

COMP1021
Introduction to Computer Science

Getting Started with Python

David Rossiter and Gibson Lam

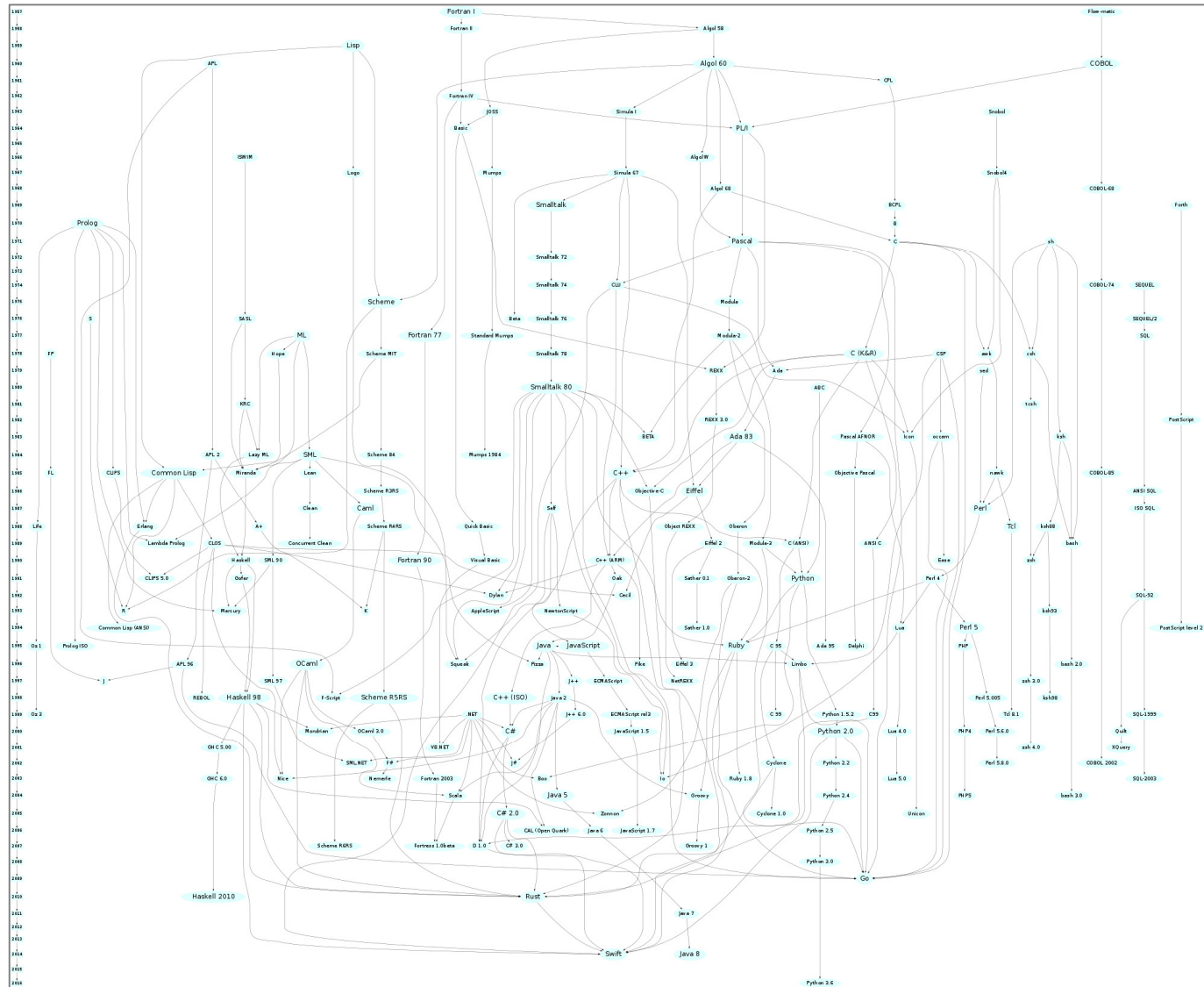
Outcomes

- After completing this presentation, you are expected to be able to:
 1. Understand the history and some background information of the Python programming language
 2. Briefly explain the difference between a compiled and an interpreted program
 3. Install Python and start using Python through the command line tool and IDLE

