# Chapter – 4
# Python Strings

## Contents

4.0 dir() Function........3
4.1 .capitlize() Method........3
4.2 .casefold() Method........4
4.3 .center() Method........4
4.4 .count() Method........5
4.5 .encode() Method........5
4.6 .endswith() Method........6
4.7 .expandtabs() Method........6
4.8 .find() Method........7
4.9 .format() Method........7
4.10 .format_map() Method........8
4.11 .index() Method........8
4.12 .isalnum() Method........9
4.13 .isalpha() Method........9
4.14 .isascii() Method........10
4.15 .isdecimal() Method........10
4.16 .isdigit() Method........11
4.17 .isidentifier() Method........11
4.18 .islower() Method........12
4.19 .isnumeric() Method........13
4.20 .isprintable() Method........13
4.21 .isspace() Method........14
4.22 .istitle() Method........14
4.23 .isupper() Method........15

4.24 .join() Method 15

4.25 .ljust() Method 16

4.26 .lower() Method 16

4.27 .lstrip() Method 16

4.28 .maketrans() Method + translate() Method 17

4.29 .partition() Method 18

4.30 .replace() Method 18

4.31 .rfind()method 18

4.32 .rindex() Method 18

4.33 .rjust() Method 19

4.34 .rpartition() method 19

4.35 .rsplit() Method 20

4.36 .rstrip() Method 20

4.37 .split() Method 20

4.38 .splitlines() Method 21

4.39 .startswith() Method 21

4.40 .strip() Method 21

4.41 .swapcase() Method 22

4.42 .title() Method 22

4.43 .upper() Method 22

4.44 .zfill() Method 23

# Chapter – 4

# Python Strings

အက္ခရာ(alphabet letter)များ a မှ z အထိ ဖြစ်သည်။ နံပါတ်(numbers)များသည် 0 မှ 9 အထိ ဖြစ်သည်။ အက္ခရာ(alphabet letter)များ နှင့် နံပါတ်(numbers)များကို ပေါင်း၍ alphanumeric(alphabet + number)ဟု ခေါ်သည်။ alphanumeric တွင် အက္ခရာ(alphabet letter)များဖြစ်သည့် a မှ z အထိနှင့် နံပါတ်(numbers)များ ဖြစ်သည့် 0 မှ 9 တို့ပါဝင်သည်။ (space) နှင့် ! # % & ? စသည့် character များသည် alphanumeric ထဲတွင် မပါဝင်ပါ။

အက္ခရာ(alphabet letter)များ၊ နံပါတ်(numbers)များ၊ space နှင့် **! # % & ?** စသည့်တို့ကို character များ ဟုခေါ်သည်။

### 4.0 dir() Function

Python3 တွင် **dir()** သည် powerful built-in function တစ်ခု ဖြစ်သည်။ Object ၏ attribute များ နှင့် method များကို ထုတ်ပေးသည်။ Functions , modules, strings, lists, dictionaries စသည်တို့သည့် object များ ဖြစ်ကြသည်။

**dir()** function သည် သတ်မှတ်ထားသည့် object ၏ method များနှင့် property များကို ဖော်ပြသည်။ မည်သည့်တန်ဖိုးကိုမှ ထုတ်ပေးလိမ့်မည်မဟုတ်ပေ။ အောက်တွင် my_string ဆိုသည့် variable တစ်ခု(object)၏ method များနှင့် property များကို ကြည့်ရန် **dir()** function ကို အသုံးပြုထားသည်။

```
my_string = "I love Python."
print(dir(my_string))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

### 4.1 .capitlize() Method

**.capitlize()** method ပထမဆုံး character သည် အက္ခရာ(alphabet letter)ဖြစ်လျှင် ထိုအက္ခရာကို အကြီးစာလုံးသို့ ပြောင်းပေးသည်။

```
my_string1 = "first letter of the sentence will be capitalized."
# capitalizes the first letter of the string.
print(my_string1.capitalize())
```

```
First letter of the sentence will be capitalized.
```

```
my_string1_a = "36 is my age."
print (my_string1_a.capitalize())
```

```
36 is my age.
```

ပထမဆုံး အလုံး(first letter)သည် digit ဖြစ်သောကြောင့် အကြီးစာလုံး ပြောင်းပေးရန် မလိုပါ။

**4.2 .casefold() Method**

**.casefold()** method အက္ခရာများအားလုံးကို အသေးစာလုံးသို့ ပြောင်းပေးသည်။ (returns a string where all the characters are lower case.)။ **.lower()** method နှင့်တူညီသော်လည်း .lower() method ထက် ပို၍ အစွမ်း ထက်သည်။ (casefold() method is stronger, more aggressive)

```
my_string2 = "THIS SENTENCE WILL BECOME LOWER-CASED"
print(my_string2.casefold())
```

```
this sentence will become lower-cased
```

```
my_string2_a = "THIS SENTENCE WILL BECOME LOWER-CASED"
print(my_string2_a.lower())
```

```
this sentence will become lower-cased
```

**4.3 .center() Method**

Character အရေအတွက် ထည့်ပေးရသည်။ ထို character အရေအတွက်၏ အလယ်တွင် string ကို ဖော်ပြပေးမည်။ အလယ်တည့်တည့်တွင် string ကို နေရာချထား လိုသည့်အခါတွင် အသုံးပြုသည်။ အောက်က ဥပမာတွင် character အရေအတွက်(၆၀)၏ အလယ်တွင် string ကို ဖော်ပြပေးမည်။

