

Python - Input, output and variables

Lecture 22 - COMPSCI1111/111G SS 2016

Today's lecture

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

What is a programming language?

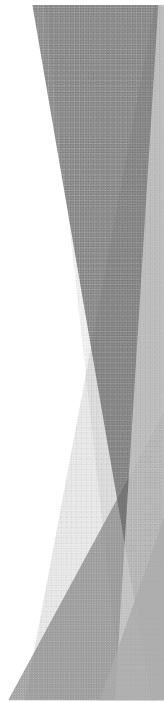
- ▶ A formal language that specifies how to perform a computational task
- ▶ Many programming languages exist:
 - ▶ Visual Basic
 - ▶ C and C++
 - ▶ C#
 - ▶ Java
 - ▶ Python
- ▶ Python was created in 1989 by Guido Van Rossum in The Netherlands

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

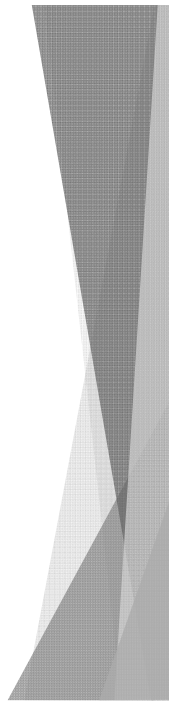
Translating code

- ▶ 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



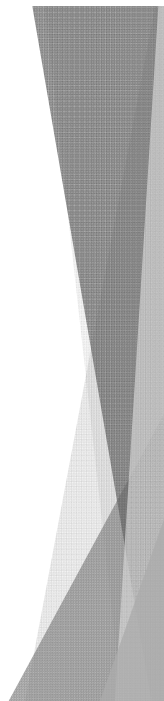
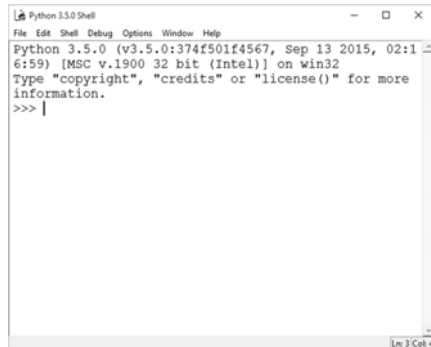
IDLE Integrated Development Environment (IDE)

- ▶ 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



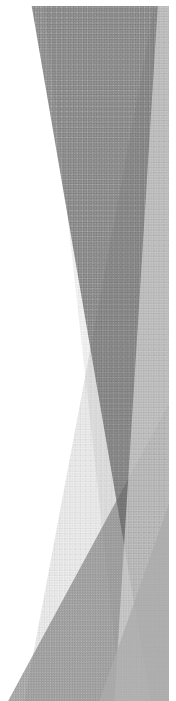
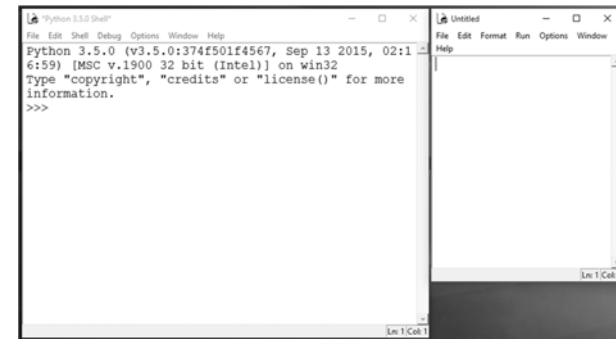
IDLE IDE

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



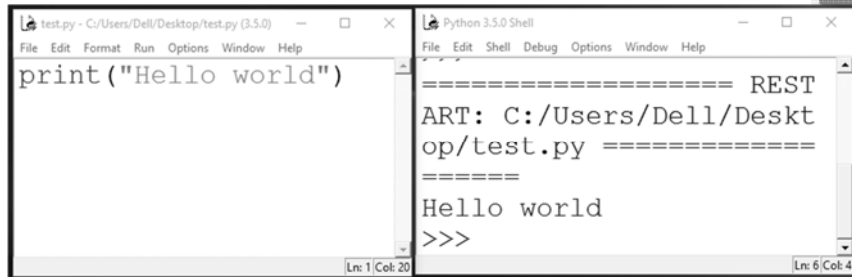
IDLE IDE

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



“Hello world”

