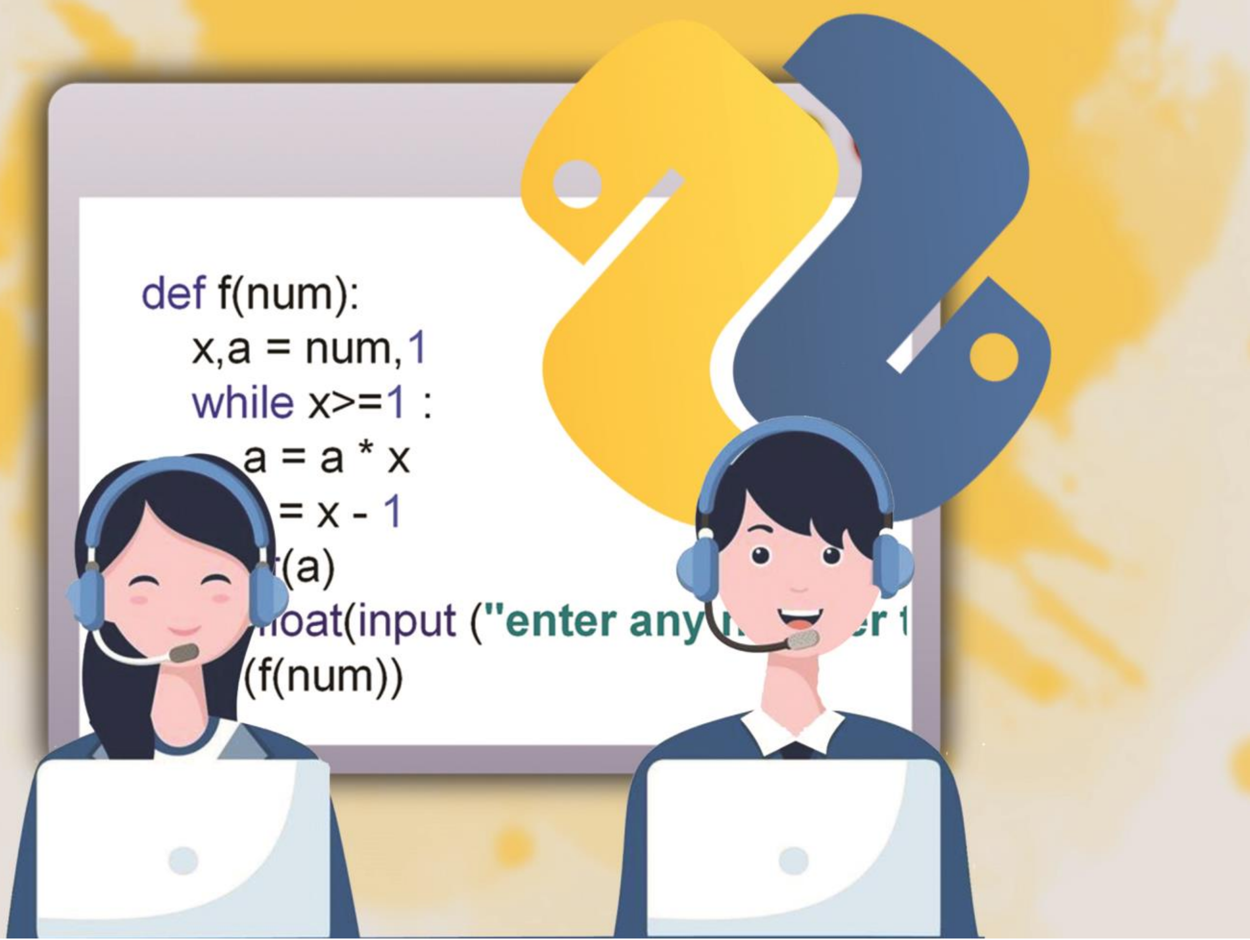
IP EDUCATION

PYTHON

The First Steps In Programming



```
def f(num):
    x,a = num,1
    while x>=1 :
        a = a * x
          = x - 1
        (a)
      loat(input ("enter any        er t
     (f(num))
```

Prepared By

Al-Joushen Company For Educational Activities

Written by:

Eng. Mohammad W. AbuFares

To get a hardcopy of this book, please feel free to contact us at 0781999744

Table of contents

Chapter 1 : Hello Python

Chapter 2 : Data types

Chapter3: Control Statements

Chapter 4: Functions

CHAPTER 2

Data Types

OBJECTIVES:

1.Students can create variables and assign their values.

