**Strings**

We’ve used strings a few times already. A string is basically text and is wrapped in either single or double quotes.

my\_string = “Here’s some text.”

my\_other\_string = ‘Here is “another” string’

**Concatenation and string formatting**

We can join multiple strings together, also known as concatenation, using a few methods. The most basic is through using the + operator.

**strings.py**

string\_one = "one"

string\_two = "two"

joined\_string = string\_one + " " + string\_two

print(joined\_string)

Technically we’ve already done a form of concatenation in the previous units, something like:

print(“apples:”, apples)

In a print statement we can print out multiple values at the same time by separating them with commas. This also puts a space between each value automatically. However this only works in print statements and sometimes you may not want that added space.

But using the + operator for concatenation can get unwieldy with longer operations. Consider the following:

**greeting.py**

name = "Will"

age = 32

print("Hello " + name + ", you are " + str(age) + " years old.")

While it is still relatively simple it’s not exactly the nicest code to read. There is a more sophisticated method we can use that looks neater and doesn’t require the use of ‘str()’ to change numbers to strings.

Python has a built-in function called ‘.format()’ which can be used to find and replace markers we set in a string. It is best demonstrated in the following example:

**greeting\_2.py**

name = "Will"

age = 32

greeting = "Hello {}, you are {} years old.".format(name, age)

print(greeting)

This is the simplest way to use ‘.format()’. It will find and replace each {} pair with the next value or variable given to it, in this case ‘name’ then ‘age’.

To find out what else you can do with ‘.format()’ you can visit https://pyformat.info/.

It’s up to you which method you use, though I would recommend choosing one and sticking to it. However, be aware that I will be using ‘.format()’ where possible in future examples.