



Python – Input, output and variables

Lecture 16 – COMPSCI111/111G S2 2019

1



Today's lecture

- ▶ What is Python?
- ▶ Displaying text on screen using `print()`
- ▶ Variables
- ▶ Numbers and basic arithmetic
- ▶ Getting input from keyboard using `input()`

2



What is a programming language?

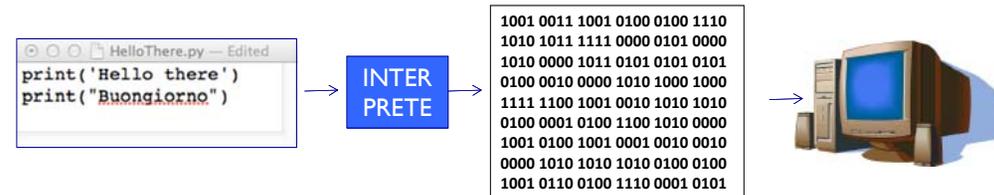
- ▶ A formal language that specifies how to perform a computational task
- ▶ Many programming languages exist:
 - ▶ Visual Basic
 - ▶ C and C++
 - ▶ C#
 - ▶ Java
 - ▶ Python
- ▶ Python was created in 1989 by Guido Van Rossum in The Netherlands

3



Statements

- ▶ A program consists of a series of commands called **statements**
- ▶ They are generally executed (ie. run) in the order they appear
- ▶ The statements must be written correctly otherwise you will get a syntax error
- ▶ Python programs are saved in files with the '.py' extension



4



Translating code

- ▶ The statements in our programs are translated into simpler instructions that the CPU can execute
- ▶ Two ways of doing this:
 - ▶ Compiler: translates the entire program file at once
 - ▶ Interpreter: repeatedly translates one line and runs it
- ▶ Python is an interpretative programming language
 - ▶ There are also compilers available for Python

5



IDLE Integrated Development Environment (IDE)

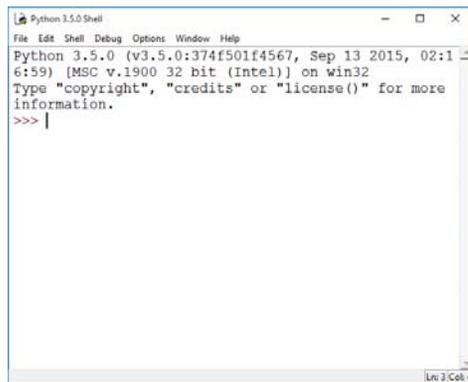
- ▶ An IDE is used by programmers to:
 - ▶ Write code
 - ▶ Check for errors
 - ▶ Translate code and run the program
- ▶ We use the IDLE IDE; a popular IDE for Python
- ▶ IDLE has a shell for the Python interpreter
- ▶ You can also create a new file that can be compiled when you've finished writing a program

6



IDLE IDE

- ▶ The interpreter allows you to type statements, translate them and see them run instantly
- ▶ Very helpful for experimentation and learning



7



Interactive Interpreter Vs Running a script

- ▶ Interactive Interpreter
 - ▶ Allows you to type statements directly at the prompt
 - ▶ Statement is executed when you hit <Enter>
 - ▶ Very useful for experimentation
 - ▶ Good for learning
- ▶ Running a Script
 - ▶ Type a sequence of statements into a file
 - ▶ Save the file with the file extension .py
 - ▶ Running the program executes each statement in turn

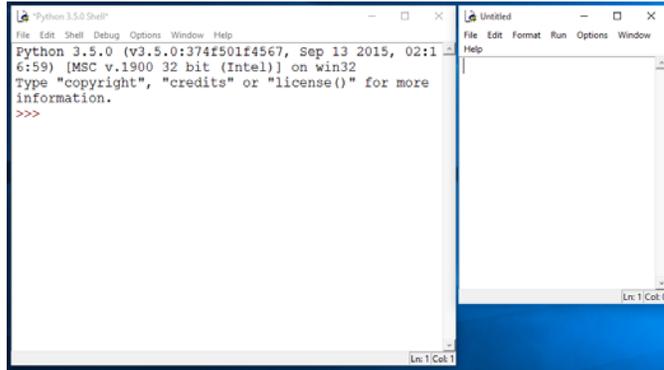


8



IDLE IDE

- ▶ Create a new program by clicking on File → New File
- ▶ Type your statements in the file, then click on Run → Run Module...