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



The image shows two windows from a Python IDE. The left window, titled 'test.py - C:/Users/Dell/Desktop/test.py (3.5.0)', contains the code `print("Hello world")`. The right window, titled 'Python 3.5.0 Shell', shows the output of running the script: a separator line, the file path `C:/Users/Dell/Desktop/test.py`, another separator line, and the text `Hello world`. The shell prompt `>>>` is visible at the bottom.

“Hello world”

- ▶ Using the Python interpreter:



The image shows a 'Python 3.5.0 Shell' window. It displays a separator line, the text `RESTART: Shell =`, another separator line, and the prompt `>>>`. The user has entered `print("Hello world")`, and the shell has responded with `Hello world`. The prompt `>>>` is now on a new line. The status bar at the bottom right indicates 'Ln: 10 Col: 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 `#`



The image shows a Python IDE window titled 'test.py - C:/Users/Dell/Desktop/test.py (3.5.0)'. The code contains two comment lines: `#Reuel's first program` and `#3/02/16`. Below them is a line of code: `print("Hello world") #Print() displays text on screen`. The status bar at the bottom right indicates 'Ln: 5 Col: 0'.

Data types

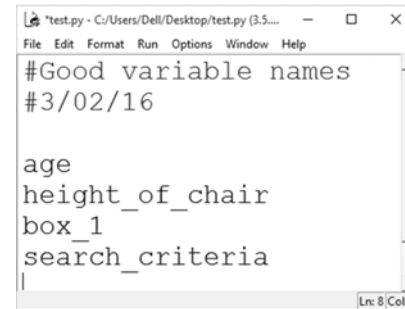
- ▶ Strings:
 - ▶ Sequence of characters
 - ▶ Plain text (ASCII or Unicode)
 - ▶ Enclosed in quote marks
 - ▶ Eg: "Hello", "Goodbye"
- ▶ Integers:
 - ▶ Whole numbers (ie. without a decimal point)
 - ▶ Eg. -100, 0, 45
- ▶ Floating point numbers:
 - ▶ Numbers with a decimal point
 - ▶ Eg. 5.2, -1.002,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



```
*test.py - C:/Users/Dell/Desktop/test.py (3.5...  
File Edit Format Run Options Window Help  
#Good variable names  
#3/02/16  
  
age  
height_of_chair  
box_1  
search_criteria  
|  
Ln: 8 Col: 0
```



```
*test.py - C:/Users/Dell/Desktop/test.py (3.5...  
File Edit Format Run Options Window Help  
#Poor variable names  
#3/02/16  
  
1_test  
age-child  
numberofrooms  
x|  
Ln: 7 Col: 1
```

Variables

- ▶ Assigning a value to a variable:



```
*test.py - C:/Users/Dell/Desktop/test.py (3.5.0)*  
File Edit Format Run Options Window Help  
age = 21  
name = "Reuel"  
height = 1.68  
course_in_ss = "Compsci111/111G"  
Ln: 6 Col: 0
```

Variables

- ▶ Changing the value in a variable:



```
*test.py - C:/Users/Dell/Desktop/test.py (3.5.0)*  
File Edit Format Run Options Window Help  
age = 30  
age = age + 1  
  
course = "Compsci"  
course = course + "111/111G"  
Ln: 13 Col: 0
```

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

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 !

