Course Unit 4: Week 5

Python Syntax, Variables, Data Types

TOPICS

Python Syntax

- Display Output
- User’s Input
- Statements
- Comments
- Indentation
- Identifiers
- Reserved Words
- Quotations

Python Variables

- Declaration
- Printing
- Deleting
- Multiple Assignment
- Naming

Python Data Types

- Number
- Set
- Dictionary
- Sequence

Objectives

1. Explain python syntax, variables, data types.
2. Simulate a program using python syntax, variables, data types.
3. Construct a simple python program.

TOPICS

Python Syntax

- Display Output
- User’s Input
- Statements
- Comments
- Indentation
- Identifiers
- Reserved Words
- Quotations

Python Variables

- Declaration
- Printing
- Deleting
- Multiple Assignment
- Naming

Python Data Types

- Number
- Set
- Dictionary
- Sequence

Python Syntax

- a set of rules that are used to create a Python Program.

```
print ( "Hello World")
```

Different modes of Python Programming

(a) Interactive Mode Programming
(b) Script Mode Programming

Display Output

- **print()** function in Python displays an output to a console or to the text stream file.

```
print('Hello World!')
print(1234)
print(True)
```

```
PS C:\Users\HOME> & C:/Users/HOME/AppData/Local/Programs/Python/Python39/python.
exe c:/Users/HOME/HW.py
Hello World!
1234
True
```

Display Output

- **print()** function can also display the values of one or more variables separated by a comma.

```
Ram
Ram 21
Name: Ram , Age: 21
```

```
name="Ram"
print(name) # display single variable


age=21
print(name, age)# display multiple variables
print("Name:", name, ", Age:",age) # display formatted output
```

User’s Input

- **input()** function is used to get the user's input. It reads the key strokes as a string object which can be referred to by a variable having a suitable name.

```
print ('What is your name?')
name = input ()
print ("Name:", name)
```

Statements

- ends with the token NEWLINE character (carriage return).
- It means each line in a Python script is a statement.

```
print('id: ', 1)
print('First Name: ', 'Steve')
print('Last Name: ', 'Jobs')
```

Multi-Line Statements

- use backslash character \ to join a statement span over multiple lines

```
print('id: ',\
1)
print('First Name: ',\
'Ste\
ve')
print('Last \
Name: ',\
'Jobs')
```

Multi-Line Statements

- Expressions in parentheses (), square brackets [ ], or curly braces { } can be spread over multiple lines without using backslashes.

```
list = [1, 2, 3, 4,
        5, 6, 7, 8,
        9, 10, 11, 12]
print (list)
```

Multi-Line Statements

Please note that the backslash character spans a single line in one logical line and multiple physical lines, but not the two different statements in one logical line.

Multiple Statements in Single Line

- Use the semicolon ; to separate multiple statements in a single line.

```
print('id: ', 1);print('First Name: ', 'Steve');print('Last Name: ', 'Jobs')
```

Comments

- the symbol # indicates the start of a comment line. It is effective till the end of the line in the editor.

```
# this is a comment

print("Hello World")

print("Welcome to Python Tutorial") #comment after a statement.
```

Comments

There is no provision to write multi-line comments, or a block comment. For multi-line comments, each line should have the # symbol at the start.

Multi-line Comments

- A triple quoted multi-line string is also treated as a comment if it is not a docstring of the function or the class.

```
'''
comment1
comment2
comment3
'''
```

Indentation

- Leading space or tab at the beginning of the line is considered as indentation level of the line, which is used to determine the group of statements. Statements with the same level of indentation considered as a group or block.

For example, functions, classes, or loops in Python contains a block of statements to be executed. Other programming languages such as C# or Java use **curly braces { }** to denote a block of code. Python uses **indentation** (a space or a tab) to denote a block of statements.

Indentation Rules

- Use the colon **:** to start a block and press Enter.
- All the lines in a block must use the same indentation, either space or a tab.
- Python recommends **four spaces** as indentation to make the code more readable. Do not mix space and tab in the same block.
- A block can have inner blocks with next level indentation.