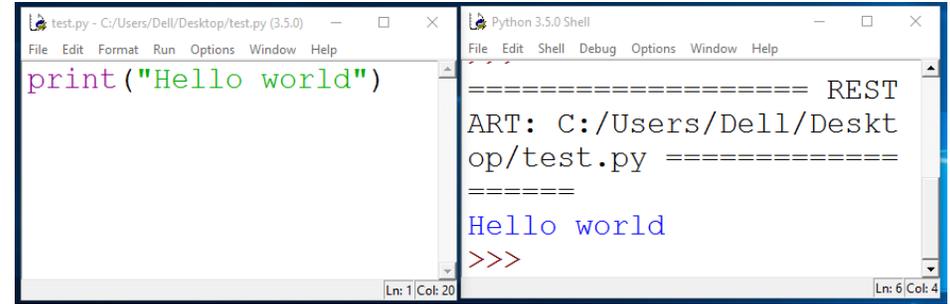


9



“Hello world”

- ▶ Traditional first program is displaying “Hello World” on screen
- ▶ To display text on screen you use the `print()` function

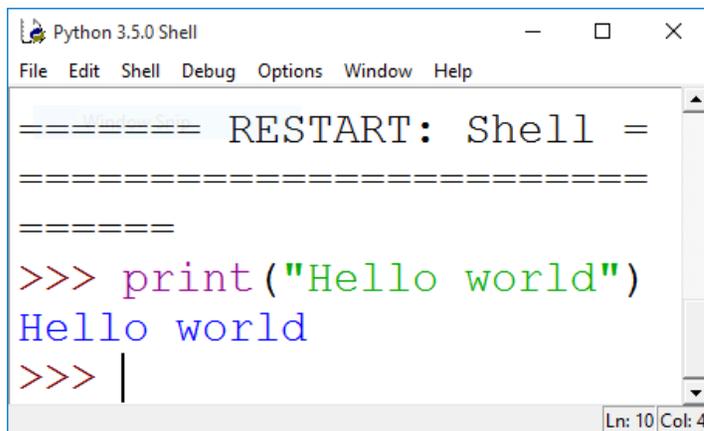


10



“Hello world”

- ▶ Using the Python interpreter:



11



Printing output

- ▶ Use the print statement

Code	Output
<code>print("This is text")</code>	This is text
<code>print(34.9)</code>	34.9

- ▶ Printing more than one thing on a single line

- ▶ Separate each thing with a comma
- ▶ Single space used between different things in the output

Code	Output
<code>print("Hello", "World")</code>	Hello World
<code>print("The year is", 2017)</code>	The year is 2017

12



Exercise 1

- ▶ What is the output produced by the following statements?

```

*Untitled*
File Edit Format Run Options Window Help
print(1,2,3,4)
print("1,2,3,4")
print("1234", 1,2)
print("1",2,3,"4")

```



Comments

- ▶ When writing a program, it is helpful to leave comments in the code
- ▶ You can write a comment in Python by typing a '#' in front of the line
- ▶ The compiler will ignore all text after the '#'

```

*test.py - C:/Users/Dell/Desktop/test.py (3.5.0)*
File Edit Format Run Options Window Help
#Reuel's first program
#3/02/16

print("Hello world") #Print() displays text on screen

```



Data types

- ▶ **Strings:**
 - ▶ Sequence of characters
 - ▶ Plain text (ASCII or Unicode)
 - ▶ Enclosed in quote marks
 - ▶ Eg: "Hello", "Goodbye"
- ▶ **Integers:**
 - ▶ Whole numbers (ie. without a decimal point)
 - ▶ Eg. -100, 0, 45
- ▶ **Floating point numbers:**
 - ▶ Numbers with a decimal point
 - ▶ Eg. 5.2, -1.002, 0.0



Variables

- ▶ A 'container' in the computer's memory in which you can store data
- ▶ A variable's value can change when the program runs
- ▶ Python variables are loosely-typed; they can hold any data type





Variables

- ▶ Rules to follow when naming your variables:
 - ▶ Names should reflect what is stored in the variable
 - ▶ Can begin with a letter or underscore (eg. '_')
 - ▶ Variable names can include numbers
 - ▶ Generally, all words are lowercase and words are separated using an underscore

The image shows two side-by-side screenshots of a Python IDE window titled '*test.py - C:/Users/Dell/Desktop/test.py (3.5.0)*'. The left window shows 'Good variable names' with code: `#Good variable names`, `#3/02/16`, `age`, `height_of_chair`, `box_1`, and `search_criteria`. The right window shows 'Poor variable names' with code: `#Poor variable names`, `#3/02/16`, `1_test`, `age-child`, `numberofrooms`, and `x`.

17



Assignment statement

- ▶ Assigning a value to a variable:

The image shows a screenshot of a Python IDE window titled '*test.py - C:/Users/Dell/Desktop/test.py (3.5.0)*'. The code contains the following assignment statements: `age = 21`, `name = "Reuel"`, `height = 1.68`, and `course_in_ss = "Compsci111/111G"`. The status bar at the bottom right indicates 'Ln: 6 Col: 0'.

