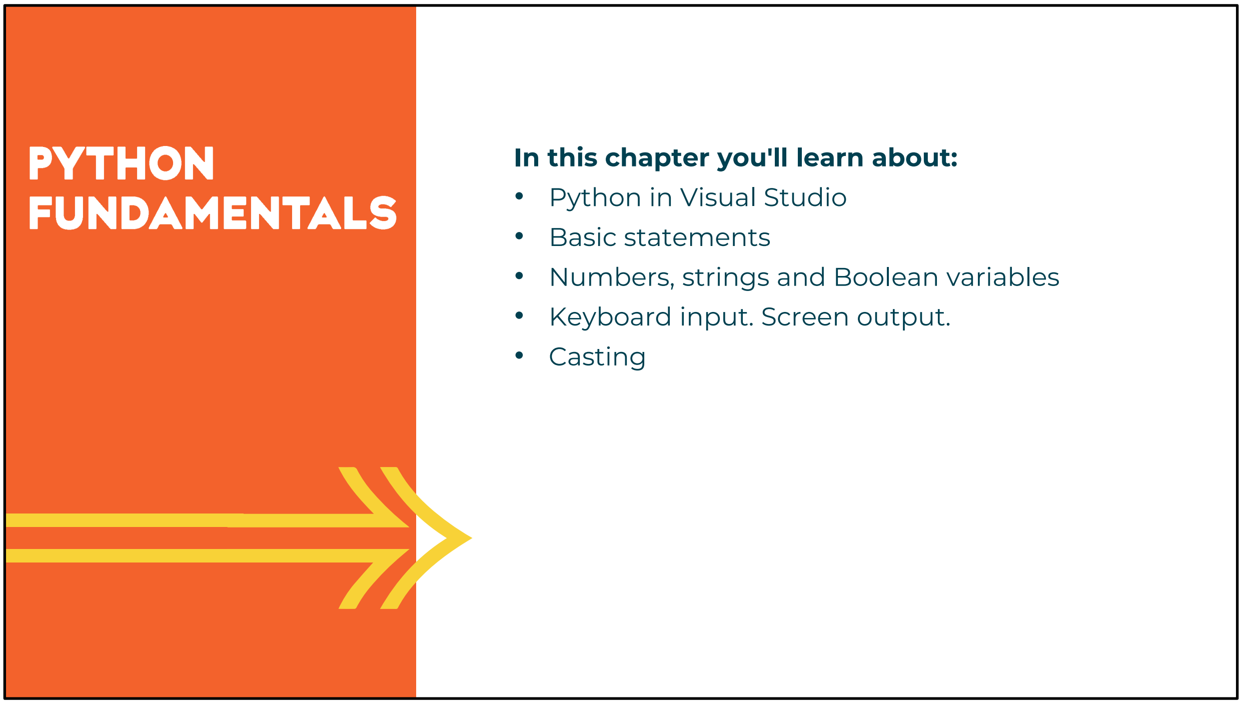
POWERING
POTENTIAL

Python programming basics



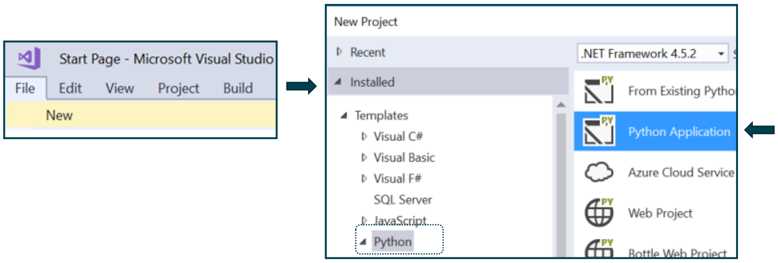
PYTHON FUNDAMENTALS

In this chapter you'll learn about:

- Python in Visual Studio
- Basic statements
- Numbers, strings and Boolean variables
- Keyboard input. Screen output.
- Casting

Why Visual Studio?

