COMP1021
Introduction to Computer Science

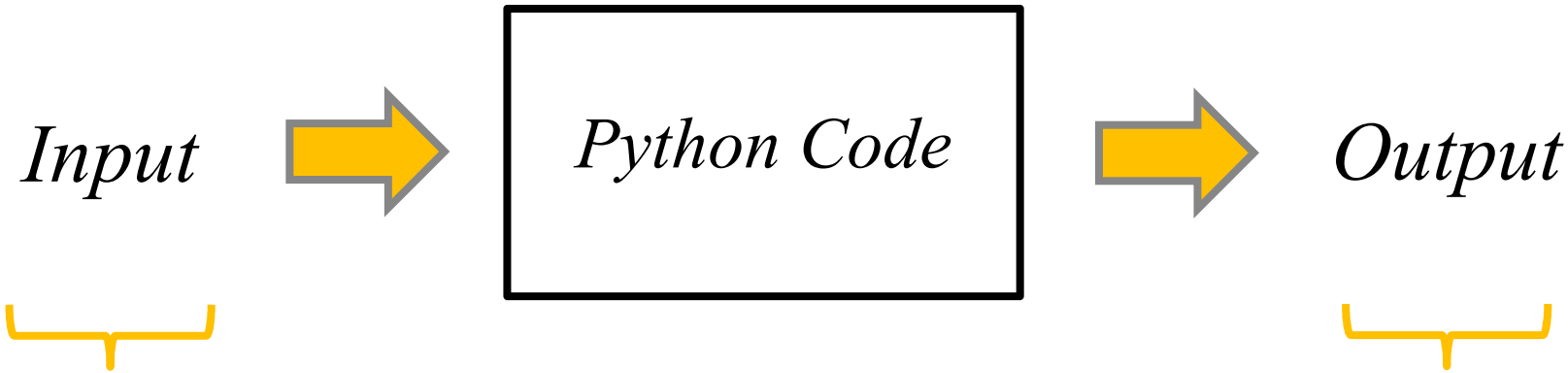
Beginning to Program Python

David Rossiter

Outcomes

- After completing this presentation, you are expected to be able to:
  1. Use Python code to do simple text input and output
  2. Use variables to store things, such as text and numbers
  3. Demonstrate running Python code as a program

Input and Output



- In this presentation we'll look at text input
- Later we will look at handling various other types of input such as mouse input

- In this presentation we'll look at text output
- Later we'll look at some graphics and music output

Text Output

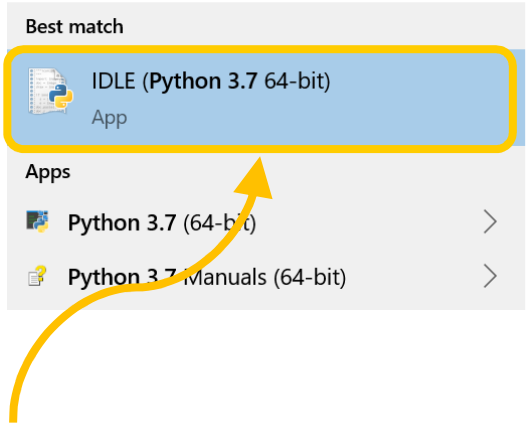
- Let's do some simple text output
- Here is a line of Python code which prints (i.e. output) a message:

```
print("It's a typhoon!")
```