Computer Programming Languages

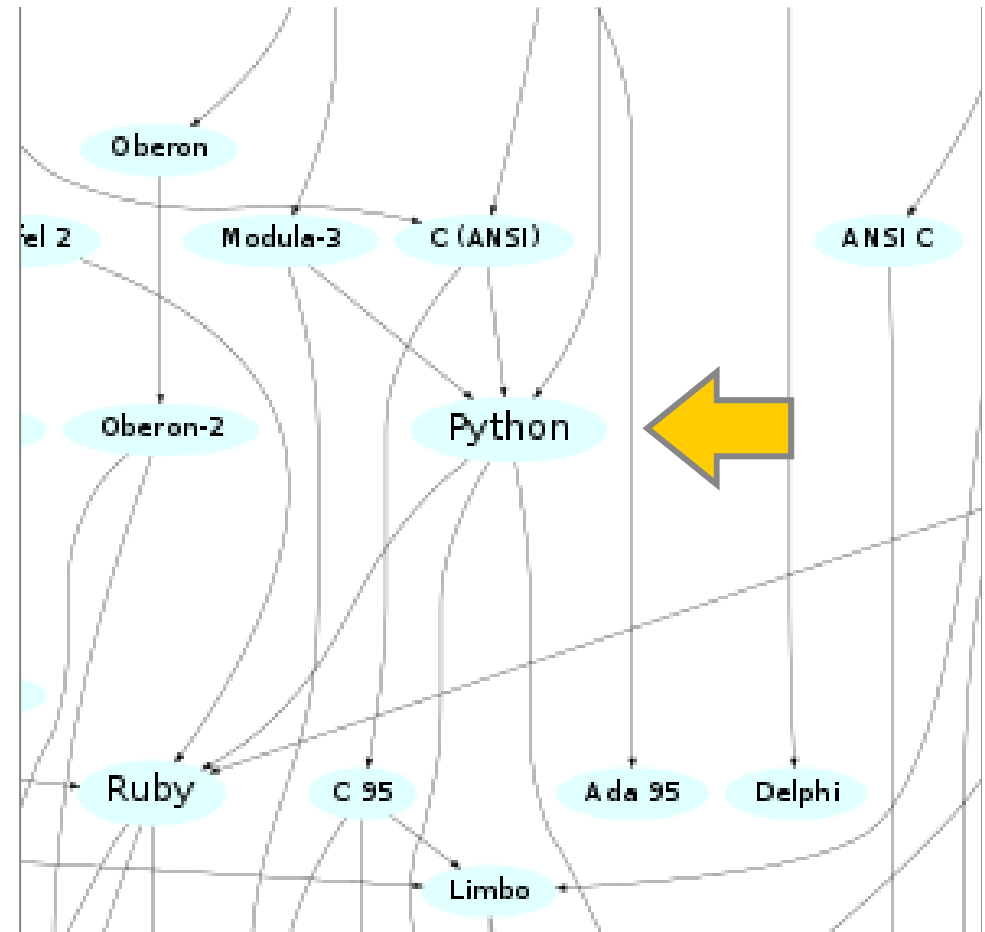
- Computer programming languages have been developed over the last 50 years
- There are hundreds of them
- For this course we will use a language called *Python*

Evolution of Programming Languages



<http://rigaux.org/language-study/diagram.html>

- Where is Python in all of that?!



- Do we need to know about all those other languages? Thankfully, no!
- But once you know Python learning another programming language is easy

Python

- Started by a guy who was bored during Christmas 1989
- He made a computer language with these qualities:
 - a language just as powerful as other languages
 - code that is almost as understandable as simple English
 - suitable for everyday tasks, so you can quickly make a useful program
 - open source, so anyone can contribute to its development