Identifiers

- a name used to identify a variable, function, class, module or other object.
- starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).
- Python does not allow punctuation characters such as @, $, and % within identifiers.

Python is a case sensitive programming language.
*Thus, **Manpower** and **manpower** are two different identifiers in Python.*

Naming Identifiers

- Python Class names start with an uppercase letter. All other identifiers start with a lowercase letter.
- Starting an identifier with a single leading underscore indicates that the identifier is **private** identifier.
- Starting an identifier with two leading underscores indicates a strongly **private** identifier.
- If the identifier also ends with two trailing underscores, the identifier is a **language-defined** special name.

Reserved Words

- These are reserved words and you cannot use them as constant or variable or any other identifier names.
- All the Python keywords contain lowercase letters only.

Reserved Words

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

Quotations

- accepts single ('), double (") and triple (''' or """) quotes to denote string literals, as long as the same type of quote starts and ends the string.
- The triple quotes are used to span the string across multiple lines.

```
word = 'word'
print (word)

sentence = "This is a sentence."
print (sentence)

paragraph = """This is a paragraph.
It is
 made up of multiple lines and
sentences."""
print (paragraph)
```

TOPICS

Python Syntax

- Display Output
- User’s Input
- Statements
- Comments
- Indentation
- Identifiers
- Reserved Words
- Quotations

Python Variables

- Declaration
- Printing
- Deleting
- Multiple Assignment
- Naming

Python Data Types

- Number
- Set
- Dictionary
- Sequence

Python Variables

- are the reserved memory locations used to store values with in a Python Program.
- This means that when you create a variable you reserve some space in the memory.

Python Variables

- Based on the data type of a variable, Python interpreter allocates memory and decides what can be stored in the reserved memory.
- Therefore, by assigning different data types to Python variables, you can store integers, decimals or characters in these variables.

Variable Declaration

- A Python variable is created automatically when you assign a value to it. The equal sign (=) is used to assign values to variables.
- The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable.

Variable Declaration

- counter = 100 # Creates an integer variable
- miles = 1000.0 # Creates a floating point variable
- name = "Zara Ali" # Creates a string variable

Printing Variables

- create a Python variable and assign a value to it, we can print it using print() function.

```
counter = 100          # Creates an integer variable
miles   = 1000.0       # Creates a floating point variable
name    = "Zara Ali"   # Creates a string variable
print (counter)
print (miles)
print (name)
```

Deleting Variables

- delete the reference to a number object by using the del statement.
- del var1[,var2[,var3[....,varN]]]]

```
counter = 100
print (counter)

del counter
print (counter)
```

Multiple Assignment

- Python allows to initialize more than one variables in a single statement.

```
>>> a=10
>>> b=10
>>> c=10
```

```
>>> a=b=c=10
>>> print (a,b,c)
10 10 10
```

Multiple Assignment

- Python allows to initialize more than one variables in a single statement.

```
>>> a=10
>>> b=20
>>> c=30
```

```
>>> a,b,c = 10,20,30
>>> print (a,b,c)
10 20 30
```

Naming Variable

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number or any special character like $, (, * % etc.
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Python variable names are case-sensitive which means Name and NAME are two different variables in Python.
- Python reserved keywords cannot be used naming the variable.

Naming Variable

If the name of variable contains multiple words, we should use these naming patterns −

- **Camel case** − First letter is a lowercase, but first letter of each subsequent word is in uppercase. For example: kmPerHour, pricePerLitre
- **Pascal case** − First letter of each word is in uppercase. For example: KmPerHour, PricePerLitre
- **Snake case** − Use single underscore (_) character to separate words. For example: km_per_hour, price_per_litre

Naming Variable

```
counter = 100
_count  = 100
name1 = "Zara"
name2 = "Nuha"
Age  = 20
zara_salary = 100000

print (counter)
print (_count)
print (name1)
print (name2)
print (Age)
print (zara_salary)
```

```
100
100
Zara
Nuha
20
100000
```

```
1counter = 100
$_count  = 100
zara-salary = 100000

print (1counter)
print ($count)
print (zara-salary)
```

```
File "main.py", line 3
    1counter = 100
    ^
SyntaxError: invalid syntax
```