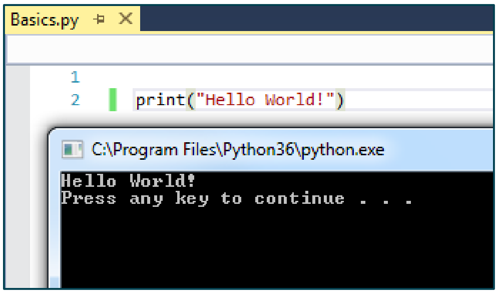
- Is a superb development environment
  - Excellent debugging and testing facilities



- There are many online compilers
  - https://repl.it/languages/python3

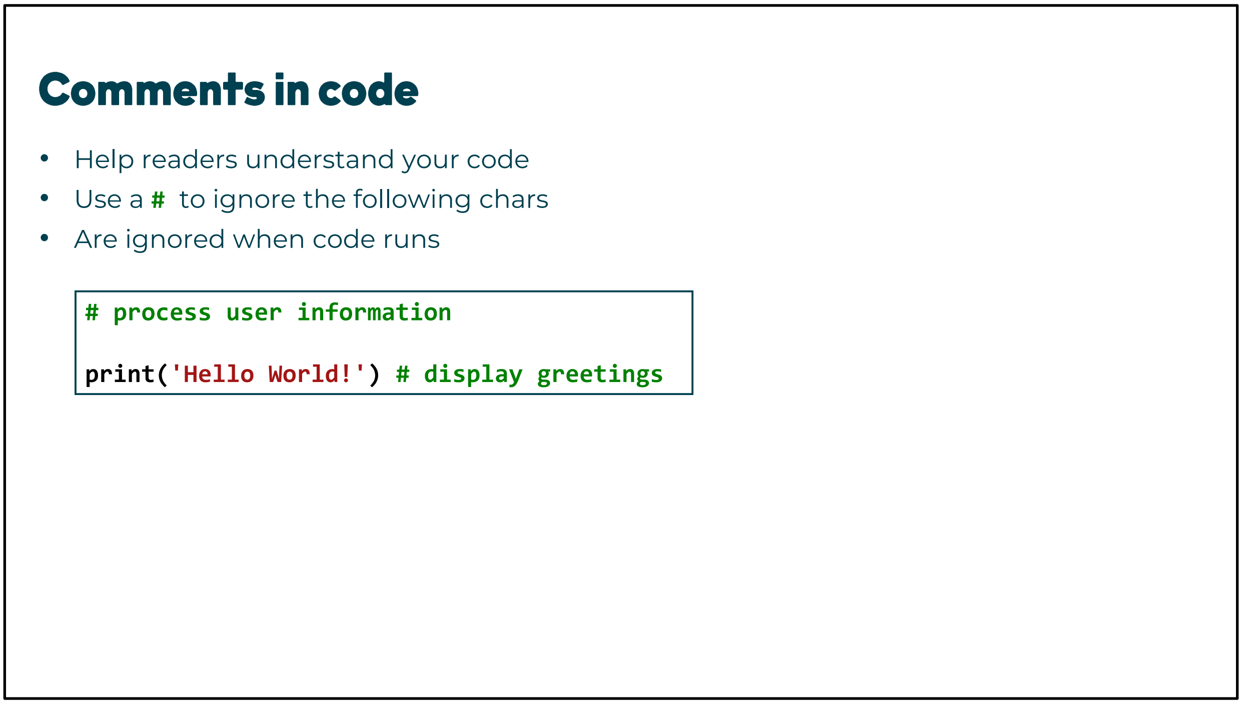
Printing to Console window

- print ('Hello World')
- Click ▶ Attach... ▾ or F5 key to run
- press *Ctrl*-F5 to run without debugging



```
print("Hello World!")
```

