```
# Combining repetition and concatenation
print('A' + 'B' * 5 + 'C')    # ABBBBBC
print(('A' + 'B') * 5 + 'C')  # ABABABABABC
```

```
# Returns original value, which is unchanged
s1 = 'Hello World'
print(s1)  # Hello World

# Returns upper case
s2 = 'helloworld'
print(s2.upper()) # HELLOWORLD

# Returns lower case letter
s3 = 'HELLOWORLD'
print(s3.lower()) # helloworld

# Returns a string each word begins with a capital letter
s4 = 'helloworld'
print(s4.title()) # Helloworld

# Returns a letter opposite
s5 = 'Hello World'
print(s5.swapcase())# hELLO wORLD
```

replace()

```
s1 = 'Hello Hello Hello'
print(s1.replace('H', 'h'))  # hello hello hello

s2 = 'Hello Hello Hello'
# replace(old, new, count)
print(s2.replace('H', 'h', 3))  # hello hello hello
print(s2.replace('H', 'h', 2))  # hello hello Hello
print(s2.replace('H', 'h', 1))  # hello Hello Hello

# Note: Count is used for occurrences of same characters and default is all occurrences
```

```
# split() method splits string into pieces and returns list object
# split uses white space as default separator
# split works with string only not with list and tuple etc...

sentence = 'Hello World Hello Python'
print(sentence.split())
# ['Hello', 'World', 'Hello', 'Python']
```

pass space in between the single quotes or else it will raise exception as ValueError: empty separator

```
s1  = "Hello World"
print(s1.split('')) # ValueError: empty separator

passing white space in between single quotes
s2  = "Hello World"
print(s2.split(' ')) # ['Hello', 'World']
```

```
#split using single colon :
mobiles = "Nokia:Samsung:LG:Moto"
print(mobiles.split(':')) # ['Nokia', 'Samsung', 'LG', 'Moto']


#split using - minus
colors = "Red-Black-Green-Yellow"
print(colors.split('-')) # ['Nokia', 'Samsung', 'LG', 'Moto']


#split using  and
programming = "javaandpythonandphpandruby"
print(programming.split('and')) # ['java', 'python', 'php', 'ruby']

# spits both the argument and white space in string
programming = "java and python and php and ruby"
print(programming.split('and')) # ['java ', ' python ', ' php ', ' ruby']
```

```
# spits using max split
programming = "java and python and php and ruby"
print(programming.split(' ', 1)) # ['java', 'and python and php and ruby']

# split the data using max split
user = "101,Sai Kiran,Hyd"
print(user.split(',',1)) # ['101', 'Sai Kiran,Hyd']
```

```
# join() method of a string joins the element of a sequence of strings from list, tuple,set

mobiles = ["Nokia", "Samsung", "MI", "Moto"]
print(mobiles) # ['Nokia', 'Samsung', 'MI', 'Moto']

print(','.join(mobiles)) # Nokia,Samsung,MI,Moto

print(' : '.join(mobiles)) # Nokia : Samsung : MI : Moto

print(''.join(mobiles)) # NokiaSamsungMIMoto
```

```
#Error
#join() method will through error if any of the elements in the sequence is not a string
lst = [1,2,3,4]
print(','.join(lst))
#TypeError: sequence item 0: expected str instance, int found

lst = ['1','2','3','4']
print(','.join(lst)) # 1,2,3,4
```

```
# diff bw join and split
mobiles = {"Nokia", "Samsung", "MI", "Moto"}
print(','.join(mobiles)) # Nokia,Samsung,MI,Moto

mobiles = "Nokia,Samsung,MI,Moto"
print(mobiles.split(',')) # ['Nokia', 'Samsung', 'MI', 'Moto']
```

```
# find() Returns the lowest index
s = "Hello learners like the code "
#    0123456789
print(s.find('l')) #2
print(s.find('l')) #2

# rfind() returns highest index
print(s.rfind('l')) # 15
```

```
#AttributeError: 'str' object has no attribute 'lfind'
print(s.lfind('l'))
```

```
# value, start, end
print(s.rfind('l',5,8)) # 6, value, start, end

# if there is no matching value we get -1
print(s.rfind('m')) # -1
```

```
# Reverse a String
s = 'HelloWorld'
print(list(reversed(s))) ['d', 'l', 'r', 'o', 'W', 'o', 'l', 'l', 'e', 'H']

for x in reversed(s):
    print(x, end = ' ') d l r o W o l l e H
```

```
# removes any whitespace from the beginning or the end
s1 = ' Hello Learners '
print(s1)
# 'string with spaces'
print(s1.strip())
```

Output

```
C:\Users\lenovo\Ap
 Hello Learners
Hello Learners
```

```
# Return a copy of the string with leading whitespace removed in left.
s2 = ' Hello Learners '
print(s2.lstrip())
```

Output

```
C:\Users\lenovo\
Hello Learners
```

```
# Return a copy of the string with leading whitespace removed in right.
s3 = ' Hello Learners '
print(s3.rstrip())
```

Output

```
C:\Users\lenovo\A
 Hello Learners
```