# Strings

A string is a sequence of characters. You can loop through the characters in a string, unlike an integer or real number. These exercises will help you manipulate strings for your programming.

## Indexing and slicing strings

The index of a string is a position compared to the rest. Programming languages often start counting at 0, so the first character in a string is position 0. To slice a string is to take a portion of it, from the front, middle or end.

```
pizza = “Hawaiian”
```

| **Position** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| **String characters** | H | a | w | a | i | i | a | n |

1. Match the code with the correct output

| **Code** | **Output** |
|---|---|
| `print(pizza[0])` | wa |
| `print(pizza[:5])` | a |
| `print(pizza[0:2])` | Hawai |
| `print(“pizza[6]”)` | Hawaiian |
| `print(pizza)` | H |
| `print(pizza[6])` | aiian |
| `print(pizza[2:4])` | pizza[6] |
| `print(pizza[3:])` | Ha |

2. Write a program to ask the user for their full name. It should output:
    a. The whole name
    b. the first character
    c. the first 5 characters
    d. the fourth, fifth and sixth characters

## Concatenating strings

This means joining multiple strings together. A plus symbol (+) is used in Python.

```
greeting = "Hello"
name = "Elizabeth"
```

3. Match the code with the correct output

| Code | Output |
|---|---|
| print(greeting + name) | error |
| print(greeting + " " + name) | Hi Elizabeth |
| print("Hi " + name) | Elizabeth Elizabeth |
| print(Hi + name) | HelloElizabeth |
| print(name + " " + name) | ElizabethElizabeth |
| print(name * 2) | Hello liz |
| print(greeting + " " + name[1:4]) | Hello Elizabeth |

4. Create a program which asks for a name, adjective, noun, verb and adverb. It will then output silly sentence back to the user, including the words input.

## Case conversion

You can check if a user has entered in data in the expected case (UPPER or lower).

The syntax is:

```
varname.upper() or varname().lower
```

For example:

```
name = "Elizabeth"

print (name.UPPER())

>>> 'ELIZABETH'
```

5. Write a program which asks for a name.
    a. It returns the name in lower case
    b. It returns the name in UPPER case
    c. It returns the name with the first letter as UPPER case and the rest lower (You will need to use string slicing as above)
6. Write a program that asks for a letter. It should check which case the letter is and return a message to tell the user. You will need to use an IF statement.

## Find the length of a string

The len() function returns an integer – the number of items in a sequence, such as a string, list or dictionary.

The syntax is:

```
name = "Elizabeth"

print (len(name))

>>> 9
```

7. Write a program which asks for a username of 6 characters in length. It should check the length of the data input and tell the user whether they have entered six characters or not.

## Checking the contents of a string

You can use str.isalpha() to check if it contains only letters and str.isdigit() to check if it contains only numbers. 'str' is the name of the string and is likely to be a variable. These methods are Boolean expressions and will therefore return `True` or `False`.

```
username = "6666ML"
```

8. Match the code with the correct output:

| Code | Output |
|---|---|
| `print("Billy".isalpha())` | `True` |
| `print(username.isalpha())` | `True` |
| `print(username[0].isdigit())` | `False` |
| `print(username[5].isalpha())` | `True` |

9. Edit your program which asks for a username. It should check that the first four characters are alpha and the fifth and sixth are digits.

## Finding data in strings

It is very similar to English to check if a value occurs within a string. `True` or `False` will be returned when you ask Python if a string contains a value.

```
name = "Elizabeth"
letter = "f"
print ("E" in name)
True
print (letter in name)
False
```

## Iterating through strings

Repeating instructions, or iteration, is often a more efficient method of coding. A 'loop' is created where certain instructions are executed and the program returns back to the beginning of the loop.

If you want to perform an action on each letter of the string, you can use a 'for' loop. These are used when you know how many iterations you need to do at the beginning of the loop.

```
name = “Elizabeth”
for letter in name:
      print (letter.upper()*5)
>>> EEEEE
>>> LLLLL
>>> IIIII
>>> ZZZZZ         #etc

#islower() is a Boolean method which returns True or False, similar
#to isdigit() and isalpha(). It checks strings or characters are #lowercase

password = “qWerTy”
newstring = ““
for letter in password:
      if letter.islower() == True:
            newstring += letter #Append letter to newstring
print(newstring)

>>> ’qery’
```

10. Why does the program output ‘qery’? Describe what the program does.

| |
|---|
| |

11. Write a program which asks for a string of data. It will code the data by substituting some of the characters. Make it as complicated as possible! Here are some ideas:

a. Make the first character uppercase
b. Replace vowels with the next one. A is replaced by E, E by I etc.
c. Replace all occurrences of S or s with $
d. Replace all numbers 1-4 with double the number

**Example:**

password1234 would become Pe$$wurd2468