A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Python : programming language

CT010-3-1 Python Programming

Topic & Structure of the lesson

- Problem Solving Using Programmed Solutions
  - Introduction to Python
  - Python as a programming language
  - Interactive vs script programming

Learning outcomes

- At the end of this lecture you should be able to:
  - Develop a problem-based strategy for creating and applying programmed solutions.

Key terms you must be able to use

- If you have mastered this topic, you should be able to use the following terms correctly in your assignments and exams:
  - Program
  - Interactive
  - Script

Computers want to be helpful...

- Computers are built for one purpose - to do things for us
- But we need to speak their language to describe what we want done
- Users have it easy - someone already put many different programs (instructions) into the computer and users just pick the ones we want to use

Users .vs. Programmers

- Users see computers as a set of tools - word processor, spreadsheet, map, todo list, etc.
- Programmers learn the computer "ways" and the computer language
- Programmers have some tools that allow them to build new tools
- Programmers sometimes write tools for lots of users and sometimes programmers write little "helpers" for themselves to automate a task

What is Code? Software? A Program?

- A sequence of stored instructions
  - It is a little piece of our intelligence in the computer
  - It is a little piece of our intelligence we can give to others - we figure something out and then we encode it and then give it to someone else to save them the time and energy of figuring it out
- A piece of creative art - particularly when we do a good job on user experience

Talking to Python

```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23
2016, 08:06:12) [MSC v.1900 64 bit (AMD64)]
on win32

Type "copyright", "credits" or "license()"
for more information.

>>>
```

```
>>> 1 + 2
3
>>> 2 * 3
6
>>> x = 1
>>> print(x)
1
>>> x = x + 1
>>> print(x)
2
>>> exit()
```

This is a good test to make sure that you have Python correctly installed. Note that quit() also works to end the interactive session.

Elements of Python

- Vocabulary / Words - Variables and Reserved words (Chapter 2)
- Sentence structure - valid syntax patterns (Chapters 3-5)
- Story structure - constructing a program for a purpose

Variable Names

- Choose a meaningful name
- Can be as long as you like
- Can contain both letters and numbers
- Cannot begin with a number
- Cannot use **keywords**
- Legal to use uppercase, however all lowercase is the convention
- Underscore can be used for multi-word variable name:

```
name_of_student
```

Keywords or Reserved Words

- Keywords are words reserved by python that have predefined meaning. You can not use reserved words as variable names / identifiers

```
False
None
True
and
as
assert
break
class
continue
def
del
elif
else
except
finally
for
from
global
if
import
in
is
lambda
nonlocal
not
or
pass
raise
return
try
while
with
yield
```

Sentences or Lines

- x = 2 ⟵ Assignment Statement
- x = x + 2 ⟵ Assignment with expression
- print(x) ⟵ Print statement

Python Scripts

Interactive Python is good for experiments and programs of 3-4 lines long

But most programs are much longer so we type them into a file and tell python to run the commands in the file.

In a sense we are “giving Python a script”
As convention, we add “.py” as the suffix on the end of these files to indicate they contain Python

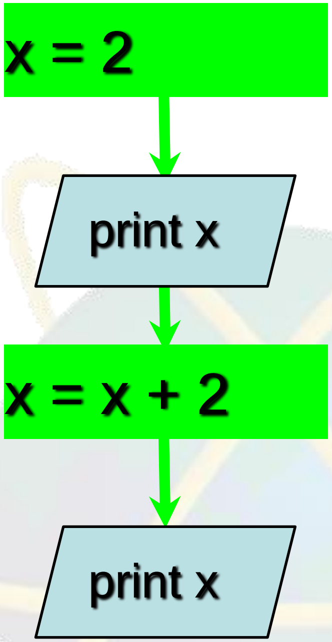
Interactive versus Script

- Interactive
  - – You type directly to Python one line at a time and it responds
- Script
  - – You enter a sequence of statements (lines) into a file using a text editor and tell Python to execute the statements in the file

3. Program Steps or Program Flow

- Like a recipe or installation instructions, a program is a **sequence of steps to be done in order**
- Some steps are **conditional** - they **may be skipped**
- Sometimes a step or **group of steps are to be repeated**
- Sometimes we store **a set of steps to be used over and over as needed in several places** throughout the program

Sequential Steps



Program:

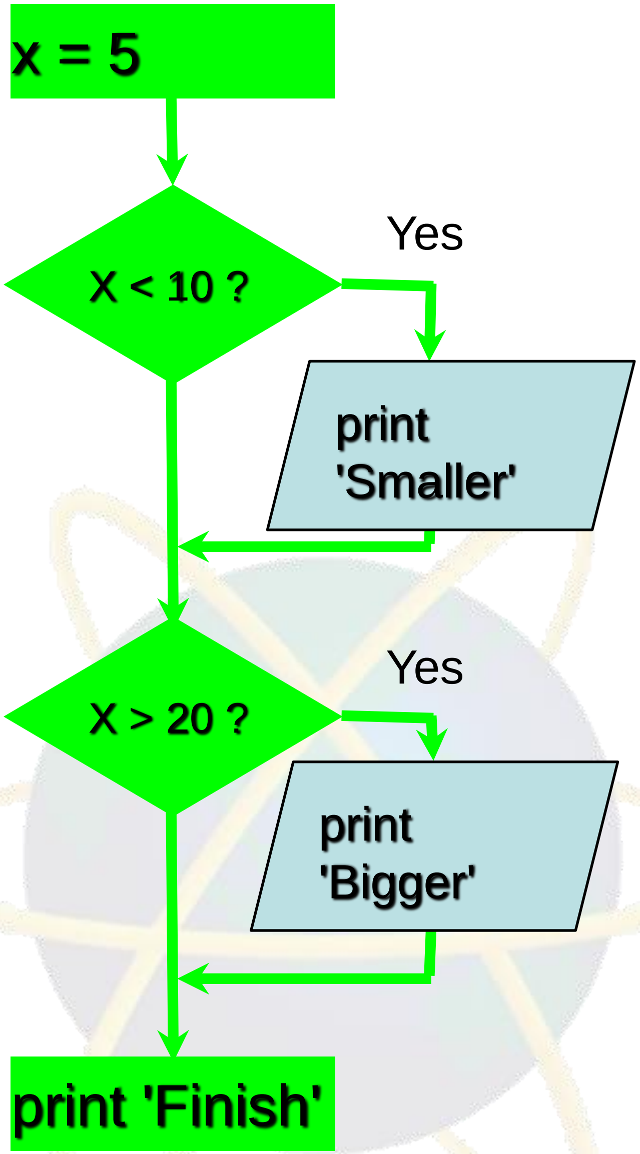
```
x = 2
print(x)
x = x + 2
print(x)
```

Output:

```
2
4
```

When a program is running, it flows from one step to the next. We as programmers set up “paths” for the program to follow.

Conditional Steps



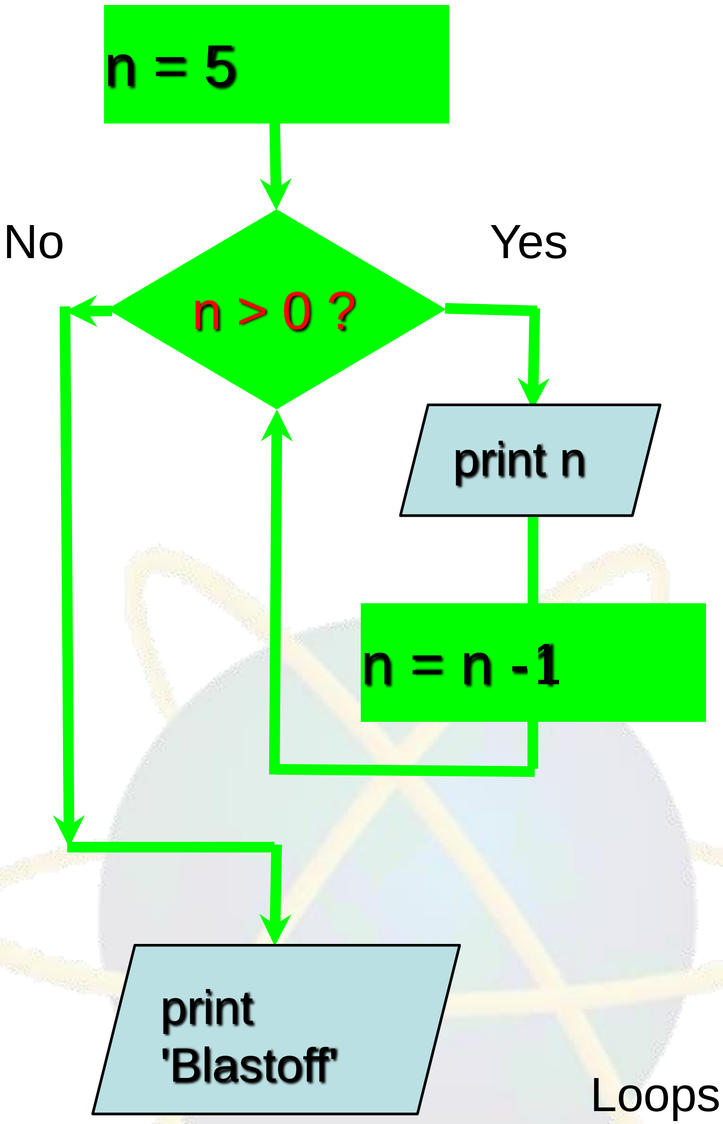
Program:

```
x = 5
if x < 10:
    print('Smaller’)
if x > 20:
     print('Bigger‘)
print( 'Finish‘)
```

Output:

```
Smaller
Finis
```

Repeated Steps



Program:

```
n = 5
while n > 0 :
    print(n)
    n = n – 1
print('Blastoff!‘)
```

Output:

```
5
4
3
2
1
Blastoff!
```

Loops (repeated steps) have iteration variables that change each time through a loop. Often these iteration variables go through a sequence of numbers.

Next Session

- Problem Solving Using Programmed Solutions
  - Variables, expressions and statements in python
    - Rules for naming variables
    - Numeric expressions
    - Operator precedence
    - Statements in python
    - Reserved words