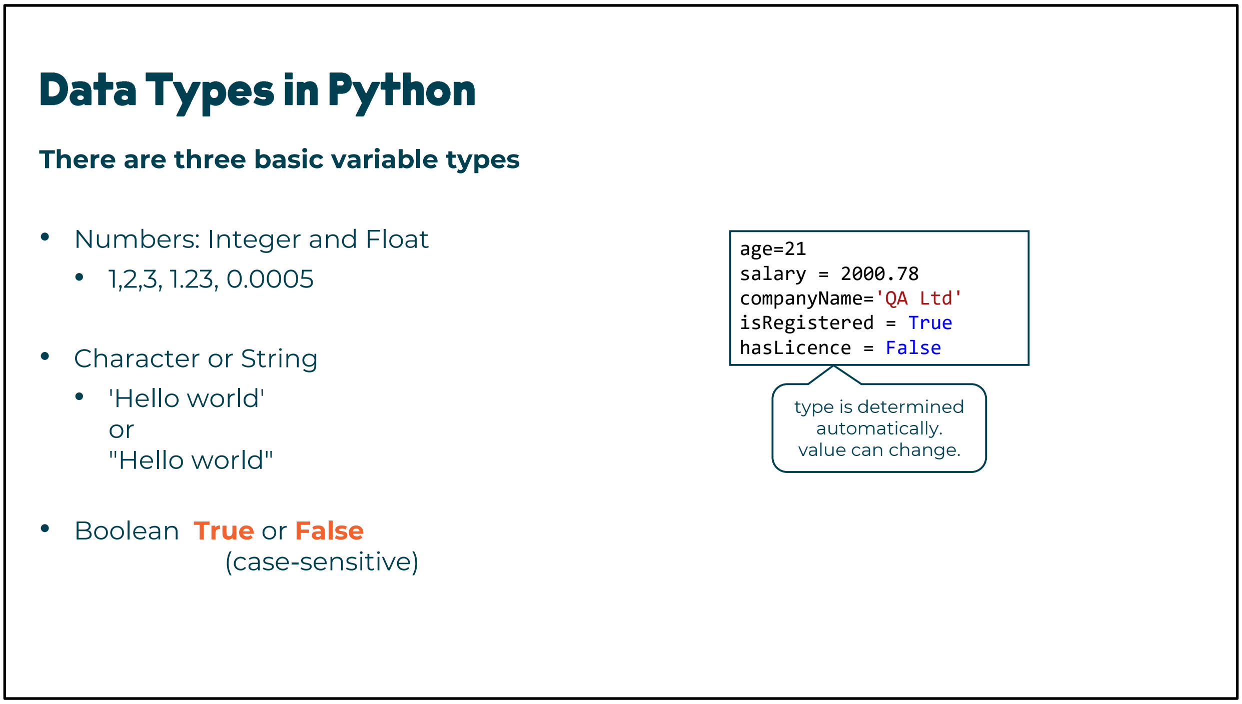
```
Hello World!
Press any key to continue . . .
```



Comments in code

- Help readers understand your code
- Use a # to ignore the following chars
- Are ignored when code runs

```
# process user information

print('Hello World!') # display greetings
```



Data Types in Python

There are three basic variable types

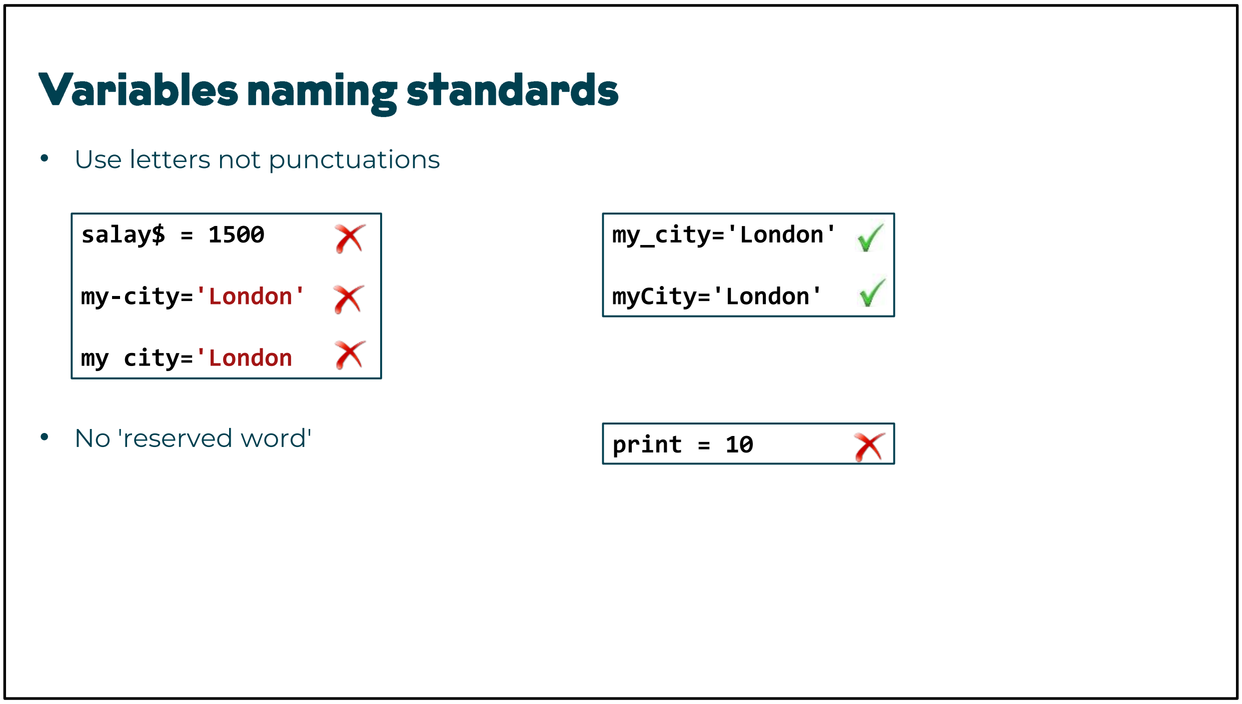
- Numbers: Integer and Float
  - 1,2,3, 1.23, 0.0005
- Character or String
  - 'Hello world'
    or
    "Hello world"
- Boolean **True** or **False**
  (case-sensitive)

```
age=21
salary = 2000.78
companyName='QA Ltd'
isRegistered = True
hasLicence = False
```

type is determined automatically.
value can change.

