Python Class 1 Assignment

1. String Replace

The **string.replace()** method in Python replaces all occurrences of a specified substring with a new value. It does not replace just the first occurrence; rather, it replaces all instances found within the string.

Here is the syntax for the **replace()** function in Python:

```
string.replace(old, new, count)
```

old: The substring to be replaced. **new:** The new substring that will replace the occurrences of old. **count (optional):** The maximum number of occurrences to replace. If not specified, all occurrences will be replaced.

If count is provided, the **replace()** method will replace the first count occurrences of the old substring with the new substring. If count is not provided, all occurrences of the old substring in the string will be replaced with the new substring.

In [1]:

```
my_string = "Hello, hello, hello, world!"

new_string = my_string.replace("hello", "hi")

print(new_string)
```

```
Hello, hi, hi, world!
```

In [2]:

```
my_string = "hello, hello, hello, world!"

new_string = my_string.replace("hello", "hi")

print(new_string)
```

```
hi, hi, hi, world!
```

In [3]:

```
my_string = "hello, hello, hello, world!"

new_string = my_string.replace("hello", "hi", 2)

print(new_string)
```

```
hi, hi, hello, world!
```

In [4]:

```
my_string = "hello, hello, hello, world!"

new_string = my_string.replace("hello", "hi", 1)

print(new_string)
```

```
hi, hello, hello, world!
```

Questions:

Q1: Given the following string, remove all the spaces using the replace() function.

```
input_string = "Hello, how are you doing?"
# Write code here to remove spaces from the string
# Output should be: "Hello,howareyoudoing?"
```

In [5]:

```
input_string = "Hello, how are you doing?"
input_string = input_string.replace(" ", "")
print(input_string)
```

```
Hello,howareyoudoing?
```

Q2: Given a string with multiple occurrences of the word "book," write a Python program to replace only the first two occurrences with "notebook."

```
input_string = "I have a book, a pen, and another book in my bag and book."
# Output should be: "I have a notebook, a pen, and another notebook in my bag and
book."
```

In [6]:

```
input_string = "I have a book, a pen, and another book in my bag and book."
input_string = input_string.replace("book", "notebook", 2)
print(input_string)
```

```
I have a notebook, a pen, and another notebook in my bag and book.
```

2. String len()

- The **len()** function in Python is used to determine the length of a sequence, such as a string, list, tuple, or other iterable. It returns the number of elements present in the given sequence.

Q1. What happens when you pass an empty string to the len() function?

In [7]:

```
input_string = ""
length_input_string = len(input_string)
print(length_input_string)
```

```
0
```

Hence, We can observe that whenever we pass a empty string to the **len()** function, it will always return 0 since the length of an empty string is zero.

Q2. How len() works in case of multi-line string?

In [8]:

```
doctype = """
This is a multi-line string
presented by Arpit Dubey
and spans across
multiple lines.
"""

length_of_doctype = len(doctype)
print(length_of_doctype)
```

```
87
```

- **len()** function works the same way for multi-line strings as it does for single-line strings. It returns the number of characters in the string, including newline characters (\n) for each line break.

3. String

Q1: Can you concatenate strings and variables in Python? If yes, how?

In [9]:
```
name = "Arpit Dubey"
message = "Hello, "+name+"!"
print(message)
```

```
Hello, Arpit Dubey!
```

In [10]:
```
name        = "Arpit Dubey"
roll_number = 7
message     = "Hello, "+name+"!"+". My roll number is"+roll_number+"."
print(message)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[10], line 3
      1 name        = "Arpit Dubey"
      2 roll_number = 7
----> 3 message     = "Hello, "+name+"!"+". My roll number is"+roll_number+"."
      4 print(message)

TypeError: can only concatenate str (not "int") to str
```

In [11]:
```
name        = "Arpit Dubey"
roll_number = 7
message     = "Hello, "+name+"!"+". My roll number is "+str(roll_number)+"."
print(message)
```

```
Hello, Arpit Dubey!. My roll number is 7.
```

- Yes, we can concatenate string and variables in python using the `+` operator but only if the data type of the data is string otherwise we have to convert it to string data type using **str()** function.

Q2: How do you format strings in Python to include variables or expressions?

In [12]:
```
name    = "Arpit"
age     = 28
message = f"My name is {name} and I am {age} years old."
print(message)
```

```
My name is Arpit and I am 28 years old.
```

In [13]:
```
name    = "Arpit"
age     = 28
message = "My name is {name} and I am {age} years old."
print(message.format(name=name, age=age))
```

```
My name is Arpit and I am 28 years old.
```

Q3: Input- abcd and Output- A_b_c_d

In [14]:
```
input_string  = "abcd"
input_string  = input_string.capitalize()
output_string = "_".join(input_string)
print(output_string)
```

```
A_b_c_d
```

Q4. How can you limit the number of splits using the maxsplit parameter?

In [15]:

```
input_string    = "10 20 30 40 50"
limited_splits = input_string.split(" ", 2)
print(limited_splits)
```

```
['10', '20', '30 40 50']
```