Basic string manipulation | Part A

A string is a sequence of characters – this might be letters, numbers, spaces or punctuation marks.

In a program we represent a string constant by putting it in quotation marks.

There are many programs we use every day that make use of strings. When you log in to your favourite website, your username is a string and your password is a string too.

We can do lots of different things with strings!

A very useful thing we can do is to join two or more strings together – this is called concatenation. It's like getting some glue and squishing the two strings together to make a new string. For example:

```
food1 = "cheese"
food2 = "cake"
new_food = food1 + food2
```

This will glue "cheese" to "cake" and result in "cheesecake"! Mmm.

The concatenation operator in Python looks the same as the plus sign. Look at this example:

```
first_name = "Donald"
last_name = "Duck"
full_name = first_name + last _name
```

This results in the string "DonaldDuck" which is not quite right – there should be a space. If you want spaces in your concatenation you have to add them yourself, like this:

```
full_name = first_name + " " + last_name
```

Notice we concatenated in a string containing a space – it just looks like a pair of empty quotes.

When you decided on your password, it probably had to be a certain length – let's say between 6 and 12 characters long. We can ask the user to type in a password and check its length using the len() function.

To use len(), we just plug the string we want to find the length of into the brackets:

len("cake") → 4
len("cheese") → 6

We can also plug in a variable that is a string, like this:

```
password = input("Enter your password")
if len(password) < 6:
   print("Password too short")
elif len(password) > 12:
  print("Password too long")
else:
  print("OK")
```

Now we can tell the user whether their password is an acceptable length.

Sometimes we might want to change a string that a user has typed in – users don't always behave themselves! We could convert the string to all capital letters:

```
print ( "hello".upper() )
```

…or all lower case letters.

Upper and lower work slightly differently to len() because they are **methods** – you plug the string in *first* and then add .lower() or .upper() on the end.

We can use the .replace() method to replace one set of characters with another:

```
phrase = "My favourite food is cake"
new_phrase = phrase.replace("cake", "cheese")
print( new_phrase )
```

This will take the phrase and look for all occurrences of the string "cake", and then replace them with the string "cheese". We end up with 'My favourite food is cheese'.