

python string method.

### 1.capitalize()

Converts the first character to upper case

ရှေ့ဆုံးစာလုံးကို အကြီးပြောင်းချင်ရင် .capitalize () ကိုသုံးပါတယ်။

အသုံးပြုပုံက .upper() .lower() တွေနဲ့အတူတူပါပဲ။

```
1 x = "nothing is forever."  
2 y = x.capitalize()  
3 print( y )
```

```
Nothing is forever.
```

### 2.casefold()

Converts string into lower case

စာလုံးအသေးပြောင်းချင်ရင်သုံးပါတယ်။

```
1 x = "Hello World"  
2 y = x.casefold()  
3 print( y )
```

```
hello world
```

### 3.center()

Returns a centered string

အလယ်မှာပဟိုထားပြီး အပိုစာလုံးတွေကို ဘေးကပြချင်ရင်  
သုံးပါတယ်။

.center(စာလုံးအရေအတွက် , အပိုစာလုံး)

```

1 x = "Hello World"
2 y = x.center(19, '*')
3 print( y )

```

xရဲ့တန်ဖိုးဟာ စာ11လုံးဖြစ်ပါတယ်။

x ကို ပဟိုထားပြီး စာ19 လုံး ထုတ်ခိုင်းလိုက်ပါတယ်။

ဘေးတစ်ဖက်စီမှာ ပြရမယ့်စာကို \* လို့ပေးလိုက်ပါတယ်။

```

****Hello World****

```

မူလ 11 လုံးကို ပဟိုထားပြီး 19လုံးဖြစ်အောင်ထုတ်ခိုင်းတဲ့အတွက်  
\* ရှစ်ခု ထပ်ထွက်လာပါတယ်။

အလွယ်ရေးချင်ရင် အောက်ကလိုရေးနိုင်ပါတယ်။

```

1 x = "Hello World"
2 y = x.center(len(x)+8, '*')
3 print( y )

```

အပေါ်က program ရဲ့ 8 နေရာမှာ 10 လို့ ပြောင်းရေးရင်  
ဘယ်ငါးလုံး ညာငါးလုံး ထွက်လာမှာပါ။

လေ့ကျင့်ခန်း ။ ။ 8 နေရာမှာ 9 ထားပြီး စမ်းကြည့်ပါ။

#### 4.count()

Returns the number of times a specified value occurs in a string

စာထဲမှာရှိတဲ့ စာလုံးအရေအတွက်သိချင်ရင် သုံးပါတယ်။

.count( မိမိရေတွက်ချင်သော စာလုံး)

စာတစ်လုံး ဒါမှမဟုတ် အတွဲလိုက် ရေတွက်လို့ရပါတယ်။

```
1 x = "Nothing is forever.We can change the future."  
2 y = x.count('i')  
3 print(y)
```

i ဆိုတဲ့စာလုံးကို ရေတွက်ခိုင်းတာပါ။

```
2
```

ဒုတိယနည်းက တည်နေရာ အကန့်အသတ်နဲ့ရေတွက်တဲ့နည်းပါ။  
စမှတ်နဲ့ဆုံးမှတ်နဲ့ ထပ်ထည့်ဖို့ လိုအပ်ပါတယ်။

```
1 x = "Nothing is forever.We can change the future."  
2 y = x.count('i', 0, 8)  
3 print(y)
```

တည်နေရာ zero ဟာ N ဖြစ်ပြီး တည်နေရာ 8 ဟာ g ဖြစ်လို့  
Nothing ဆိုတဲ့ စာထဲက i ကိုပဲ ရေတွက်မှာပါ။

```
1
```

5.encode().....I don't know clearly.

6.endswith()

Returns true if the string ends with the specified value

မိမိသတ်မှတ်ထားသော စာလုံးများနှင့်အဆုံးသတ်/မသတ် စစ်ဆေးရာတွင်သုံးသည်။

.endswith(စစ်မည့်စာ)

```
1 x = "I love python."  
2 y = x.endswith("python.")  
3 print( y )
```

အဆုံးသတ်လျှင် True ဖြစ်ပြီး မဟုတ်ခဲ့လျှင် False တန်ဖိုးထွက်မည်။

python. သည် x တန်ဖိုး၏ အဆုံးသတ်ဖြစ်၍ True ဟုထွက်သည်။

```
True
```

စမှတ် ဆုံးမှတ်ထည့်ပြီးလည်းစစ်နိုင်သည်။

.endswith(စစ်မည့်စာ , စမှတ် , ဆုံးမှတ်)

```
1 x = "I love python."  
2 y = x.endswith("e" ,0 ,6)  
3 print( y )
```

