



MODERN DIGITAL SKILLS
CHAPTER 7:
INTRODUCTION TO PYTHON PROGRAMMING LANGUAGE

© All Rights Reserved. The University Of Jordan

1

- What is Python
- Start using Python
- Defining Variables
- Data Types in Python
- Statements in Python
 - 1. Assignment Statement
 - 2. Print Statement
 - 3. Input Statement
 - 4. Comment statement
 - Expressions
 - 1. Arithmetic expressions
 - 2. Relational expressions
 - 3. Logical expressions
- Operations on Strings

CONTENTS



What is Python?



- **Python** is a high-level, interpreted programming language known for its simplicity and readability.
- Python was created by **Guido van Rossum** in the late 1980s and was officially released in **1991**. and it was named after the British comedy show "**Monty Python's Flying Circus**," not the snake! 🐍
- Python is an **open source: this means that** it is free to use and has a large community supporting its development.

Start using Python?



Option 1: Install Python

- Download Python from python.org.

Option 2: Use Python Online

- Go to online editors like
 - <https://www.programiz.com/python-programming/online-compiler/>
 - or <https://www.online-python.com/>
 - or https://www.onlinegdb.com/online_python_compiler

Let's start programming

Defining variables

Variable: is a label for a location in memory used to hold a value which can be changed during program execution

Example:

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

Output

5

- **Note : Print statement is used to produce output on the screen**

5

© All Rights Reserved The University Of Jordan

Let's start programming

Defining variables

Note that: variable names are **case-sensitive**.

Example

In this Python code, there are two different variables a, A

```
A = 10
a = "Maha"
print (A)
print(a)
```

```
A = 10
a = "Maha"
print(A)
print(a)
```

Output

10
Maha

© All Rights Reserved The University Of Jordan

6

Let's start programming

Data Types in Python

- **Data Types** used to define the type of a **variable**. It defines what type of data we want to store in a variable.

The most common data types in Python are:

1. Strings
2. Integers
3. Floating-point
4. Boolean

Let's start programming

Data Types in Python: *Strings*

- **String** is series of characters .

- **Characters** can be:

1. **Letters**: a.. z , A..Z
2. **Digits**: 0,1,2,3,4,5,6,7,8,9
3. **Symbols**: @,\$,*,.,-,#,.....

“jordan” “JORDAN” “Jordan”
 “189” “564342” “9023”
 “@#\$” “@@@” “\$\$\$”

- Strings in Python are surrounded by:
 - *Single quotation*: (')
 - *Double quotation marks*(")

Example:

- "Amman-Jordan, street No. 101" → String
- 'Hello' → String
- "Hello" → Undetermined

Let's start programming

Data Types in Python: *Integers*

Integers Represent complete integer values (positives or negatives)

Examples: the following are integers:

12 , 7 , -10 , -55, 16

- 1.5 not an integer
- "12" not an Integer

9

© All Rights Reserved The University Of Jordan

Let's start programming

Data Types in Python: *Floating numbers*

Floating numbers are used to store real numbers using **Decimal Points** or **Exponential Notations**

Examples: The following are Floating point numbers

Decimal Points

45.5 -67.89 144.96 0.3 .3 3.0

Exponential Notations

7.8×10^2 9.567×10^{-4}

:

10

© All Rights Reserved The University Of Jordan

Let's start programming

Data Types in Python: Boolean

Boolean is used to store Boolean values which can be either **True** or **False**.

- **For example,**

- The result of comparing two numbers is a Boolean value

```
2>3
False
```

```
4 >= 2+2
True
```

11

© All Rights Reserved The University Of Jordan

Let's start programming

Statements in Python



A **statement** in Python is a command that is written by the programmer to tell the computer to do something, like **input**, **processing**, **storing**, **output**.