```
my_string3 = " I love Python."
print(my_string3.center(60))
print("----original string below----")
print(my_string3)
```

```
                         I love Python.
----original string below----
 I love Python.
```

အောက်က ဥပမာတွင် character အရေအတွက်(၂၀)၏ အလယ်တွင် string ကို ဖော်ပြပေးမည်။ အရှေ့နှင့်နောက် ကျန်နေရာများတွင် "O" ဖြင့် ဖြည့်ပေးမည်။

```
my_string3 = "banana"
print(my_string3)
x = my_string3.center(20, "O")
print(x)
```

```
banana
OOOOOOObananaOOOOOOO
```

Character အရေအတွက် ဘယ်လောက်ထည့်ပေးရမယ်ဆိုတာ စဥ်းစားစရာမလိုဘဲ "@" ကို အရှေ့(၂)လုံး၊အနောက်(၂)လုံး ပေါင်း (၄)လုံးထည့်မယ်ဆိုပြီးလည်း ရေးလို့ရသည်။

```
print(my_string3.center(len(my_string3)+4, "@"))
```

```
@@banana@@
```

**4.4 .count() Method**

စာကြောင်း သို့မဟုတ် စာလုံး(word)အတွင်း၌ မိမိထည့်ပေးလိုက်သည့် အက္ခရာ(character) အရေအတွက်ကို ဖော်ပြပေးသည်။ String အတွက်သာမက number, list, tuple စသည့် အမျိုးအစားများ အတွင်းရှိ မိမိသတ်မှတ် ပေးလိုက်သည့် အက္ခရာ၏ အရေအတွက်ကိုလည်း ဖော်ပြ ပေးနိုင်သည်။

```
my_string4 = "fdsGJsufH23edHHHe23edffosidgs23edHe66623ed6hw23edgw"
# returns the count of string "23" in the original string.
print(my_string4.count("H"))
```

```
5
```

မိမိသတ်မှတ်ပေးသည့် အစနေရာ(starting index)မှ စ၍ အဆုံးနေရာ(ending index)အတွင်း၌သာ ရေတွက်ခြင်း(counting) ပြုလုပ်နိုင်သည်။ `("string being serached", starting index (optional), ending index(optional))` ။ အောက်တွင် အစ index နံပါတ်နှင့် အဆုံး index နံပါတ်ထည့် ပေးပြီး ရေတွက်ခိုင်းသည်။

```
my_string4 = "fdfJsufH23edHHHHye23edffosidgs23eduiae66623ed6hw23edg0w"
print(my_string4.count("H", 0, 20))
```

```
5
```

**4.5 .encode() Method**

ထည့်ပေးသည့် string ကို သတ်မှတ်လိုက်သည့် encoding မျိုးဖြင့် encode လုပ်ပေးသည်။ ဘာမှ မသတ်မှတ်ပေးလျှင် UTF-8 ဖြင့် encode လုပ်ပေးသည်။(If no encoding is specified, UTF-8 will be used.)

```
my_string5 = "My name is Ståle"
x=my_string5.encode(encoding="utf-8", errors="strict")
print(x)
```

```
print("----original string below----")
print(my_string5)
```

```
b'My name is St\xc3\xa5le'
----original string below----
My name is Ståle
```

```
print(my_string5.encode(encoding="ascii",errors="backslashreplace"))
```

```
b'My name is St\\xe5le'
```

### 4.6 .endswith() Method

အဆုံး အက္ခရာများကို စစ်ပေးသည်။ သတ်မှတ်ထားသည့် character နှင့် အဆုံးသတ်သည် ဆိုလျှင် True ပြန်ပေးသည်။ ထိုသို့မဟုတ်လျှင် False ပေးသည်။ (returns True if the string ends with the specified value, otherwise False.) အစနှင့် အဆုံးထည့်ပေးလျှင် ထိုအစနှင့် အဆုံးအတွင်း၌သာ ရှာသည်။

**string.endswith(value, start, end)**

```
my_string6 = "I love Python."
print(my_string6.endswith("hon."))
```

```
True
```

### 4.7 .expandtabs() Method

ပုံမှန်အားဖြင့် \t သည် tab တစ်ခုစာ ခြားပေးသည်။ Tab တစ်ခုသည် space (၄)နေရာ(4 spaces)နှင့် ညီသည်။ **.expandtabs()** method ကိုသုံး၍ မိမိအလိုရှိသည့် space အရေအတွက်အတိုင်း ခြားပေးသည်။ ဥပမာ .expandtabs(10) သည် space (၁၀)နေရာ(10 spaces) ခြားပေးသည်။ string.expandtabs(tabsize)

```
my_string7 = "I love \tPython."
print(my_string7.expandtabs(1))
print("----original string below----")
print(my_string7)
```

```
I love    Python.
----original string below----
I love Python.
```

```
txt = "H\te\tl\tl\to"
# character တစ်လုံးနှင့် တစ်လုံးအကြားတွင် space  နေရာစာ လုံးဝမထားရန်
x = txt.expandtabs(0)
# character တစ်လုံးနှင့် တစ်လုံးအကြားတွင် space  (၃)နေရာစာ ခြားပေးရန်
x3 = txt.expandtabs(3)
print(x)
```

```
print(x3)
```

```
Hello
H   e   l   l   o
```

**4.8 .find() Method**

**.find()** method သည် ရှာခိုင်းသည့် အက္ခရာ ၊ စာလုံး(string)၏ ပထမဆုံး တွေ့ရှိသည့် တည်ရှိရာ နေရာကို ဖော်ပြ ပေးသည်။ (finds the first occurrence of the specified value.)။ ထိုအက္ခရာ ၊ ထိုစာလုံး မရှိပါက -1 ထုတ်ပေးသည်။ (returns -1 if the value is not found.)