TOPICS

Python Syntax

- Display Output
- User’s Input
- Statements
- Comments
- Indentation
- Identifiers
- Reserved Words
- Quotations

Python Variables

- Declaration
- Printing
- Deleting
- Multiple Assignment
- Naming

Python Data Types

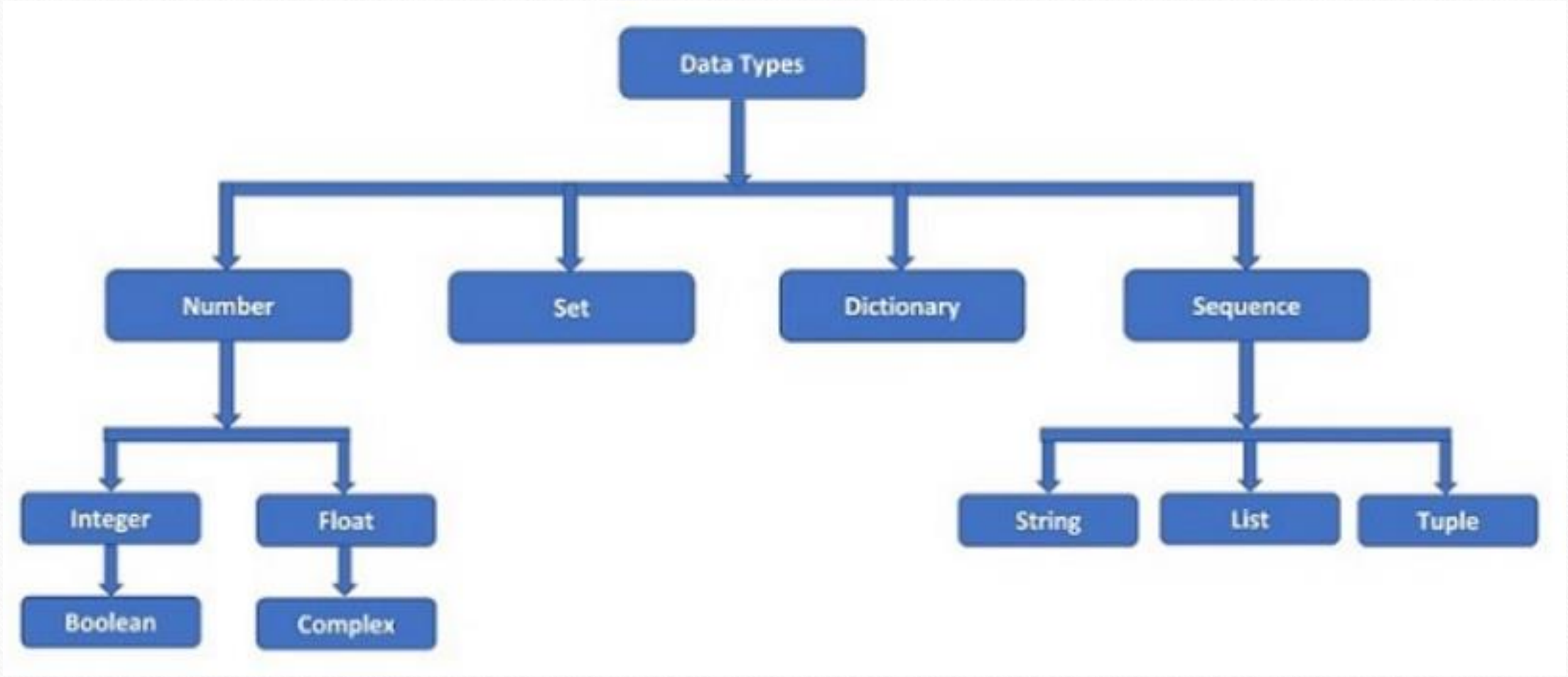
- Number
- Set
- Dictionary
- Sequence

Computer is a data processing device. Computer stores the data in its memory and processes it as per the given program. Data is a representation of facts about a certain object.