Some statements in python	
Statement	Purpose
1. Assignment statement	<i>Process and store data</i>
2. Input statement	<i>Input data</i>
3. Print statement	<i>Output information</i>
4. Comment statement	<i>Make the code more readable</i>

12

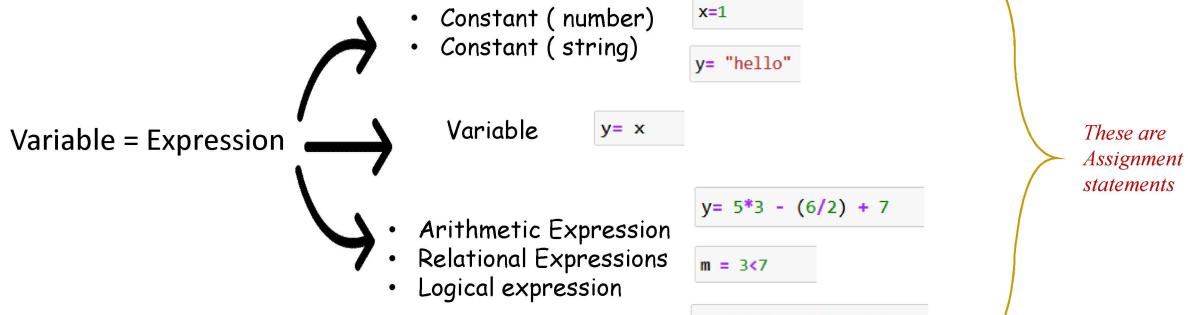
© All Rights Reserved The University Of Jordan

Let's start programming

1. Assignment statement

Assignment statement is a statement used to assign a value into a variable

General form:



© All Rights Reserved The University Of Jordan

13

Let's start programming

1. Assignment statement

Exercise

Which of the following is a Correct assignment statement?

Statement	Assignment statement?
<code>y = x + 3</code>	Yes
<code>x + 1 = y + 2</code>	No
<code>y + 2 = 5</code>	No
<code>"nice" = k</code>	No
<code>k = "nice"</code>	Yes
<code>W = True</code>	Yes

Let's start programming

1. Assignment Statements: *Increment and Decrement*

- **Increment** Increasing a variable's value by a fixed amount
- **Decrement** Decreasing a variable's value by a fixed amount

Examples: Write the assignment statement needed for each case in the following:

1. <i>Increment x by 3 :</i>	$x=x+3$	<i>or</i>	$x+=3$
2. <i>Increment y by 7:</i>	$y= y+7$	<i>or</i>	$y+=7$
3. <i>Decrement x by 4:</i>	$x= x-4$	<i>or</i>	$x-=4$
4. <i>Decrement y by 1:</i>	$y=y-1$	<i>or</i>	$y-=1$

15

© All Rights Reserved The University Of Jordan

Let's start programming

1. Assignment Statements: *Exercises*

Exercise 1: Determine the data type of the variable X in each of the following examples?

Example	Data type
$X = "Good Job"$	<i>String</i>
$X= 5$	<i>Integer</i>
$X= True$	<i>Boolean</i>
$X = 2.3$	<i>Floating - point</i>

16

© All Rights Reserved The University Of Jordan

Let's start programming

1. Assignment Statements: *Exercises*

Exercise 2:

What is the Data type of variable W in each assignment statement in the following?

Statement	Data type of W
W = "True"	String
W = True	Boolean
W = 5	Integer
W = 5.25	Floating number
W = 6+2	Integer
W = 6 / 4	Floating number
W = "123"	String
W = 123	Integer

Let's start programming

1. Assignment Statements: *Exercises*

Exercise 3: Find the output of the following Python code?

```

X = 20
Y = 30
X=X+1
Y=Y+2
X = X+Y
print (X)
print (Y)

```

Solution:

53
32

Exercise 4: Which assignment statement will store the value of x into y

- a) y = x
- b) x = y

Solution:

(a) is the correct answer

Let's start programming

1. Assignment Statements: *Exercises*

Exercise 5: answer each of the following:

- Define a variable count and assign the integer value 5 to the label
`count = 5`
- Define a variable name and assign the string value Mary to the label
`name = "Mary"`
- Define a variable num and assign the floating-point value 1.5 to the label
`num=1.5`
- Define a variable test and assign the Boolean value True to the label
`test = True`

19

© All Rights Reserved The University Of Jordan

Let's start programming

2. Print (Output) statements



- **Print statements** are used to produce output on the screen

Different ways to use:

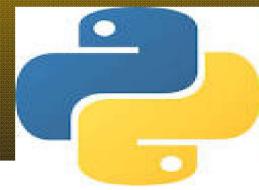
- `print ("Message")`
- `print (expression)`
- `print (item1, item2, ... , itemn)`
- `print ()` #prints a blank line

© All Rights Reserved The University Of Jordan

20

Let's start programming

2. Print (Output) statements



Example: Find the output of this Python code:

```
print("Good Morning")
print(45-10)
print()
print (5, "ok", -1)
```

```
print("Good Morning")
print(45-10)
print()
print (5, "ok", -1)
```

Good Morning
35
5 ok -1

Output →

Good Morning
35
5 ok -1

© All Rights Reserved The University Of Jordan

21

Let's start programming

3. Input statement

Input statement is a statement that reads input as a string by default.

General form:

```
variableName = Input("text")
```

Examples

- **Example1: input a text (string)**
name = input("Enter your name:")
print (name)
- **Example2: input an integer number use the function int**
grade = int(input("Enter your grade:"))
print (grade)
- **Example3: input a floating-point number use the function float**
salary = float(input("Enter your salary:"))
print (salary)

```
: name = input("Enter your name:")
grade = int(input("Enter your grade:"))
salary = float(input("Enter your salary:"))

print (name)
print (grade)
print (salary)
```

Enter your name:Ahmad
Enter your grade:85
Enter your salary:500
Ahmad
85
500.0

Let's start programming

4. Comment statement

Comment statements are used to make the python code more readable.

Comments are ignored by the python compiler during execution.

Use `#` symbol to write a comment.

Examples: Find the output of this Python code:

Example1:

```
# This is a comment
print("Welcome")
```

Output: Welcome

```
# This is a comment
print("Welcome")
```

Welcome

Example2:

```
print(500) # Good salary
```

Output: 500

```
print(500) # Good salary
```

500

3

Let's start programming

Expressions (Operations) in Assignment statements

➤ **Expressions (Operations) in Python are :**

- 1. Arithmetic Expression
- 2. Relational Expressions
- 3. Logical expression

Let's start programming

Expressions in Assignment statements: **Arithmetic Expression**

- **Arithmetic expression:** a calculation using numbers and math symbols (+, -, *, /) that results in a **numeric value** (integer or float).
 - **Example:** $5 + 3 * 2 \rightarrow 11$ (integer).

Operation	Operator
<i>Unary minus</i>	-
<i>Exponentiation</i>	**
<i>Multiplication</i>	*
<i>Division</i>	/
<i>Remainder</i>	%
<i>Addition</i>	+
<i>Subtraction</i>	-

25

© All Rights Reserved The University Of Jordan

Let's start programming

Expressions in Assignment statements: **Arithmetic Expression**

- **Multiplication and Division**

Expression	Result
$3 * 2$	6
$-3 * 2$	-6
$3 * -2$	-6
$-3 * -2$	6
$4 / 2$	2.0
$-4 / -2$	2.0
$-4 / 2$	-2.0
$4 / -2$	-2.0
$5 / 2$	2.5

26

© All Rights Reserved The University Of Jordan

Let's start programming

Expressions in Assignment statements: **Arithmetic Expression**

- **Remainder (Modulus) %**

Expression	Result
5 % 2	1
14 % 5	4
15 % 4	3
6 % 2	0
4 % 4	0
7 % 1	0
2 % 6	2
15 % 100	15