```
my_string8 = "I love Python."
print(my_string8.find("Py"))      # Py သည် (၇)လုံးမြောက် character ဖြစ်သည်။
print(my_string8.find("My"))      # My ဆိုသည့် character မရှိသောကြောင့် -1 ပေးလိမ့်မည်။
```

```
7
-1
```

**4.9 .format() Method**

**.format()** method ကို သုံး၍ string ကို မိမိ အလိုရှိသလို ပုံစံချ(format လုပ်) နိုင်သည်။ တွန့်ကွင်း (curly brackets: {}) သည် placeholder ဖြစ်သည်။ String ၏ placeholder အတွင်း၌ မိမိ အလိုရှိသည့် တန်ဖိုးများ(specified value) ထည့်နိုင်သည်။ Output များကို ဖော်ပြရာတွင် အလွန် အသုံးဝင်သည်။

```
my_string9 = "I love {} Python {}."
print(my_string9.format("placeholder1", "placeholder2"))
```

```
I love placeholder1 Python placeholder2.
```

`{price:.2f}` တွင် `.2f` သည် ဒသမနောက်၌ ကိန်း(၂)လုံးသာ ဖော်ပြပေးမည်ဟု ဆိုလိုသည်။

```
my_string9_a = "For only {price:.2f} dollars!"
print(my_string9_a.format(price = 49))
```

```
For only 49.00 dollars!
```

```
myorder = "I have a {carname}, it is a {model}."
print(myorder.format(carname = "Ford", model = "Mustang"))
```

```
I have a Ford, it is a Mustang.
```

အောက်တွင် formatted string ဥပမာတစ်ခုကို အပြည့်အစုံ ဖော်ပြထားသည်။

```
quantity = 3
item_no = 567
price = 49
```

```
myorder = "I want {} pieces of item number {} for {:.2f} dollars."
print(myorder.format(quantity, item_no, price))
```

```
I want 3 pieces of item number 567 for 49.00 dollars.
```

Index number များ ကို အသုံးပြု၍လည်း လုပ်နိုင်သည်။ ထို index number ကို တွန့်ကွင်း(curly brackets { }) အတွင်း ထည့်ပေးရမည်။ value များကို သက်ဆိုင်သည့် placeholder မှန်ကန်စွာ ထည့်ပေးရန် လိုသည်။ အောက်တွင် ဥပမာ ဖြင့် ဖော်ပြထားသည်။

```
age = 36
name = "John"
txt = "His name is {1}. {1} is {0} years old."
print(txt.format(age, name))
```

```
His name is John. John is 36 years old.
```

### 4.10 .format_map() Method

`.format_map()` method ကို dictionary များ၏ key နှင့် value ဖော်ပြရာတွင် အသုံးပြုသည်။ (used to return an dictionary key's value.)။ Dictionary တွင် အတွဲလိုက်ရှိသည့် key နှင့် value ကို အတူတကွ ဖော်ပြရန်အတွက် သုံးသည်။

```
my_dict = {"name" : "Kaung",
           "birth city": "Yangon" }
my_string10 = "{name}'s place of birth is {birth city}."
print(my_string10.format_map(my_dict)) # .format_map()ထဲသို့ dictionary ထည့်
```

```
Kaung's place of birth is Yangon.
```

my_dict ကို `.format_map()`ထဲထည့်ပေးသည်။ ထည့်ပေးထားသည့် key များမှ တဆင့် value များကို ဖော်ပြသည်။

### 4.11 .index() Method

ထည့်ပေးသည့် character ၏ ပထမဆုံးတွေ့သည့် နေရာ(index နံပါတ်)ကို ဖော်ပြ ပေးသည်။ (returns the position at the first occurrence of the specified value.)။ "I love Python." တွင် o (၂)လုံးပါရှိရာ မည့်သည့်နေရာအတွင်း ရှာပေးပါဟု မပြောလျှင် အစမှ အဆုံးအတွင်း ရှာပေးပြီး ပထမဆုံး တွေ့သည့်နေရာကို ဖော်ပြပေးသည်။ ဒုတိယ o အတွက် အစ နှင့် အဆုံးနေရာ သတ်မှတ်ပေးသည်။

```
my_string11 = "I love Python."
print(my_string11.index("o"))            # first occurrence
print(my_string11.index("o", 5, 12))     # 5 မှ 12 အတွင်း ရှာပေးရန်
```

```
3
11
```

### 4.12 .isalnum() Method

**.isalnum()** method သည် string တစ်ခုထဲတွင် alphanumeric တွေသာ ပါဝင်နေသည် ဟုတ်မဟုတ် စစ်ပေးသည်။(Checks whether all the characters in a given string are alphanumeric or not.) Character များအားလုံးသည် alphanumeric ဖြစ်နေလျှင် True or False(boolean) ပြန်ပေးသည်။

Alphanumeric တွင် အက္ခရာ(alphabet letter)များဖြစ်သည့် a မှ z အထိနှင့် နံပါတ် (numbers)များ ဖြစ်သည့် 0 မှ 9 အတွင်း တို့ ပါဝင်သည်။ (space) နှင့် ! # % & ? စသည့် character များသည် alphanumeric တွင် မပါဝင်ပါ။

```
my_string12 = "LovePython3"
print(my_string12)
print(my_string12.isalnum())
```

