CITY
UNIVERSITY OF LONDON
EST 1894

STEM Digital Academy

School of Science & Technology

www.city.ac.uk

# Input and Output

Programming and Algorithms

Lecture by

Dr Daniil Osudin

```
n = 3
for i in range(1,n+1):
    print("Hello World!")
```

```
Hello World!
Hello World!
Hello World!
```

# What will we Cover?

- Input and output in Python
- Using input and output in programs
- Writing programs with user input

# Examples

| Input | Processing | Output |
|---|---|---|
| Recipe ingredients | Cooking | Cake |
| Dirty dishes<br>Washing up liquid | Washing | Clean dishes |
| Materials | Construction | Building |
| Closed door<br>Key | Unlock | Open door |

# Input and Output Uses

Output:

- See the results of the calculations
- Save the processed data
- Create a graphical representation of the data

Input:

- Operate on the data provided by the user (rather than pre-defined values in the code)
- Change the behaviour of the program while it is executing

# Examples

- Setting the temperature on the AC
- Entering a phone number to call a specific person
- Entering GPS coordinates into a navigator and being given a route
- Entering numbers and operations on a calculator and being displayed the results

# Output in Python I

`print()` statement

- Handles output to the console
- Contains one or more outputs within the brackets `()`
- Outputs can be of type string, int, float, Boolean or an expression

# Output in Python II

`print()` statement examples

```
x = 1
print(x)
```

```
1
```

Int

```
b = True
print(b)
```

```
True
```

Boolean

```
x = 1
print(1+x)
```

```
2
```

Expression

```
y = 1.0
print(y)
```

```
1.0
```

Float

```
c = "a"
print(c)
```

```
a
```

String

```
x = 1
print(type(x))
```

```
<class 'int'>
```

CITY UNIVERSITY OF LONDON EST 1894

# Output in Python III

`print()` statement with multiple outputs

- Multiple outputs are separated by a comma ,
- In the output console space is added automatically between multiple outputs

```
print(1, "a")
```

```
1 a
```

```
print(2>1, type(2.0))
```

```
True <class 'float'>
```

Space added automatically

# Input in Python I

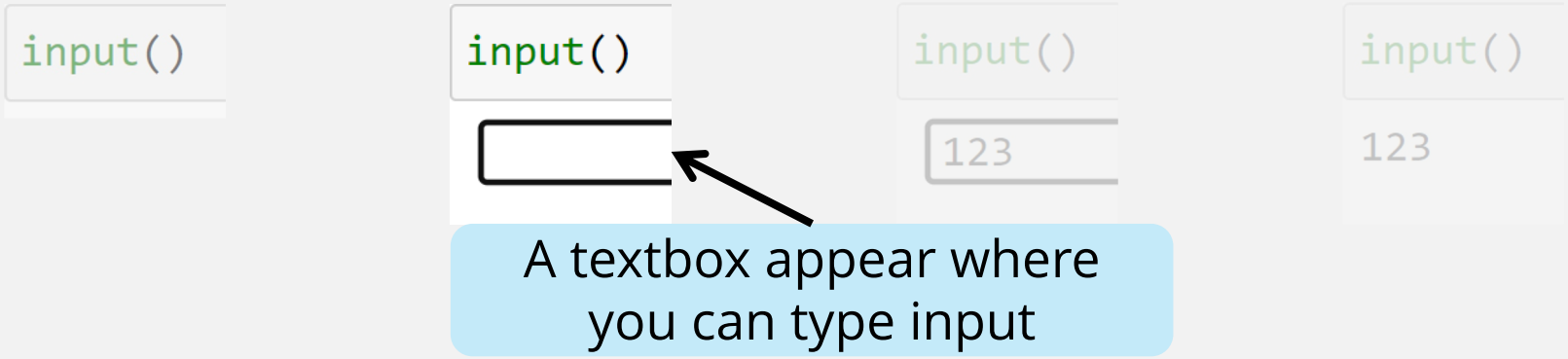
`input()` statement

- Handles input from the console typed by the user
- Program waits until user types the input followed by the return key (usually looks like this: ↵ )

# Input in Python I

`input()` statement

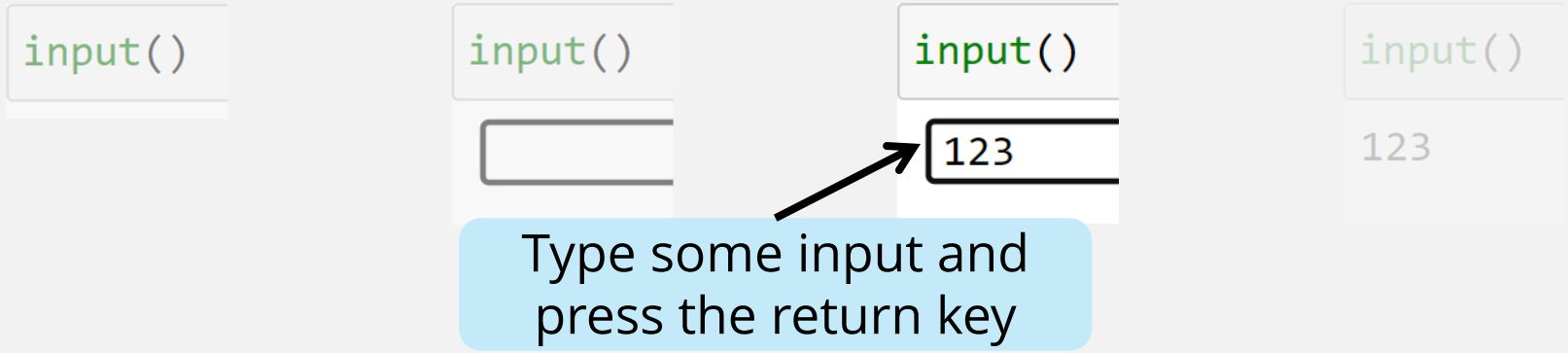
- Handles input from the console typed by the user
- Program waits until user types the input followed by the return key (usually looks like this: ↵ )



