



PALAWAN STATE UNIVERSITY
College of Sciences



CC 2/L COMPUTER PROGRAMMING 1

**Variables, Statements
and Expressions**



MODULE 02



Table of Contents

Content	Page
Heading 1	1
Heading 2	2
Heading 3	3
Heading 4	4
Heading 5	5
Heading 6	6
Heading 1	7
Heading 1	8
Heading 1	9
Heading 1	10
Heading 1	11
Heading 1	12



Learning Objectives

After going through in this module, you should be able to:

- ✓ Identify variables that reference an object of a particular type.
- ✓ Convert type into another type
- ✓ Simulate evaluation of an expression and assignment statement.
- ✓ Use reassignment to increment a variable (accumulator pattern)
- ✓ Get input from a user and convert the input to the appropriate type.
- ✓ Identify the following types of values: strings, integers, floats, functions.
- ✓ Recognize valid vs. invalid variable names.
- ✓ Write an assignment statement.



Overview

1.1 Foundations of Python Programming

A value type in computer programming in Python is a coded object that involves memory allocation directly where it is created. Value types are commonly contrasted to reference types that instead act as pointers to a value that is stored elsewhere (Techopedia, 2021).

Information types and variables in Python are diverse in certain perspectives from other programming dialects. There are whole numbers, floating point numbers, strings, and some more, yet things are not equivalent to in C or C++. Python gives control information types like records as a real piece of the language (Dharaiya, 2019).

In this module you will learn the basic building blocks of Python programs. Such as literals, operators, function calls, and variables.

- 1.Literals - like numbers and character strings
- 2.Operators - like + and *
- 3.Function calls - which take values as input/s and compute new values
- 4.Variables - which save values so they can be used later in the program



Discussion

1.2 Values and Datatypes

A value is one of the fundamental things - like a word or a number - that a program manipulates. It represents a certain variable that is used for declaration.

Examples:

result = 2+3, 5 is the value of result when you add the 2 + 3

word = "Hello World!", where the value of word is "Hello World"

These objects are classified into different data types.

The different basic data types are:

- integer - whole numbers (e.g. 4, -1, 3, 0)
- float - numbers with decimal (e.g. 1.25, 1.0, -5.7)
- strings - enclosed in quotation marks (e.g. "Hello World", "PSU")

[]

```
1      x = 5
2      base = 12.5
3      name = 'Adree'
```



Discussion

1.3 Operators and Operands

Operators are special tokens that represent computations like addition, subtraction, multiplication, and division. The values the operator works on are called **operands**.

```
1      20 + 32
2      5 ** 2
3      5 + 9) * (15-7)
4      print(7+5)

12
```

Line 2. Double asterisk ** is the token for exponentiation

Line 3. Asterisk * is the token for multiplication

Line 4. Gets the sum of 7 and 5, then prints the result.

Remember: If you want to see the results of the computation, use the word print. print (7+5) is used in line 4 to print the sum 12, then run the program.



Discussion

Operators and Operands

Line 1,2,& 4. The division operator / produces a floating point result even if the result is an integer. For example $4/2$ is 2.0.

Line 3 & 5. If you want truncated division, which ignores the remainder, you can use the // operator. For example $5//2$ is 2.

```
[]
```

```
1      print(9/5)
2      print(5/9)
3      print(9//5)
4      print(7.0/3.0)
5      print(7.0//3.0)
```

```
1.8
0.5555555555555556
1
2.3333333333333335
2.0
```



Discussion

Operators and Operands

The modulus operators work on integers and yields the remainder when the first operand is divided by the second. The modulus operator is a percent sign %.

```
[ ]
```

```
1      print(7//3)
```

```
2      print(7%3)
```

```
2
```

```
1
```




Learning Check

Exercise 01

1. Assign a value of 8 to a variable named Number.

```
</>
```

2. Assign a value of 'Puerto Princesa City' to a variable named City.

```
</>
```

3. Assign a value of '3.1416 to a variable named Pi.

```
</>
```



Discussion

Tuples

A tuple is like a list, it can be a collection of items of any type, but instead of using square brackets, tuples are represented using parenthesis.

The code below is a sample declaration of a tuple.

```
ict1 = ("ICT1", "Computer Fundamentals and Programming",  
2, "Laboratory", "Wed 8:00-11:00", "NIT 3")
```

The difference between lists and tuples is that a tuple is immutable, meaning its content can't be changed after the tuple is created.

Index Operator: Working with the Characters of a String

As mentioned, Strings can be defined as a sequential collection of characters. To select a single character from a string, Python uses square brackets to enclose the index. The characters in a string are accessed by their position or index value. For example, "Palawan" has seven characters, and it is indexed left to right from position 0 to position 6.

0	1	2	3	4	5	6
P	a	l	a	w	a	n
-7	-6	-5	-4	-3	-2	-1

You can also use negative numbers as position or index value where -1 is the rightmost index and so on.



Discussion

Look at the example below,

```
1 school = "Palawan State University"
2 m = school[2]
3 print(m)
4
5 lastchar = school[-1]
6 print(lastchar)
```

Line 2. This expression selects the character at index 2 from the value of the variable named **school**. The letter at index zero (0) of "Palawan State University" is P. So at position [2] we have letter l.



Learning Check

Exercise 02

1. Given the statement `school = "Palawan State University"`, get the capital letter 'S' and assign it to a variable named **letter**.

```
</>
```

2. Using the variable named `school` in the previous number, get the 3rd letter 'a' from the left. Use a negative number as an index.

```
</>
```

3. What is printed by the following statements?

```
sentence = "python rocks"  
print(sentence[2] + sentence[-4])
```

4. What is printed by the following statements?

```
alist = [3, 67, "cat", [56, 57, "dog"], [ ], 3.14, False]  
print(alist[5])
```



Discussion

Index Operator: Accessing Elements of a List or Tuple

Same as in the string, you will use square brackets to access the elements of list or tuple. Remember that indices start at zero (0), any integer expression can be used as an index. Negative index value will locate items from the right.

