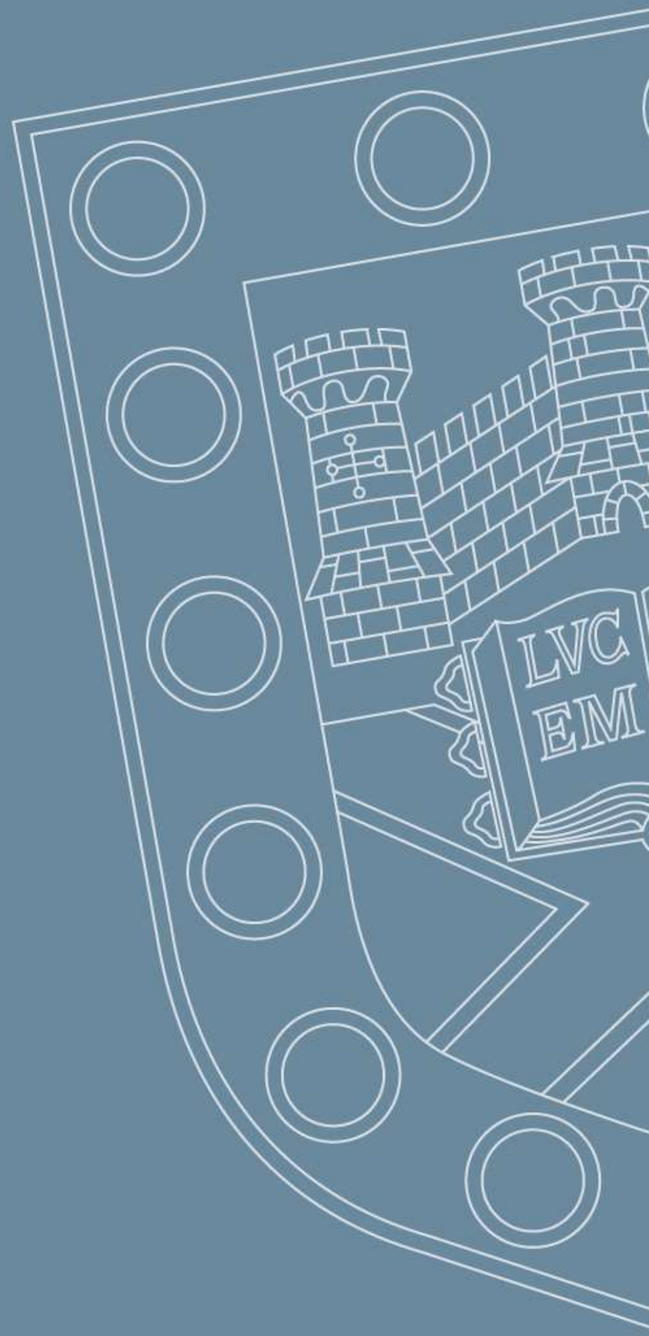
UNIVERSITY OF EXETER

Python programming

Variables



institute of CODING

VARIABLES

Variables, literals and comments

A computer program is nothing more than lines of text.

Symbols and punctuation do more work than we're used too.

Here's an example.

```
# An example
message = "This is fine"
```

VARIABLES

Variables, literals and comments

```
# An example
message = "This is fine"
```

# indicates the rest of the first line is a comment

message is the name of a variable

The words enclosed in quotation marks " are a string literal

VARIABLES

Types - numbers

```
# integer numbers
n = 12
p = 0b1100
q = 0xc
# floating point (real) numbers
val = 0.01
val = 1e-2

# complex numbers
vec = 1.4142+1.4142j
```

VARIABLES

Types - strings

```
msg_from = "Alice"
msg_to = 'Bob'
msg_body = '''Use three quotes (' or ") to start, and
end, a multi-line string. '''
empty = ""
```

VARIABLES

Types – Boolean (logical or truth)

```
send_message = True
await_reply = False
```

VARIABLES

Naming variables

In these examples I’ve used a variety of names for variables.

Python doesn’t care what you call your variables, but there are a few rules, and also best practice.

- Only use letters, numbers and underscore
- Start with a letter. Underscore allowed, but ..
- Avoid words that have meaning to Python, or start with underscore. Python keywords are not allowed.

VARIABLES

An exercise

Assign values to two variables.

Name the variables -

`name_1`

`name_2`

Can we write a program that will swap the values over?

VARIABLES

An exercise

```
name_1 = "Bob"

name_2 = "Alice"
```

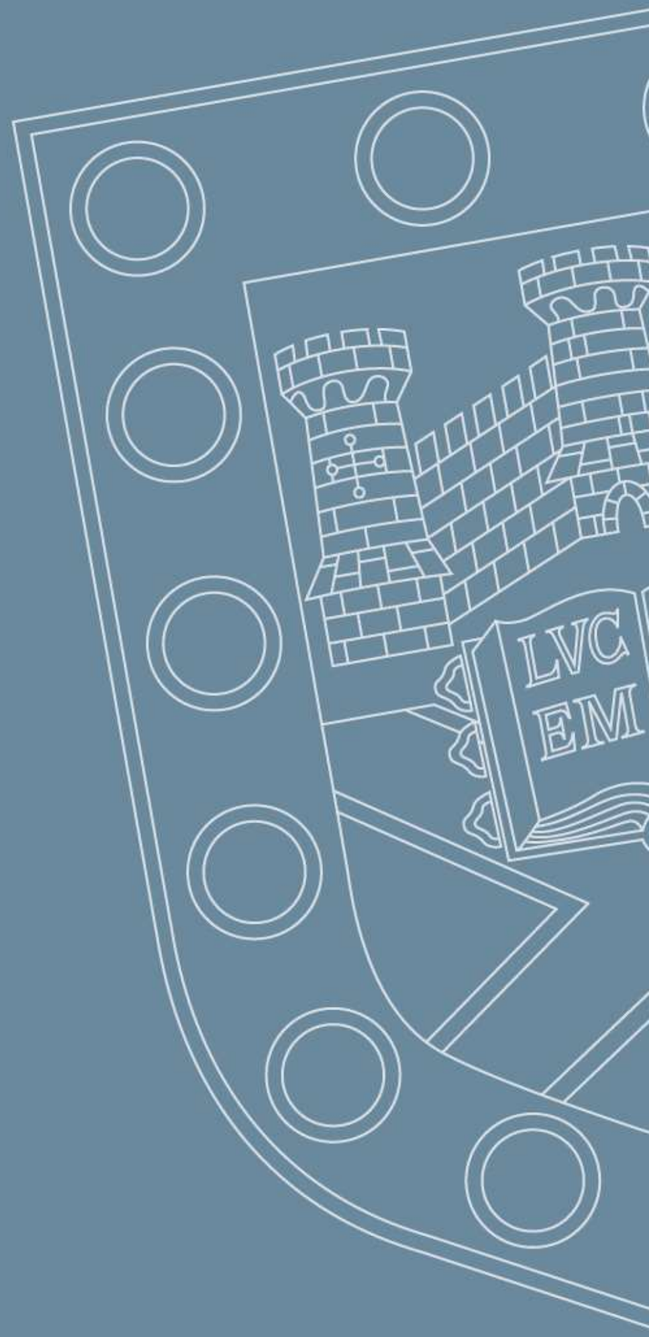
To show the value of variables, or literals, we use **print** statements.

```
print(name_1)

print(name_2)
```

Now write a program to swap the values.

UNIVERSITY OF EXETER



institute of CODING