တည်နေရာ zero မှ 6 ထိသည် I love ဖြစ်ပြီး e နှင့် ဆုံးသဖြင့် True ဖြစ်သည်။

(Tuple နှင့်သုံးလိုပါက စာနေရာတွင် ပြောင်းထည့်ရုံသာဖြစ်သည်။)

7.expandtabs() Sets the tab size of the string  
စာအတွင်းရှိ tab (space bar) ၏ အရွယ်အစားကို သတ်မှတ်ရာ  
တွင်သုံးသည်။

.expandtabs(tab size)

```
1 x = "Hello\tWorld"
2 y = x.expandtabs(20)
3 print( y )
```

select tab ဟုအရှည်ကောက်ခေါ်သော \t ၏ နေရာတွင်  
space bar အချက် 20 ဝင်လာမည်ဖြစ်သည်။

```
Hello                World
```

8.find() Searches the string for a specified  
value and returns the position of where it was  
found

မိမိရှာလိုသော စာလုံးရဲ့ တည်နေရာကို သိချင်ရင်သုံးပါတယ်။

.find(ရှာလိုသောစာ)

```
1 x = "Hello World"
2 y = x.find('o')
3 print( y )
```

x တန်ဖိုးထဲက အိုရဲ့ တည်နေရာကို ရှာခိုင်းတာပါ။

```
4
```

အိုဟာ တည်နေရာ 4 မှာရှိလို့ 4ကို ထုတ်ပေးပါတယ်။



### 9.format() Formats specified values in a string

စာတွေကို format ချို့ သုံးပါတယ်။

.format(format ချမည့်စာ)

```
1 x = "Hello {}"  
2 print(x)  
3  
4 y = x.format('a')  
5 print( y )
```

format နောက်က ကွင်းထဲက စာဟာ x တန်ဖိုးထဲက တွန့်ကွင်း  
နေရာမှာ ဝင်မှာပါ။ မြင်သာအောင်လို့ format မချောင် x တန်ဖိုးနဲ့  
x ကို a နဲ့ format ချလိုက်တဲ့ y တန်ဖိုး နှစ်ခုလုံးကို print ထုတ်ပြ  
ထားပါတယ်။

```
Hello {}  
Hello a
```

format နဲ့အများကြီးသုံးလို့ရပါတယ်။

User ဖြေတဲ့နာမည်နဲ့ နှုတ်ပြန်ဆက်တာကို နမူနာရေးပြထားပါတယ်။

```
1 x = "Hello {} . Nice to meet you."  
2 print(x.format(input('What is your name ?\nAnswer  
')))
```

```
What is your name ?  
Answer : Myo Thant Zin  
Hello Myo Thant Zin . Nice to meet you.
```

ကွက်လပ်ထဲ မိုးသန့်ဇင်လို့ ထည့်လိုက်လို့ hello ဆိုပြီးပြန်ပြတာပါ။

## 10.format\_map()

Formats specified values in a string

တည်နေရာပေးပြီး format ချရင် သုံးပါတယ်။

.format\_map({တည်နေရာ : ထည့်မည့်စာ})

```
1 x = "Hello {a}.Nice to meet you."  
2 y = x.format_map({'a' : 'mg mg'})  
3 print(y)
```

တည်နေရာ a ကို mg mg ဆိုတဲ့ တန်ဖိုးထည့်လိုက်တာပါ။

```
Hello mg mg.Nice to meet you.
```

{ တည်နေရာ: ထည့်မည့်စာ }

အထက်ပါတွန့်ကွင်းကို dictionary လို့ခေါ်ပါတယ်။

dictionary ကို သိခြားရှင်းပြမှာပါ။

## 11.index()

Searches the string for a specified value and returns the position of where it was found

မိမိရှာလိုသော စာလုံးရဲ့တည်နေရာကို သိချင်ရင်သုံးပါတယ်။

.index(ရှာလိုသောစာ)

နံပါတ် 8 find( ) နဲ့တူတူပါပဲ။

```
1 x = "Hello Wotld"
2 y = x.index('o')
3 print(y)
```

အိုရဲ့ တည်နေရာကို ရှာတာပါ။

```
4
```

x.index(ရှာမည့်စာ , စမှတ် , ဆုံးမှတ်)

x.find(ရှာမည့်စာ , စမှတ် , ဆုံးမှတ်)