Guido van Rossum

- He has been working for Google up to 2012
- He is now working for Dropbox



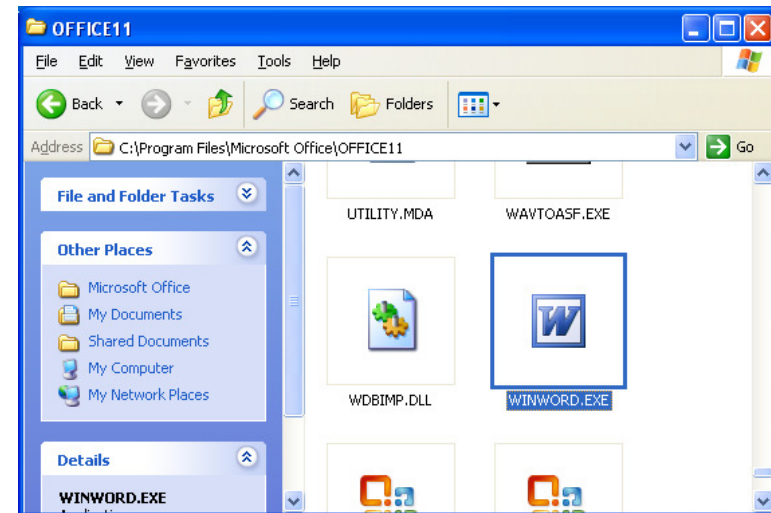
Why is it called Python?

- He called his language Python because of his favourite TV comedy program, 'Monty Python's Flying Circus'



Running a Program

- Most of the programs on your computer are .exe files
- For example, if you run Microsoft Word, it means the file *winword.exe* is being executed
- Microsoft Word is a program that has been *compiled* into winword.exe, so it can be directly run by a computer



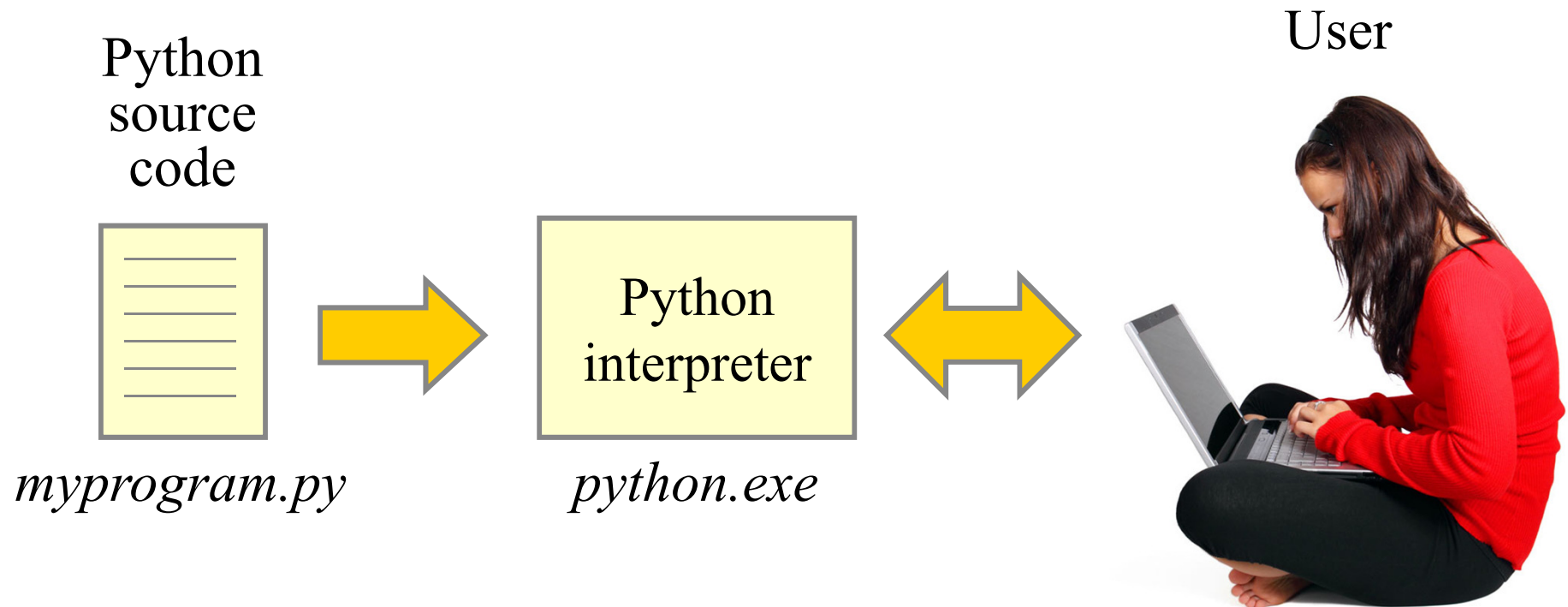
Running a Compiled Program



- A computer can understand and directly run an exe file

Executing a Python Program

- Python programs are a bit different, they have to be ‘given’ to a Python interpreter for execution



