Program

Write a Python program to find the area of a rectangle given that its length is 10 units and breadth is 20 units.

```
#To find the area of a rectangle
length = 10
breadth = 20
area = length * breadth
print(area)
Output:
200
```

Data Types

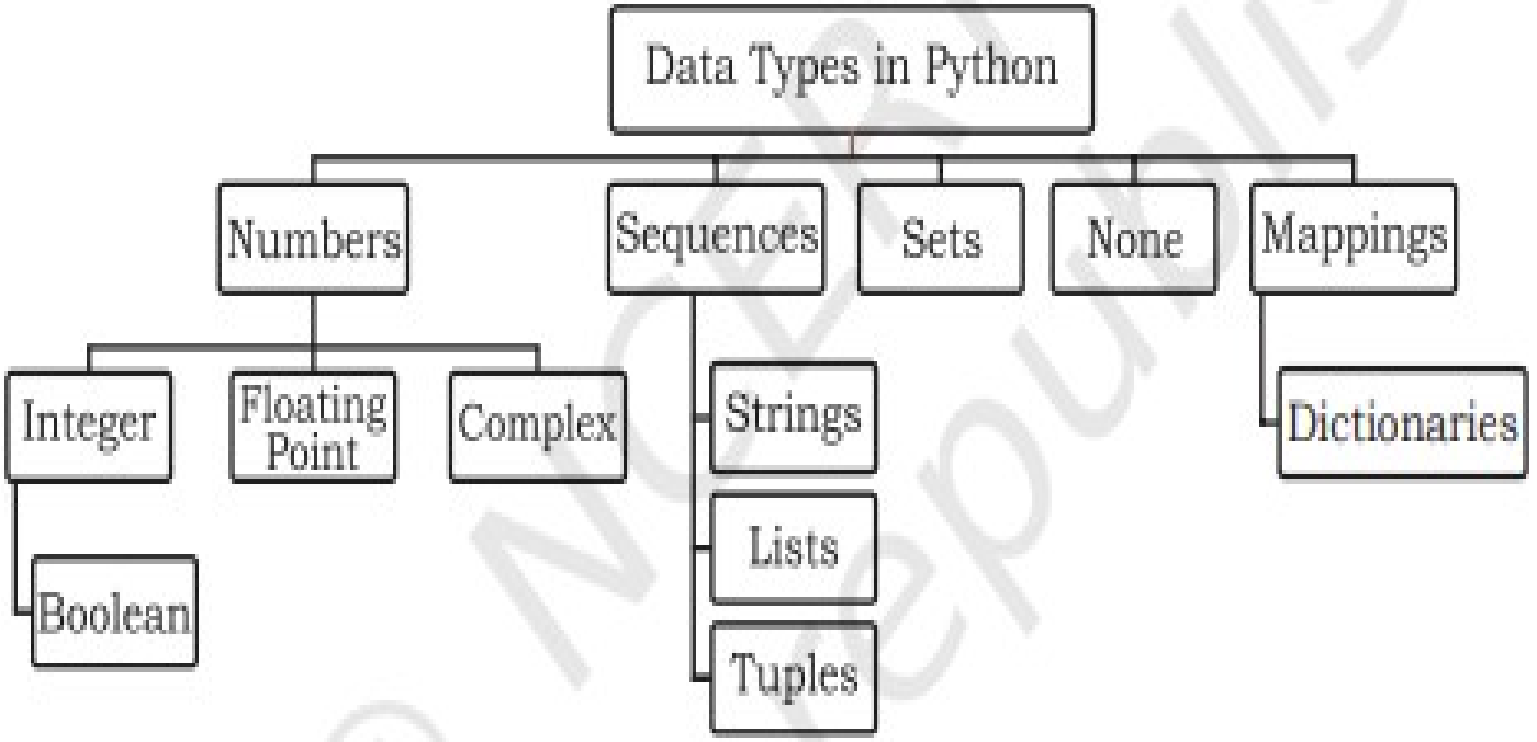


Figure 5.6: Different data types in Python

Number

Type/ Class	Description	Examples
int	integer numbers	-12, -3, 0, 125, 2
float	real or floating point numbers	-2.04, 4.0, 14.23
complex	complex numbers	3 + 4i, 2 - 2i

Number cont..

Boolean data type (bool) is a subtype of integer. It is a unique data type, consisting of two constants, True and False. Boolean True value is non-zero, non-null and non-empty. Boolean False is the value zero.

Example

```
>>> num1 = 10
>>> type(num1)
<class 'int'>

>>> num2 = -1210
>>> type(num2)
<class 'int'>

>>> var1 = True
>>> type(var1)
<class 'bool'>

>>> float1 = -1921.9
>>> type(float1)
<class 'float'>
>>> float2 = -9.8*10**2
>>> print(float2, type(float2))
-980.0000000000001 <class 'float'>

>>> var2 = -3+7.2j
>>> print(var2, type(var2))
(-3+7.2j) <class 'complex'>
```

Variables of simple data types like integers, float, boolean, etc., hold single values. But such variables are not useful to hold a long list of information, for example, names of the months in a year, names of students in a class, names and numbers in a phone book or the list of artefacts in a museum. For this, Python provides data types like tuples, lists, dictionaries and sets.

