### Ques. Python Strings?

- Once a string object has been created, it cannot be changed. “Modifying” that string creates a whole new object in memory.
- 'hello' is the same as "hello".

### Ques. Assign String to a Variable?

```
a = "Hello"
print(a)

Output:- Hello
```

### Ques. Multiline Strings?

- You can assign a multiline string to a variable by using three quotes:

```
a = """Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua."""
print(a)

Output:-
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
```

### Ques. Slicing Strings?

```
b = "Hello, World!"

print(b[2:5])   # Slice from the start position and end position
Output:- llo

print(b[:5])    # Slice From the Start: Get the characters from the start to
position (5 not included)
Output:- Hello

print(b[2:])    # Slice To the End: Get the characters from position 2, and all
the way to the end.
Output:- llo, World!

print(b[-5:-2]) # Negative Indexing: Get the characters from position 2, and all
```

```
the way to the end.
Output:- orl
```

## Ques .Modify Strings?

- The **replace()** method replaces a string with another string.

```
a = "Hello, World!"
print(a.replace("H", "J"))

Output:- Jello, World!
```

- The **split()** method splits the string into substrings if it finds instances of the separator

```
a = "Hello, World!"
b = a.split(",")

Output:- ['Hello', ' World!']
```

- **Upper Case**

```
b = "Hello, World!"
print(a.upper())

Output:- HELLO, WORLD!
```

- **Lower Case**

```
b = "Hello, World!"
print(a.lower())

Output:- hello, world!
```

- **strip() method**:- The strip() method removes any whitespace from the beginning or the end.

```
b = "      Hello, World!      "
print(a.strip())

Output:- Hello, World!
```

## String Concatenation

- To concatenate, or combine, two strings you can use the **plus(+)** operator.

```
a = "Hello"
b = "World"
c = a + b
print(c)

Output:- HelloWorld
```

- To add a space between them, add a **" "**

```
a = "Hello"
b = "World"
c = a+" "+b

print(c)

Output:- Hello World
```

### Format - Strings

```
# String Format:- As we learned in the Python Variables chapter, we cannot combine
strings and numbers like this.
age = 36
txt = "My name is John, I am " + age
print(txt)

Output:-
Traceback (most recent call last):
File "demo_string_format_error.py", line 2, in <module>
   txt = "My name is John, I am " + age
TypeError: must be str, not int


# 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:
  age = 36
  txt = "My name is John, and I am {}"
  print(txt.format(age))

  output:- My name is John, and I am 36


# The format() method takes unlimited number of arguments, and are placed into the
respective placeholders:
quantity = 3
itemno = 567
```

```
price = 49.95
myorder = "I want {} pieces of item {} for {} dollars."
print(myorder.format(quantity, itemno, price))

output:- I want 3 pieces of item 567 for 49.95 dollars.

# You can use index numbers {0} to be sure the arguments are placed in the correct
placeholders:

quantity = 3
itemno = 567
price = 49.95
myorder = "I want to pay {2} dollars for {0} pieces of item {1}."
print(myorder.format(quantity, itemno, price))

Output:- I want to pay 49.95 dollars for 3 pieces of item 567
```

### Ques. What is an f-string and how do you use it?

New in python 3.6, f-strings make string interpolation really easy. Using f-strings is similar to using format(). and F-strings are denoted by an f before the opening quote.

```
name = 'Chris'
food = 'creme brulee'
print(f'Hello. My name is {name} and I like {food}.')

Output:- Hello. My name is Chris and I like creme brulee.
```

### Ques. Escape Characters?

- An escape character is a backslash \ followed by the character you want to insert.

```
txt = "We are the so-called \"Vikings\" from the north."
print(txt)

Output:- We are the so-called "Vikings" from the north.

# Single Quote
txt = 'It\'s alright.'
print(txt)

Output:- It's alright.

# Backslash
txt = "This will insert one \\ (backslash)."
print(txt)

Output:- This will insert one \ (backslash).
```

```
# New Line
txt = "Hello\nWorld!"
print(txt)

Output:-
Hello
World!

# Carriage Return
txt = "Hello\rWorld!"
print(txt)

Output:-
Hello
World!

# Tab
txt = "Hello\tWorld!"
print(txt)

Output:- Hello   World!

# Backspace
txt = "Hello \bWorld!"
print(txt)

Output:- HelloWorld!

# Octal value
txt = "\110\145\154\154\157"
print(txt)

Output:- Hello

# Hex value
txt = "\x48\x65\x6c\x6c\x6f"
print(txt)

Output:- Hello
```

### Ques. How would you confirm that 2 strings have the same identity?

The **is** operator returns True if 2 names point to the same location in memory.

```
animals          = ['python','gopher']
more_animals     = animals

print(animals == more_animals) #=> True
print(animals is more_animals) #=> True
Output:-
True
```

```
True
-------------------------------------------------
animals           = ['python','gopher']
even_more_animals = ['python','gopher']
print(animals == even_more_animals) #=> True
print(animals is even_more_animals) #=> False
Output:-
True
False
```

