# 01 Python Basics

### Plan for this Lecture 1. Introduction to Python and first program (Output)

COM4008-Programming-Concepts / 01 Python Basics

- 2. Input

## 3. Variables, Types, and Values

1.1. What is Python?

express concepts in a few lines of code." • Guido Van Roosum

"Python is a high level programming language, and its core design philosophy is all about code readability and a syntax which allows programmers to

1.2. History of Python

• Guido named the language after UK comedy series Monty Python.

Python was originally developed by Dutch programmer Guido van Rossum throughout the 1980s.

 Python 0.9.0 was released in 1991 Python 2 was released in 2000 Python 3.0 was released in 2008 Python has gained in popularity thanks data science and national school curriculums.

Beautiful is better than ugly.

 Complex is better than complicated. Readability counts.

1.4. Advantages of Python

• **Readable**: intuitive and strict syntax • **Productive**: saves a lot of code

• Web Scraping: e.g. Scrapy

• Computer Vision: e.g. OpenCV Machine Learning: Sci-kit Learn, NLTK, TensorFlow

In [ ]: print("Hello Nick") print("Your ID is 12345678")

• We can store data (value) by assigning it (=) to a memory location. That memory location is then referred to by a variable name.

**15** 

2.2 Variables in C / Java

A=15;

String name = "Nick" 2.3 Variables in Python

• We can store data (value) by assigning it = to a memory location. That memory location is then referred to by a variable name.

• Remember that the = sign in programming is different to mathematics (which compares two sides of the equation)!

2.4 Output of values held in variables In [ ]: name = "Nick" print('Hello', name) In [ ]: name = "Nick" print("Hello " + name) In [ ]: name = "Nick" print("Hello " + name) In Jupyter you can also just state the variable name to reveal the contents:

Variable name

Assignment

In [ ]: name = "Nick" name

In [ ]: name = "Nick"

In [ ]: name = "Nick"

In [ ]: age = 30

print("You are", age)

age = "Nick"

name = "Nick"

In []: age = int(30)

print(age)

print(age)

str2 = 30

In [ ]: str1 = "Nick is" str2 = 30

str3 = "years old"

str3 = "years old"

print(str1, str2, str3)

print(str1 + str2 + str3)

age = float(30)

print(type(name))

first\_mark + second\_mark

2.11 Case Sensitivity

What will the output be of the below:

age = 30

age = 30

### 2.8 C-format specifiers • The C programming language was first developed in the 1970s by Dennis Ritchie at Bell Labs.

• Python is dynamically typed (type checked at run-time).

Python can also utilise these 'format specifiers' for outputing data.

print(f"My name is {name} and I am {age} years old.")

2.9 Dynamic vs Static typing

• Python variables can be initially assigned a value of one type, and then be reassigned a value of a different type!

• C is a strongly typed language and required the format of the data to be included in the 'output stream'.

%u

%f

%e

%E

%g

**Syntax Description** 

**Unsigned Integer** 

Single character

Floating point as m.ddddd

Floating point as m.ddddde+/-xx

Floating point as m.dddddE+/-xx

%d, %i Integer

- print("You are", age) In [ ]: type(age) In [ ]: age = 30 print(type(age))
- In [ ]: age = 30 print(age) Age = 35print(Age) **Quiz Questions:**

#### 3. Input! In [ ]: input("Please enter your name") In [ ]: name = input("Please enter your name: ") print("Hello", name)

print(type(name))

In [25]: # age = 18 # print("You are", age, "years old") In [ ]: age = 18 #declare age variable

# for single line comments

Exercise 2: Now declare a variable called name. Assign your firstname as the value for this variable. Print this value to the screen to check that it has been assigned to the variable.

In [ ]: # Write your solution here. Exericse 5: Now use a function in Python to ask the user to enter their name. This name should be stored in an appropriate variable. Once stored, print the value of this variable (what the user entered) to the screen.

Extension: Also ask the user to enter the same number of feet and check this converts back to the original number of miles. In [ ]: # Write your solution here. Exercise 9:

For reference, there are 5280 feet in 1.0 mile.

For simplicity, it would be easier if they enter this in the format of an integer (int). For example, May would be the 5th month of the year. Challenge: Instead of asking for integers, ask the user to enter their month of birth as a string (str)! In [ ]: # Write your solution here.

Extension: Check whether this is a leap year, and if it is, remember to make an adjustment. In [ ]: # Write your solution here. (Bonus) Exercise 13:

(Bonus) Exercise 15:

nbviewer version: 8b013f7 nbconvert version: 7.2.3 Rendered 8 minutes ago

(Bonus) Exercise 11:

output this to the screen.

(Bonus) Exercise 14: Define a function called 'get\_input'. The function should prompt the user to enter an input and then return this to the calling location. Test this works by calling the

Extension 1: You may wish to extend this further to create a separate function which will 'get' (return) integers. This would be a separate function from the 'get\_input' function, which would by default return strings (str). Extension 2: You may wish to incorporate this function into the scripts which process the user's DOB.

to the program's specification. In [ ]: # Write your solution here. Delivered by Fastly, Rendered by OVHcloud

1.3. Philosophy of Python First posted in 1999, and later published in the Zen of Python:

• Explicit is better than implicit. • Simple is better than complex.

• Portable: for every operating system • **Reloaded**: it comes with many libraries 1.5. Uses of Python • Web Development: Frameworks such as Django and Flask • Data Analysis: Libraries such as NumPy, Pandas, Matplotlib • Internet of Things: Raspberry Pi + Python

• Game Development: PyGame 1.6 Output in Python (first program!) In [ ]: print("Hello World") 2.1 Variables • Variables are a way to store data temporarily in our programs. • Just like the word variable implies, this data can vary; it can change throughout the course of the program.