```
LovePython3
True
```

```
my_string12_a = "I love Python"  #"I love Python" တွင် space များ ပါဝင်နေသည်။
print(my_string12_a)
print(my_string12_a.isalnum())
```

```
I love Python
False
```

### 4.13 .isalpha() Method

String ထဲရှိ character များအားလုံးသည် a မှ z အထိ အက္ခရာ(alphabet letter)ထဲမှ ဖြစ်နေလျှင် True ပြန်ပေးသည်။ အက္ခရာ(alphabet letter) မဟုတ်သည့် character များမှာ 0 မှ 9 အထိ နံပါတ် (numbers) များနှင့် (space) နှင့် ! # % & ? စသည့်တို့ ဖြစ်သည်။

```
my_string13 = "Python" # alphabet
print(my_string13)
print(my_string13.isalpha())    # alphabet များ ဟုတ် မဟုတ် စစ်ပေးရန်
```

```
Python
True
```

```
my_string13_a = "I love Python" # "space" သည် alphabet မဟုတ်ပါ။
print(my_string13_a)
print(my_string13_a.isalpha())
```

```
I love Python
```

```
False
```

### 4.14 .isascii() Method

String တစ်ခုအတွင်းရှိ character များအားလုံးသည် ASCII character များ ဟုတ် မဟုတ် စစ်ပေးသည်။ (checks whether all characters in the string are ASCII or not.)။ ASCII character ဆိုသည်မှာ print ထုတ်၍ ရသည့် characters (that have a printable representation) ဖြစ်သည်။ U+0000 မှ U+007F အတွင်း ဖြစ်သည်။ ဥပမာ- ° ကို \xb0 အဖြစ်သို့ ပြောင်းပေးသည်။

```
my_string14 = "I love Python."
print(my_string14)
# checks whether all characters in the string are ASCII or not.
print(my_string14.isascii())
```

```
I love Python.
True
```

```
my_string14_a = "Æðþʊ"        # non-ASCII characters
print(my_string14_a)
print(my_string14_a.isascii())
```

```
Æðþʊ
False
```

```
print(ascii('°'))
print(type(ascii('°')))
```

```
'\xb0'
<class 'str'>
```

### 4.15 .isdecimal() Method

**.isdecimal()** method သည် string တစ်ခုအတွင်းရှိ character များ အားလုံးသည် digits နှင့် numeric character များ ဖြစ်နေလျှင် True ပြန်ပေးသည်။ ထိုသို့မဟုတ်လျှင် False ပြန်ပေးသည်။

```
my_string15 = "64329"
print(my_string15)
print(my_string15.isdecimal())
```

```
64329
True
```

```
my_string15_a ="64f3 29"   # alphabets နှင့် space များ ပါဝင်နေသောကြောင့် False
print(my_string15_a)
print(my_string15_a.isdecimal())
```

```
64f3 29
False
```

```
my_string15_b = "\u0033"          #unicode for 3
print(my_string15_b.isdecimal())
```

```
True
```

```
my_string15_c = "\u0030"           #unicode for 0
print(my_string15_c.isdecimal())
```

```
True
```

```
my_string15_d = "\u0047"           #unicode for G
print(my_string15_d.isdecimal())
```

```
False
```

**4.16 .isdigit() Method**

Character များ အားလုံးသည် digit များ ဖြစ်နေလျှင် True ပြန်ပေးသည်။ ထိုသို့မဟုတ်လျှင် False ပြန်ပေးသည်။ (returns True if all the characters are digits, otherwise False.)

```
my_string16 = "6574292"
print(my_string16)
print(my_string16.isdigit())
```

```
6574292
True
```

```
my_string16_a = "837hghhh  4834"    # includes spaces and alphabet.
print(my_string16_a)
print(my_string16_a.isdigit())
```

```
837hghhh  4834
False
```

```
my_string16_b = "\u0030"            #unicode for 0
print(my_string16_b.isdigit())
```

```
True
```

```
my_string16_c = "\u00B2"           #unicode for 2
print(my_string16_c.isdigit())
```

```
True
```

**4.17 .isidentifier() Method**

String သည် valid identifier တစ်ခု ဖြစ်လျှင် True ပြန်ပေးသည်။ ထို့သို့ မဖြစ်ခဲ့ လျှင် False ပြန်ပေးသည်။ (returns True if the string is a valid identifier, otherwise False.)။ String တစ်ခုသည် alphanumeric letters (a-z) and (0-9) ဖြစ်နေလျှင်သော်လည်းကောင်း၊ underscores (_) ပါဝင်နေလျှင် သော်လည်းကောင်း valid identifier ဖြစ်သည်။ Valid identifier တစ်ခု ဖြစ်ရန် number ဖြင့် မ စ,ရ သို့မဟုတ် space မပါဝင်စေရ။

```
my_string17 = "I love Python."
print(my_string17.isidentifier())
```

```
False
```

```
a = "MyFolder"
print(a.isidentifier())
```

```
True
```

```
b = "Demo002"
print(b.isidentifier())
```

```
True
```

```
c = "2bring"
print(c.isidentifier())
```

```
False
```

```
d = "my demo"
print(d.isidentifier())
```

```
False
```

### 4.18 .islower() Method

Character များ အားလုံးသည် အသေးစာလုံး(lower case)များ ဖြစ်နေလျှင် True ပြန်ပေးသည်။ ထိုသို့ မဟုတ်လျှင် False ပြန်ပေးသည်။ နံပါတ်များ(numbers)၊ သင်္ကေတများ(symbols )နှင့် space များကို မစစ်ဆေးပါ။ Alphabet များကို သာ စစ်ဆေး ပါသည်။