### Ques. How would you check if each word in a string begins with a capital letter?

The **istitle()** function checks if each word is capitalized.

```
print( 'The Hilton'.istitle() ) #=> True
print( 'The dog'.istitle() ) #=> False
print( 'sticky rice'.istitle() ) #=> False
```

### Ques. Check if a string contains a specific substring?

The **in** operator will return **True** if a string contains a substring.

```
str = "My name is mohit saxena"

print( 'saxena' in str ) #=> True
print( 'car' in 'The worlds fastest plane' ) #=> False
```

### Ques. Count the number of a specific character in a string?

- **count()** will return the number of occurrences of a specific character.

```
print('The first president of the organization..'.count('o')) # 3

Output:- 3
```

### Ques. Count number of characters in a string

```
 string = "My Name is Mohit Saxena";
count = 0;

for i in range(0, len(string)):
    if(string[i] != ' '):
        count = count + 1;
print("Total number of characters in a string: " + str(count));
```

```
Output:- 19
```

### Ques. Remove vowels from a string?

```
string = 'Hello mohit saxena'
vowels = ('a','e','i','o','u')
new_string = ''
for ele in string:
    if ele not in vowels:
        new_string = new_string + ele
print(new_string)

# Using comprehension
print(''.join([c for c in string if c not in vowels]))

Output:- Hll mht sxn
```

### Ques. When would you use rfind()?

- **rfind()** is like find() but it starts searching from the right of a string and return the first matching substring.

```
story = 'The price is right said Bob. The price is right.'
story.rfind('is')

Output:- 39
```

### Ques. Extract numbers from string?

```
new_string = "Germany26China47Australia88"

emp_str = ""
for m in new_string:
    if m.isdigit():
        emp_str = emp_str + m
print("Find numbers from string:",emp_str)

Output:- Find numbers from string: 264788

# 2 Example
new_str = "Micheal 89 George 94"

emp_lis = []
for z in new_str.split():
   if z.isdigit():
      emp_lis.append(int(z))

print("Find number in string:",emp_lis)

Output:- Find number in string: [89, 94]
```

### Ques. Get a Substring of a String?

```
my_string = "I love python."

print(my_string[2:6])   # You need to specify the starting index and the ending
index of the substring. In this case, love starts at index 2 and ends at index 6.

print(my_string[2:])    # All the text from index 2 to the end are selected

print(my_string[:-1])   # All the text before the last index is selected.

Output:-
love
love python.
I love python
```

### Ques. Find repeated characters in a string python?

```
string = "Great responsibility";

for i in range(0, len(string)):
    count = 1;
    for j in range(i+1, len(string)):
        if(string[i] == string[j] and string[i] != ' '):
            count = count + 1;
            string = string[:j] + '0' + string[j+1:];

    if(count > 1 and string[i] != '0'):
        print(string[i]);

output:- r
e
t
s
i
```

### Ques. program to count the frequency of each character?

```
# using "in" operater
str1 = input ("Enter the string: ")
```

```
d = dict()
for c in str1:
    if c in d:
        d[c] = d[c] + 1
    else:
        d[c] = 1
print(d)

Output:-
Enter the string: HheLlo
{'H': 1, 'h': 1, 'e': 1, 'L': 1, 'l': 1, 'o': 1}
Enter the string: Hello My name is Mohit Saxena
{'H': 1, 'e': 3, 'l': 2, 'o': 2, ' ': 5, 'M': 2, 'y': 1, 'n': 2, 'a': 3, 'm': 1,
'i': 2, 's': 1, 'h': 1, 't': 1, 'S': 1, 'x': 1}
----------------------------------------------------------------------------

# Use of “get()” function
```

### Ques. Looping Through a String?

```
for x in "Mohit":
  print(x)

Output:-
M
o
h
i
t
```

### Ques. String Length?

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

```
a = "Hello, World!"
print(len(a))

Output:- 13
```

### Ques. Check String?

- we can use the keyword in.

```
txt = "The best things in life are free!"
print("free" in txt)
```

```
Output:- True
```

- Use it in an if statement:

```
txt = "The best things in life are free!"
if "free" in txt:
  print("Yes, 'free' is present.")

Output:- Yes, 'free' is present.
```

- Check if NOT

```
txt = "The best things in life are free!"
print("expensive" not in txt)

output:- True
```

### Ques. Reversed the String?

- using reversed() function

```
string = "Hello, World!"
print("".join(reversed(string)))

Output:- !dlroW ,olleH
```

- Using slice method and through function

```
txt = "Hello World"[::-1]
print(txt)

# Using Function
def my_function(x):
  return x[::-1]

mytxt = my_function("Hello World")

print(mytxt)

Output:- dlroW olleH
```