အပေါ်ကလို သုံးလည်းရပါတယ်။



## 12.isalnum()

Returns True if all characters in the string are alphanumeric

စာနဲ့ရောထားတဲ့ ဂဏန်းပါမပါစစ်ရင်သုံးပါတယ်။

.isalnum() ၏ အရှည်ကောက်က is alphanumeric ပါ။

```
1 x = "Hello1"  
2 y = x.isalnum()  
3 print( y )
```

```
True
```

## 13.isalpha()

Returns True if all characters in the string are in the alphabet

string ထဲက စာအားလုံးဟာ အက္ခရာ(a to z)ဖြစ်မဖြစ် စစ်ရင်သုံးပါသည်။

```
1 x = "Hello World * "  
2 y = x.isalpha()  
3 print(y)
```

a to z မဟုတ်တဲ့ \* သင်္ကေတ ပါလို့ x တန်ဖိုးဟာ alphabet သီးသန့်မဟုတ်တော့ပါဘူး။

```
False
```

#### 14.isdecimal()

Returns True if all characters in the string are decimals

ဧကထဲမှာ 0 to 9 ဆိုတဲ့ ဆယ်လီစိတ်ကိန်းတွေပဲပါမပါ စစ်ရင်သုံးပါတယ်။

```
1 x = "16531580"  
2 y = x.isdecimal()  
3 print(y)
```

x တန်ဖိုးထဲမှာ 0 to 9 decimal ကိန်းတွေပဲ ထည့်ထားပါတယ်။

```
True
```

#### 15.isdigit()

Returns True if all characters in the string are digits

ဧကထဲမှာ 0 ,1 ဆိုတဲ့ digitalကိန်းတွေပဲပါမပါ စစ်ရင်သုံးပါတယ်။

```
1 x = "1101001010"  
2 y = x.isdigit()  
3 print( y )
```

zero နဲ့ one ပဲ ထည့်ထားပြီး စစ်ပါတယ်။ရလဒ်ကို y တန်ဖိုးအဖြစ် သိမ်းလိုက်ပြီး ပြန်ပြပါတယ်။

```
True
```

### 16.isidentifier()

Returns True if the string is an identifier

ကိုယ်စားပြု စာ ဟုတ်မဟုတ် စစ်ရင်သုံးပါတယ်။

space တွေ number တွေမပါရပါဘူး။

စာသီးသန့် ကပ်ပြီးရေးရင် identifier ပါ။

```
1 x = "Car"
2 y = x.isidentifier()
3 print( y )
```

စာသီးသန့်ကပ်ရေးထားပါတယ်။car

```
True
```

### 17.islower()

Returns True if all characters in the string are lower case

စာအားလုံး အသေးစာလုံးဟုတ်မဟုတ်စစ်တာပါ။

a to z အကြီးစာလုံးမပါရင် True ဖြစ်ပါတယ်။

```
1 x = "abcdefg h23"
2 y = x.islower()
3 print( y )
```

A to Z မပါပါဘူး။

```
True
```

### 18.isnumeric()

Returns True if all characters in the string are numeric

စာအားလုံး number ဟုတ်မဟုတ် စစ်ချင်ရင် သုံးပါတယ်။

နံပါတ်အဖြစ်သုံးတာမှန်သမျှ လက်ခံပါတယ်။ ရောမကိန်းတွေလည်းရပါတယ်။ အပိုင်းကိန်းလည်းရပါတယ်။

```
1 x = "\u00bd"
2 print(x)
3 y = x.isnumeric()
4 print(y)
```

u00bd ဟာ 1 by 2 ဆိုတဲ့အပိုင်းကိန်းပါ။

သူ့ကို print ထုတ်ရင် 1/2 လို့ပြမှာပါ။

```
1/2
True
```

အပိုင်းကိန်းဟာ number ဖြစ်ပါတယ်။

```
1 x = "\u00b1"
2 print(x)
3 y = x.isnumeric()
4 print(y)
```

```
±
False
```

u00b1 ဟာ ± ဖြစ်ပြီး number ထဲမှာ မပါပါ။

### 19.isprintable()

Returns True if all characters in the string are printable

စာအဖြစ် print ထုတ်လို့ရမရသိချင်ရင် သုံးပါတယ်။

တော်တော်များများကို print ထုတ်လို့ရပါတယ်။ ဒါပေမယ့်

\ သင်္ကေတကို ထုတ်မရပါဘူး။

သူ့ရဲ့ code နံပါတ်က 27 ပါ။

```
1 x = chr(27)
2 y = x.isprintable()
3 print(y)
```

character code 27 = \

```
True
```

### 20.isspace()

Returns True if all characters in the string are whitespaces

စာထဲမှာ space တွေပဲ ပါမပါ စစ်တာပါ။

```
1 x = "      "
2 y = x.isspace()
3 print( y )
```

```
True
```