The best way to learn to code is to actually write the code yourself. You should do your best to follow along, writing code with me during the presentation. Example code is available in the Files section of Canvas.

You do need to know the basics of how programs work. They rely on *commands* and the *data* that the commands operate on. Data can be *literals*, i.e. actual numbers or pieces of text, held in *variables.*



Variables naming standards

- Use letters not punctuations

```
salay$ = 1500          ✗

my-city='London'       ✗

my city='London        ✗
```

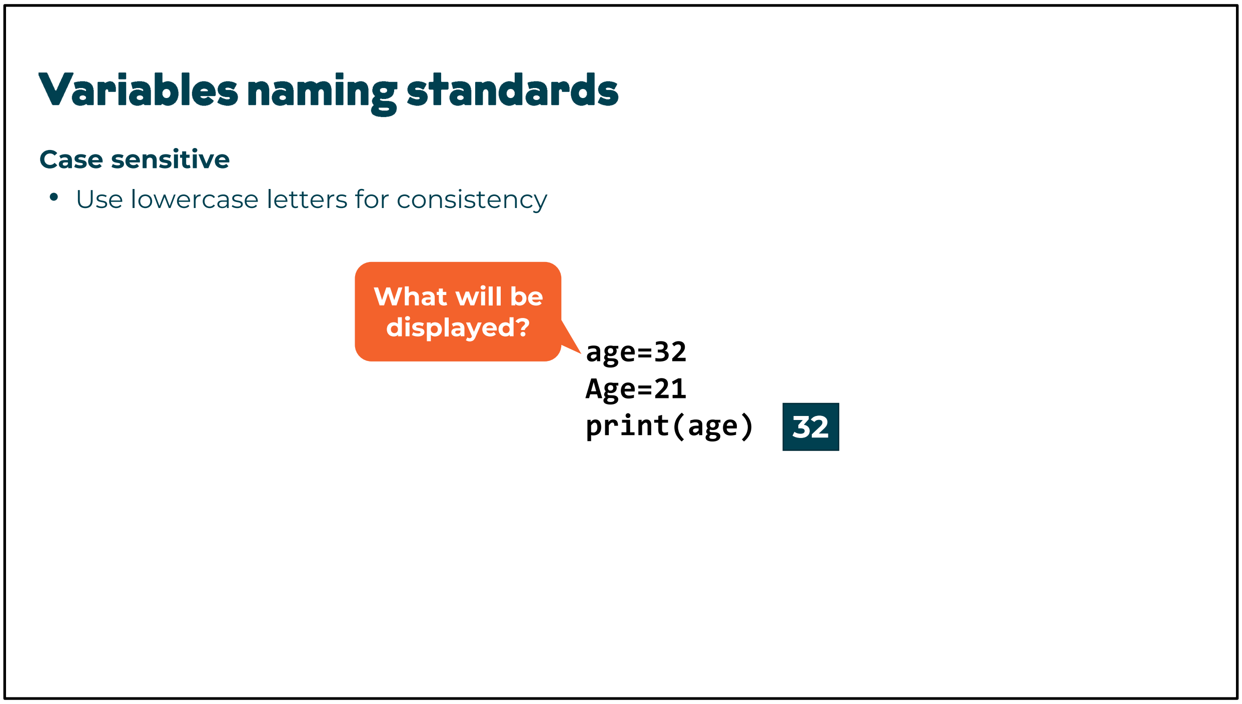
```
my_city='London'   ✓

myCity='London'    ✓
```

- No 'reserved word'

```
print = 10       ✗
```

A Python identifier is a name used to identify a variable, function, class, module or other object (more on functions, classes & modules later). An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).

Python does not allow punctuation characters such as @, $, and % within identifiers. Like most (but not all) programming languages, Python is case sensitive so **Manpower** and **manpower** are two different identifiers in Python.