- We say that Python code is *interpreted*
- This is the most common way that Python is used

Python is Interpreted

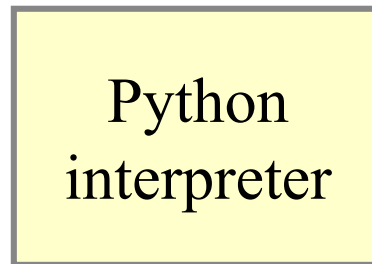
- *Interpreted* means that each line of code is given to the interpreter and executed, one by one

User

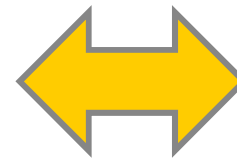
Python
source code

```
def add5(x):  
    return x+5  
  
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]),ast[0])  
    print '%s (label="%s"' % (nodename, label),  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '%s"' % ast[1]  
        print '['  
    else:  
        print '['  
        children = []  
        for n, child in enumerate(ast[1:]):  
            children.append(dotwrite(child))  
        print '%s -> {' % nodename,  
        for name in children:  
            print '%s' % name,
```

myprogram.py



python.exe



- In this course we won't look at the difference between interpreting and compiling any further

Different Versions of Python

- There are two versions of Python being used: Python version 2 and Python version 3
- We will use the more recent version, version 3
- Although Python version 2 is older, it is still used by many people around the world (no-one uses version 1)
- Python 3.7 has been installed in all the CS department labs and should be in all the computer barns
- You can run it *virtually* – details a few slides later
- You can install it in your own computer, see next slide

How to Install Python 3.7?

- To install Python on your own PC/Mac system, you can get the installation files from the COMP1021 web site:



Getting Started with Python

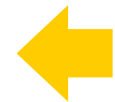
[B&W, colour] [Google Docs]



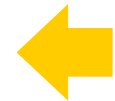
Different ways to run Python [here](#)



- For this course we will use the programming language Python
- Here's some places where you can run Python
 1. The computers in the **CS lab room** where we will have the taught lab sessions already have Python installed so you don't need to install it yourself in that room
 2. The **ITSC Computer Barns** also have Python already installed
 3. You can also use Python in the **Virtual Barn environment** of ITSC:
 - <http://itsc.ust.hk/services/academic-teaching-support/facilities/virtual-barn/overview/>
 4. If you want to do COMP1021 work on **your own computer**, you can install Python on it
 - You should install these files (from [here](#)), which are the same versions we use on the course:
 - Python for Windows - [python-3.7.3.exe](#)
 - Python for Windows (64 bits) - [python-3.7.3-amd64.exe](#)
 - Python for MacOS X (10.6 or later) - [python-3.7.3-macosx10.6.pkg](#)
 - Python for MacOS X (10.9 or later) - [python-3.7.3-macosx10.9.pkg](#)