## 21.istitle()

Returns True if the string follows the rules of a title

စာရေးတဲ့ပုံစံကို လိုက်နာမှုရှိမရှိစစ်တဲ့အခါ သုံးပါတယ်။

```
1 x = "I am a man!"  
2 y = x.istitle()  
3 print(y)
```

I am a man. ဟာ စာရေးတဲ့ပုံစံကို လိုက်နာပါတယ်။

```
False
```

x တန်ဖိုးမှာ full stop ' . ' မပါဘဲကျန်နေလို့ false ထွက်ပါတယ်။

## 22.isupper()

Returns True if all characters in the string are upper case

A to Z စာလုံးတွေအားလုံး အကြီးဖြစ်မဖြစ် စစ်တဲ့အခါ သုံးပါတယ်။

No.17 (islower()) ရဲ့ ဆန့်ကျင်ဘက်ပါ။)

### 23.join()

Joins the elements of an iterable to the end of the string

စာတစ်ခုနဲ့တစ်ခု ရောချပ်ရင်းသုံးပါတယ်။

```
1 x = "Hello World"
2 z = '/'.join(x)
3 print(z)
```

Hello World ဆိုတဲ့ စာကို / တွေထည့်ခိုင်းတဲ့ program ဝါ။

```
H/e/l/l/o/ /W/o/r/l/d
```

### 24.ljust()

Returns a left justified version of the string method no.3 center() နဲ့အတူတူပါပဲ။

ဒါပေမယ့် ljust() က ဘယ်ဘက်ကိုမူသေထားပြီးညာမှာဖြည့်ပါတယ်။

```
1 x = "Hello World"
2 y = x.ljust(len(x)+8 , '*' )
3 print(y)
```

```
Hello World*****
```

### 25.rjust()

ညာဘက်ကိုမူသေထားပြီး ဘယ်မှာဖြည့်ပါတယ်။

```
1 x = "Hello World"
2 y = x.rjust(len(x)+8 , '*' )
3 print(y)
```

```
*****Hello World
```

## 26.lstrip()

Returns a left trim version of the string

ဘယ်ဘက်က စာလုံးကိုဖျက်ချင်ရင်သုံးပါတယ်။

.lstrip(ဖျက်မည့်စာ)

```
1 x = "****Hello World****"
2 print("Original x = ", x)
3 print("After left striping of star, x = ", x.lstrip(
  ""))
```

x.lstrip("")လို့ရေးလိုက်လို့ x တန်ဖိုးထဲက ဘယ်ဘက်က star တွေကို ဖျက်ပြီးပြမှာပါ။

```
Original x = ****Hello World****
After left striping of star, x = Hello World****
```

## 27.rstrip()

ညာဘက်က စာလုံးကိုဖျက်ချင်ရင်သုံးပါတယ်။

.rstrip(ဖျက်မည့်စာ)

```
1 x = "****Hello World****"
2 print("Original x = ", x)
3 print("After right| striping of star, x = ", x.
  rstrip(""))
```

```
Original x = ****Hello World****
After right striping of star, x = ****Hello World
```

x.rstrip("") လို့ရေးတဲ့အတွက် ညာဘက်က star တွေကို ဖျက်ပြီးပြပါတယ်။

28.strip()ဘယ်ရော ညာရော အစွန်ကနေစဖျက်ပါတယ်။

x.strip("")

လို့ပြောင်းရေးရင် Hello World ပဲကျန်မှာပါ။

29.partition() Returns a tuple where the string is parted into three parts

စာကို သုံးပိုင်းခွဲချင်ရင် သုံးပါတယ်။

.partition(အလယ်ပိုင်း)

```
1 x = "AppleBananaOrange"
2 y = x.partition("Banana")
3 print(y)
```

partition နောက်က ကွင်းထဲက စာကို အလယ်ပိုင်းအဖြစ်ထားပြီး

သူ့ရဲ့ဘယ်အခြမ်းနဲ့ ညာအခြမ်းကို တစ်ပိုင်းစီအဖြစ်ခွဲပါတယ်။

အလယ်ပိုင်းအဖြစ်သတ်မှတ်တဲ့စာကို စတွေ့တာနဲ့ သုံးပိုင်းခွဲပါတယ်။

```
('Apple', 'Banana', 'Orange')
```

