Objective 3:

Understand string manipulation functions

In this objective you learn how to extract and change data that has been input.

Tasks

1. Try entering the following commands and see what happens:

```
#Working with strings
forename=input("Enter your surname: ")
forename_uppercase=forename.upper()
print("Your name in capital letters is:",forename_uppercase)
```

2. Change the program to ask for an email address, outputting the string in lowercase using `.lower()` instead of `.upper()`.

3. Try entering the following commands and see what happens:

```
#Len returns the number of characters in a string
surname = input("Enter your surname: ")
length_name = len(surname)
print("There are",length_name,"letters in your name.")
```

4. Try entering the following commands and see what happens:

```
#[:?] returns a number of characters to the left of a string
sentence = "I saw a wolf in the forest. A lonely wolf."
characters = sentence[:5]
print(characters)
```

5. Change the program to output the last 12 characters in the sentence using `[-12:]` instead of `[:5]`.

6. Try entering the following commands and see what happens:

```
#[start:end] returns a number of characters in the middle of a string
sentence = "I saw a wolf in the forest. A lonely wolf."
characters = sentence[20:26]
print(characters)
```

7. Try entering the following commands and see what happens:

```
#find returns the location of one string inside another
sentence = "I saw a wolf in the forest. A lonely wolf."
print(sentence)
word = input("Enter the word to find: ")
position = sentence.find(word)
print("The word",word,"is at character",position)
```

Objective 3: Key learning points

String manipulation functions

- Strings can be manipulated using built in functions and methods to extract characters from the left, right or middle of a string.
- You can find if one string exists inside another string.
- A built in function takes data to use in parenthesis (brackets), called a **parameter** and returns a result. E.g. `int(parameter)`
- A method (signified with a dot) applies an operation to itself. E.g. `"Hello".upper()`

Note, it is possible for functions and methods to return their value to another command rather than a variable. For example: `print("Hello World".find("World"))` works because the resultant value from find becomes the parameter for print.

It is common in programming to use as few variables as necessary in order to conserve memory. This makes the program more efficient, but often more difficult to understand.

Objective 3: Key words

.upper()

Example code: `x = y.upper()`

Purpose: To turn a string into uppercase.

x is the name of the variable to return the result to. y is the original variable.

.lower()

Example code: `x = y.lower()`

Purpose: To turn a string into lowercase.

len()

Example code: `x = Len(y)`

Purpose: To return the number of characters in a string.

x becomes the number of characters in y.

str()

Example code: `x = Str(4)`

Purpose: To return a string from a number.

If you wanted to output an integer or float variable with a string, you need to convert the number to a string first. E.g. to output 4 * 3 =

This would not work: question = number1 + "*" + number2 + "="

Instead, you need to cast the numbers to strings like this:
question = str(number1) + "*" + str(number2) + "="

[:?]

Example code: `x = y[:z]`

Purpose: To return characters to the left of a string.

x becomes the characters. y is the string to extract characters from. z is the number of characters to extract from the left.

[-?:]

Example code: `x = y[-z:]`

Purpose: To return characters to the right of a string.

x becomes the characters. y is the string to extract characters from. z is the number of characters to extract from the right.

[?:?]

Example code: `x = y[w:z]`

Purpose: To extract characters from the middle of a string.

x becomes the middle characters of y, starting from position w, up to z number of characters.

.find()

Example code: `x = y.find(z)`

Purpose: To find the position of substring z in y.

X becomes the position in the string y where z can be found. E.g. x = “Hello World”.find(”World”) returns 6 as the letter W is the sixth character in the string (starting at zero).