PC users



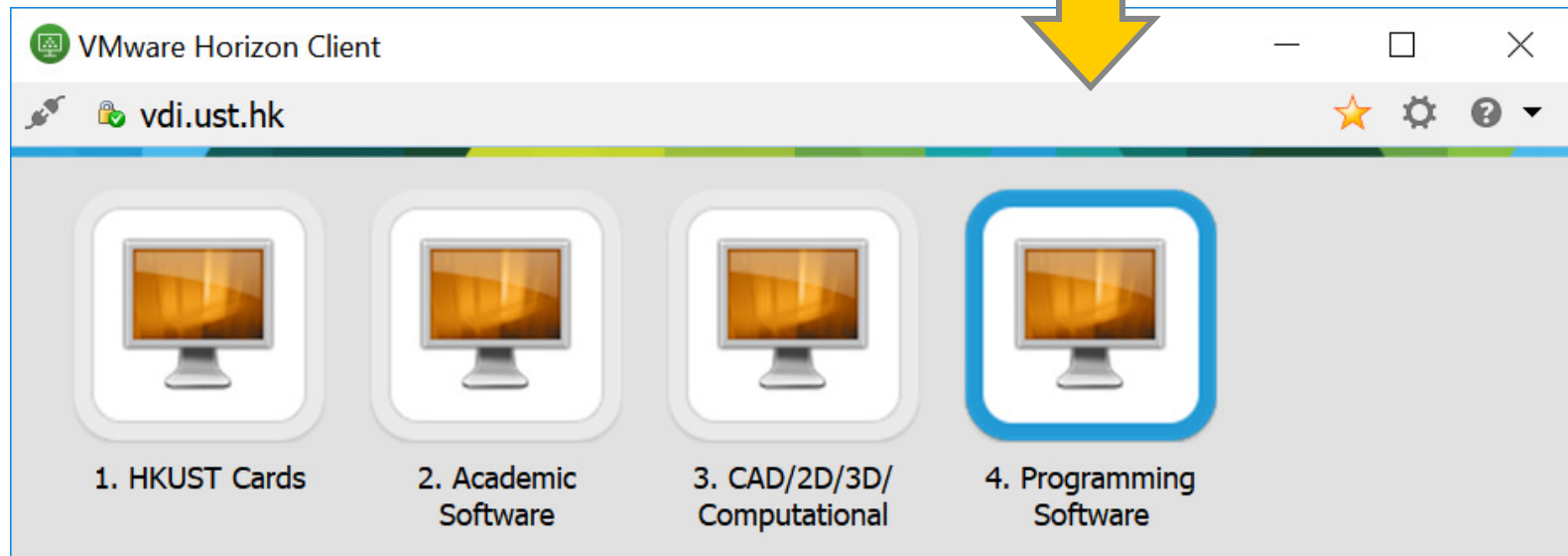
Mac users

The ITSC Virtual Barn

- The ITSC virtual barn lets you access Python (and other software) anywhere:

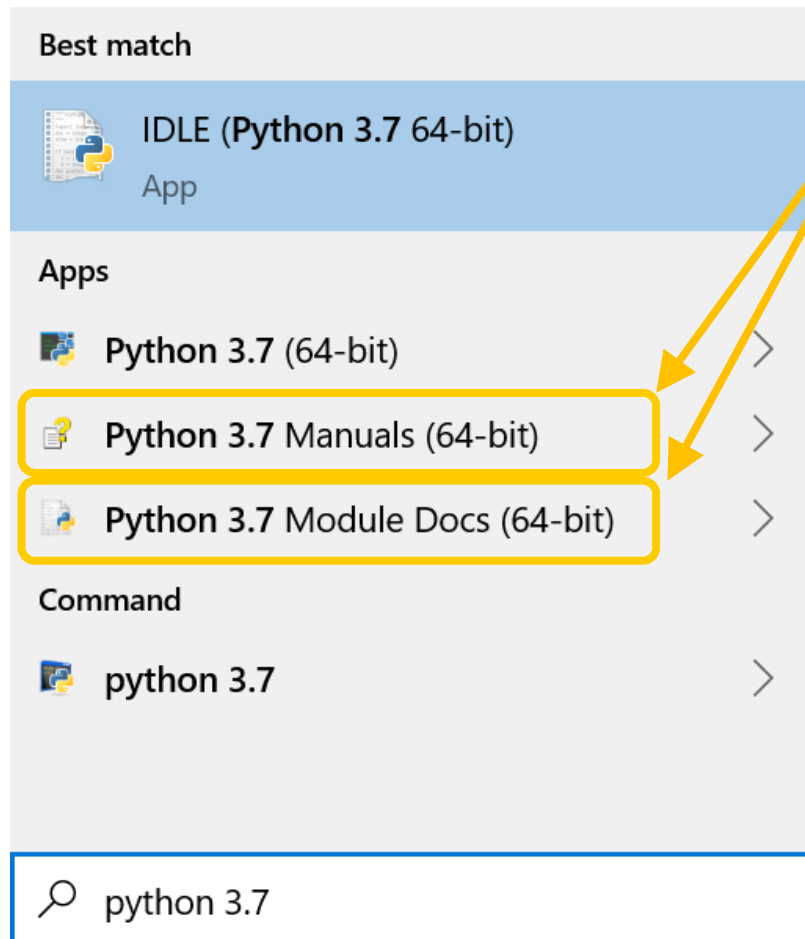
<http://itsc.ust.hk/services/academic-teaching-support/facilities/virtual-barn/overview/>