Data type represents a kind of value and determines what operations can be done on it.

Python Data Types

- are used to define the type of a variable.
- It defines what type of data we are going to store in a variable.
- The data stored in memory can be of many types.


Data Types
Number
Set
Dictionary
Sequence
Integer
Float
Boolean
Complex
String
List
Tuple

Python Numeric Data Type

```
var1 = 1          # int data type
var2 = True       # bool data type
var3 = 10.023     # float data type
var4 = 10+3j      # complex data type
```

Python Numeric Data Type

- Python numeric data types store numeric values. Number objects are created when you assign a value to them.
- Python supports four different numerical types and each of them have built-in classes in Python library, called **int**, **bool**, **float** and **complex**

Python Numeric Data Type

Python's standard library has a built-in function **type()**, which returns the class of the given object.

```
>>> type(123)
<class 'int'>
>>> type(9.99)
<class 'float'>
```

Python Numeric Data Type

A complex number is made up of two parts - **real** and **imaginary**. They are separated by '+' or '-' signs. The imaginary part is suffixed by 'j' which is the imaginary number. The square root of -1 ($\sqrt{-1}$), is defined as imaginary number. Complex number in Python is represented as x+yj, where x is the real part, and y is the imaginary part. So, 5+6j is a complex number.

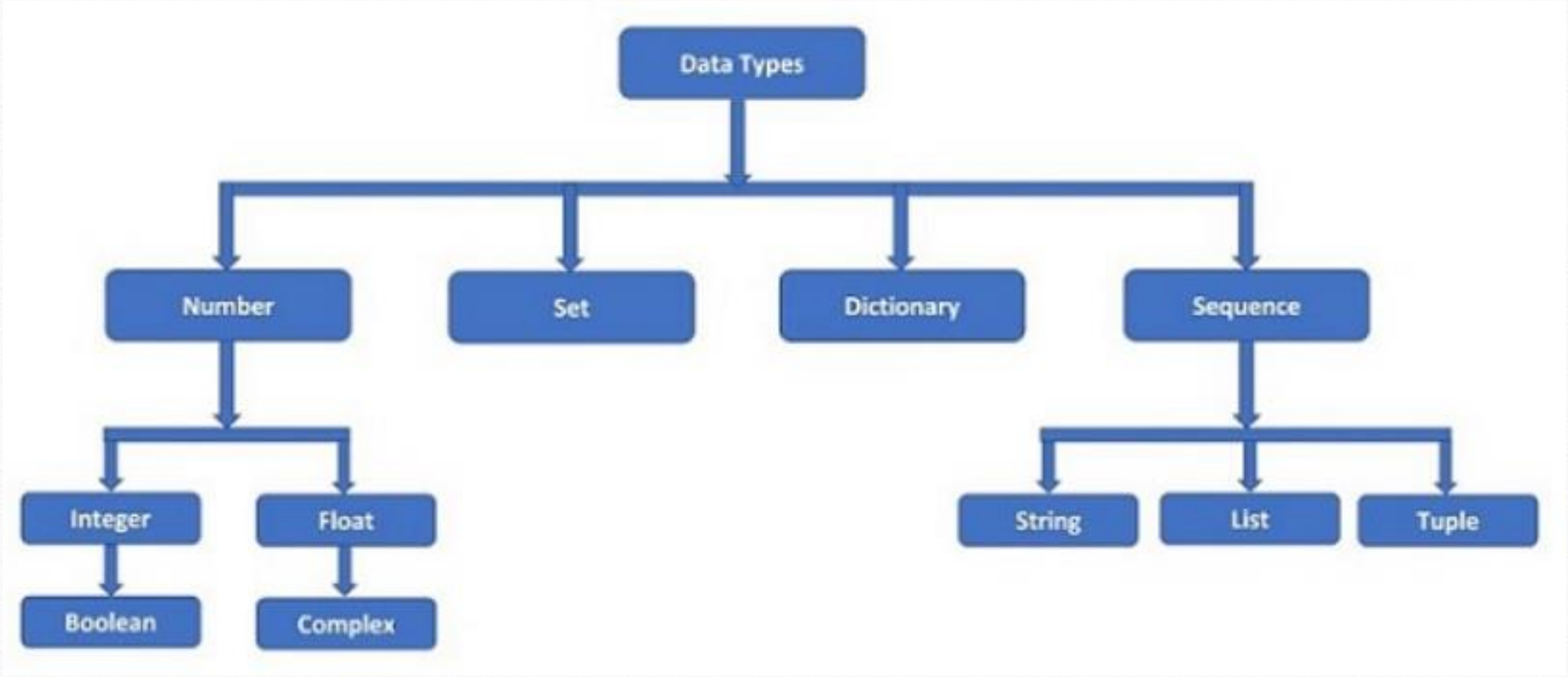
```
>>> type(5+6j)
<class 'complex'>
```

