

Python Programming Language

1

Lecture 3: Variables, String and Operator

Hafsa Sultana

Computer Science and Engineering

Khulna University

Today's Learning Objectives

- Keywords
- Expressions,
- Delete Variable
- Variable Scope,
- Local Variables,
- Global Variables,
- Casting
- Advanced user input,
- Constants
- String
- Operator
- Practical Exercises,
- Debugging Tips,

Keywords

Python has a set of keywords that are reserved words that cannot be used as variable names, function names, or any other identifiers:

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

Expressions

- A combination of values, variables, operators, and function calls that can be evaluated to produce a result.
- An expression always returns a value.
- Examples:
 - $2 + 3$
 - $x * y$
 - `len("hello")`

Types of Expressions

- Arithmetic Expressions:

Examples: $5 + 3$, $10 - 2$, $7 * 4$, $8 / 2$

- String Expressions:

Examples: `"Hello, " + "World!"`, `"Python" * 3`

- Boolean Expressions:

Examples: $5 > 3$, $10 == 10$, $x != y$

Variable Features

- Redeclaration
- Multiple Assignment

- ```
num1 = 5
num1 = 3.45
```

```
num1, num2, stri = 4, 5, "okay"
num1 = num2 = num3 = 10
```

```
print(num1, num2, num3)
```

- ```
x, y, z = "Orange", "Banana", "Cherry"  
print(x)  
print(y)  
print(z)
```

Delete Variable

- `Num1 = 10`
- `print(id(num1))` #Shows the RAM location where this is saved
- `del num1`
- `print(num1)`
- **TRY to Delete a variable**

Variable Scope

Determines where a variable can be used.
Local vs Global variables.

Local Variables

- Declared inside a function.
- Only accessible within that function.
- Example:

```
def my_function():  
    x = 10  
    print(x)
```

Global Variables

- Declared outside any function.
- Accessible anywhere in the code.
- Example:

```
x = 10
```

```
def my_function():  
    print(x)
```

Casting

Converting a variable from one type to another.

Example: `str(3)` converts integer 3 to string '3'

```
x = int(1)
y = int(2.8)
z = int("3")
print(x)
print(y)
print(z)
```

```
x = float(1)    # x will be 1.0
y = float(2.8)  # y will be 2.8
z = float("3")  # z will be 3.0
w = float("4.2") # w will be 4.2
print(x)
print(y)
print(z)
print(w)
```

```
x = str("s1") # x will be 's1'
y = str(2)    # y will be '2'
z = str(3.0)  # z will be '3.0'
print(x)
print(y)
print(z)
```

Advanced User Input

- Use `input()` function to get input from the user.
- Converting input to other data types.
- Example:
- `name = input('What is your name? ')`
- `print('Hello, ' + name)`

Constant

- Python doesn't have any formally defined constants,
- However you can indicate a variable to be treated as a constant by using all-caps names with underscores.
- For example, the name `PI_VALUE` indicates that you don't want the variable redefined or changed in any way.

```
PI_VALUE = 3.1416
```

String

- A string is a non-numeric data type
- **Slicing:** Subsets of strings can be taken using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.
- **Concatenation:** The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator in Python.

String

```
str = 'Hello World!'
```

```
print (str)      # Prints complete string
print (str[0])   # Prints first character of the string
print (str[2:5]) # Prints characters starting from 3rd to 5th
print (str[2:])  # Prints string starting from 3rd character
print (str * 2)  # Prints string two times
print (str + "TEST") # Prints concatenated string
```

String

Output:

- `print (str)` # Hello World!
- `print (str[0])` # H
- `print (str[2:5])` #llo
- `print (str[2:])` # llo World!
- `print (str * 2)` # Hello World!Hello World!
- `print (str + "TEST")` # Hello World!TEST

Python Operators

Operators are used to perform operations on variables and values.

➤ Examples: +, -, *, /, ==, !=

➤ **Types of Operators:**

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

Python Operators

1. Arithmetic Operators:

Operator	Name	Example
+	Addition	$a + b = 30$
-	Subtraction	$a - b = -10$
*	Multiplication	$a * b = 200$
/	Division	$b / a = 2$
%	Modulus	$b \% a = 0$
**	Exponent	$a ** b = 10 ** 20$
//	Floor Division	$9 // 2 = 4$

Arithmetic Operators:

Operator	Name	Example
+	Addition	$a + b = 30$
-	Subtraction	$a - b = -10$
*	Multiplication	$a * b = 200$
/	Division	$b / a = 2$
%	Modulus	$b \% a = 0$
**	Exponent	$a ** b = 10 ** 20$
//	Floor Division	$9 // 2 = 4$

Comparison Operators:

Operator	Name	Example
==	Equal	(a == b) is not true.
!=	Not equal	(a != b) is true.
>	Greater than	(a > b) is not true.
<	Less than	(a < b) is true.
>=	Greater than or equal to	(a >= b) is not true.
<=	Less than or equal to	(a <= b) is true.

Practical Exercises

1. Square of a Number

Input: Enter a number: 5

Output: 25

2. Power of a number

Input: Enter the number: 4

Enter the power number:3

Output: 64

Practical Exercises

3. Concatenate Two Strings

Input: Enter the first word: Hello

Enter the second word: World

Output: HelloWorld

4. Print the First and Last Character

Input: Enter a word: Python

Output: First character: P

Last character: n

Practical Exercises

5. Extract a Substring

Input: Enter the String: " Believe in Yourself!"

Output: The area is: Believe

6. Skip Characters in a String

Input: Enter the String: "hello, How are you??"

Output: The area is: How are you??"

Practical Exercises

7. Draw:

```
*  
**  
***  
****  
*****
```

8. Draw:

```
*  
  
***  
  
*****  
  
*****  
  
*****
```

9. Draw:

```
*  
***  
*****  
*****  
*****  
*****  
*****
```

```
*****  
*****  
***  
*
```

10. Draw:

```
*****  
* *  
* *  
* *  
*****
```

11. Draw:

```
* *  
* *  
* *  
*  
* *  
* *
```

12. Draw:

```
*  
* *  
* *  
* *  
* *  
* *  
* *  
* *  
* *  
*
```

13. Draw:
(W)

```
* * * *  
* * * * *  
* * * * *  
* * *  
* * *
```


Any Question?

References

- <https://www.w3schools.com/python/>
- <https://www.futurelearn.com/info/courses/introduction-to-programming-with-python-fourth-rev-/0/steps/264867>
- <https://www.tutorialspoint.com/python/>