- You can find Python 3.7 in the ‘Programming Software’ pool



After Installing Python 3.7

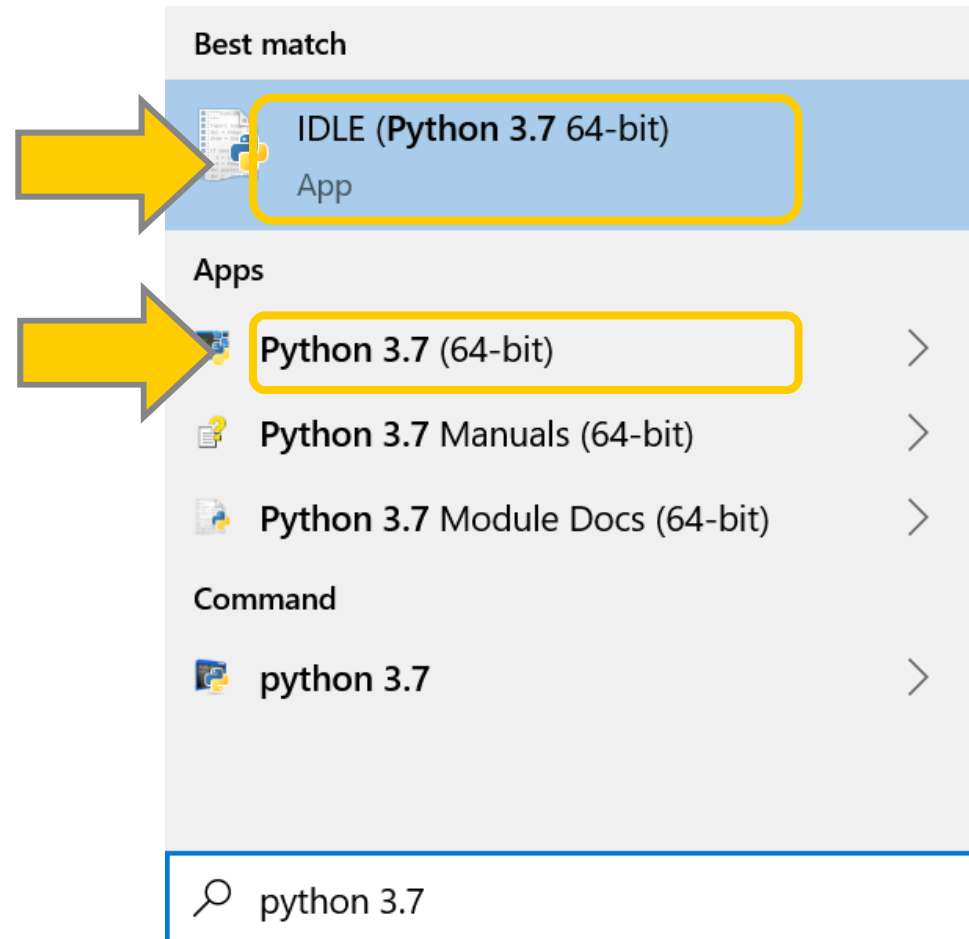
(These images are from a PC)



- After installing, you will see several options if you search for Python 3.7 in Windows
- Here is some documentation about Python
 - Probably you won't need to look at this, the COMP1021 notes and labs should be enough
- *Note - you probably won't see this in the CS labs, but you will see it if you install Python in your own computer*