Python Numeric Data Type

A Boolean number has only two possible values, as represented by the keywords, **True** and **False**. They correspond to integer 1 and 0 respectively.

```
>>> type (True)
<class 'bool'>
>>> type(False)
<class 'bool'>
```

int	bool	float	complex
10	True	0.0	3.14j
0O777	False	15.20	45.j
-786		-21.9	9.322e-36j
080		32.3+e18	.876j
0x17		-90.	-.6545+0J
-0x260		-32.54e100	3e+26J
0x69		70.2-E12	4.53e-7j

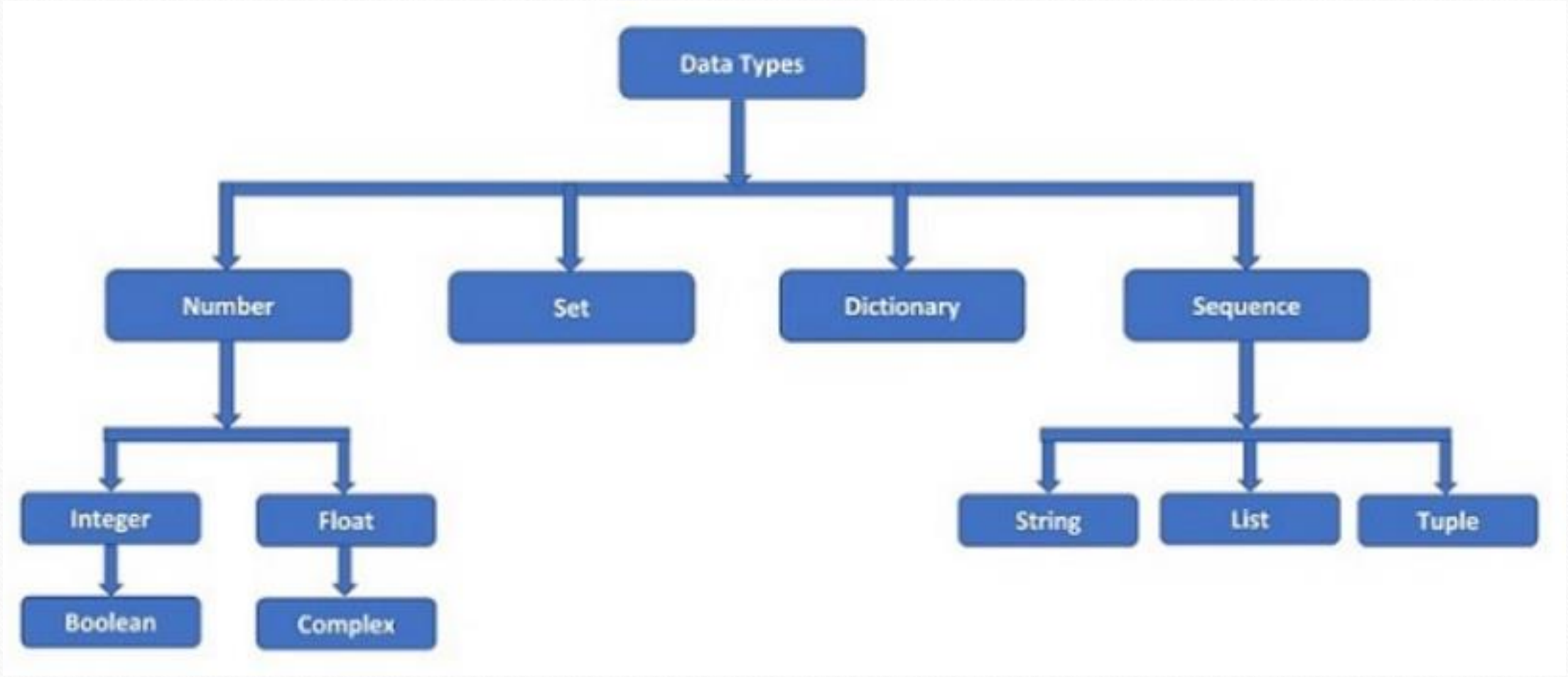

Data Types
Number
Set
Dictionary
Sequence
Integer
Float
Boolean
Complex
String
List
Tuple

Python Set Data Type

- Set is a Python implementation of set as defined in Mathematics.
- A set in Python is a collection, but is not an indexed or ordered collection like sequence data types.

Python Set Data Type

- An object cannot appear more than once in a set, whereas in sequence data types, same object can appear more than once.
- Comma separated items in a set are put inside curly brackets or braces {}. Items in the set collection can be of different data types.