```
my_string18 = "I love Python."
print(my_string18.islower())
```

```
False
```

```
a = "Hello world!"
print(a.islower())
```

```
False
```

```
b = "hello 123"
```

```
print(b.islower())
```

```
True
```

```
c = "mynameisPeter"
print(c.islower())
```

```
False
```

**4.19 .isnumeric() Method**

Character များ အားလုံးသည် နံပါတ်များ(numeric (0-9)) ဖြစ်နေလျှင် True ပြန်ပေးသည်။ ထိုသို့မဟုတ်လျှင် False ပြန်ပေးသည်။ ထပ်ကိန်းများနှင့် အပိုင်းကိန်းများကိုလည်း နံပါတ်များ(numeric)အဖြစ်သတ်မှတ်သည်။ (Exponents, like ² and ¾ are also considered to be numeric values.)

```
my_string19 = "I love Python."
print(my_string19.isnumeric())
```

```
False
```

```
a = "\u0030"          # unicode for 0
print(a.isnumeric())
```

```
True
```

```
b = "\u00B2"          # unicode for &sup2;
print(b.isnumeric())
```

```
True
```

```
c = "10km2"
print(c.isnumeric())
```

```
False
```

**4.20 .isprintable() Method**

Character များ အားလုံးသည် ပုံနှိပ်ဖော်ပြနိုင်သည့် စာလုံးများ ဖြစ်နေလျှင် True ပြန်ပေးသည်။ ထိုသို့မဟုတ်လျှင် False ပြန်ပေးသည်။ (returns True if all the characters are printable, otherwise False.) ပုံနှိပ် ဖော်ပြနိုင်သည့် စာလုံးများ ဟုတ် မဟုတ် စစ်ဆေးရန် သုံးသည်။

```
my_string20 = "I love Python."
# checks whether the string can be printed or not.
print(my_string20.isprintable())
```

```
True
```

```
my_string20_a = "Hello!\nAre you       #1?"
x = my_string20_a .isprintable()
```

```
print(x)
```

```
False
```

### 4.21 .isspace() Method

String တစ်ခုအတွင်းရှိ character များ အားလုံးသည် whitespaces ဖြစ်နေလျှင် True ပြန်ပေးသည်။ ထိုသို့မဟုတ်လျှင် False ပြန်ပေးသည်။(returns True if all the characters in a string are whitespaces, otherwise False.)

```
my_string21 = "I love Python."
# checks whether the string includes only spaces or not.
print(my_string21.isspace())
```

```
False
```

```
my_string2_a = "      "
print(my_string2_a.isspace())
```

```
True
```

```
my_string2_b = "   K  "
print(my_string2_b.isspace())
```

```
False
```

### 4.22 .istitle() Method

စာလုံးများအားလုံးသည် အကြီးစာလုံး(upper case letter)ဖြင့် စပြီး ကျန်စာလုံးများသည် အသေး စာလုံး(lower case letters) ဖြစ်လျှင် True ပြန်ပေးသည်။ ထိုသို့မဟုတ်လျှင် False ပြန်ပေးသည်။ သင်္ကေတများ(symbols)နှင့် နံပါတ်များ(numbers)နှင့် မသက်ဆိုင်ပါ။(Symbols and numbers are ignored.)

```
my_string22 = " I love Python.  "
print(my_string22)
print(my_string22.istitle())    # love ၏ ပထမဆုံး အက္ခရာသည် အကြီးစာလုံးမဟုတ်ပါ။
```

```
 I love Python.
False
```

```
my_string22_a = " I love Python.  "
print(my_string22_a.istitle())
```

```
False
```

```
a = "HELLO, AND WELCOME TO MY WORLD"
print(a.istitle())
```

```
False
```

```
b = "Hello"
print(b.istitle())
```

True

```
c = "22 Names"
print(c.istitle())
```

True

```
d = "This Is %'!?"
print(d.istitle())
```

True

**4.23 .isupper() Method**

Character များ အားလုံးသည် အကြီးစာလုံး(upper case)ဖြစ်နေလျှင် True ပြန်ပေးသည်။ ထိုသို့ မဟုတ်လျှင် False ပြန်ပေးသည်။ သင်္ကေတများ(Symbols)နှင့် နံပါတ်များ(numbers)နှင့် မသက်ဆိုင်ပါ။ အက္ခရာများ(alphabet characters)ကိုသာ စစ်ဆေးသည်။ (Numbers, symbols and spaces are not checked, only alphabet characters.)

```
my_string23 = "KAUNG HTAT NYUNT"
print(my_string23.isupper())
```

True

```
my_string23_a = "Kaung Htat Nyunt"
print(my_string23_a.isupper())
```

False

```
a = "@WORLD"
print(a.isupper())
```

True

```
b = "HELLO 123"
print(b.isupper())
```

True

**4.24 .join() Method**

String မှ character တစ်လုံးချင်းစီ(iterable items)ကို ယူ၍ သတ်မှတ်ထားသည့် character နှင့် ပေါင်းပေးသည်။(takes all items in an iterable and joins them into one string.)