# Input in Python I

`input()` statement

- Handles input from the console typed by the user
- Program waits until user types the input followed by the return key (usually looks like this: ↵ )

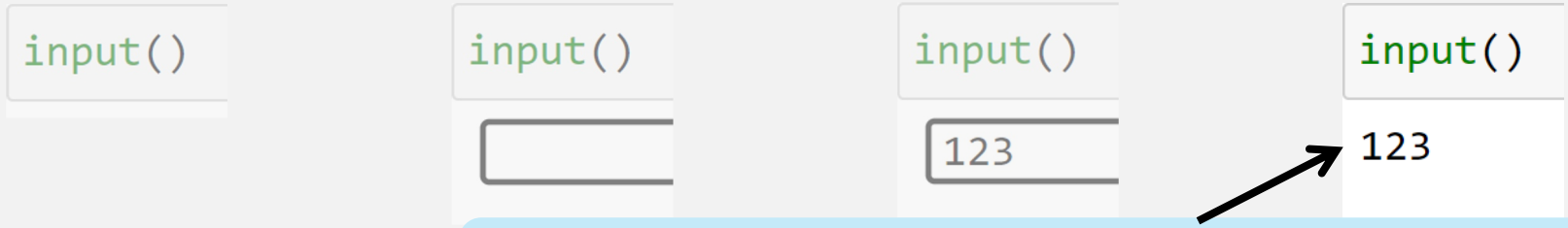



CITY UNIVERSITY OF LONDON EST 1894

# Input in Python I

`input()` statement

- Handles input from the console typed by the user
- Program waits until user types the input followed by the return key (usually looks like this: ↵ )




CITY UNIVERSITY OF LONDON EST 1894

# Input in Python II

`input()` statement

- User's input usually stored in a variable using assignment statement
- Can display a message for the user by entering it in the brackets `()`

# Input in Python III

## `input()` statement examples

```
a = input("Please enter a number: ")
print("You entered:", a)
```

```
Please enter a number: 1
```

```
a = input("Please enter a letter: ")
print("You entered:", a)
```

```
Please enter a letter: a
```

```
x = input("Please enter a number: ")
print("You entered:", x)
```

```
Please enter a number: 1
You entered: 1
```

```
a = input("Please enter a letter: ")
print("You entered:", a)
```

```
Please enter a letter: a
You entered: a
```

CITY UNIVERSITY OF LONDON EST 1894

# Input in Python III

## `input()` statement examples

```
a = input("Please enter a number: ")
print("You entered:", a)
```

```
Please enter a number: 1
```

```
a = input("Please enter a letter: ")
print("You entered:", a)
```

```
Please enter a letter: a
```

```
x = input("Please enter a number: ")
print("You entered:", x)
```

```
Please enter a number: 1
You entered: 1
```

```
a = input("Please enter a letter: ")
print("You entered:", a)
```

```
Please enter a letter: a
You entered: a
```

CITY UNIVERSITY OF LONDON EST 1894

# Input in Python IV

`input()` statement input type

- input() function call returns a string

```
x = int(input("Please enter a number: "))
print("You entered:", x)
```

```
Please enter a number: a
```

- For numeric input, explicit conversion is required
- A 'TypeError' will occur otherwise when the numeric input is manipulated in the program

CITY
UNIVERSITY OF LONDON
EST 1894

# Input in Python V

Type error when manipulating input

```
x = input("Please enter a number: ")
x += 1
print("You entered:", x)
```

```
Please enter a number: 1
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_14652\2317829844.py in <module>
      1 x = input("Please enter a number: ")
----> 2 x += 1
      3 print("You entered:", x)

TypeError: can only concatenate str (not "int") to str
```

Error location

Error description

Error type

# Input in Python VI

Constraining the type returned by `input()`

```
x = int(input("Please enter a number: "))
print("You entered:", x)
```

```
Please enter a number: a
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_9860\1505648100.py in <module>
----> 1 x = int(input("Please enter a number: "))
      2 print("You entered:", x)

ValueError: invalid literal for int() with base 10: 'a'
```

Error location

Error type

Error description

# Try It Yourself

Enter and run the following statements in the python environment:

```
x = input()
print(x)
```

```
n = float(input())
n = n / 2
print(n)
```

```
w = int(input())
w = w + 9
print(w)
```

```
b = int(input())
print(not b)
```