```
"""
String:
A String is a collection of characters.
In Python, A single character is also a String.
A string in python can be enclosed inside a double quotes or
single quotes.
Multi Line Strings are enclosed in between \""" or '''

Characteristics:
1. Strings are ordered.
2. Strings are immutable.
3. Strings allow duplicates.
4. Strings are case sensitive.

Example:
"""
a = "Python"
print(type(a))
a = 'Program'
print(type(a))
a = "A"
print(type(a))

a = """
This
is
a
Multi
Line
String
"""
print(type(a))

a = '''
This
is
a
Multi
Line
String
'''
print(type(a))

a = "aabbccdd"
print(a)

a = "Python"
b = "python"
print(a == b) # False

a = "1234"
print(type(a)) # <class 'str'>

a = "True"
print(type(a))
a = bool(a)
print(a) # True

a = "False"
```

```
print(type(a))
a = bool(a)
print(a) # True

a = ""
print(type(a))
a = bool(a)
print(a) # False

# 0, Empty List, Empty Tuple, Empty String => False
a = 0
if a:
    print("Condition is True")
else:
    print("Condition is False")

a = -5
if a:
    print("Condition is True")
else:
    print("Condition is False")

a = []
if a:
    print("Condition is True")
else:
    print("Condition is False")

a = ()
if a:
    print("Condition is True")
else:
    print("Condition is False")

a = ""
if a:
    print("Condition is True")
else:
    print("Condition is False")

a = " "
if a:
    print("Condition is True")
else:
    print("Condition is False")


"""
Indexing
Syntax:
<string_variable_name>[<index>]
"""
a = "Programming"
# Positive Indexing
print(a[0]) # P
print(type(a[0])) # <class 'str'>

print(a[5]) # a
print(a[9]) # n
```

```
# print(a[100]) # IndexError: string index out of range

# Negative Indexing
print(a[-1]) # g
print(a[-3]) # i
print(a[-7]) # r

# print(a[-40]) # IndexError: string index out of range


"""
Slicing:
Extracting a part of the string (substring)
Syntax:
<string_variable_name>[[start]:[stop]:[step]]
start is optional. Default value for start is 0.
stop is optional. Default value for stop is length of string.
step is optional. Default value for step is 1.
"""
a = "Python Programming"
print(a[2: 7]) # thon
print(a[0: 5]) # Pytho
print(a[: 5])

print(len(a)) #
print(a[11: 18]) # ramming
print(a[11: ]) # ramming

print(a[:]) # Python Programming

# Negative Indexing in Slicing
a = "Python Programming"
print(a[-7: -2]) # rammi
print(a[-9: -5]) # ogra
print("Value = ", a[-8: -15])

print(a[2: 10: 2]) # to r
print(a[4: 15: 3]) # oPgm

print(len(a[9: 17: -1])) # 0
print(a[-5: -14: -2]) # mroPn

print(len(a[-10: -2: -1])) # 0


# String Methods
"""
1. lower()
2. upper()
3. casefold()
4. swapcase()
5. title()
6. capitalize()
7. isupper()
8. islower()
9. isalpha()
10. isnumeric()
11. isdigit()
12. isdecimal()
```

```
13. isalnum()
14. isspace()
15. isprintable()
16. strip()
    lstrip()
    rstrip()
17. split()
18. join()
19. splitline()
20. index()
    rindex()
21. find()
22. rfind()
23. partition()
    rpartition()
24. zfill()
25. center()
26. count()
27. startswith()
28. endswith()
29. format()
30. replace()
"""
a = "PYTHON"
print(a.lower())
print(a)

a = "Python"
print(a.lower())
print(a)

a = "P y%t$#h&o@n"
print(a.lower())
print(a)


a = "python"
print(a.upper())
print(a)

a = "pYTHon"
print(a.upper())
print(a)

a = "P Y%T$#h&o@n"
print(a.upper())
print(a)

a = "Python"
print(a.casefold())

a = 'ß'
print(a.lower())
print(a.casefold())

a = "PyThOn"
print(a.swapcase())

a = "P Y%T$#h&o@n"
```

```
print(a.swapcase())

a = "the sun rises in the east"
print(a.title())

print(a.capitalize())

a = "PYTHON"
print(a.isupper())
a = "Python"
print(a.isupper())
a = "P Y"
print(a.isupper())
a = "PY%$^#"
print(a.isupper())


a = "python"
print(a.islower())
a = "Python"
print(a.islower())
a = "p y"
print(a.islower())
a = "py%$^#"
print(a.islower())


a = "Python"
print(a.isalpha())
a = "Python Program"
print(a.isalpha())
a = "P$%yejrh"
print(a.isalpha())

# https://stackoverflow.com/questions/44891070/whats-the-difference-between-str-isdigit-isnumeric-and-isdecimal-in-pyth
a = "1233445"
print(a.isdigit())
print(a.isnumeric())
print(a.isdecimal())

print("--------")
a = 'ⅢⅧ'
print(a.isdigit())
print(a.isnumeric())
print(a.isdecimal())
print("----------")

a = '½'
print(a.isdigit())
print(a.isnumeric())
print(a.isdecimal())

a = "Ptrrt"
print(a.isalnum())
a = "13434"
print(a.isalnum())
a = "ksjsk2323"
print(a.isalnum())
```

```
a = "Ptrrt3434 "
print(a.isalnum())
a = "Ptrrt233$"
print(a.isalnum())

a = "          "
print(a.isspace())
a = "          $"
print(a.isspace())
a = "          a"
print(a.isspace())

a = "        Python    Program         "
print(a.lstrip())
print(a.rstrip())
print(a.strip())

a = "The Sun rises in the east"
print(a.split(' '))

a = ['The', 'Sun', 'rises', 'in', 'the', 'east']
print(' '.join(a))
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] # 1*2*3*......9*10
# a_str = "*".join(str(a))
a_str_list = [str(number) for number in a]
print(a_str_list)
a_str = "*".join(a_str_list)
print(a_str)

a = "Python"
print(a.center(10))
print(a.center(10, '*'))
print(a.center(11, '&'))

a = "12345"
print(a.zfill(8))

a = """I am learning Python.Python is a Programming Language.