```
my_string24 = "IlovePython"
print("--".join(my_string24))    # "--" နှင့် ဆက်သည်။
print("----original string below----")
```

```
print(my_string24)
```

```
--I--l--o--v--e--P--y--t--h--o--n
----original string below----
IlovePython
```

```
my_string24 = "IlovePython"

print(" ".join(my_string24))    # " "(space) နှင့် ဆက်သည်။

# it is joined using -- here in my example.
```

```
I l o v e P y t h o n
```

**4.25 .ljust() Method**

String ကို ဘယ်ညှိပေးသည်။ (will left align the string, using a specified character (space is default) as the fill character.)။ သတ်မှတ်ထားသည့် character အရေအတွက် ပြည့်အောင် ထည့်ပေးသည့် character ဖြင့် ဖြည့်ပေးသည်။

```
my_string25 = "I love Python."

print(my_string25.ljust(50,'*'))    # character အရေအတွက် 50 အတွင်း
```

```
I love Python.************************************
----original string below----
I love Python.
```

```
txt = "banana"

x = txt.ljust(20, "_")

print(x)
```

```
banana______________
```

**4.26 .lower() Method**

String တစ်ခုမှ အကြီးစာလုံး(uppercase)character များကို အသေးစာလုံး (lowercase) အဖြစ်သို့ ပြောင်းပေးသည်။ String တွင် အကြီးစာလုံး(uppercase character)များ မပါရှိပါက မူလ string အတိုင်းသာ ဖြစ်လိမ့်မည်။

```
my_string26 = "I love Python."

print(my_string26.lower())

print("----original string below----")

print(my_string26)
```

```
i love python.
----original string below----
I love Python.
```

**4.27 .lstrip() Method**

Character ၏ အရှေ့တွင် ရှိသည့် မလိုလားအပ်သည့် space, comma စသည်တို့ကို ဖယ်ရှားပေးသည်။ (removes any leading characters (space is the default leading character to remove))

```
my_string27 = "          I love Python.  " # space များကို ဖယ်ရှားပေးရန်
print(my_string27.lstrip())
print("----original string below----")
print(my_string27)
```

```
I love Python.
----original string below----
          I love Python.
```

```
txt = ",,,,,ssaaww.....banana"  " # .,a,s,w များကို ဖယ်ရှားပေးရန်
x = txt.lstrip(",.asw")
print(x)
```

```
banana
```

**4.28 .maketrans() Method + translate() Method**

**.maketrans()** method သည် translate() function တွင် သုံးရန်အတွက် translate table ကို ထုတ်ပေးသည်။ (This method returns a translate table to be used translate() function.)။ Translation လုပ်ပြီး dictionary ထဲတွင် key, value pair အဖြစ် သိမ်းဆည်းထားသည်။

I love Python." မှ I ကို 1 ဖြင့်လည်းကောင်း v ကို 2 ဖြင့်လည်းကောင်း P ကို 3 ဖြင့်လည်းကောင်း y ကို 1 ဖြင့်လည်းကောင်း အစားထိုးမည်။ `I love Python.` → `1 lo2e 34thon.`

```
my_string28 = "I love Python."
translation = my_string28.maketrans("IvPy", "1234")
print(translation)  # "IvPy" နှင့် "1234" ကို  key, value pair ဖြစ်ပြောင်းသည်။
# character တစ်ခုချင်းစီကို Unicode representation အဖြစ် translation လုပ်သည်။
print("73: 49  is ", chr(73),chr(49))
print(my_string28.translate(translation))
print("----original string below----")
print(my_string28)
```

```
{73: 49, 118: 50, 80: 51, 121: 52}
I
1 lo2e 34thon.
----original string below----
I love Python.
```

**4.29 .partition() Method**

**.partition()** method သည် သတ်မှတ်ပေးသည့်(specified) string ကို ရှာဖွေပြီး string ကို tuple (၃)ခု ဖြစ်အောင်ခွဲသည်။ သတ်မှတ်ပေးသည့်(specified string)ကို နောက်ဆုံးတွေ့သည့်နေရာကို မူတည်၍ tuple (၃)ခု ဖြစ်အောင်ခွဲသည်။ သတ်မှတ်ပေးသည့်(specified string) မတိုင်ခင်နေရာ(everything before the "match") သတ်မှတ်ပေးသည့်(specified string) ပါသည့်နေရာ( element contains the specified string.) သတ်မှတ် ပေးသည့်(specified string) ပါသည့်နေရာ နောက်ပိုင်း(everything after the "match")

**string.partition(value)**

```
my_string29 = " I love Python."
print(my_string29.partition("love"))
```

```
(' I ', 'love', ' Python.')
```

```
my_string29_a = "Python and Machine Learning for Myanmar Engineers"
print(my_string29_a.partition("for"))
```

```
('Python and Machine Learning ', 'for', ' Myanmar Engineers')
```

**4.30 .replace() Method**

စာလုံး(phrase)တစ်ခုကို တခြားစာလုံး(phrase) ဖြင့် အစားထိုးသည်။ အရှေ့စာလုံးသည် အစားထိုးခံရမည့် စာလုံး(phrase)ဖြစ်သည်။ အနောက် စာလုံးသည် အစားထိုးမည့် စာလုံး(phrase)ဖြစ်သည်။

```
my_string30 = " I love Python.  "
print(my_string30.replace("Python", "apple"))
```

```
I love apple.
```

**4.31 .rfind()method**

နောက်ဆုံးတွေ့ခဲ့သည့်နေရာ(index နံပါတ်)ကို ဖော်ပြပေးသည်။ အတူညီဆုံးဖြစ်သည့်နေရာကို ရှာပေးသည်။) တူသည့်စာကို ရှာမတွေ့ပါက -1 ထုတ်ပေးသည်။ (returns -1 if the value is not found.) .rfind() method သည် .rindex() method ခပ်ဆင်ဆင်နှင့်တူညီသည်။