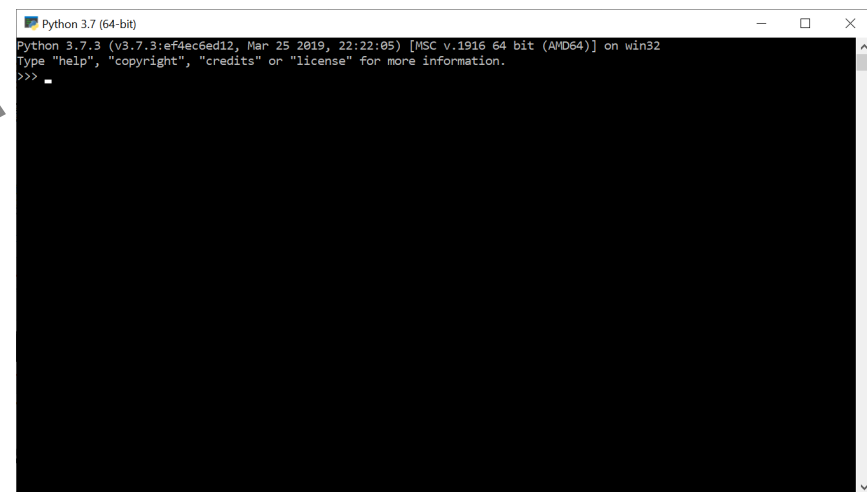
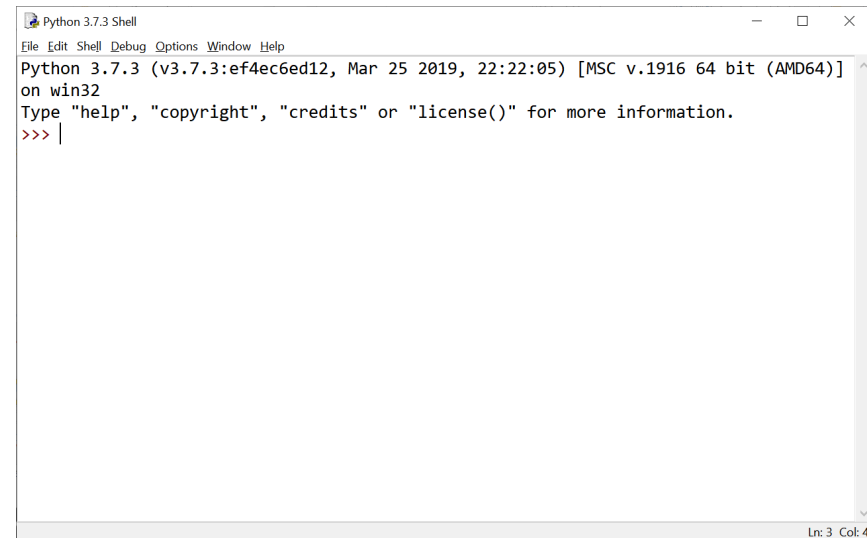
Using Python

- Let's look at how we can start using Python
- There are two ways we will look at now

