#  Lab 10

#  Strings

A string is a sequence of characters.

A character is simply a symbol. For example, the English language has 26 characters.

Computers do not deal with characters, they deal with numbers (binary). Even though you may see characters on your screen, internally it is stored and manipulated as a combination of 0s and 1s.

This conversion of character to a number is called encoding, and the reverse process is decoding. ASCII and Unicode are some of the popular encodings used.

In Python, a string is a sequence of Unicode characters. Unicode was introduced to include every character in all languages and bring uniformity in encoding.

**Creating a string**

Strings can be created by enclosing characters inside a single quote or double-quotes. Even triple quotes can be used in Python but generally used to represent multiline strings and docstrings.

**Program 1**

*# all of the following are equivalent*

*my\_string = 'Hello'*

*print(my\_string)*

*my\_string = "Hello"*

*print(my\_string)*

*my\_string = '''Hello'''*

*print(my\_string)*

*# triple quotes string can extend multiple lines*

*my\_string = """Hello, welcome to*

 *the world of Python"""*

*print(my\_string)*

**Output:**

**Accessing characters in a string**



We can access individual characters using indexing and a range of characters using slicing. Index starts from 0. Trying to access a character out of index range will raise an IndexError. The index must be an integer. We can't use floats or other types, this will result into TypeError.

Python allows negative indexing for its sequences.

The index of -1 refers to the last item, -2 to the second last item and so on. We can access a range of items in a string by using the slicing operator :(colon).

**Program 2**

*#Accessing string characters in Python*

*str = 'programiz'*

*print('str = ', str)*

*#first character*

*print('str[0] = ', str[0])*

*#last character*

*print('str[-1] = ', str[-1])*

*#slicing 2nd to 5th character*

*print('str[1:5] = ', str[1:5])*

*#slicing 6th to 2nd last character*

*print('str[5:-2] = ', str[5:-2])*

**Output:**

**Change or delete a string**

Strings are immutable. This means that elements of a string cannot be changed once they have been assigned. We can simply reassign different strings to the same name.

We cannot delete or remove characters from a string. But deleting the string entirely is possible using the del keyword.

The replace() method replaces a string with another string:

**Program 3**

*a = "Hello, World!"*

*print(a.replace("Hello", “Hi"))*

**Output**

**String Operations**

**Slice String**

You can return a range of characters by using the slice syntax.

Specify the start index and the end index, separated by a colon, to return a part of the string.

**Program 4**

*b = "Hello, World!"*

*print(b[2:5])*

**Output**

**String Length**

To get the length of a string, use the len() function.

**Program 5**

*a = "Hello, World!"*

*print(len(a))*

**Output**

**String Concatenation**

To concatenate, or combine, two strings you can use the + operator.

**Program 6**

*a = "Hello"*

*b = "World"*

*c = a + b*

*print(c)*

**Output**

**Check String**

To check if a certain phrase or character is present in a string, we can use the keywords in or not in.

**Program 7**

*txt = "The rain in Spain stays mainly in the plain"*

*x = "ain" in txt*

*print(x)*

**Output**

**Upper & lower case**

The lower() method returns the string in lower case.

The upper() method returns the string in upper case:

**Program 8**

*a = "Hello, World!"*

*print(a.lower())*

*b = "Bye, World!"*

*print(b.upper())*

**Output**

**String Format**

we can combine strings and numbers by using the format() method!

The format() method takes the passed arguments, formats them, and places them in the string where the placeholders {} are.

**Program 9**

*age = 36*

*txt = "My name is John, and I am {}"*

*print(txt.format(age))*

**Output**

**Lab Task**

Create your own string and reapply all the operations in today’s lab on it.

*Marks Obtained: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Instructor’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*