Variables naming standards

Case sensitive

- Use lowercase letters for consistency

What will be displayed?

```
age=32
Age=21
print(age)
```

32

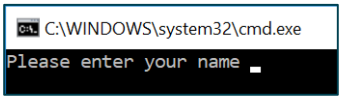
A Python identifier is a name used to identify a variable, function, class, module or other object (more on functions, classes & modules later). An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).

Python does not allow punctuation characters such as @, $, and % within identifiers. Like most (but not all) programming languages, Python is case sensitive so **Manpower** and **manpower** are two different identifiers in Python.

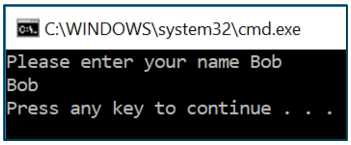
User input using keyboard

- **Input with a prompt**
  `input`***(<prompt>)***

```
name = input('Please enter your name ')
print(name)
```



```
Please enter your name _
```



```
Please enter your name Bob
Bob
Press any key to continue . . .
```

Python in Visual Studio does not have the interactive command mode of the IDLE editor so we must run every line of code as a program. This is not normally an issue.
This does mean that we need to output program results to see what has happened. For this we use the Visual Studio output window.

Our input device is the keyboard. In this lesson we will see how to use both these devices.

Putting strings together

```
username = 'Bob'
print('Hello' , username)
print('Hello' + username)
print('Hello ' + username)
```

```
Hello Bob
HelloBob
Hello Bob
```

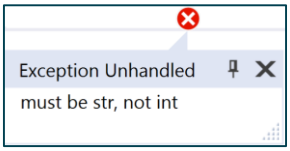
Cannot add numbers and strings

```
age = 21
print('Your age is ' + age)
Message = 'Your age is ' + age
```

✗

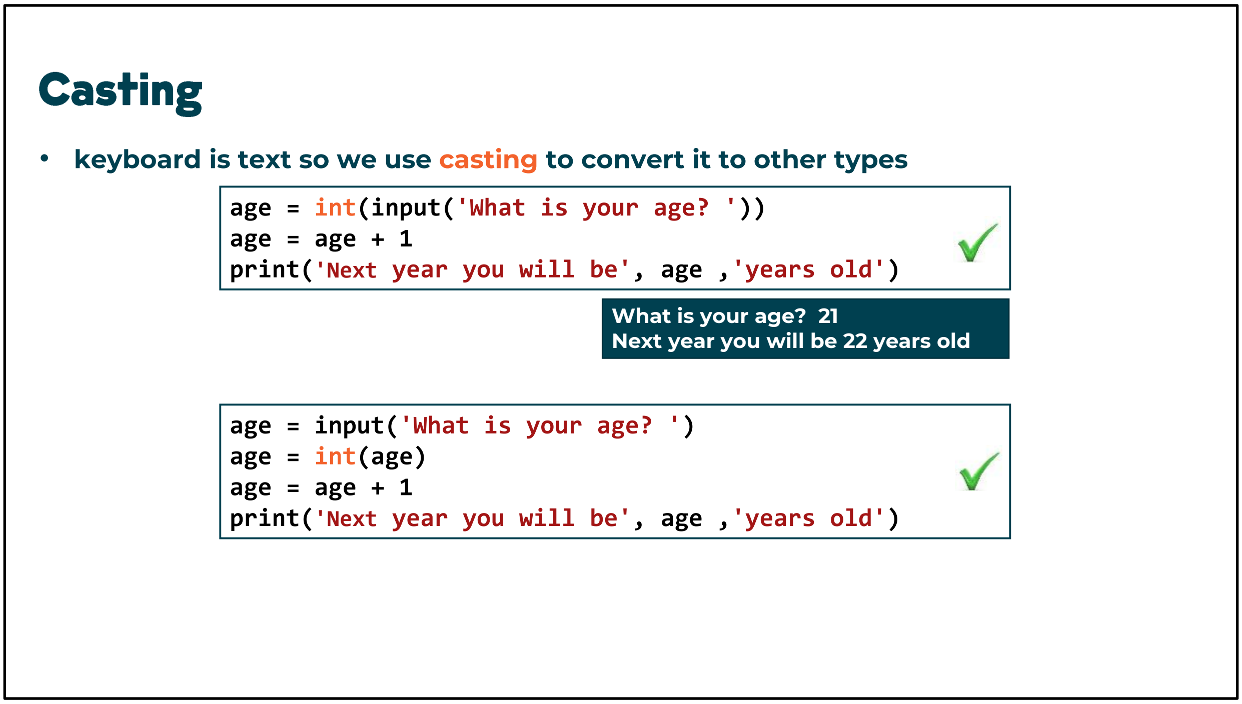
keyboard input is always text... even if it 'looks' like a number

```
age = input('Please enter your age ')
age = age + 1
```