2. Students understand what types of data are in Python, and can use them in programming as needed.

3. Students have the ability to perform operations on string data.

4. Students have the ability to perform operations on Boolean data. Which includes comparison, logic, identity, and "in" operators.
Students have the ability to perform operations on Numeric data. This includes a full understanding of arithmetic operators, expressions, and knowing the order of priorities.

SUBJECTS

- 2.1 Variables
- 2.2: Data types
- 2.3 String Operations
- 2.4 Boolean Operations
- 2.5 Numeric Operations

The main job of a computer program is process data. This data can be different . But in computer's memory is always converted into a specific sequence of binary digits. The data may be decimal or integer numbers, letters, string, or array.

In Python, like in all programming languages, data types are used to classify one particular type of data. This is important because the specific data type you use will determine what values you can assign to it and what you can do to it (including what operations you can perform on it).

In Python, we can use data directly, but programming process must be organized , so we use variables to store data and make processes more than once during the execution of the program.

Variable: A name that is stored in a computer's memory (RAM), in which a specified value (which is data) is stored. All types of data can be stored in the variable

creating variables :

The process of creating variables aims to facilitate access to data or to save or use it within the program.

To make a variable in the program, we start with the following steps:

A. **Naming variables**: Follow the rules when naming variable.

1. The variable is as short and clear as possible.
2. The variable consists of letters and numbers, and that it begins with a letter.
3. It is not allowed to write symbols in the variable except for the symbol (_).
4. A variable cannot be written with the same reserved words in Python.The table shows reserved words in Python.

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		

In Python, it does not distinguish between capital and small letters, for example the variable job is not the same as JOB or Job. So always try used to writing variables in small letters.

TasK 1

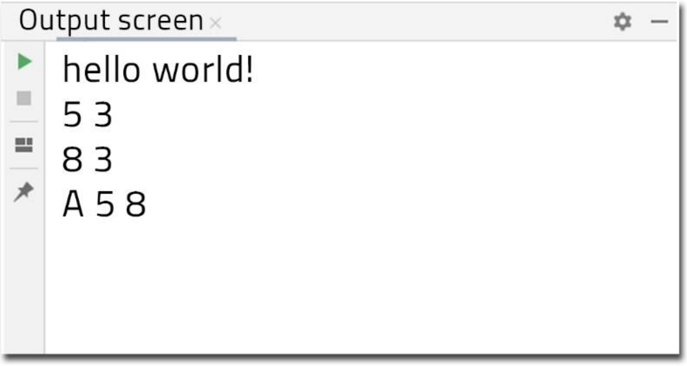
which of these variables is incorrect and why?

I	Form	none	IF	1x	Z1	Z_5
#3	A&B	Global	sum	Print	go to	class

B. Assignment statement:

After naming the variable, its value (which is one of the data types (numeric or text)) must be stored, variable's value is stored by use the sign (=) ,then putting the value, to understand this executing the following program:

```
X= 1
Y= 5.5
B= "hello, world"
print (X)
print (B)
print (X+Y)
```