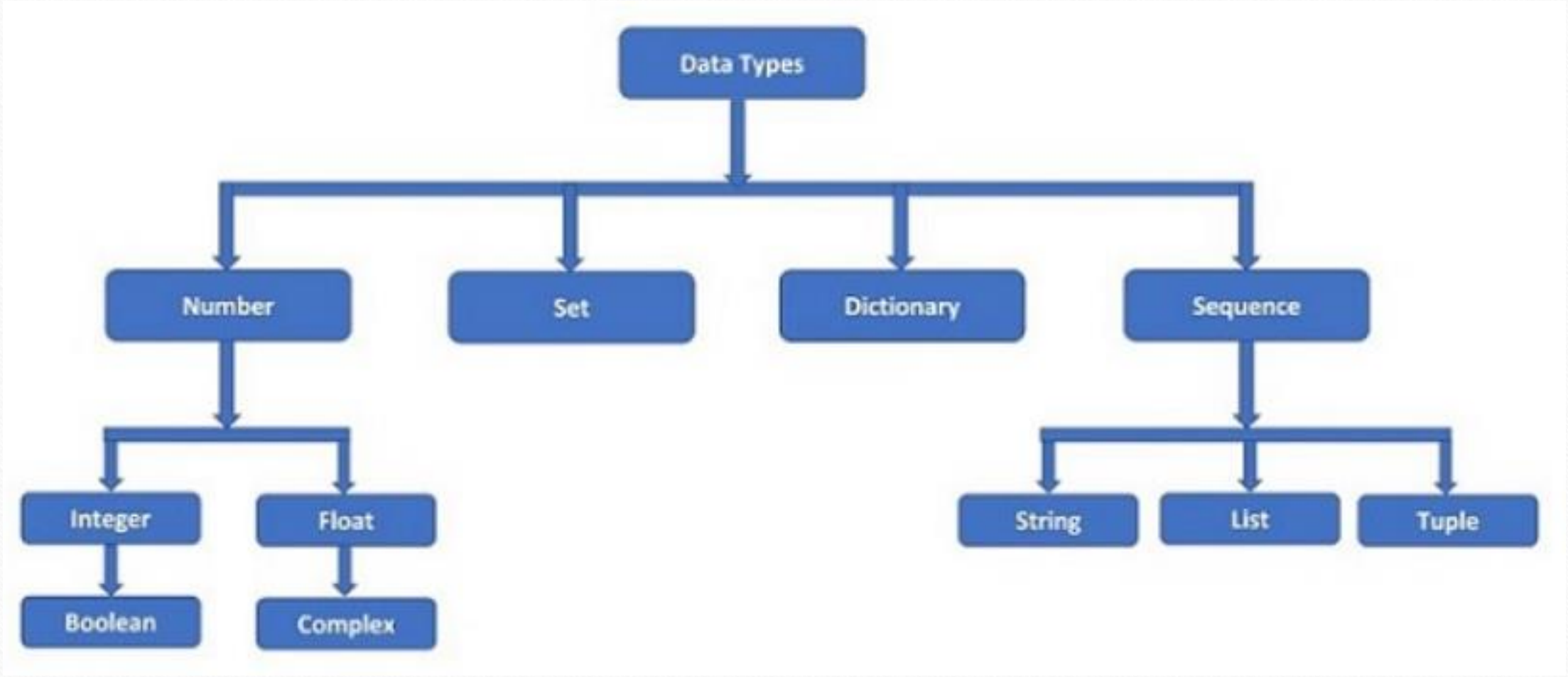

Data Types
Number
Set
Dictionary
Sequence
Integer
Float
Boolean
Complex
String
List
Tuple

Python Dictionary Data Type

- Python dictionaries are kind of hash table type.
- A dictionary key can be almost any Python type, but are usually numbers or strings.
- Values, on the other hand, can be any arbitrary Python object.

Python Dictionary Data Type

- Python dictionary is like associative arrays or hashes found in Perl and consist of key: value pairs.
- The pairs are separated by comma and put inside curly brackets {}.
- To establish mapping between key and value, the semicolon':' symbol is put between the two.


Data Types
Number
Set
Dictionary
Sequence
Integer
Float
Boolean
Complex
String
List
Tuple

Python Sequence Data Type

- Sequence is a collection data type.
- It is an ordered collection of items.
- Items in the sequence have a positional index starting with 0.
- It is conceptually similar to an array in C or C++.

Python Sequence Data Type

There are following three sequence data types defined in Python.

- String Data Type
- List Data Type
- Tuple Data Type

Any Questions?

References

Learn Python Programming. (2023). https://www.tutorialsteacher.com/python

Python Tutorial. (2022). https://www.w3resource.com/python/python-tutorial.php

Python Tutorial. (n.d.). https://www.tutorialspoint.com/python/index.htm

Python Tutorial. (n.d.). https://www.w3schools.com/python/default.asp

Unit Output 4

Answer the following questions in a 1/2 sheet of paper. Write the complete program and not just the answers on the blanks.

1

Insert the missing part of the code below to output "Hello World".

```