Exception Unhandled

must be str, not int



Casting

- keyboard is text so we use **casting** to convert it to other types

```
age = int(input('What is your age? '))
age = age + 1
print('Next year you will be', age ,'years old')
```

✓

```
What is your age?  21
Next year you will be 22 years old
```

```
age = input('What is your age? ')
age = int(age)
age = age + 1
print('Next year you will be', age ,'years old')
```

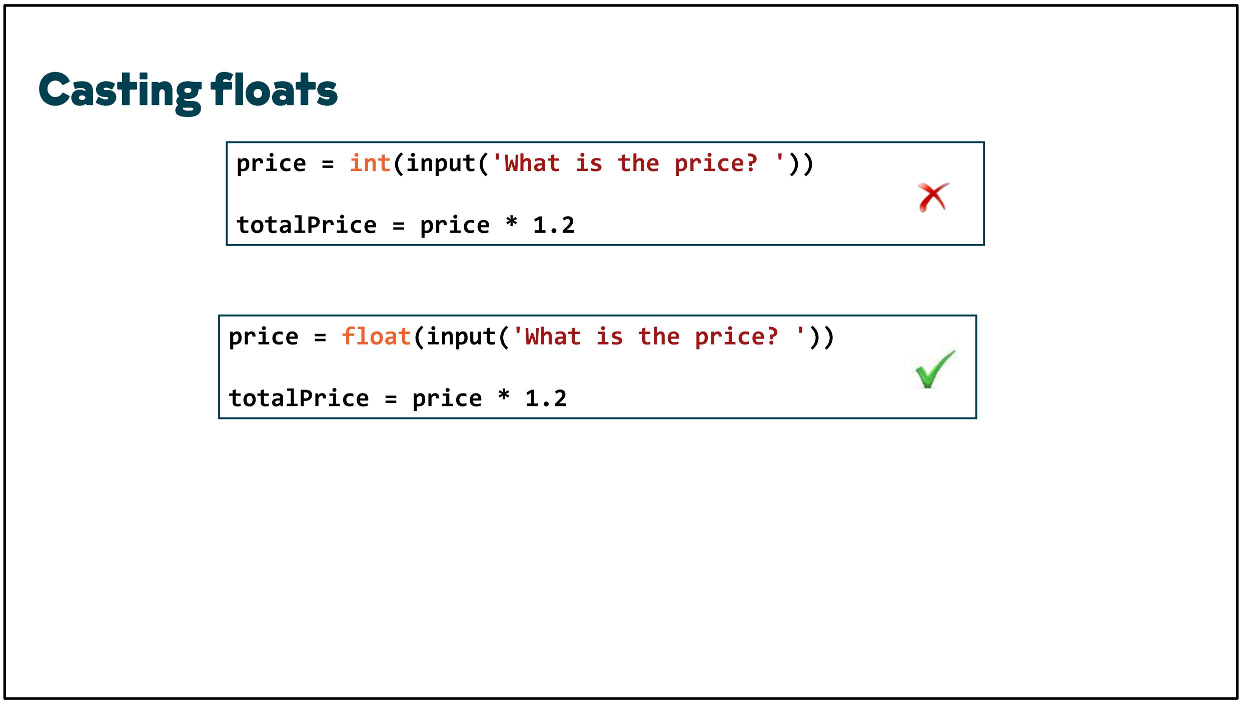
✓

The slide shows two methods of casting. The first example is more common.

Casting a number to string

- Use the str() function

```
## Find the average of a few numbers

total = 1 + 3 + 5 + 7 + 9 + 11
average = total / 6

print("Total is = " + str(total))
print("Average is = " + str(average))
```



