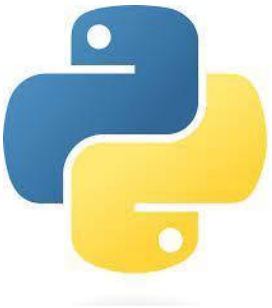
Code Clock

Tutorial 3: Strings

Learn.to.code



Programming with Python

@ QUB

Strings

Strings are one the most efficient tools for handling text data. A python string is a list of characters in order. A character is anything you can type on the keyboard in one keystroke, like a letter, a number, or a backslash. Python strings are immutable i..e they cannot be changed. Python recognize as strings everything that is delimited by quotation marks (" " or ' ').

String Manipulation

To manipulate strings, we can use some of Pythons built-in methods.

Create a String

To create a string with given characters, you can assign the characters to a variable after enclosing them in double quotes or single quotes as shown below.

Enter the following code to print a string word

```
text = "Code Clock: Time to Code"
print (text)
```

Access characters in a String

To access characters of a string, we can use the python indexing operator [ ] i.e. square brackets to access characters in a string as shown below.

Enter the following code to print the first letter of the variable "text"

```
text = "Code Clock: Time to Code"
letter=text[0]
print ("letter")
```

Find Length of a String

To find the length of a string, we can use the len() function. The len() function takes a string as input argument and returns the length of the string as shown below.

Enter the following code to which counts and prints the number of characters in the variable "text"

```
text = "Code Clock: Time to Code"
print("The variable word contains " + len(text)+ " characters")
```

Find a Character in a String

To find the index of a character in a string, we can use the find() method. The find() method, when invoked on a string, takes the character as its input argument and returns the index of first occurrence of the character as shown below.

Enter the following code to which will reveal the location of the character "k" in the variable "text"

```
text = "Code Clock: Time to Code"
print ("The character k is found at location " +
str(text.find("k"))
# find the character k in the string
```

You can also perform string manipulation in python to find the frequency of a character in the string. For this, we can use the count() method. The count() method, when invoked on a string, takes a character as its input argument and returns the frequency of the character as shown below.

Enter the following code which counts and prints the number of times the character 'c' appears in the variable "text"

```
text = "code clock: time to code"
# count how many times 'c' is in the string text
print("The character \'c\' was found " + str(text.count('c')) + "
in the string word"
```

You can also find the index of a character or a substring in a string using the index() method. The index() method, when invoked on a string, takes a character or substring as its input argument and returns the index of first occurrence of character or substring as shown below.

Enter the following code which will find the index of a word in a string

```
text = "code clock: time to code"
# find the letters code in the string
Print("The word code can be found "at index " +
str(text.index("code")))
```

Count the number of spaces in a string

To count the number of spaces in a string, you can pass the space character to the count() method as shown below.

Enter the following code which will count the number of whitespaces in the string text

```
text = "code clock: time to code"
print ("The number of whitespaces in the string text is  " + str(text.count(' ')))
```

String Slicing

To perform string manipulation in Python, you can use the syntax

string_name[start_index : end_index]

to get a substring of a string. Here, the slicing operation gives us a substring containing characters from start_index to end_index-1 of the string string_name.

Enter the following code which will slice the text variable using parameters

```
text = "code clock: time to code"

print text[0] #get one char of the text
print text[0:1] #get one char of the text (same as above)
print text[0:3] #get the first three char
print text[:3] #get the first three char
print text[-3:] #get the last three char
print text[3:] #get all but the three first char
print text[:-3] #get all but the three last character

text = "code clock: time to code"

start=0
end=8
print(text[start:end]) # items start through end-1
print(text[start:]) # items start through the rest of the list
print(text[:end]) # items from the beginning through end-1
print(text[:]) # a copy of the whole list
```

Split Strings in Python

You can split a string using the split() method to perform string manipulation in Python. The split() method, when invoked on a string, takes a character as its input argument. After execution, it splits the string at the specified character and returns a list of substrings as shown below.

Enter the following code which will split the string on the character ':'

```
text = "code clock: time to code"
# Split on colon
print(text.split(':'))
```

Check if a string Startswith or Endswith a character

To check if a string starts with or ends with a specific character, you can use the startswith() or the endswith() method respectively.

The startswith() method, when invoked on a string, takes a character as input argument. If the string starts with the given character, it returns True. Otherwise, it returns False.

The endswith() method, when invoked on a string, takes a character as input argument. If the string ends with the given character, it returns True. Otherwise, it returns False. You can observe this in the following example.

Enter the following code which will return either true or false if the string starts or ends with the character specified

```
text = "code clock: time to code"
#Returns true/false if string starts/ends with
print(text.startswith("c"))
print(text.endswith("e"))
print(text.endswith("w"))
```

Repeat Strings Multiple Times

You can repeat a string multiple times using the multiplication operator. When we multiply any given string or character by a positive number N, it is repeated N times. You can observe this in the following example.

Enter the following code which print the string ten times

```
# prints ten dots
print "."* 10
```

Replace Substring in a String in Python

You can also replace a substring with another substring using the replace() method.