18



Assignment statement

- ▶ Changing the value in a variable:

The image shows a screenshot of a Python IDE window titled '*test.py - C:/Users/Dell/Desktop/test.py (3.5.0)*'. The code shows: `age = 30`, `age = age + 1`, `course = "Compsci"`, and `course = course + "111/111G"`. The status bar at the bottom right indicates 'Ln: 13 Col: 0'.

19



Exercise 2

- ▶ What is the output produced by the following statements?

The image shows a screenshot of a Python IDE window titled '*L22.py - C:/Users/kng001.UOA/Data/Te...'. The code contains: `height = 10`, `width = 20`, `area = height * width`, and `print("Area =", area)`. The status bar at the bottom right indicates 'Ln: 6 Col: 0'.

20



Arithmetic operations

Operation	Symbol	Example
Exponent	**	2 ** 3 = 8
Multiply	*	2 * 2 = 4
Divide	/	10 / 3 = 3.333
Divide (integer)	//	10 // 3 = 3
Remainder	%	10 % 3 = 1
Add	+	8 + 9 = 17
Subtract	-	9 - 7 = 2

21



Print() function

- ▶ Used to display information on the screen

Code	Output
<code>print("This is text")</code>	This is text
<code>print(10 / 3)</code> <code>print(2 ** 5)</code>	3.3333333333333335 32
<code>age = 21</code> <code>print("You are", age, "years old")</code>	You are 21 years old
<code>age = age * 2</code> <code>print("You are actually", age, "!")</code>	You are actually 42 !

22



Print() function

- ▶ Concatenation: this involves joining two or more strings together

```

a = "Hello "
b = "big "
c = "world"
print(a + b + c + "!")
Hello big world!

```

- ▶ Repetition: lets you print a string multiple times

```

a = "Compsci111"
print(a * 3)
Compsci111Compsci111Comp
sci111

```

23



Exercise 3

- ▶ What is the output for the following code?

```

a = 5
b = 10
print("This", "is", "a", "program")
print(5 ** 2)
print("This", "is", a, "program")
print("Result:", 50 / 2 * b)

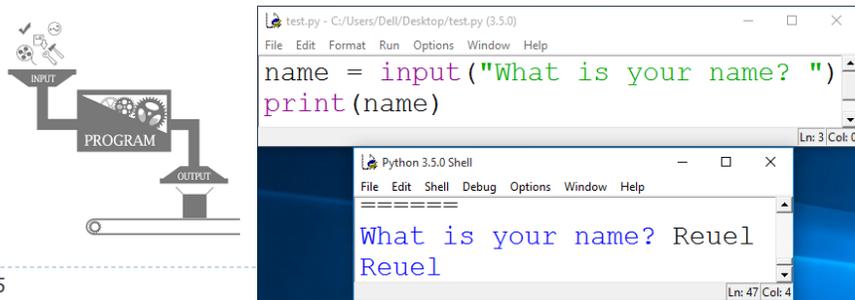
```

24



Getting input

- ▶ Primary source of input for our programs will be the keyboard
- ▶ The `input()` function:
 - ▶ Prints a prompt for the user to read
 - ▶ Captures the user's keystrokes
 - ▶ When the user presses 'Enter', stores the string in a variable



25



Getting input

- ▶ Converting the string value returned by `input()` to an integer or floating point value
 - ▶ You need to do this when you want the actual numerical value the user is entering
- ▶ `age = int(input("Enter your age: "))`
- ▶ `height = float(input("Enter your height: "))`
- ▶ `height = height + 1.5`

26



Exercise 4

- ▶ Write a Python program that converts feet to metres. The conversion formula is:

$$1 \text{ foot} = 0.3048 \text{ metres}$$

- ▶ Your program's output should look like this:

```

Enter feet: 34
34 feet = 10.3632 metres.

```

- ▶ You will need to use:
 - ▶ Variables
 - ▶ Arithmetic operator
 - ▶ `input()` and `print()`

27



Algorithm

Prompt for the value

Create a variable and set the value
(`feet_to_metres = 0.3048`)

Calculate the corresponding value

print the result

28



Summary

- ▶ Python programs consist of statements that are translated by an interpreter or compiler into instructions that the CPU can execute
- ▶ We've discussed the Python programming language and its features:
 - ▶ `print()`
 - ▶ Data types: `string`, `int`, `float`
 - ▶ Arithmetic operators
 - ▶ Variables and variable naming conventions
 - ▶ `input()` and `int()`, `float()`