LECTURER: TAI LE QUY

Introduction to Programming with Python

INTRODUCTION TO PROGRAMMING WITH PYTHON
TOPIC OUTLINE

Introduction to Python	**1**
Variables and Data Types	**2**
Statements	**3**
Functions	**4**
Errors and Exceptions	**5**

Modules and Packages

6

UNIT 2

Variables and Data Types

STUDY GOALS

- Declare variables and assign value to them
- Work with various data types: numbers, strings and characters
- Store and work with collections of data
- Perform basic file input/output operations

EXPLAIN SIMPLY

1. Why do we use variables?
2. What is the result of this expression: 80 % 25?
3. How would you put quotes into a string in Python?

VARIABLES AND VALUE ASSIGNMENT

Definition

In programming, a variable is a data item whose value can change during the execution of a program. A variable is declared (or created) by giving it a name and assigning a value to it:

$$\underbrace{\text{player\_height}}_{\text{Variable name}} = \underbrace{1.85}_{\text{Value}}$$

0 cm

Naming convention

- Begin with a letter or underscore
- Contain only letters, numbers, and underscores
- Be all lowercase
- Be descriptive (multi-word variable name is permissible)
- Have words in a multi-word name separated with underscores

NUMBERS

Types

Type	Notation	Example
Integer	int	`player_goals = 5`
Real number	float	`player_height = 1.85`
Complex number	complex	`z = 3 - 4j`

Operations

Operation	Operator	Example	Result
Addition	+	`2 + 8`	`10`
Subtraction	-	`2 - 8`	`-6`
Multiplication	*	`2 * 8`	`16`
Division	/	`2 / 8`	`0.25`
Floor division	//	`2 // 8`	`0`
Modulus	%	`2 % 8`	`2`
Exponentiation	**	`2 ** 8`	`256`

STRINGS

Declaration

Notation	Example
`""` or `''`	`player_club = "Liverpool"`

Operations

Operation	Operator	Example	Result
Get length of a string	len(variable)	`len(player_club)`	`9`
Find position of a character	.index(…)	`player_club.index('l')`	`8`
Count occurrences	.count(…)	`player_club.count('o')`	`2`
Convert to lowercase	.lower()	`player_club.lower()`	`'liverpool'`
Convert to uppercase	.upper()	`player_club.upper()`	`'LIVERPOOL'`
Add two strings	+	`"Nice" + "day"`	`'Niceday'`
Duplicate string	*	`"Bang " * 2`	`'Bang Bang '`
Get subtring between a & b	[a:b]	`player_club[5:]`	`'pool'`

Types of collections

A collection is a container of data items. There are 4 types of collection in Python.

Type	Description	
Set { }	Characteristics	Elements are unordered and changeable. Duplicates are not allowed.
	Example	`colors_set = {'red', 'blue'}`

	Operations	`colors_set.add(125)` ➔ `{'blue', 125, 'red'}`
		`colors_set.remove('red')` ➔ `{'blue', 125}`
		`colors_set.clear()` ➔ `set()`

List [ ]	Characteristics	Elements are ordered and changeable. Duplicates are allowed.
	Example	`colors_list = ['red', 'blue']`

	Operations	`colors_list.append(125)` ➔ `['red', 'blue', 125]`
		`colors_list.remove('red')` ➔ `['blue', 125]`
		`colors_list.insert(0,125)` ➔ `[125, 'blue', 125]`
		`colors_list.count(125)` ➔ `2`
		`colors_list.clear()` ➔ `[]`

Further reading: Lutz, 2013.

Types of collections (cont.)

Type	Description	
tuple ( )	Characteristics	Elements are ordered and unchangeable. Duplicates are allowed.
	Example	`colors_tuple = ('red', 'blue')`

	Operations	`colors_tuple.index('red')` ➔ `0`
		`colors_tuple.count('blue')` ➔ `1`

Dictionary { }	Characteristics	Elements are key-value pairs. They are unordered and changeable. Keys must be unique (no duplicate).
	Example	`colors_dict = {'red':56, 'blue':'sky'}`

	Operations	`colors_dict['red']` ➔ `56`
		`colors_dict['blue']=28` ➔ `{'red':56, 'blue':28}`
		`colors_dict['green’]=99`
		➔ `{'red':56, 'blue':28, 'green':99}`

Further reading: Lutz, 2013.

File operations

Python has built-in functions to create, read and write to files. The process is as follows:

Open a file ➔ Write to the file ➔ Close the file

Syntax	Description
`open(filename, mode)`	Opens and reads the content of a file called filename, where mode can be one of the following strings: ‘w’ stands for ‘write’. If the file does not exist, create it. If the file exists, overwrite the file with new content (see `.write()`) ‘a’ stands for ‘append’. If the file does not exist, create it. If the file exists, append the content to the file (see `.write()`) ‘r’ stands for ‘read’. If the file does not exist, return an error. The file will be read-only and therefore cannot be written to.
	Example: `my_file = open('myfile.txt', 'a')`

Further reading: Lutz, 2013.

File operations (cont.)

Syntax	Description
`.read()`	Reads and displays the content of the file. This operation is permissible only when the file was opened with the ‘r’ mode.
	Example: `my_file = open('myfile.txt', 'r')` `my_file.read()`
`.write(text)`	Writes `text` to the opened file. If the file was opened with the ‘w’ mode, this operation overwrites the existing content. If the file was opened with the ‘a’ mode, this operation appends `text` to the existing content. Note that the file will not be updated until the file is closed (see below).
	Example: `my_file = open('myfile.txt', 'a')` `my_file.write('To be or not to be.')`
`.close()`	Closes and saves the opened file.

	Example: `my_file = open('myfile.txt', 'a')`
	`my_file.write('To be or not to be.')`
	`my_file.close()`

Further reading: Lutz, 2013.

REVIEW STUDY GOALS

- Declare variables and assign value to them
- Work with various data types: numbers, strings and characters
- Store and work with collections of data
- Perform basic file input/output operations

SESSION 2

TRANSFER TASK

TRANSFER TASK

1. Use Python to answer the following:
   a) Find the integer part and remainder of the division 1587÷27
   b) Get the unique values from the list [11, 21, 21, 5, 72, 81, 72, 5, 11, 21,72, 5]
   c) Display the first and last names of ‘Einstein, Albert’ and ‘Pasteur, Louis’
2. Create a collection that stores the following information:

Country	**Population**	**Capital**	**Calling code**
China	1,413 million	Beijing	+86
Germany	84.36 million	Berlin	+49
India	1,408 million	New Delhi	+91
United Kingdom	67.33 million	London	+44

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.

The results will be discussed in plenary.

LEARNING CONTROL QUESTIONS

1. If I have a string “my_string” with the value "Python is so powerful!" how do I substring to get just “is so”?

LEARNING CONTROL QUESTIONS

2. I want to open and write to a file called “thatfile.txt”. If the file does not exist, I want to create it. If it does exist, I want to write at the end of the file, preserving what’s already in the file. How do I do that?

LEARNING CONTROL QUESTIONS

3. What is the result of this equation: 80 % 25?

LIST OF SOURCES

Lutz, M. (2013). Introducing Python Object Types. *Learning Python* (5th ed.). O’Reilly.