The replace() method, when invoked on a string, takes the substring to replaced as its first input argument and the replacement string as its second input argument.

After execution, it replaces the specified substring with the replacement string and returns a modified string. You can perform string manipulation in Python using the replace() method as shown below.

Enter the following code which replaces one word with another in a string

```
text = "code clock: time to code"
print(text.replace("code", "learn"))
```

Changing Upper and Lower Case Strings

You can convert string into uppercase, lowercase, and title case using the upper(), lower(), and title() method.

The upper() method, when invoked on a string, changes the string into uppercase and returns the modified string.

The lower() method, when invoked on a string, changes the string into lowercase and returns the modified string.

The title() method, when invoked on a string, changes the string into the title case and returns the modified string.

You can also capitalise a string or swap the capitalization of the characters in the string using the capitalise() and the swapcase() method.

The capitalise() method, when invoked on a string, capitalises the first character of the string and returns the modified string.

The swapcase() method, when invoked on a string, changes the lowercase characters into uppercase and vice versa. After execution, it returns the modified string.

Enter the following code which which uses these various string functions

```
text = "code clock: time to code"

print(text.upper())
print(text.lower())
print(text.title())
print(text.capitalize())
print(text.swapcase())
```

Reverse a String in Python

To reverse a string, you can use the reversed() function and the join() method.

The reversed() function takes a string as its input argument and returns a list containing the characters of the input string in a reverse order.

The join() method, when invoked on a separator string, takes a list of characters as its input argument and joins the characters of the list using the separator. After execution, it returns the resultant string.

To reverse a string using the reversed() function and the join() method, we will first create a list of characters in reverse order using the reversed() function. Then we will use an empty string as a separator and invoke the join() method on the empty string with the list of characters as its input argument. After execution of the join() method, we will get a new reversed string as shown below.

Enter the following code which reverses a string

```
text = "code clock: time to code"
print ''.join(reversed(text))
```

Strip a String in Python

Python strings have the strip(), lstrip(), rstrip() methods for removing any character from both ends of a string.

The strip() method, when invoked on a string, takes a character as its input argument and removes the character from the start (left) and end(right) of the string. If the characters to be removed are not specified then white-space characters will be removed.

The lstrip() method, when invoked on a string, takes a character as its input argument and removes the character from start (left) of the string.

The rstrip() method, when invoked on a string, takes a character as its input argument and removes the character from the end(right) of the string.

```
text = "code clock: time to code \n\n"
```

You can Strip off newline characters from end of the string by passing “\n” as input argument to the rstrip() method.

Enter the following code which strips a string of trailing new lines

```
print text.strip('\n')
```

The following functions provide other options for removing leading or trailing characters

```
strip() #removes from both ends
lstrip() #removes leading characters (Left-strip)
rstrip() #removes trailing characters (Right-strip)
```

Enter the following code which removes leading/trailing spaces

```
text = " code clock: time to code "
print(text)
print (text.strip())
print (text.lstrip())
print (text.rstrip())
```

Concatenate Strings in Python

To concatenate strings in Python use the “+” operator as shown below.

```
"Code " + "Clock"
"Code " + "Clock" + "Time to code"
```

As discussed above, The join() method, when invoked on a separator string, takes a list of characters as its input argument and joins the characters of the list using the separator. After execution, it returns the resultant string.

Enter the following code which adds a specified character between every character

```
text = "code clock: time to code"

#add a ":" between every char
print ":".join(text)

# add a whitespace between every char
print " ".join(text)
```

Checking if a string meets a certain criteria

A string in Python can be tested for truth value. The return type will be in Boolean value (True or False)

```
text = "Code Clock: Time to Code"

text.isalnum() #check if all char are alphanumeric
text.isalpha() #check if all char in the string are alphabetic
text.isdigit() #test if string is all digits
text.istitle() #test if string is all title words
text.isupper() #test if string is all upper case
text.islower() #test if string is all lower case
text.isspace() #test if string is all spaces
text.endswith('c') #test if string endswith a d
text.startswith('e') #test if string startswith H
```

Challenge:

1. Write a program which takes a four digit input and then multiplies the first two numbers with the second two numbers.
2. Write a program which counts the number of words in sentence
3. Write a program which checks if a word is a palindrome i.e. the same forwards as it backwards e.g. racecar
4. Write a program which counts the number of vowels, consonants and spaces in a sentence.
5. Write the code which ensures that a post-code is inputted in the correct format i.e. BT24 6YH
6. Write the code which ensures that a national insurance number is entered in an appropriate format i.e. BB123456 B
7. Write a program which checks to make sure a user enters an email in the correct format:
   - Must contain the '@' symbol
   - Must not contain a space
   - Must have letters/numbers before @ symbol
   - Must contain a '.' Symbol after the @
8. Write the code which ensures that a password meets the following criteria:
   - Must contain a letter
   - Must contain a number
   - Must contain an upper case letter
   - Must contain a lower case letter
   - Must be between 8 and 15 characters
   - Must contain one of the following special charactes ['@','#','?','£']
   - Should not contain the word "password"