Try to predict what will be printed out by the following code, then run the code in your IDE to check your prediction.

```
numbers = [17, 123, 87, 34, 66, 8398, 44]
print(numbers[2])
print(numbers[9-8])
print(numbers[-2])
```



Learning Check

Exercise 03

1. Given the statement `numbers = [17, 123, 87, 34, 66, 8398, 44]`, assign the sum of elements whose values are 87, 34, and 44 to a variable named **sum**. After the assignment statement, write a code that displays the value of sum.

```
</>
```



Discussion

The len Function

In Python, the **len** is a predefined function that returns the number of characters in a string.

```
1 fruit = "Banana"
2 print(len(fruit))
```

Line 2. This code will display the length of a variable named **fruit**. It will return a number six (6) because 'Banana' has six characters.

To get the last letter of a string using the len function in the expression.

```
1 fruit = "Banana"
2 len_fruit = len(fruit)
3 lastch = fruit[len_fruit-1]
4 print(lastch)
```

Line 2. This code will assign 6 to variable **len_fruit** because that is the length of a string "Banana".

Line 3. This code will select the character in index 5 in a string "Banana" and assign it to a variable named **lastch**.

Line 4. This code will print the value of a variable named **lastch**.

Using **len_fruit** as an index will cause runtime error because there is no letter at index position 6 in "Banana", since indexing starts counting at zero. To get the last character, you have to subtract 1 from the length.

Typically, you can combine lines 2 and 3 to a single line statement.

```
lastch = fruit[len(fruit)-1]
```



Discussion

You can also use the **len** function to access the middle character of the string.

```
fruit = "grape"  
midchar = fruit[len(fruit)//2]
```

When the **len** function is used in a list it will return the number of items in the list.

```
alist = ["hello", 2.0, 5]  
print(len(alist))  
print(len(alist[0]))
```



Learning Check

Exercise 04

1. What is printed by the following statements?

```
alist = [1, 57, "Reid", 3.1416, False]  
print(len(alist))
```

2. Assign the number of elements in pet_names to the variable named output.

```
pet_names = ['Charlie', 'Max', 'Buddy', 'Milo', 'Reid',  
'Blanket', 'Luna', 'Nixie', 'Yuki', 'Leo']
```

```
</>
```



Discussion

The Slice Operator

“He”, “Hell”, “e”, and “el” are some of the substrings of the word “Hello”. A substring of a string is called a **slice**. Selecting a slice is similar to selecting a character.

Look at the examples below.

```
1 sentence = "If you don't know where you are, a map won't help.  
-Watts Humphrey"  
2 print(sentence[0:6])  
3 print(sentence[7:12])  
4 print(sentence[17:21])
```

Line 1. This code will get the substring “If you” from the sentence; characters in index 0 to 5.

Line 2. This code will get the substring “don’t” from the sentence; characters in index 7 to 11.

Line 3. This code will get the substring “ whe” from the sentence; characters in index 17 to 20.

The slice operator [n:m] returns part of the string starting with a character at index n and going up to but not including the character at index m.

If you omit the first index (n) before the colon, the slice starts at the beginning of the string. If you omit the second index (m) the slice goes to the end of the string.

```
1 fruit = "banana"  
2 print(fruit[:3])  
3 print(fruit[3:])
```

Line 2. This code will get the substring “ban” from “banana”.

Line 3. This code will get the substring “ana” from “banana”.



Learning Check

Exercise 05

1. Write a code that slices the “quietly and clearly” from the sentence below and put it in a variable named **excerpt_des**.

```
desiderata = “Speak your truth quietly and clearly; and listen to others, even to the dull and the ignorant; they too have their story.”
```

</>

2. Write a code that slices the word “and” from the value of variable **excerpt_des**.

</>



Discussion

List Slices

The slice operation used in a string will work on list elements too. If you will use slice operation in list this will return a list of elements.

```
1 a_list = ['a', 'b', 'c', 'd', 'e', 'f']
2 print(a_list[1:3])
3 print(a_list[:4])
4 print(a_list[3:])
5 print(a_list[:])
```

Line 2. This code will get the items in a_list; from index 1 to 2.

Line 3. This code will get the items in a_list; from index 0 to 3.

Line 4. This code will get the items in a_list; from index 3 onwards.

Line 5. This code will get all items in a_list.

Tuple Slices

You can't modify the elements of a tuple, but you can make a variable reference a new tuple holding different values.

If you will use slice operation to tuple it will return a new tuple.

```
1 julia = ("Julia", "Roberts", 1967, "Duplicity", 2009,
"Actress", "Atlanta, Georgia")
2 print(julia[2])
3 print(julia[2:6])
4 print(len(julia))
5 julia = julia[:3] + ("Eat Pray Love", 2010) + julia[5:]
print(julia)
```

Line 2. This code will print an element in position index 2 from tuple julia.

Line 3. This code will print elements of tuple julia from index 2 to 5.

Line 4. This code will print the number of elements in tuple julia.

Line 5. This code will form another tuple; elements are from index 0 to 2, "Eat Pray Love", 2010, and elements from 5 onwards.



Rubrics

COPY AND PASTE YOUR RUBRICS HERE



References

References:

Techpedia. (2021). Value Type. Retrieved from:
<https://www.techopedia.com/definition/25556/value-type>:

Dharaiya, Divyesh. (2019, August 16). The Importance of Values and Variables-Python. Retrieved from: <https://medium.com/python-concepts/the-importance-of-values-and-variables-python-86e7545e9358>