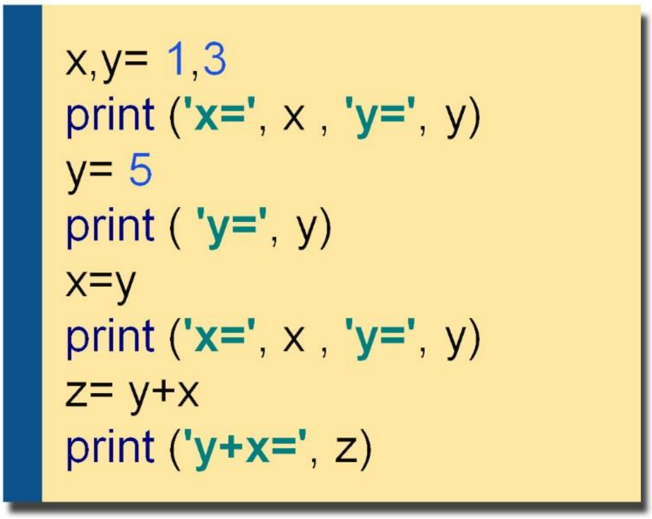


```
hello world!
5 3
8 3
A 5 8
```

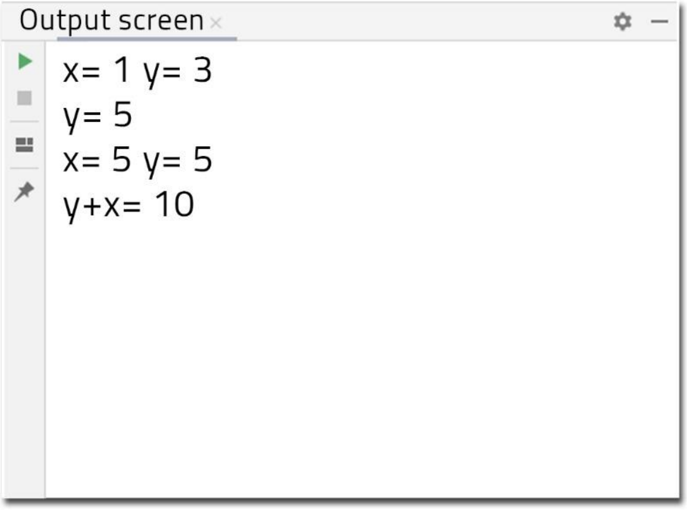
In the previous program, three variables were made: the first variable is x and its value is 1, which is a real number, the second is the variable y and its value is5.5 which is a decimal number, and the third variable B has a value of "hello, world", which is a text value.

When using the print function, it shows the value of the variable, while the last print function found the result of mathematical operation for two variables.

Do executing the following program:



```
x,y= 1,3
print ('x=', x , 'y=', y)
y= 5
print ( 'y=', y)
x=y
print ('x=', x , 'y=', y)
z= y+x
print ('y+x=', z)
```



```
x= 1 y= 3
y= 5
x= 5 y= 5
y+x= 10
```

In the previous program, we conclude that it is possible to create more than one variable in one line and store their values, using a comma (,) between each variable and its value, as in the example
A, B, C = 1, 2, 3 , where the value of variable A is 1, variable B is 2, and variable C is 3.

The variable value can be changed in subsequent lines. The value of the variable Y has been changed from 3 (in the first line) to 5 (the third).

The value of a variable can be recorded using another variable or the result of a process as in:

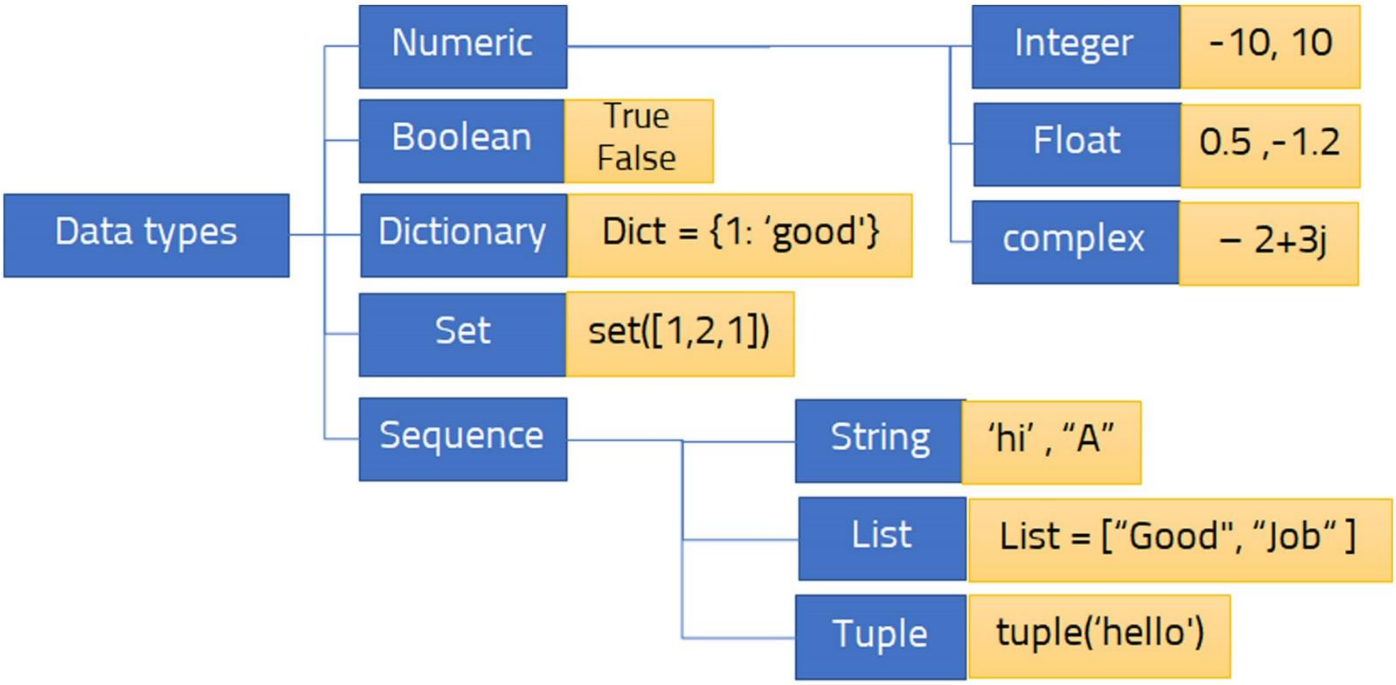
The fifth line where the value of variable x has changed to the value of variable.