27

© All Rights Reserved The University Of Jordan

Let's start programming

Expressions in Assignment statements: **Arithmetic Expression**

- **Exponentiation ****

Expression	Result
2**2	4
2**3	8
3**2	9
6**1	6
1**6	1
2**0	1
-2**2	-4
-2**3	-8
2**-2	0.25
2**-1	0.5

28

© All Rights Reserved The University Of Jordan

Let's start programming

Expressions in Assignment statements: **Arithmetic Expression**

- **Arithmetic expression priorities** are the precedence rules of operators

- 1) ()
- 2) **
- 3) - unary minus
- 4) * , / , % left to right
- 5) + , - left to right

Ex1:

$$\begin{aligned} & \underline{10/2} * 3 \% 5 \\ & \underline{5} * 3 \% 5 \\ & \underline{15} \% 5 \\ & 0 \end{aligned}$$

Ex2:

$$\begin{aligned} & \underline{7-2}+1 \\ & \underline{5}+1 \\ & 6 \end{aligned}$$

29

© All Rights Reserved The University Of Jordan

Let's start programming

Expressions in Assignment statements: **Arithmetic Expression**

- **Exercises**

Ex1.

$$\begin{aligned} A &= 2 * 3 + 6 ** 2 - (\underline{4 + 2}) * 2 \\ A &= 2 * 3 + \underline{6 ** 2} - 6 * 2 \\ A &= \underline{2 * 3} + 36 - 6 * 2 \\ A &= \underline{6 + 36} - 12 \\ A &= \underline{42} - 12 \\ A &= 30 \end{aligned}$$

30

© All Rights Reserved The University Of Jordan

Let's start programming

Expressions in Assignment statements: **Arithmetic Expression**

- **Exercises**

Ex4.

$$Z = (\underline{2 + 1}) ^\star\star ((\underline{4 + 8}) / 4) + 1$$

$$\begin{array}{c} 3 \\ (\underline{12}) \\ (\underline{12 / 4}) \\ 3 \end{array}$$

$$Z = 3 ^\star\star 3 + 1$$

$$Z = \underline{3 ^\star\star 3} + 1$$

$$Z = \underline{27} + 1$$

$$Z = 28$$

33

© All Rights Reserved The University Of Jordan

Let's start programming

Expressions in Assignment statements: **Relational Expression**

- **Relational expression:** a comparison between values using symbols like $>$, $<$, $==$, $!=$ to check relationships that results in a **Boolean value** .
- **Example:** $5 > 3 \rightarrow$ (True).

Operators	Meaning	Example	Result
$<$	Less than	$5 < 2$	FALSE
$>$	Greater than	$5 > 2$	TRUE
\leq	Less than or equal to	$5 \leq 2$	FALSE
\geq	Greater than or equal to	$5 \geq 2$	TRUE
$==$	Equal to	$5 == 2$	FALSE
$!=$	Not equal to	$5 != 2$	TRUE

© All Rights Reserved The University Of Jordan

Relational Operators Priorities: All Operators have the same Priorities

Let's start programming

Expressions in Assignment statements: **Logical Expression**

- **Logical expression:** a statement that uses logical operators **And**, **Or**, **Not** to combine conditions to make decisions.
- **Example:** $(5 > 3) \text{ and } (2 < 4) \rightarrow (\text{True})$.

- **Logical operators:** And , Or , Not
- **Logical Operators priorities:**
 1. Not (highest)
 2. And
 3. Or (lowest)

Let's start programming

Expressions in Assignment statements: **Logical Expression**

- **Logical Truth Tables**

X	Y	X and Y
True	True	True
True	False	False
False	True	False
False	False	False

X	Y	X or Y
True	True	True
True	False	True
False	True	True
False	False	False

X	Not X
True	False
False	True

Let's start programming

Expressions in Assignment statements: **Expressions**

- **Priorities for all Operators**

1. Start with **Arithmetic operators**
2. Then by **Relational Operators**
3. Finally: **Logical Operators**

1. ()
2. **
3. - unary minus
4. * / %
5. + -
6. <, >, <=, >=, ==, !=
7. not
8. and
9. or

Let's start programming

Expressions in Assignment statements: **Logical Expression**

- **Exercises**

Ex1.