```
my_string31 = "I love Python."
print(my_string31.rfind("Py"))
```

```
7
```

```
my_string31 = "I love Python and Python."
print(my_string31.rfind("Py"))
```

```
18
```

**4.32 .rindex() Method**

နောက်ဆုံးတွေ့ခဲ့သည့်နေရာ(index နံပါတ်)ကို ဖော်ပြပေးသည်။ အတူညီဆုံးဖြစ်သည့် substring ၏ နေရာ(index နံပါတ်)ကို ရှာပေးသည်။ တူသည့်စာကို ရှာမတွေ့ပါက -1 ထုတ်ပေးသည်။ (returns -1 if the value is not found.) rindex() method သည် rfind() method ခပ်ဆင်ဆင်နှင့်တူညီသည်။

```
my_string32 = "I love Python."
print(my_string32.rindex("Py"))
```

```
7
```

```
txt = "Hello, welcome to my world."
x = txt.rindex("o")
print(x)
```

```
22
```

**4.33 .rjust() Method**

ညာညှိရန် character အရေအတွက် ထည့်ပေးရသည်။ ထို character အရေအတွက် နေရာတွင် ညာညှိ(right justified) ပေးသည်။ တခြား characters များလည်း ထည့်နိုင်သည်။

```
my_string33 = " I love Python.  "
print(my_string33.rjust(50, "*"))
# စာလုံးရေ (၅၀)စာနေရာ(50 characters long) တွင် ညာညှိ(right justified) ပေးသည်။
```

```
********************************* I love Python.
```

```
my_string33 = " I love Python.  "
print(my_string33.rjust(50, " "))# character (၅၀) နေရာတွင် space ဖြင့် ညာညှိသည်။
```

**4.34 .rpartition() method**

**.rpartition()** method သည် သတ်မှတ်ပေးသည့်(specified) string ကို ရှာဖွေပြီး string ကို tuple (၃)ခု ဖြစ်အောင် ခွဲသည်။ သတ်မှတ်ပေးသည့်(specified string)ကို နောက်ဆုံးတွေ့သည့်နေရာကို မူတည်၍ tuple ၃ခု ဖြစ်အောင် ခွဲသည်။ (၁) သတ်မှတ်ပေးသည့်(specified string) မတိုင်ခင်နေရာ(everything before the "match")။ (၂) သတ်မှတ်ပေးသည့်(specified string) ပါသည့်နေရာ( element contains the specified string.)။ (၃) သတ်မှတ်ပေးသည့်(specified string) ပါသည့်နေရာ နောက်ပိုင်း(everything after the "match")

```
string.rpartition(value)
```

```
my_string34 = "I am a student learning Computer Science. Computer Scien
ce is an amazing discipline. "
print(my_string34.rpartition("Computer"))
print("----original string below----")
print(my_string34)
```

```
('I am a student learning Computer Science. ', 'Computer', ' Science
 is an amazing discipline. ')
----original string below----
I am a student learning Computer Science. Computer Science is an ama
zing discipline.
```

**4.35 .rsplit() Method**

String တစ်ခုကို list တစ်ခု အဖြစ်ခွဲပေးသည်။ ညာဘက်စ၍ ခွဲပေးသည်။ (splits a string into a list, starting from the right.) "max" ကို မသတ်မှတ်ပေးလျှင် split() method မှ ထုတ်ပေးသည့် ရလဒ်နှင့် တူညီလိမ့်မည်။

Note: When maxsplit is specified, the list will contain the specified number of elements plus one.

```
my_string35 = "I love Python."
print(my_string35.rsplit(" "))


my_string35_a= "apple, banana, cherry"
print(my_string35_a.rsplit(",", 2))
```

```
['I', 'love', 'Python.']
['apple', ' banana', ' cherry']
```

**4.36 .rstrip() Method**

String တစ်ခုရှိ အဆုံး(trailing) character များကို ဖယ်ရှားပေးသည်။ String ၏ နောက်ဆုံး၌ ရှိသည့် space သည် ဖယ်ရှားပေး ရမည့်အရာ ဖြစ်သည်။

```
my_string36 = " I love Python....."
print(my_string36.rstrip("."))          # "." ကို ဖယ်ရှားပေးရန်
```

```
I love Python
```

```
my_string36_a = "banana,,,,,ssaaww....."
print(my_string36_a.rstrip(",.asw")) # ",", ., a, s, w ကို ဖယ်ရှားပေးရန်
```

```
banan
```

**4.37 .split() Method**

String တစ်ခုကို list တစ်ခုအဖြစ် ခွဲပေးသည်။ မိမိလိုချင်သည့် separator ကို သတ်မှတ် ပေးနိုင်သည်။ Default separator သည် whitespace ဖြစ်သည်။

`string.split(separator, maxsplit)`

```
my_string37 = "I love Python."
print(my_string37.split())
print(my_string37.split(" "))          # same result
```

```
['I', 'love', 'Python.']
['I', 'love', 'Python.']
```

**4.38 .splitlines() Method**

String တစ်ခုကို list တစ်ခု အဖြစ် ခွဲပေးသည်။(splits a string into a list. ) စာကြောင်းအဆုံးတွင် line break ဖြစ်သည်။ (The splitting is done at line breaks.)