       Python is easy to learn."""
print(a.index("Python")) # 14
print(a.rindex("Python")) # 72

print(a.index("Python", 15)) # 21
print(a.index("Python", 15, 27)) # 21

print(a.find("Python")) # 14
print(a.find("Python", 15)) # 21
print(a.find("Python", 15, 27)) # 21

print(a.find("kljskdjskfjs")) # -1
print(a.find("python", 10)) #

print(a.rfind("Python")) # 63

a = "Python*Program"
print(a.partition("*")) # ('Python', '*', 'Program')
a = "Python*Program*"
print(a.partition('*')) # ('Python', '*', 'Program*')
print(a.rpartition('*')) # ('Python*Program', '*', '')
print(a.partition('&')) # ('Python*Program*', '', '')

print(a.count('*')) # 2
print(a.count('&')) # 0

a = "Programming Language"
print(a.startswith("Program")) # True
print(a.startswith("Python")) # False

a = "Python Programming Language"
print(a.startswith("Program")) # False
print(a.startswith("Python")) # True

print(a.endswith("Language")) # True
print(a.endswith("Program")) # False

a = "I am learning {} Programming Language and {} framework"
print(a.format("Python", "Django"))

a = "I am learning {} Programming Language and {} framework. {} is easy to learn."
print(a.format("Python", "Django", "Python"))

a = "I am learning {0} Programming Language and {1} framework. {0} is easy to
learn."
print(a.format("Python", "Django"))

a = "I am learning {language} Programming Language and {framework} framework.
{language} is easy to learn."
print(a.format(language="Python", framework="Django"))

d = {"language": "Python", "framework": "Django"}
a = "I am learning {language} Programming Language and {framework} framework.
{language} is easy to learn."
a.format_map(d)

# ASCII - American  Standard Code for Information Interchange
a = "if"
print(a.isidentifier()) # True
a = "list"
print(a.isidentifier()) # True

a = "I am learning Python. Python is a Programming Language"
print(a.replace("Python", "Java")) # I am learning Java. Java is a Programming
Language

print(a)
a.replace("Python", "Java")
print(a)


# Operators with Strings:
# + - Concatenation, * - Repetition
a = "Hello"
b = "World"
c = a + " " + b
print(c)

a = 'Python '
b = a * 3
```

```
print(b)


# Iterating through the string
a = "Python Program"
for char in a:
    print(char, end=' ')
print()

index = 0
while index < len(a):
    print(a[index], end=' ')
    index += 1
print()

"""
Escape Sequences
\n - new line, \t - tab space, \r - carriage return, \f - form feed, \\, \' - ', \"
-, \b - backspace "
"""
# I am learning "Python"
# a = "I am learning "Python""
a = 'I am learning "Python"'
print(a)
# I am learning 'Python'
a = "I am learning 'Python'"
print(a)
# I'm learning "Python"
# a = "I'm learning "Python""
# a = 'I'm learning "Python"'
a = "I'm learning \"Python\""
print(a)
a = 'I\'m learning "Python"'
print(a)
a = "Hello \n World!!!"
print(a)
a = "Hello \t World!!!"
print(a)
a = "Hello\bWorld!!!"
print(a)
a = "Hello \r World!!!"
print(a)

# Raw String
a = r"Hello \n\r\t World!!!"
print(a)

a = 10
b = 20
c = f"a value is {a} and b value is {b}"
print(c)
```