      ("Hello World")
```

2

This example misses indentations to be correct.

Insert the missing indentation to make the code correct:

```
if 5 > 2:
print("Five is greater than two!")
```

3

Comments in Python are written with a special character, which one?

```
This is a comment
```

4

Use a multiline string to make the a multiline comment:

```
"""
This is a comment
written in
more than just one line
"""
```

5

Create a variable named `carname` and assign the value `Volvo` to it.

```
        = "      "
```

6

Create a variable named `x` and assign the value `50` to it.

```
 = 
```

7

Display the sum of `5 + 10`, using two variables: `x` and `y`.

```
  = 
y = 10
print(x  y)
```

8

Create a variable called `z`, assign `x + y` to it, and display the result.

```
x = 5
y = 10
  = x + y
print( )
```

9

Remove the illegal characters in the variable name:

```
2my-first_name = "John"
```

10

Insert the correct syntax to assign the same value to all three variables in one code line.

```
x   y   z   "Orange"
```

11

The following code example would print the data type of x, what data type would that be?

```
x = 5
print(type(x))
```

12

The following code example would print the data type of x, what data type would that be?

```
x = "Hello World"
print(type(x))
```

13

The following code example would print the data type of x, what data type would that be?

```
x = 20.5
print(type(x))
```

14

The following code example would print the data type of x, what data type would that be?

```
x = True
print(type(x))
```

15

Insert the correct keyword to make the variable x belong to the global scope.

```
def myfunc():
    _____ x
  x = "fantastic"
```