Strings

Explanation

`String` is the technical name for text. To define a block of code as a string, you need to include it in either double quotes (") or single quotes ('). It doesn't matter which you use so long as you are consistent.

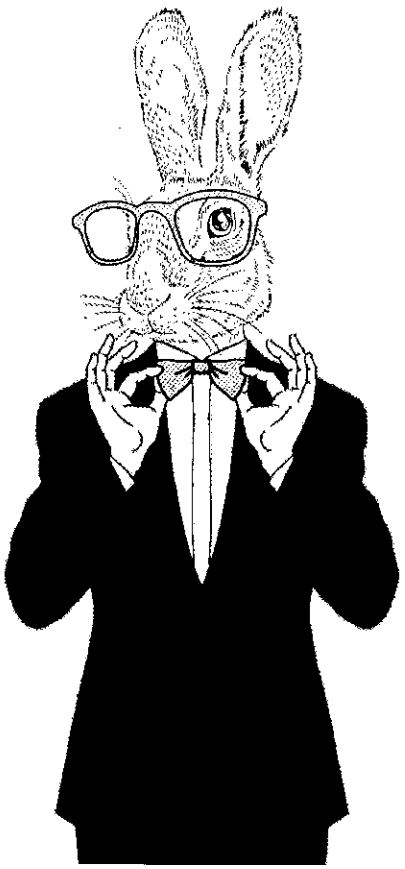
There are some characters you need to be particularly careful with when inputting them into strings. These include:

" ' \

That is because these symbols have special meanings in Python and it can get confusing if you use them in a string.

If you want to use one of these symbols you need to precede it with a backslash symbol and then Python will know to ignore the symbol and will treat it as normal text that is to be displayed.

Symbol	How to type this into a Python string
"	\"
'	\'
\	\\



Strings and Numbers as Variables

If you define a variable as a string, even if it only contains numbers, you cannot later use that string as part of a calculation. If you want to use a variable that has been defined as a string in a calculation, you will have to convert the string to a number before it can be used.

```
num = input("Enter a number: ")
total = num + 10
print(total)
```

In this example, the author has asked for a number, but has not defined it as a numeric value and when the program is run they will get the following error:

```
Enter a number: 45
Traceback (most recent call last):
  File "C:/Python34/CHALLENGES/String/example.py", line 2, in <module>
    total = num + 10
TypeError: Can't convert 'int' object to str implicitly
>>>
```

Although this error message looks scary, it is simply saying that the line **`total = num + 10`** isn't working as the variable num is defined as a string.

This problem can be solved in one of two ways. You can either define it as a number when the variable is being originally created, using this line:

```
num = int(input("Enter a number: "))
```

or you can convert it to a number after it has been created using this line:

```
num = int(num)
```

The same can happen with strings.

```
name = input("Enter a name: ")
num = int(input("Enter a number: "))
ID = name + num
print(ID)
```

In this program, the user is asked to enter their name and a number. They want it joined together and with strings the addition symbol is used as **concatenation**. When this code is run you will get a similar error message to before:

```
Enter a name: Bob
Enter a number: 23
Traceback (most recent call last):
  File "C:/Python34/CHALLENGES/String/example.py", line 3, in <module>
    ID = name + num
TypeError: Can't convert 'int' object to str implicitly
>>>
```

To get around this, either don't define the variable as a number in the first place or convert it to a string afterwards using the line:

```
num = str(num)
```

Multiple-Line Strings

If you want to input a string across multiple lines you can either use the line break (**\n**) or you can enclose the entire thing in triple quotes. This will preserve the formatting of the text.

```
address="""123 Long Lane
Oldtown
AB1 23CD"""
print(address)
```

Example Code

Please note: In the following examples, the terms word, phrase, name, firstname and surname are all variable names.

`len(word)`
Finds the length of the variable called word.

`print(word.capitalize())`
Displays the variable so only the first word has a capital letter at the beginning and everything else is in lower case.

`name = firstname+surname`
Joins the first name and surname together without a space between them, known as concatenation

`word.upper()`
Changes the string into upper case.

`word.lower()`
Changes the string into lower case.

`word.title()`
Changes a phrase so that every word has a capital letter at the beginning with the rest of the letters in the word in lower case (i.e.Title Case).

```
text = " This is some text. "
print(text.strip(" "))
```
Removes extra characters (in this case spaces) from the start and end of a string.

`print ("Hello world"[7:10])`
Each letter is assigned an index number to identify its position in the phrase, including the space. Python starts counting from 0, not 1.

0	1	2	3	4	5	6	7	8	9	10
H	e	l	l	o		w	o	r	l	d

Therefore, in this example, it would display the value of positions 7, 8 and 9 ,which is "orl".

Don't forget that you can reuse previous programs to save time when you are making new programs. Simply use SAVE AS and give it a new name.