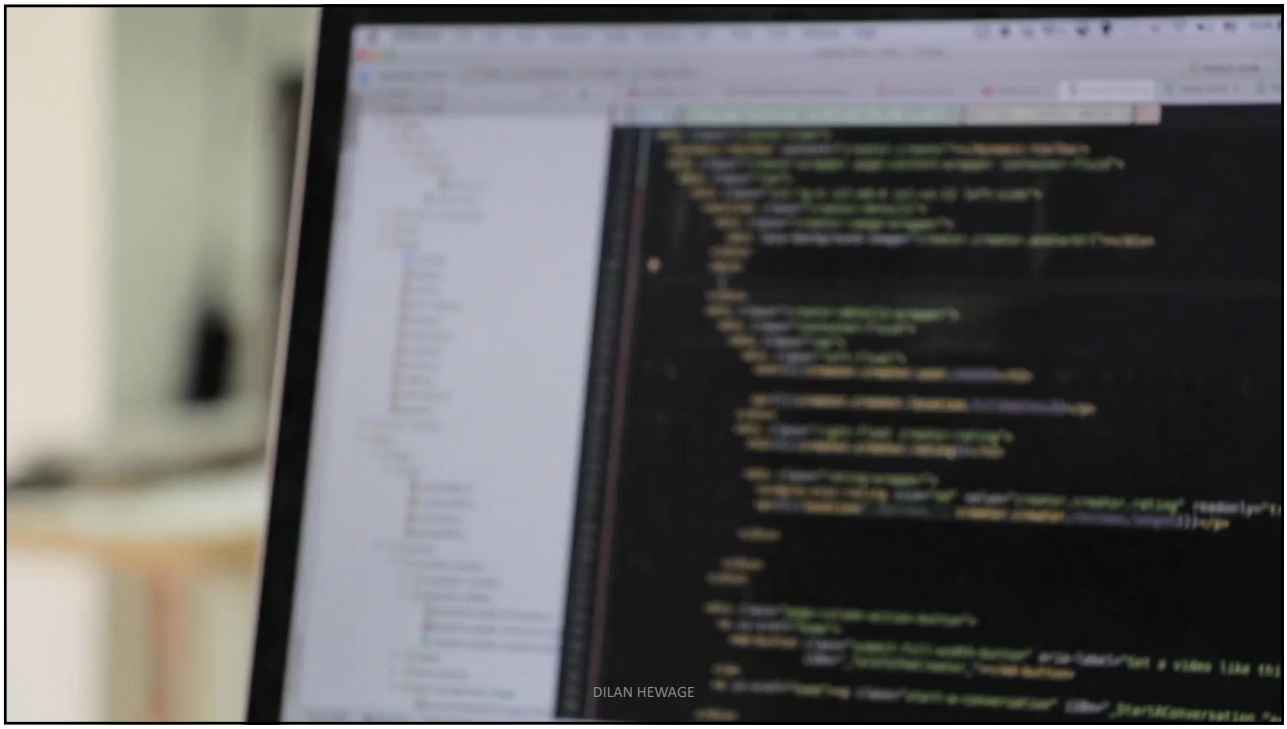



1

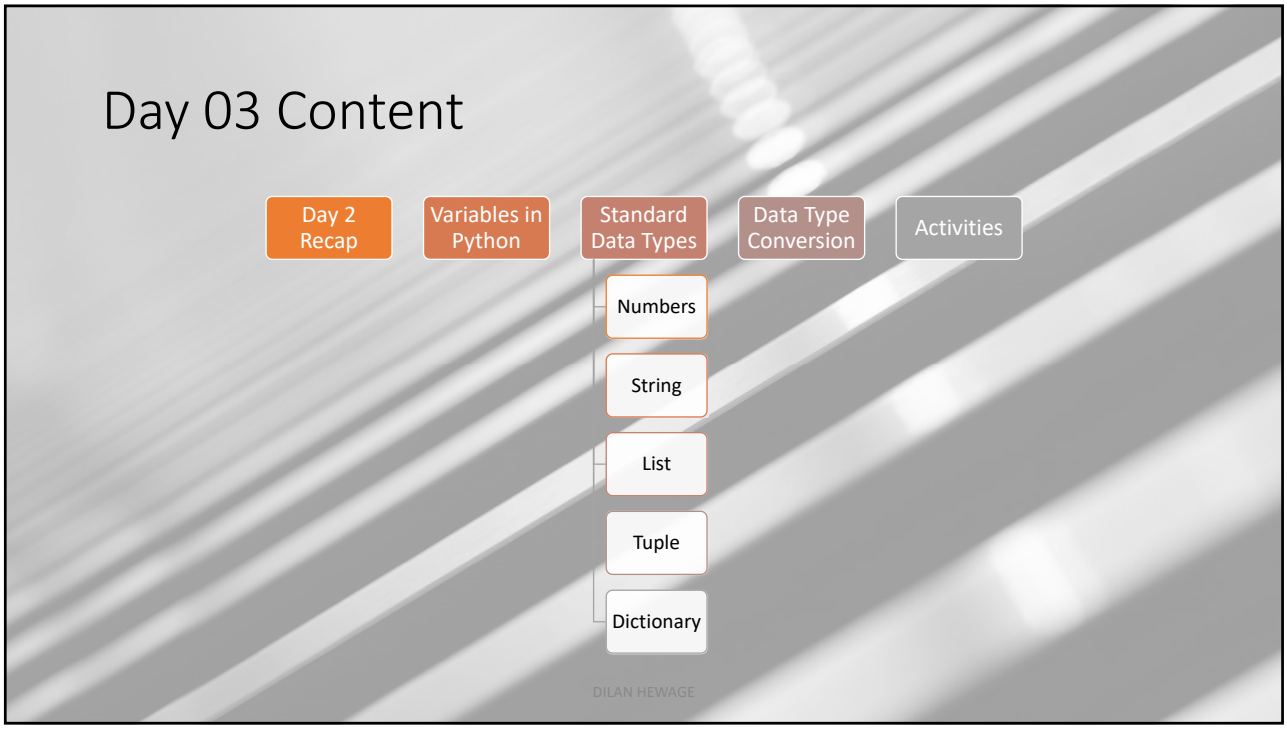
ඇල්ගොරිදම සංවර්ධනය සහ පයිතන් ක්‍රමලේඛනය

# Algorithm Development & Python Programming

Workshop Day - 3

DILAN HEWAGE
BIT – University of Colombo / CIDM – APIDM
Founder of WiTeds Solutions
Co-Founder of Print Arcade

2


Day 03 Content
Day 2 Recap
Variables in Python
Standard Data Types
Data Type Conversion
Activities
Numbers
String
List
Tuple
Dictionary
DILAN HEWAGE

3


Day 2 Recap
2 දින සාරාංශය
DILAN HEWAGE

4

# Variables in Python

# පයිතන් විචල්‍යන්

DILAN HEWAGE

5

## Variables in Python

**A variable :** Unlike other programming languages, Python has no command for declaring a variable.

- can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).
- A variable is created the moment you first assign a value to it
- 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, 0-9, and _ )
- Variable names are case-sensitive (age, Age and AGE are three different variables)

- විචල්‍යයක්: වෙනත් ක්‍රමලේඛන භාෂාවන් මෙන් නොව, විචල්‍යයක් ප්‍රකාශ කිරීම සඳහා පයිතන්ට විධානයක් නොමැත.
- කෙටි නමක් (x සහ y වැනි) හෝ වඩාත් විස්තරාත්මක නමක් (age, carname, total_volume) තිබිය හැක.