Print() function

► Concatenation: this involves joining two or more strings together

```

test.py - C:/Users/Dell/Desktop/test.py (3.5.0)
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
a = "Hello "
b = "big "
c = "world"
print(a + b + c + "!")
Hello big world!
>>> |
Ln: 4 Col: 22
Ln: 34 Col: 4
    
```

► Repetition: lets you print a string multiple times

```

test.py - C:/Users/Dell/Desktop/test.py (3.5.0)
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
a = "Compsc111"
print(a * 3)
Compsc111Compsc111Compsc111
>>> |
Ln: 2 Col: 12
Ln: 37 Col: 4
    
```

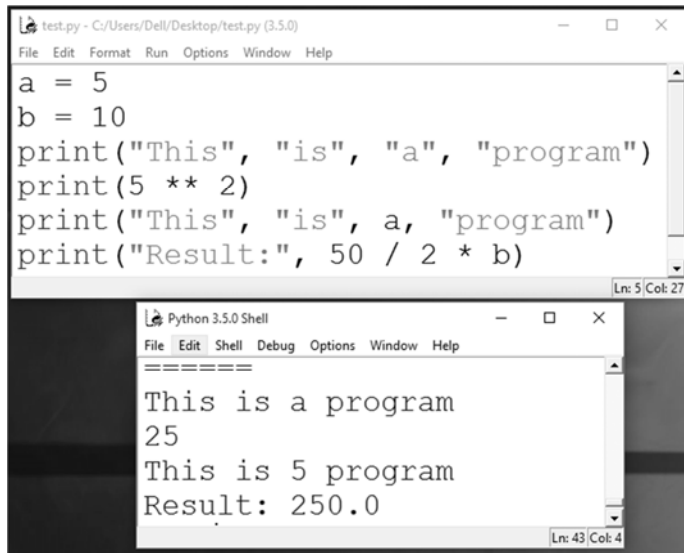
Exercise

► What is the output for the following print() statements:

```

Untitled
File Edit Format Run Options Window Help
a = 5
b = 10
print("This", "is", "a", "program")
print(5 ** 2)
print("This", "is", a, "program")
print("Result:", 50 / 2 * b)
Ln: 11 Col: 0
    
```

Exercises



The screenshot shows a Python IDE window titled 'test.py - C:/Users/Dell/Desktop/test.py (3.5.0)'. The code in the editor is:

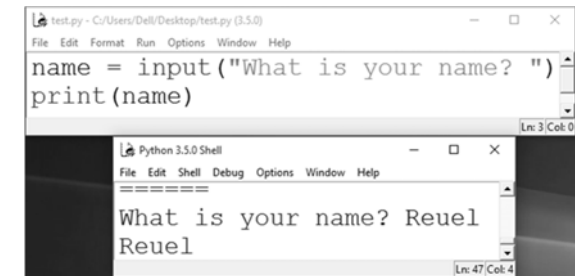
```
a = 5
b = 10
print("This", "is", "a", "program")
print(5 ** 2)
print("This", "is", a, "program")
print("Result:", 50 / 2 * b)
```

Below the code, the Python 3.5.0 Shell window shows the output:

```
=====
This is a program
25
This is 5 program
Result: 250.0
```

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



The screenshot shows a Python IDE window titled 'test.py - C:/Users/Dell/Desktop/test.py (3.5.0)'. The code in the editor is:

```
name = input("What is your name? ")
print(name)
```

Below the code, the Python 3.5.0 Shell window shows the output:

```
=====
What is your name? Reuel
Reuel
```

Getting input

- ▶ You convert 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`

Exercise

- ▶ Write a Python program that converts feet to metres. The conversion formula is:
$$1 \text{ foot} = 0.3048 \text{ meters}$$
- ▶ Your program's output should look like this:

```
Enter feet: 34
34 feet is equal to 10.3632 metres
```
- ▶ You will need to use:
 - ▶ Variables
 - ▶ Arithmetic operator
 - ▶ `input()` and `print()`

Exercise

```
feet = int(input("Enter feet: "))  
feet_to_metres = 0.3048  
metres = feet * feet_to_metres  
print(feet, "feet is equal to", metres,  
      "metres.")
```

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 convention
 - ▶ `input()` and `int()`, `float()`