IT Department

Arithmetic Expressions, Input/Output

Soma Solaiman Zadeh

Programming I (IT 117)

Fall 2024 - 2025

Week 2

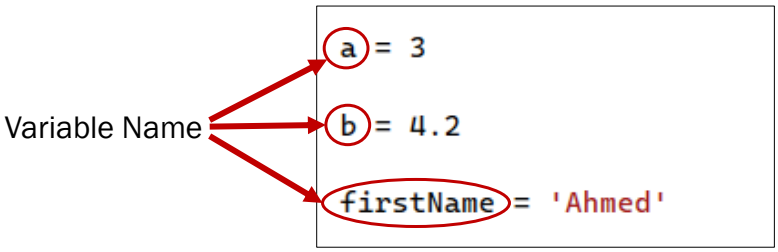
December 23, 2024

Outline

- **Variables** in Python
- Basic **Data Types** in Python
- Typecasting
- **Expressions**
- **Arithmetic Expressions** and **Arithmetic Operators**
- **Input/Output** in Python

Variables in Python

- Variables are containers where data values can be stored within the computer's memory.
- The **assignment** operator; **"="** symbol, is the operator that is used to assign values to variables in Python



Variable Names

- Rules for naming variables in Python:
  - A variable name must start with a **letter** or the **underscore(_)** character.
  - A variable name cannot start with a **number**.
  - A variable name can only contain **alpha-numeric** characters and **underscores** (A-Z, a-z, 0-9, and _).
  - A variable name cannot contain a **space** character.
  - Variable names are case-sensitive (Bruce, bruce and BRUCE are three different variables)
  - A variable name cannot be the same as a **Python keyword**.

Variable Names – Examples

- Which variable name is not correct?

❌ **3Name =** 'Omed Ali Hassan' → variable name cannot start with a number.

✅ **_myName =** 'Lana'

✅ **my_Name =** 'Lana'

✅ **Age** = 34

❌ **first name** = 'Kani' → variable name cannot contain space.

❌ **firstName&lastName** = 'Kani Asaad' → variable name cannot contain &.

❌ **True** = 4 → True is a Python keyword and is not allowed for variable name.

Basic Data Types in Python

- When you assign a value to a variable, Python knows the data type of the variable.
- Some basic data types in Python:

Data Type	Description	Example
int	Used for integer numbers.	639
float	Used for float numbers.	7.65
str	Used for string values, such as characters, words and text. (string values are surrounded by single or double quotes.)	"IT Department!" or 'IT Department!'
bool	Used for Boolean values.	True or False

Typecasting

- **Typecasting** is the process of converting one data type to another data type.
- Some typecasting methods in Python:

Typecasting Method	Description	Example
int( )	Converting a value type to an **integer** number.	**int**(5.7) → 5
float( )	Converting a value type to a **float** number.	**float**(6) → 6.0
Str( )	Converting a value type to a **string** value.	**str**(54) → '54'
bool( )	Converting a value type to a **boolean** value.	**bool**('Hi') → **True**

type( ) Method

- Whenever you want to know a variable's data type, you can use **type( )** method.
- Example:

a = 12

b = 9.25

c = "Hello Everyone!"

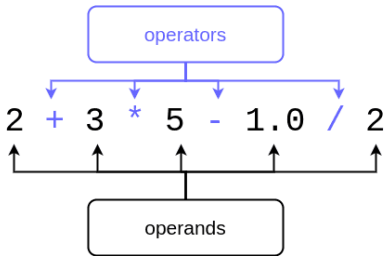
type(a) → int

type(b) → float

type(c) → str

Expression

- An expression in Python is a combination of operators and operands.



  - An expression has a value, which has a type.
  - Syntax for a simple expression → <operand> **<operator>** <operand>
  - Expression Example: 4*5+3

Arithmetic Expressions

- An **arithmetic expression** is built up using numbers, arithmetic operators (such as +, -, * and /) and parentheses, "(" and ")".
- Examples of Arithmetic Expressions:
  - 6+9-2
  - (15/2)*6+7
  - 50 - (5*2**3) / 10

Arithmetic Operators

Arithmetic Operation	Arithmetic Operator	Description	Example
Addition	+	It is used to add or sum two values.	**7 + 3 → 10**
Subtraction	-	It is used to subtract a value from another.	**7 – 3 → 4**
Multiplication	*	It returns the product of two numbers.	**4 * 2 → 8**
Exponentiation	**	The power operator raises the left operand to the power of the right operand.	**4 ** 2 → 16**
Division	/	It divides the first number by another and gets the result as a **floating-point** number.	**7 / 2 → 3.5** **8 / 2 → 4.0**
Floor Division	//	It divides the first number by another and **rounds down** the result to the nearest integer number.	**7 // 2 → 3**
Modulus (Remainder)	%	It returns the **remainder** of dividing the left operand by the right operand.	**7 % 3 → 1**

Arithmetic Operators Precedence

- **Operator precedence** is the order in which operations are performed in an arithmetic expression.