The seventh line where the value of z is the result of the process (x + y).

When writing the variables on one line, it is not required that they be of the same type of data, for example A, B = "python ", 10, so the variable A is a text variable, and the variable B is a numerical variable.

In Python, like in all programming languages, data types are used to classify one particular type of data. Data types are the classification of data items. It represents the kind of value that tells what operations can be performed on a particular data. We will go over the important data types that are native to Python. This is not an exhaustive investigation of data types, but will help you become familiar with what options you have available to you in Python.

The standard or built-in data type of Python:



Data types
Numeric
Integer -10, 10
Float 0.5 ,-1.2
complex – 2+3j
Boolean True False
Dictionary Dict = {1: 'good'}
Set set([1,2,1])
Sequence
String 'hi' , "A"
List List = ["Good", "Job"]
Tuple tuple('hello')

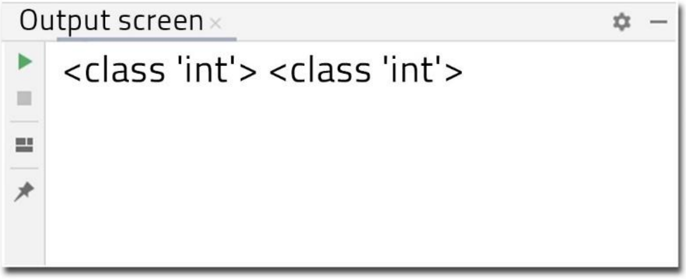
1. Numeric data :

Python has three built-in numeric data types : integers, float point numbers , and complex numbers. In this level , you'll learn about integers and float numbers, which are the two most commonly used number types.

A. I**ntegers Numbers (int)** : it is a whole number with no decimal places. For example, 1 is an integer, but 1.0 isn't. It represents the positive and negative numbers (Ex. -5, 5). The name in python for the integer data type is(int). We use integers , as we can store them in variables and perform operations on them.

To find out what type of data is used, we use (type()) function, for example :

```
x= 3
y=-100000
print (type(x) ,
type(y))
```



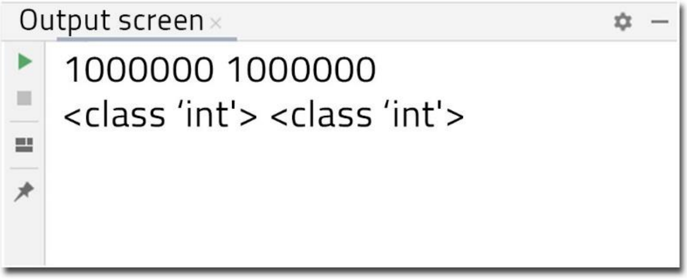
```
<class 'int'> <class 'int'>
```

Notice that, in the previous example the output of the print function was integer data.

When you write large numbers by hand, you typically group digits into bundles of three separated by comma . The number 1,000,000 is a lot easier to read than if it was written 1000000.

In Python, you can’t use commas to group digits , but you can use underscores (_). Both of the following are valid ways to represent the number one million as an integer literal:

```
x= 1000000
y= 1_000_000
print(x ,y)
print (type(x), type(x))
```



```
1000000 1000000
<class ‘int'> <class ‘int'>
```

In Python, there is no limit for the value of an integer, so you can write and use an infinity number.

□ TasK 1

?

Try writing the largest integer number ever you know, then make sure that Python can handle it without a problem using the print function.

□ TasK 2

?

Which of these numbers is integer?

3,000.5 500 -20 1_05_000 1_200.5

-700.2 10’ 5e10 10e-10 - 15e2