(3<=10) and (6 != 6)
True and False
False

Ex2.

(3<=10) or (6 != 6)
True or False
True

Ex3.

not (3<=4)
not True
False

Ex4.

not(3!=3)
not False
True

Let's start programming

Expressions in Assignment statements: **Expressions**

- **Exercises**

Ex1. Find the value of X

$X = 2 + 3 > 1 \text{ and } 5 \% 2 != 9 / 3 \text{ or } 5 + 4 == 8$

$X = 5 > 1 \text{ and } 1 != 3 \text{ or } 9 == 8$

$X = \text{True and True or False}$

$X = \text{True or False}$

$X = \text{True}$

Recall:

1. Start with **Arithmetic operators**
2. Then by **Relational Operators**
3. Finally: **Logical Operators**

```
X = 2 + 3 > 1 and 5 % 2 != 9 / 3 or 5 + 4 == 8
print(X)
```

True

39

Let's start programming

Expressions in Assignment statements: **Expressions**

- **Exercises**

Ex2. Find the value of Z, given that X = 1, and Y = 5

$Z = X <= Y-4 \text{ and not } X*5 != Y \text{ or } X+2 < (X+Y)/2$

$Z = 1 <= 5-4 \text{ and not } 1*5 != 5 \text{ or } 1+2 < (1+5)/2$

$Z = 1 <= 1 \text{ and not } 5 != 5 \text{ or } 3 < 3$

$Z = \text{True and not False or False}$

$Z = \text{True and True or False}$

$Z = \text{True or False}$

$Z = \text{True}$

Recall:

1. Start with **Arithmetic operators**
2. Then by **Relational Operators**
3. Finally: **Logical Operators**

```
X=1
Y=5
Z = X <= Y-4 and not X*5 != Y or X+2 < (X+Y)/2
print(Z)
```

True

Let's start programming

Expressions in Assignment statements: **Expressions**

- **Exercises**

Ex3. Find the value of Y

$Y = 3 >= 9 \text{ or } (5+1) != 2 \text{ and not } ((7-3) < 2 \text{ or not False})$

6

$(4 < 2 \text{ or not False})$
 $(\text{False or not False})$
 (False or True)

True

$Y = 3 >= 9 \text{ or } 6 != 2 \text{ and not True}$

$Y = \text{False or True and not True}$

$Y = \text{False or True and False}$

$Y = \text{False or False}$

$Y = \text{False}$

Recall:

1. Start with **Arithmetic operators**
2. Then by **Relational Operators**
3. Finally: **Logical Operators**

```
Y = 3 >= 9 or (5+1) != 2 and not ((7-3) < 2 or not False)
print(Y)
```

False



Tracing examples

What is the output of the following Python codes:

Python Code	Output
<pre>x = 7 y = 3 z = (x % y == 1) or (x + y > 2) and not (x != y**2) print(z)</pre>	
<pre>a = 4 + 3 * 2 b = (4 + 3) * 2 c = a + b/2 - ((a*2+5) ** (0.5)) print(a, b, c)</pre>	



Tracing examples/Answers

What is the output of the following Python codes:

Python Code	Memory	Output
<pre>x = 7 y = 3 z = (x % y == 1) or (x + y > 2) and not (x != y**2) print(z)</pre>		True
<pre>a = 4 + 3 * 2 b = (4 + 3) * 2 c = a + b/2 - ((a*2+5) ** (0.5)) print(a, b, c)</pre>		10 14 12.0

© All Rights Reserved The University Of Jordan

43