Sequence

- (A) String -String is a group of characters. These characters may be alphabets, digits or special characters including spaces. String values are enclosed either in single quotation marks (e.g., ‘Hello’) or in double quotation marks (e.g., “Hello”). The quotes are not a part of the string, they are used to mark the beginning and end of the string for the
- interpreter. For example,
- >>> str1 = 'Hello Friend'
- >>> str2 = "452"

Note:---

We cannot perform numerical operations on strings, even when the string contains a numeric value, as in str2.

(B) List

List is a sequence of items separated by commas and the items are enclosed in square brackets [ ].

Example

```
#To create a list
>>> list1 = [5, 3.4, "New Delhi", "20C", 45]
#print the elements of the list list1
>>> print(list1)
[5, 3.4, 'New Delhi', '20C', 45]
```

(C) Tuple

Tuple is a sequence of items separated by commas and items are enclosed in parenthesis ( ). This is unlike list, where values are enclosed in brackets [ ]. Once created, **we cannot change the tuple.**

```
#create a tuple tuple1
>>> tuple1 = (10, 20, "Apple", 3.4, 'a')
#print the elements of the tuple tuple1
>>> print(tuple1)
(10, 20, "Apple", 3.4, 'a')
```

Set is an unordered collection of items separated by commas and the items are enclosed in curly brackets { }. A set is similar to list, except that it cannot have duplicate entries. Once created, elements of a set cannot be changed.

Example

```
#create a set
>>> set1 = {10,20,3.14,"New Delhi"}
>>> print(type(set1))
<class 'set'>
>>> print(set1)
{10, 20, 3.14, "New Delhi"}
#duplicate elements are not included in set
>>> set2 = {1,2,1,3}
>>> print(set2)
{1, 2, 3}
```

None

- **None is a special data type** with a single value. It is used to signify the absence of value in a situation. None supports no special operations, and it is neither False nor 0 (zero).

Dictionary

Dictionary in Python holds data items in key-value pairs. Items in a dictionary are enclosed in curly brackets { }.Dictionaries permit faster access to data. Every key is separated from its value using a colon (:) sign. The key : value pairs of a dictionary can be accessed using the key. The keys are usually strings and their values can be any data type. In order to access any value in the dictionary, we have to specify its key in square brackets[ ].

```
#create a dictionary
>>> dict1 = {'Fruit':'Apple', 'Climate':'Cold', 'Price(kg)':120}
>>> print(dict1)
{'Fruit': 'Apple', 'Climate': 'Cold','Price(kg)': 120}
>>> print(dict1['Price(kg)'])
120
```

Mutable and Immutable Data Types

Python data types can be classified into mutable and immutable as shown in Figure 5.7.

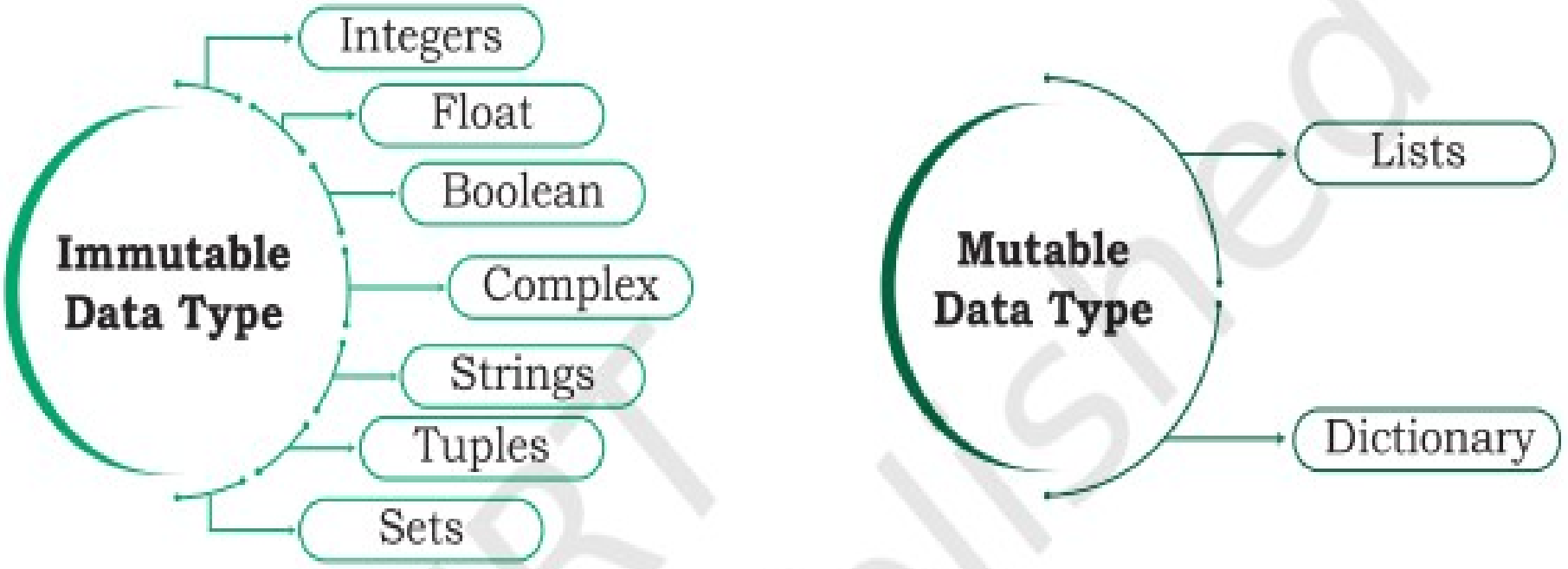


Figure 5.7: Classification of data types