Casting floats

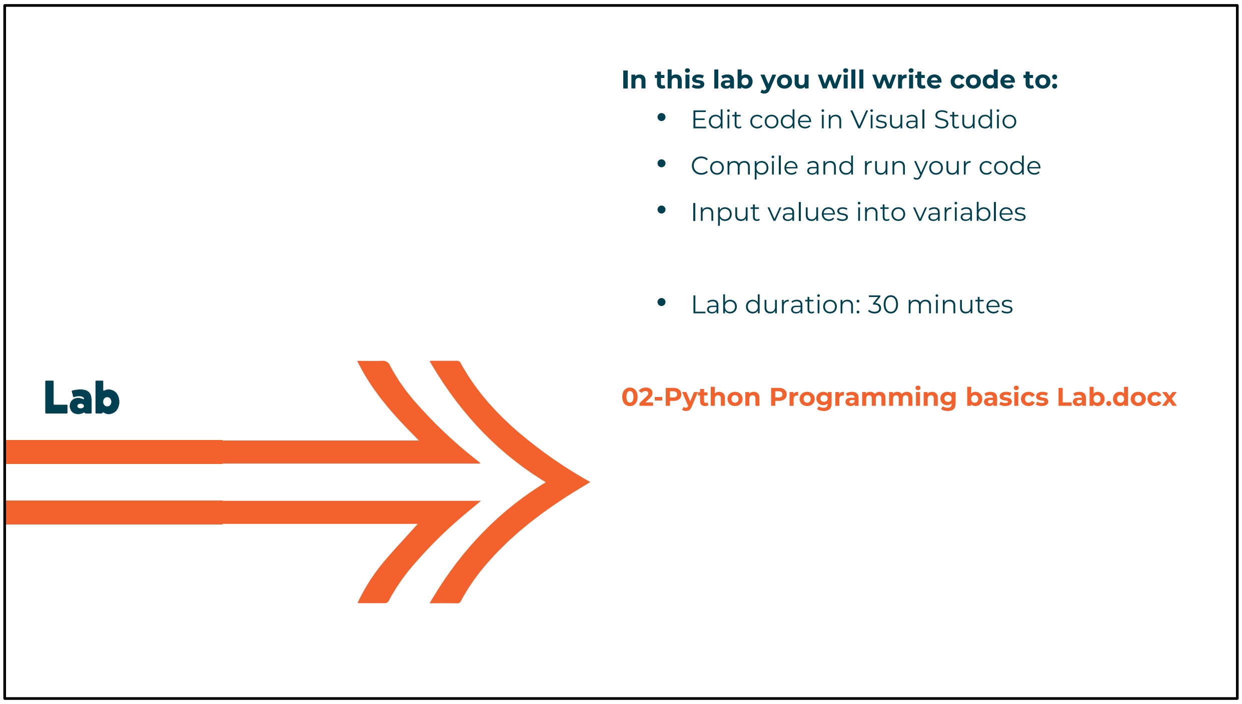
```
price = int(input('What is the price? '))

totalPrice = price * 1.2
```

✗

```
price = float(input('What is the price? '))

totalPrice = price * 1.2
```

✓

Integers cannot take part in float operations. Cast to float to perform such operation as seen on the slide.