- ඔබ මුලින් එයට වටිනාකමක් ලබා දුන් මොහොතේ සිට විචල්‍යයක් නිර්මාණය වේ
- විචල්ය නාමයක් අකුරකින් හෝ අවධාරනය කළ අක්ෂරයකින් ආරම්භ විය යුතුය
- **විචල්ය නාමයකට අංකයකින් ආරම්භ කළ නොහැක**
- විචල්‍ය නාමයකට අඩංගු විය හැක්කේ ඇල්ෆා සංඛ්‍යාත්මක අක්ෂර සහ යටි ඉරි (A-z, 0-9, සහ _)
- විචල්‍ය නම් සිද්ධි සංවේදී වේ (age, Age සහ AGE වෙනස් විචල්‍ය තුනක් වේ)

DILAN HEWAGE

6

## Variables in Python Contd.

- Examples

```
#!/usr/bin/python

counter = 100          # An integer assignment
miles   = 1000.0       # A floating point
name    = "John"       # A string

print counter
print miles
print name
```

DILAN HEWAGE

7

## Variables in Python Contd.

- Multiple Assignment

```
a = b = c = 1
```

- Python allows you to assign a single value to several variables simultaneously.
- an integer object is created with the value 1, and all three variables are assigned to the same memory location
- you can also assign multiple objects to multiple variables.

```
a,b,c = 1,2,"john"
```

- Here, two integer objects with values 1 and 2 are assigned to variables a and b respectively, and one string object with the value "john" is assigned to the variable c.