```
string.startswith(value, start, end)
```

```
my_string38 = " I love Python.\n I plan to learn Python because it is f
un.\n These lines will be separated. "
print(my_string38)
print(my_string38.splitlines(True))
print(my_string38.splitlines(False))
```

```
I love Python.
 I plan to learn Python because it is fun.
 These lines will be separated.
[' I love Python.\n', ' I plan to learn Python because it is fun.\n'
, ' These lines will be separated. ']
[' I love Python.', ' I plan to learn Python because it is fun.', '
These lines will be separated. ']
```

**4.39 .startswith() Method**

String တစ်ခုသည် သတ်မှတ်ထားသည့် character ဖြင့် စထားလျှင် True ပြန်ပေးသည်။ ထိုသို့ မဟုတ်လျှင် False ပြန်ပေးသည်။( returns true if the string begins with the characters, and false if not.) အကြီးစာလုံး အသေးစာလုံးမှန်ရန် လိုအပ်သည်။ case sensitive ဖြစ်သည်။)

```
string.startswith(value, start, end)
```

```
my_string39 = "I love Python.  "
print(my_string39.startswith("i"))
```

```
False
```

```
my_string39_a = "Hello, welcome to my world."
# ရှာရမည့် အစနေရာနှင့် အဆုံးနေရာ ထည့်ပေးပြီး ရှာခိုင်းသည်။
print(my_string39_a.startswith("wel", 7, 20))
```

```
True
```

**4.40 .strip() Method**

String ၏ အစနှင့် အဆုံးနေရာတို့တွင် ရှိနေသည့် space များကို ဖယ်ရှားပေးသည်။ (Remove spaces at the beginning and at the end of the string)

```
my_string40 = "         I love Python.         "
```

```
print(my_string40.strip())
```

```
I love Python.
```

### 4.41 .swapcase() Method

String တစ်ခုရှိ အကြီးစာလုံးများ(upper case letters)ကို အသေးစားလုံး(lower case )များ အဖြစ်သို့ လည်းကောင်း အသေးစားလုံး(lower case)များကို အကြီးစာလုံးများ(upper case letters)ဖြစ်သို့ လည်းကောင်း ပြောင်းပေးသည်။ (returns a string where all the upper case letters are lower case and vice versa.) `string.swapcase()`

```
my_string41 = "I love Python."
print(my_string41.swapcase())
```

```
i LOVE pYTHON.
```

### 4.42 .title() Method

String တစ်ခုကို ခေါင်းစဉ်(title)အဖြစ် ပြောင်းပေးသည်။ ခေါင်းစဉ်(title)ဆိုသည်မှာ စာလုံး၏ အစ အက္ခရာသည် အကြီးစာလုံး ဖြစ်ရမည်။ စာလုံး(word)တိုင်း၏ ပထမဆုံး character ကို အကြီးစာလုံး(upper case) အဖြစ် ပြောင်းပေးသည်။ Report များပြင်ဆင်သည့်အခါ ခေါင်းစဉ်(title)တပ်ရန် format ချသည့် အခါတွင် သုံးသည်။

စာလုံး(word)တွင် သင်္ကေတများ(symbols)နှင့် နံပါတ်များ(numbers)ပါဝင်နေပါက ထို symbol သို့မဟုတ် နံပါတ်(number) နောက် character ကို ပြောင်းပေးသည်။ (If the word contains a number or a symbol, the first letter after that will be converted to upper case.)

```
my_string42 = "I love Python.   "
print(my_string42.title())
```

```
I Love Python.
```

```
my_string42_a = "Welcome to my 2nd world"
print(my_string42_a .title())
```

```
Welcome To My 2Nd World
```

### 4.43 .upper() Method

String တစ်ခု အတွင်းရှိ character များအားလုံးကို အကြီးစာလုံး(upper case) ဖြစ်အောင် ပြောင်းပေးသည်။ သင်္ကေတများ(symbols)နှင့် နံပါတ်(numbers)များ မသက်ဆိုင်ပါ။ (Symbols and numbers are ignored.)

```
my_string44 = "I love Python."
print(my_string44.upper())        # အားလုံးကို အကြီးစာလုံးဖြစ်အောင် ပြောင်းပေးသည်။
```

```
I LOVE PYTHON.
```

```
my_string44_a = "this is string example....wow!!!"
print (my_string44_a.upper())
```

```
THIS IS STRING EXAMPLE....WOW!!!
```

**4.44 .zfill() Method**

String ၏ အစတွင် zeros(0) ထည့်ပေးသည်။ မိမိသတ်မှတ်ထားသည့် အရှည်(specified length) ပြည့်အောင် ထည့်ပေးသည်။ (adds zeros (0) at the beginning of the string, until it reaches the specified length.)

**.len()** method ကို သုံး၍ မိမိအလိုရှိသည့် zeros (0) အရေအတွက်အတိုင်း ထည့်ပေါင်းပေး နိုင်သည်။ String ၏ အရှည်(length) မည်မျှရှိသည်ကို ရေတွက်နေရန် မလိုတော့ပေ။ String ၏ အရှည်သည် မိမိ သတ်မှတ် ထားသည့် အရှည်(specified length)ထက်ပိုများနေပါက မထည့်ပေးပါ။ (If the value of the len parameter is less than the length of the string, no filling is done.)

```
string.zfill(len)
```

```
my_string45 = "I love Python."
print(my_string45.zfill(30))
print(my_string45.zfill(len(my_string45)+3))
print(my_string45.zfill(10))
```

```
0000000000000000I love Python.
000I love Python.
I love Python.
```

```
a = "hello"
print(a.zfill(10))
```

```
00000hello
```

```
b = "welcome to the jungle"
print(b.zfill(10))
```

```
welcome to the jungle
```

```
c = "10.000"
print(c.zfill(10))
```

```
000010.000
```

-End-