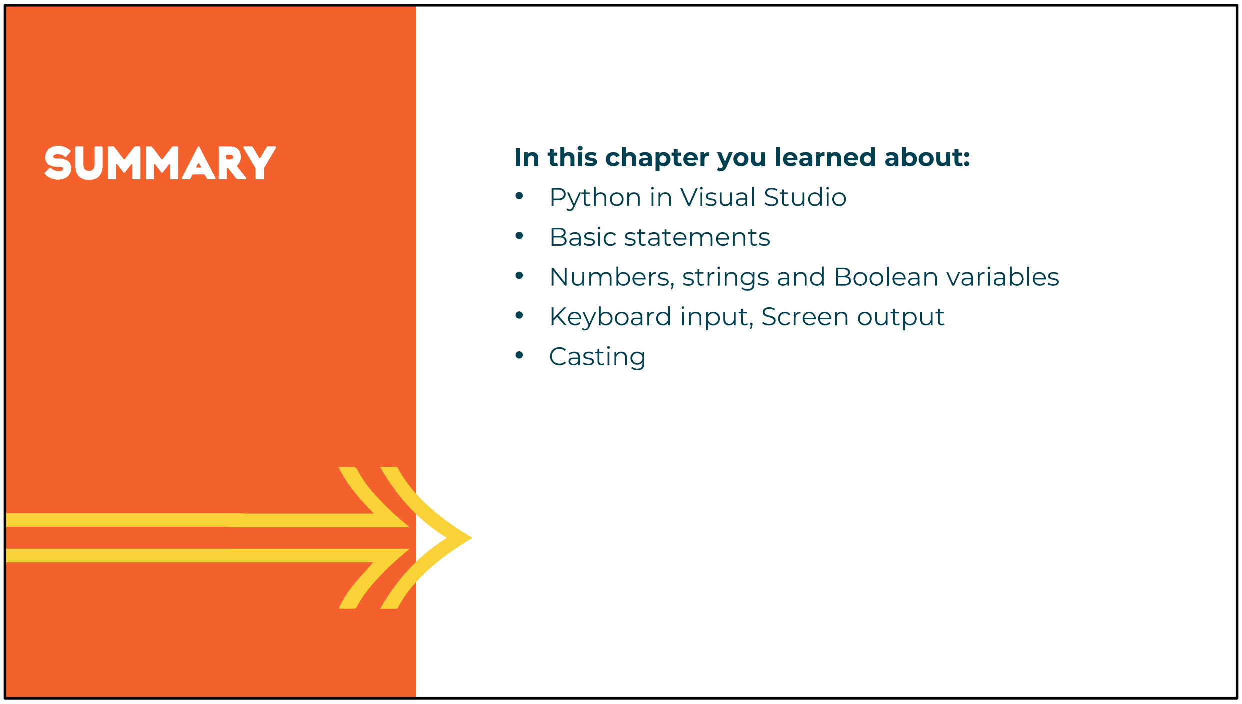
Lab

In this lab you will write code to:

- Edit code in Visual Studio
- Compile and run your code
- Input values into variables

- Lab duration: 30 minutes

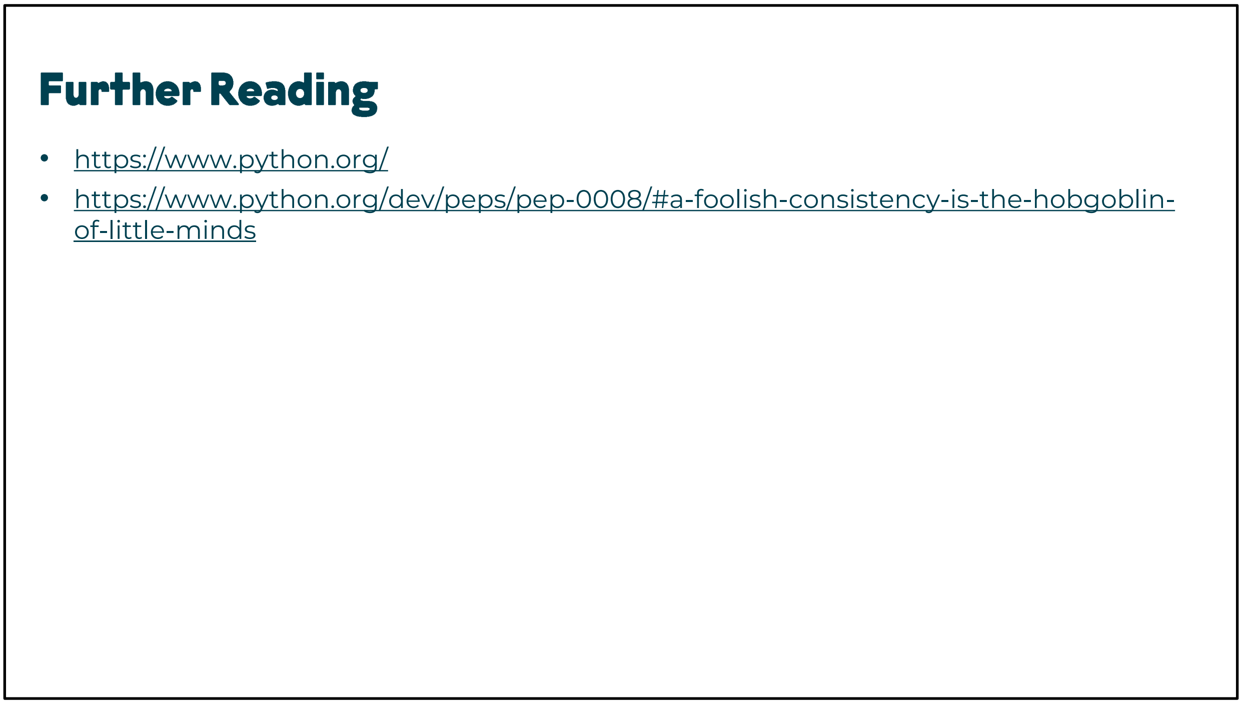
02-Python Programming basics Lab.docx



SUMMARY

In this chapter you learned about:

- Python in Visual Studio
- Basic statements
- Numbers, strings and Boolean variables
- Keyboard input, Screen output
- Casting



Further Reading

- https://www.python.org/
- https://www.python.org/dev/peps/pep-0008/#a-foolish-consistency-is-the-hobgoblin-of-little-minds

POWERING POTENTIAL

Thank you