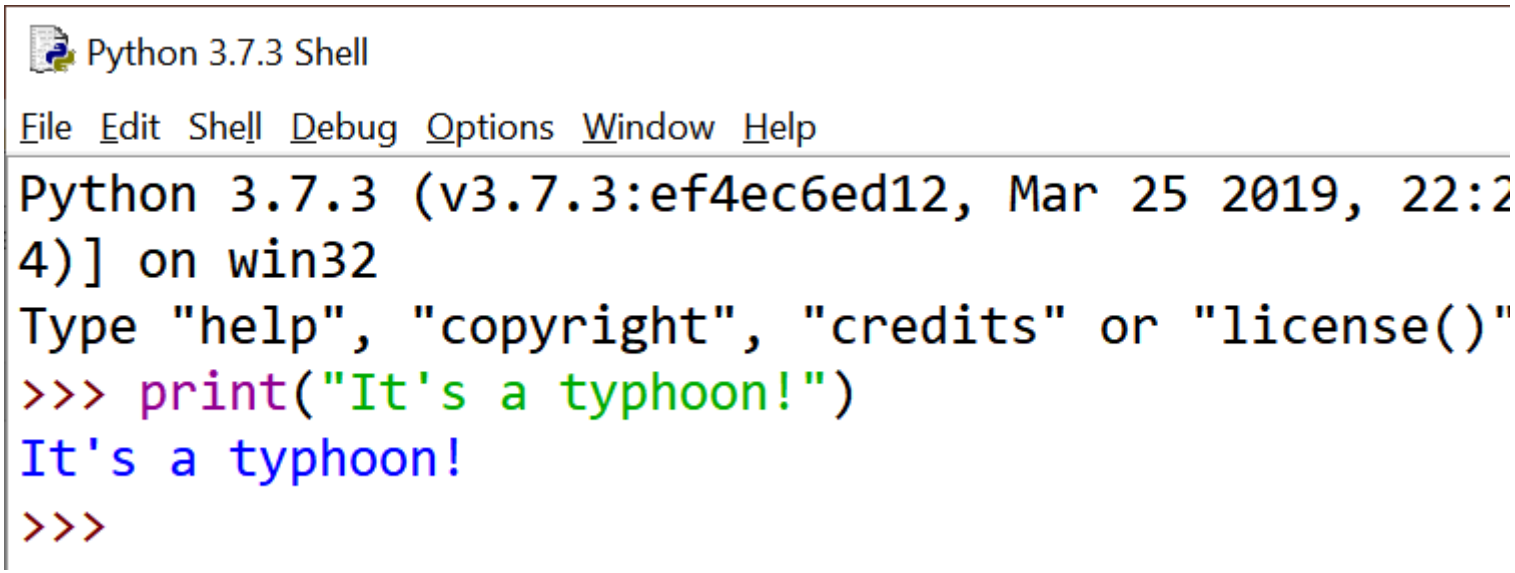
- This is the print command that asks Python to show something on the screen
- You put the message you want to show inside a pair of parentheses, i.e. `()`
- This is the message that we want to show on the screen
- When you use text in Python code, you need to enclose the text using a pair of double-quotes, i.e. `""`

Text Output

```
print("It's a typhoon!")
```



- If we type the code directly into the shell, it immediately gets executed and the result is shown:



- You can tell Python to print anything you like

Text Input

- Let's do some simple text input
- Here is a line of Python code which shows a message and lets the user enter something:

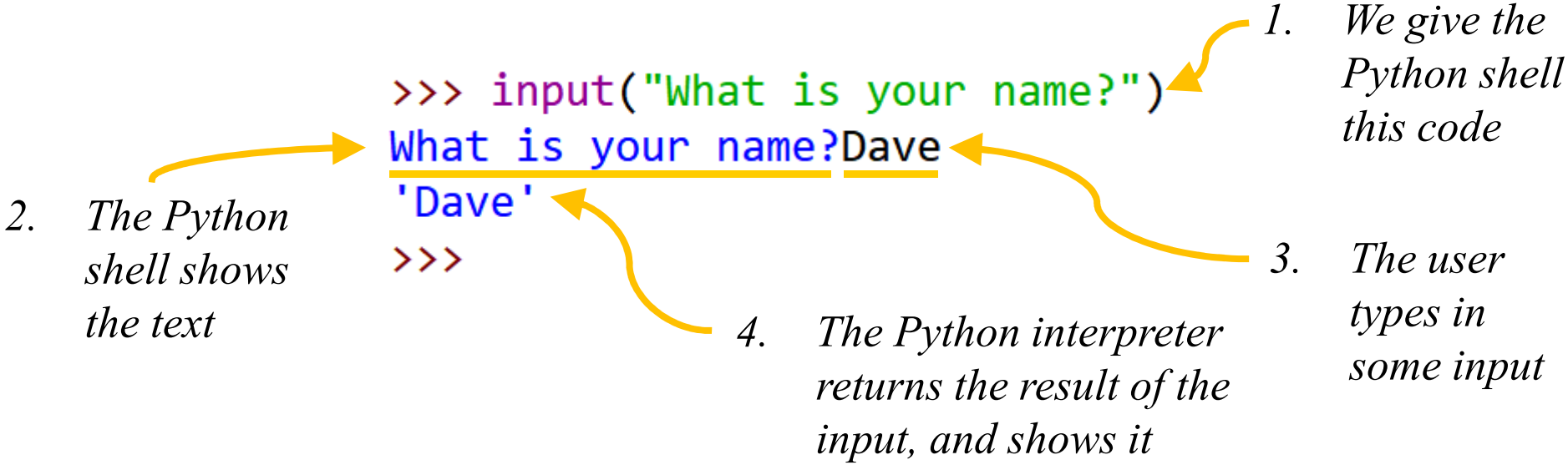
```
input("What is your name?")
```

- This is the input command which:
  - asks Python to show something on the screen, and
  - returns the thing that the user types
- This is the message that we want to show on the screen

Text Input

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

- If we type this code directly into the shell, it is immediately executed, the message is shown and the result the user types is returned:



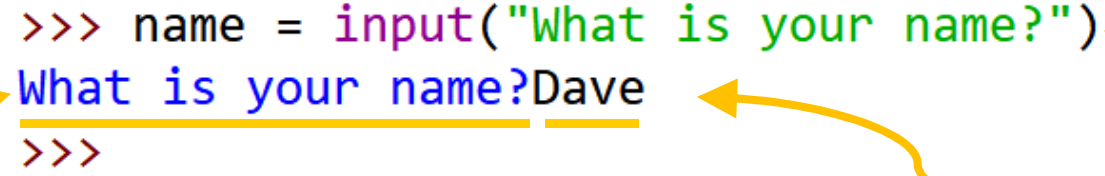
Remembering Things

- In the example we just looked at, whatever the user typed is completely ‘forgotten’ after Python has finished processing the input command
- Although you can still see things on the screen, Python doesn’t remember that
- It will be very useful if we have some way to remember what the user types
- To do that we need to use a *variable* to store it

Using a Variable

- You can think of a variable as a box
- Here is some code which stores whatever text the user enters in a variable:

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



1. *We give the Python interpreter this code*
2. *The Python interpreter shows the text*
3. *The user types in some input*
4. *The Python interpreter stores the input in the variable called 'name'*

Accessing the Variable

- If we want to use whatever is in the variable, we simply use the name of the variable
- For example, let's use `print()` to show what's in the variable:

```
>>> print(name)
Dave
```

- We could mix it with some text, like this:

```
>>> print("Your name is", name)
Your name is Dave
```

or this:

```
>>> print("Your name is", name, "and that's a great name!")
Your name is Dave and that's a great name!
```

What About Entering Numbers?

- If we want to get a number from the user, we can use the same input function `input()`
- However, `input()` always produces text
- You will encounter a problem if you try to treat the variable as if it has a number, like this:

```
>>> money = input("How much money do you have in your pocket?")
How much money do you have in your pocket?100
>>> print(money)
100
>>> moremoney = money + 5
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    moremoney = money + 5
TypeError: can only concatenate str (not "int") to str
>>>
```

Converting Text into a Number

- What we can do is to take the input from the user, and then convert it to a number using `int()`
- `int()` means 'convert this into an integer'
- After it has been converted, you can add, subtract, multiply, etc, the number stored in the variable

```
>>> money = input("How much money do you have in your pocket?")
How much money do you have in your pocket?100
>>> print(money)
100
>>> money = int(money)
>>> print(money)
100
>>> moremoney = money + 5
>>> print(moremoney)
105
```

Convert the *text* "100" into an *integer* 100

Generating a (Random) Number

- Sometimes it is useful to ask Python to give you some random numbers
- There are several ways to do that in Python
- One of them is to use the `random.randint()` command
- First, we need to use this code:

```
import random
```

- This code tells Python that we want to use a command related to random numbers

Generating a (Random) Number

- Then we can use `random.randint()` to generate a random number within a particular range, like this:

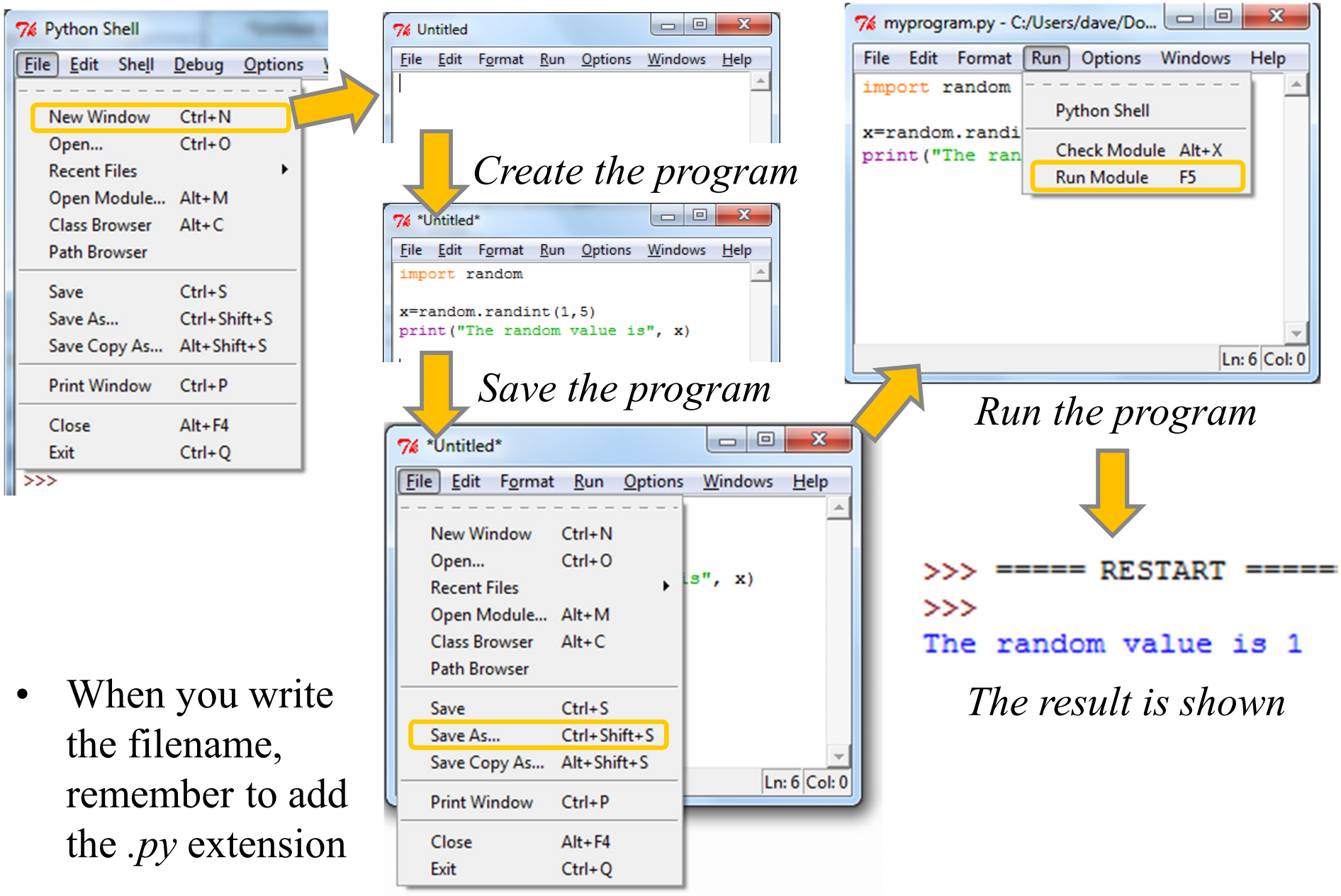
```
>>> import random
>>> random.randint(1, 10)
1
>>> random.randint(1, 10)
3
>>> random.randint(1, 10)
9
>>> random.randint(1, 10)
1
>>> random.randint(1, 10)
2
```

- We will use this technique to generate random numbers for games later

Putting Lines of Code Together

- Typing lines of code in the shell is OK but you may want to run the same lines of code many times
- You will go crazy if you have to keep typing them!
- It makes sense to put all the lines of code together into a single file of Python code
- That file, usually containing many lines of code, is called a *program*

Making and Running a Program



- When you write the filename, remember to add the *.py* extension