Precedence	Operator	Type
Highest ↓ **Lowest**	**( )**	Parentheses
	**	Exponentiation
	* / % //	Multiplication, division, modulus, floor division
	+ -	Addition, subtraction

Arithmetic Operators Precedence - Example

- In arithmetic expressions, the order of operations is as follows:
  1. Expressions between **parentheses**
  2. **Exponentiations**
  3. **Multiplication** and **Division** (from left to right)
  4. **Addition** and **Subtraction** (from left to right)

Example – What is the result of the following expression?

3 * 5 – 7 + 3 ** 2 – **(8 / 2)** → 3 * 5 – 7 + **3 ** 2** – 4

→ **3 * 5** – 7 + 9 – 4 → **15 – 7** + 9 – 4

→ **8 + 9** – 4 → **17 – 4** → **13**

Operators on Strings

- Generally, you cannot use arithmetic operators on strings.
- Specifically, the **+** operator and ***** operator work with strings, but not for addition and multiplication.

Operator	Description	Example	Example's Result
+	It is for concatenating or joining the strings. (Both operands must be strings.)	"IT" + "Department"	ITDepartment
		"IT" + " " + "Department"	IT Department
*	It is for repetition of the same string. (one operand must be a string and another one integer.)	"Python" * 3	PythonPythonPython

Input and Output in Python

- **Output** in Python
  - The **print( )** function is used to display output (text, variables, and expressions) on the console.
- **Input** in Python
  - The **input( )** function is used to take user input from the keyboard.
  - **input( )** function returns the user input ***in the form of a string***.

print( ) Function

- The **print()** function is used to display the output of your code.
- The Python **print()** function takes in any number of parameters in the parentheses ( ) and prints them out on one line of text.
- Put comma (**,**) between parameters, which are text, variables or calculation on variables.
- It is also possible to use some special characters such as single quote (‘), double quote (“), new line, tab and backslash (\) in print( ) function.

print( ) Function – Examples

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

Hello World!

```
firstname = 'Kate'
age = 20

print('My name is' , firstname)
```

My name is Kate

```
print('My name is' , firstname , 'and I was born in', 2025-age)
```

My name is Kate and I was born in 2005

Special Characters and Escape Character (\ )

- There are some special characters that might appear in the output text, such as new line, tab, backslash, single quote and double quote.
- We can use an escape character **** which is designed for these special characters.

Symbol	Description	Example	Output
\'	A single ‘	**print** (“I am from \'IT\' department”)	I am from ‘IT’ department
\”	A single “	**print** (“I am from \”IT\” department”)	I am from “IT” department
\\	A single \	**print** (“C:\\myFolder\\main.py”)	C:\myFolder\main.py
\t	Tab	**print** (“IT\tDepartment”)	IT Department
\n	A new line	**print** (‘IT\nDepartment’)	IT Department

print( ) Function – Examples

```
print("A\nB\nC")
```

```
A
B
C
```

```
print ("Arithmetic Operators\n*\t**\t/\t//")
```

```
Arithmetic Operators
*       **      /       //
```

```
print ('Welcome to \"TIU\"!\nIT Department\tFall Semester\t2024\\2025')
```

```
Welcome to "TIU"!
IT Department    Fall Semester    2024\2025
```

input( ) Function

- Whenever you want to ask a user to enter a value from the keyboard, you can use **input( )** function, and the entered value goes in a Python variable.

  firstname = **input(**"Please enter your first name: "**)**

- **input( )** function always returns a **string** value. Even if you enter a number, it is considered as string value.
  - So, if any user wants to take input as **int** or **float**, we need to typecast it (convert its data type).

input( ) Function – Example

- The parameter to **input( )** is a message that prints out, letting the user know what they are supposed to enter.

```
firstname = input("Please enter your first name: ")

age = int(input('Please enter your age'))

print('My name is',firstname,'and I am',age,'years old!')
```

- In this example, first the message **"Please enter your first name: "** is shown to the user and waiting for user to enter their first name. then user press enter on keyboard and the second message is shown to user asking for entering their age. Finally, according to user inputs, the output is displayed.

Comments in Python

- When you write a program, it is recommended to use **comments** to explain each part of your code to make it more readable.
- Comments are lines of the code that are not executed by the interpreter.
- You can add a hashtag **#** at the beginning of a line to make it a comment.

```
# This code is for calculating sum of two numbers
a = 6
b = 10
result = a + b
print("Sum of", a, "and", b, "is equal to", result)
```