• The value is stored at a memory address (RAM, which is temporary memory) and can then be accessed through the variable name.

• The opposite of a value changing throughout the program would be a constant, which never changes; it has a fixed value.

• Python does not require the type to be stated! Although it is good to be aware of types... • It is good practice to consider the name of your variable (storage area) and of course what value you will store there: name = "Nick"

• In C, C++, C# and Java, the programmer would have to first define the data type, then a meaningful name for the variable:

# 2.5 Multiple variables print("Hello", name, "you are", age) 2.7 f-strings (Formatted String Literals) • Introduced in Python 3.6, f-strings provide an efficient and readable way to embed expressions inside string literals. • You prefix the string with f and insert variables or expressions inside curly braces { }. • This method is more widely known as **string interpolation** which embeds expressions or variable values inside a string.

String, or any object that be converted via str() Insert a literal % character In [ ]: print("%s is %d years old" % (name, age) )

Use %e for exponents <-4 or >+5, otherwise use %f

2.10 Type Casting In [ ]: first\_mark = 60 second\_mark = "40" first\_mark + second\_mark In [ ]: first\_mark = 60 second\_mark = int("40")

• Statically typed languages (such as C, C++, C# and Java, which check type at compile-time) only allow values of the same type to be assigned.

In [ ]: first\_mark = "60" second\_mark = "40" first\_mark + second\_mark In [ ]: str1 = "Nick is"

Use docstrings carefully: Docstrings (triple-quoted strings) are not true comments but can be used for multi-line commenting in some cases. Just ensure it doesn't

This Jupyter Notebook contains exercises for you to practice the basics of Python. You'll see a mixture of markdown cells and interactive code cells. Attempt the following exercises, which slowly build in complexity. If you get stuck, check back to the introductory Python lecture recording here or view the W3Schools page on Python Variables, which includes examples, exercises and quizzes to help your

Now declare two variables. Assign the value 10 to the first variable. Then assign the value 40 to the second variable. Then add the variables together and print out

Now ask the user to enter two numbers, and store these inputs in two variables. Once stored, print out the average of the two numbers.

Design a program that asks the user to enter values for the length and height of a rectangle and then outputs the perimeter and area of the rectangle to the screen.

Write a program that will calculate the how much interest will paid on money invested in a savings account. Select an appropriate interest rate (based on the current

markets at time of reading) and consider how you will represent this in your program. The user should enter the deposit amount and the program should display how

Further to the previous exercise, write some code that asks the user to enter their date of birth (DOB). Now calculate how old the user is from their date of birth and

Continuing from the previous exercise, can you add conditional statements that will evaluate whether the input is 'valid'. For example, if the user was asked to type

Extension: Think about the appropriate formatting for currency. For example, consider how to print the output with two decimal places for pound sterling.

Ask the user to enter the month that they were born in. Calculate how many months it will be until their next birthday (from today's date).

Hint: If you get stuck, first check which type of data is being returned from the Python function.

Extension: ask the user to also type in a unit: cm, m, inches etc. Include this unit in the output.

number of feet to the screen. Test this for a number of inputs to check the conversion works correctly.

#### In [ ]: age = input("Please enter your age: ") print("You are", age, "years old") print("the value", age, "is", type(age)) In [ ]: age = int(input("Please enter your age: ")) print("You are", age, "years old")

print("the value", age, "is", type(age))

They are known as docstrings in Python.

• Code comments are useful for documentation purposes.

for block summary comments (three quotation marks)

They can also be used for beginners to test and debug their code.

**Comments in Python** 

In [ ]: print("the value", name, "is", type(name))

print("You are", age, "years old") In [ ]: '''print("hello")'''

In [ ]: print("hello")

In [ ]: def get\_input():

get\_input()

understanding.

In []: # Write your solution here.

In [ ]: # Write your solution here.

In [ ]: # Write your solution here.

In [ ]: # Write your solution here.

Exercise 7:

Exercise 6:

Exercise 3:

Exercise 4:

the result.

Exercise 1:

def add(x, y):

print( add(5, 5))

return x + y

 $\Pi\Pi\Pi\Pi$ 

accidentally get treated as documentation.

"""This is a class method comment.""" return input("Please enter your name: ")

- Write a line of code in Python that prints 'Hello World' to the screen. In [3]: # Write your Python code in this interactive code cell and 'run' the code to check it works.
- Providing you have completed the previous two exercises, you should have three variables with assigned values. Now write code in the cell below which will print out the individual values for these variables declared previously. In terms of formatting, print out one value for each line. You could use the '\n' escape character for a 'new line' Note: You won't need to re-declare these variables as the values should still be stored in memory until the session is closed.
- In [ ]: # Write your solution here. Exercise 8: Write a script that will convert miles to feet. First ask the user to enter a quantity of miles. Store this in an appropriate variable. Then convert and output (print) the

In [ ]: # Write your solution here. Exercise 10:

much money the user will have after the interest is paid at the end of the year.

In [ ]: # Write your solution here. (Bonus) Exercise 12:

Extend the previous exercise to calculate how many days it is until the user's next birthday from the DOB entered.

Write a script which will process the user's DOB and output which day of the week they were born. In [ ]: # Write your solution here.

function a few times, prompting the user for different types of data.

In [ ]: # Write your solution here.

in their date of birth, check to see whether this is the format required (this could be DD/MM/YYYY). Output a message to prompt the user if the format entered is not

nbviewer GitHub repository.