

# Python Strings

## Help Sheet

- A **character** is anything you can type on a keyboard - letters, numbers, symbols, even spaces!

Examples: H ; 8 p ? b

- A **string** is a list of characters in order.



- To create a string in Python, we have to give it a name.

```
In [1]: text = "This is a string."
```

In this example, the name of the string is **text**.

To create the string we put the contents in **inverted commas ""**.

- We can get the computer to **print** our string using the command:

```
In [3]: print(text)
This is a string.
```

- Each character in a string is labelled by its position. Instead of starting on 1, strings start counting from 0!



- We can ask the computer to tell us what character is in each position using **square brackets [ ]**. For example, if we wanted to know what character is in position 0:

```
In [11]: print(text[0])
T
```

Or what character is in position 5:

```
In [29]: print(text[5])
i
```

- We can find the length of our string (ie. how many characters) using the **len()** command.

```
In [26]: print(len(text))
17
```

- We can get the computer to change the string to upper and lower case using the **upper()** and **lower()** commands.

```
In [20]: ▶ print(text.upper())
print(text.lower())
THIS IS A STRING.
this is a string.
```

- We can see how many times a certain character has appeared in a string by using the **count()** command.

```
In [24]: ▶ print(text.count("s"))
print(text.count("."))
3
1
```

Here we can see the "s" character appeared 3 times, and the "." character appeared once.

- We can replace characters in the string with other characters using the **replace()** command. In the brackets, we first put the characters we want to be replaced, and then the characters we want to replace them.

```
In [25]: ▶ print(text)
print(text.replace("a", "my"))
This is a string.
This is my string.
```