DILAN HEWAGE

8

# Assigning Variables in Python

- Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.
- The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable

```
counter = 100          # An integer assignment
miles   = 1000.0       # A floating point
name    = "John"       # A string

print (counter)
print (miles)
print (name)
```

DILAN HEWAGE

9

# Multiple Variable Assignment

- Python allows you to assign a single value to several variables simultaneously.

```
a = b = c = 1
```

Here, an integer object is created with the value 1, and all the three variables are assigned to the same memory location. You can also assign multiple objects to multiple variables. For example –

```
a, b, c = 1, 2, "john"
```

Here, two integer objects with values 1 and 2 are assigned to the variables a and b respectively, and one string object with the value "john" is assigned to the variable c.

DILAN HEWAGE

10

# Standard Data Types

## සම්මත දත්ත ප්‍රරූප

DILAN HEWAGE

11

## Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

- Numbers
- Strings
- List
- Tuple
- Dictionary



12

# Python Numbers

- Python supports three different numerical types –
- int (signed integers)
- float (floating point real values)
- complex (complex numbers)
- All integers in Python3 are represented as long integers. Hence, there is no separate number type as long

Number data types store numeric values. Number objects are created when you assign a value to them. For example –

```
var1 = 1
var2 = 10
```

You can also delete the reference to a number object by using the **del** statement. The syntax of the **del** statement is –

```
del var1[,var2[,var3[....,varN]]]
```

You can delete a single object or multiple objects by using the **del** statement.

For example –

```
del var
del var_a, var_b
```

| int | float | complex |
|---|---|---|
| 10 | 0.0 | 3.14j |
| 100 | 15.20 | 45.j |
| -786 | -21.9 | 9.322e-36j |

DILAN HEWAGE

13

# Python Strings

- Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows either pair of single or double quotes. 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 to the end.
- The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator

```
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
```

This will produce the following result –

```
Hello World!
H
llo
llo World!
Hello World!Hello World!
Hello World!TEST
```

DILAN HEWAGE

14

# Python Lists

- Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One of the differences between them is that all the items belonging to a list can be of different data type.
- The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

```
list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
tinylist = [123, 'john']

print (list)          # Prints complete list
print (list[0])       # Prints first element of the list
print (list[1:3])     # Prints elements starting from 2nd till 3rd
print (list[2:])      # Prints elements starting from 3rd element
print (tinylist * 2)  # Prints list two times
print (list + tinylist) # Prints concatenated lists
```

This produces the following result –

```
['abcd', 786, 2.23, 'john', 70.200000000000003]
abcd
[786, 2.23]
[2.23, 'john', 70.200000000000003]
[123, 'john', 123, 'john']
['abcd', 786, 2.23, 'john', 70.200000000000003, 123, 'john']
```

DILAN HEWAGE

15

# Python Dictionary

- Python's dictionaries are kind of hash-table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.
- Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([])

```
dict = {}
dict['one'] = "This is one"
dict[2]     = "This is two"

tinydict = {'name': 'john','code':6734, 'dept': 'sales'}

print (dict['one'])       # Prints value for 'one' key
print (dict[2])           # Prints value for 2 key
print (tinydict)          # Prints complete dictionary
print (tinydict.keys())   # Prints all the keys
print (tinydict.values()) # Prints all the values
```

This produces the following result –

```
This is one
This is two
{'name': 'john', 'dept': 'sales', 'code': 6734}
dict_keys(['name', 'dept', 'code'])
dict_values(['john', 'sales', 6734])
```

DILAN HEWAGE

16

# Data Type Conversion

## දත්ත ප්‍රරූප පරිවර්තනය

DILAN HEWAGE

17

# End of the Day 3

Have you learned something?
See you all again on day 3

18