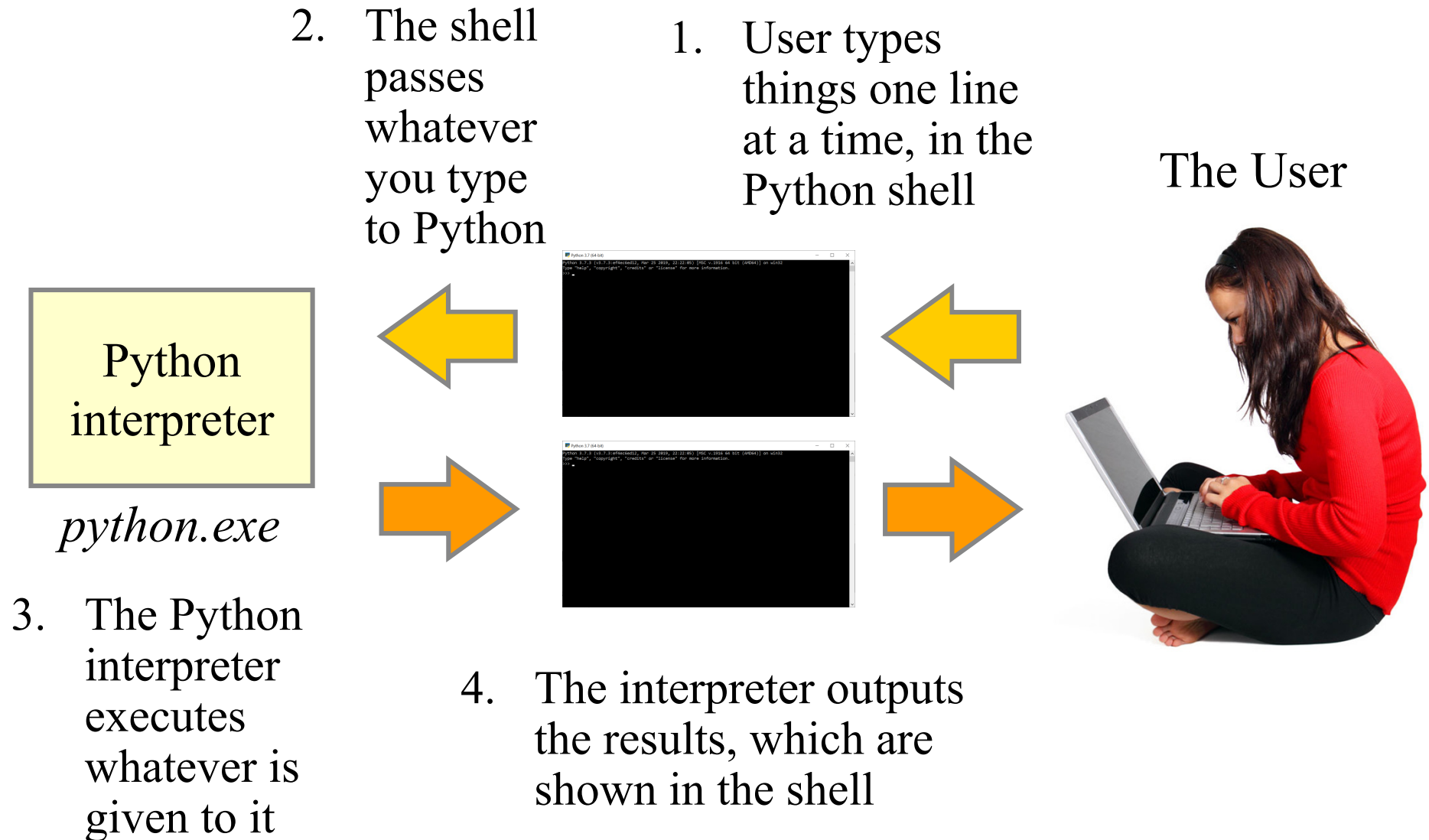


Using Python

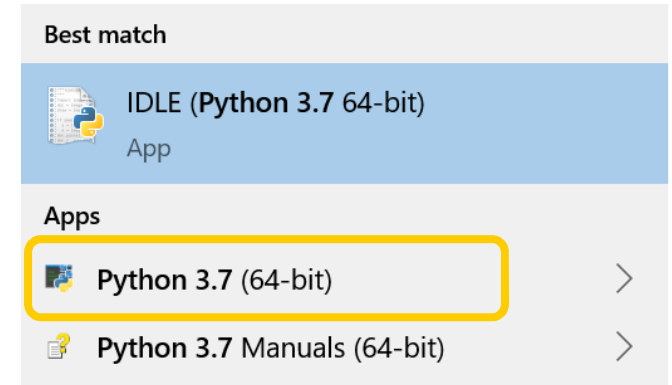
- Both options give you a *shell*



The Basic Idea of Using a Python Shell



Using a Python Shell – Some Simple Python Code



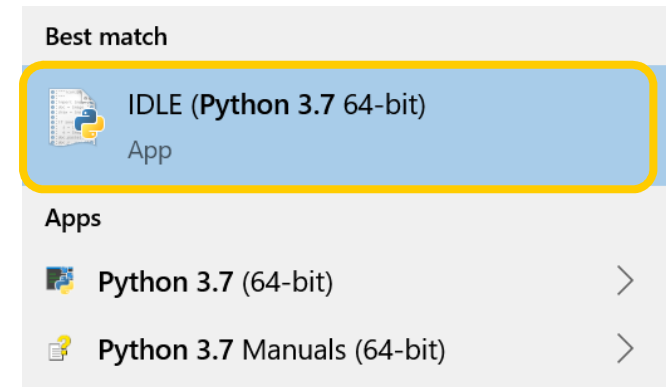
```
Python 3.7 (64-bit)

>>>
>>> print(1000 * 21)
21000
>>> print("there are these many seconds in one lecture:", 60 * 50)
there are these many seconds in one lecture: 3000
>>>
>>> print("Payment for 8 weeks, 15 hours per week =", 8 * 15 * 45)
Payment for 8 weeks, 15 hours per week = 5400
>>>
```

>>> is generated by the shell,
it means ‘this is where your input is shown’

Using the IDLE Environment

- The IDLE environment is better
- One reason is that colours are automatically used, which is sometimes very helpful for understanding
- We'll see other useful features of IDLE soon, especially in the lab work that we'll do



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("there are these many seconds in one lecture:", 60 * 50)
there are these many seconds in one lecture: 3000
>>>
>>> print("Payment for 8 weeks, 15 hours per week =", 8 * 15 * 45)
Payment for 8 weeks, 15 hours per week = 5400
>>>
```