30.rpartition() right partition

အပေါ်ကဟာနဲ့ တူပါတယ်။ဒါပေမယ့် အလယ်ပိုင်းအဖြစ်ခွဲမယ့်စာက နှ

စ်ခုထပ်နေရင် နောက်ဆုံးတစ်ခုကို အလယ်ပိုင်းအဖြစ်ယူပါတယ်။

```
1 x = "AppleBananaBananaOrange"
2 y = x.rpartition("Banana")
3 print(y)
```

banana နှစ်ခုထပ်ပြီးထည့်ထားပါတယ်။

```
('AppleBanana', 'Banana', 'Orange')
```

### 31.replace()

Returns a string where a specified value is replaced with a specified value

စာလုံးတစ်ခုကို အသစ်တစ်ခုနဲ့ အစားထိုးပြန်ရင် သုံးပါတယ်။

.replace(စာလုံးဟောင်း စာလုံးသစ်)

```
1 x = "Hello World"
2 y = x.replace("World", "Mg Mg")
3 print(y)
```

World နေရာမှာ Mg Mg ကို အစားထိုးတာပါ။

```
Hello Mg Mg
```

### 32.rfind()

Searches the string for a specified value and returns the last position of where it was found

မိမိရှာလိုသော စာလုံးရဲ့ နောက်ဆုံးတည်နေရာကို သိချင်ရင်သုံးပါတယ်။

.rfind(ရှာလိုသောစာ)

No.8 find()နဲ့တူပါတယ်။

find() က ပထမဆုံးတွေ့တဲ့တည်နေရာကိုပြပြီး right find က နောက်ဆုံး တည်နေရာကိုပြပါတယ်။

```
1 x = "Hello World"
2 y = x.rfind('o')
3 print(y)
```

အိုဟာ တည်နေရာ 4 နဲ့ တည်နေရာ 7 မှာ ရှိပါတယ်။ အိုနှစ်ခုထဲက ညာဘက်က အိုရဲ့ တည်နေရာကိုသာ ပြမှာပါ။

```
7
```



33.rindex() Searches the string for a specified value and returns the last position of where it was found

No.8 index( ) နဲ့ အတူတူပါပဲ။သူက ညာဘက်ဆုံးက index ကိုပြမှာပါ။

34.split() Splits the string at the specified separator, and returns a list

အပိုင်းခွဲရန်သုံးသည်။lesson.17 တွင်ကြည့်ရန်။

35.rspllit()

34.split() နှင့်တူသည်။

```
1 x = "apple,orange,banana"
2 y = x.split(',')
3 z = x.rspllit(',')
4 print("After left splitting ',' : ", y)
5 print("After right splitting ',' : ", z)
```

```
After left splitting ',' : ['apple', 'orange,banana']
After right splitting ',' : ['apple,orange', 'banana']
```

split လုပ်မည့်အရေအတွက်ကို တစ်ကြိမ်ပေးထားသည့်အတွက် split() သည် ပထမဆုံးတွေ့သော comma ၏ ဘယ်ဘက်နှင့် ညာဘက်ကို နှစ်ပိုင်းပိုင်းသည်။

rspllit() သည် နောက်ဆုံးတွေ့သော comma ၏ ဘယ်နှင့်ညာကို နှစ်ခုခွဲသည်။

### 36.splitlines()

Splits the string at line breaks and returns a list

စာတွေကို တစ်ကြောင်းချင်းခွဲထုတ်ရင် သုံးပါတယ်။

.splitlines() ဟာ

.split('\n') နဲ့တူပါတယ်။

### 37.startswith()

Returns true if the string starts with the specified value

မိမိသတ်မှတ်ထားသော စာလုံးနှင့် စထားခြင်းရှိ/မရှိ စစ်ရင်သုံးပါတယ်  
အသေးစိတ်သိချင်ရင် method no.6 ကိုဖတ်ပါ။

```
1 x = "I love python."  
2 y = x.startswith('I')  
3 print(y)
```

x ရဲ့ တန်ဖိုးဟာ I နဲ့ စထားလားစစ်ပါတယ်။

```
True
```

### 38.swapcase()

Swaps cases, lower case becomes upper case and vice versa

စာလုံးအကြီးကို အသေးပြောင်း အသေးကို အကြီးပြောင်း

```
1 x = "Hello World"
2 y = x.swapcase()
3 print(y)
```

```
hELLO wORLD
```

### 39.title()

Converts the first character of each word to upper case

word တစ်ခုချင်းရဲ့ ရှေ့စာလုံးကို အကြီးပြောင်းပါတယ်။

```
1 x = "python basic course"
2 y = x.title()
3 print(y)
```

```
Python Basic Course
```

40.lower() Converts a string into lower case

41.upper() Converts a string into upper case

စာလုံးအကြီးအသေးပြောင်းတာက သင်ပြီးသားပါ။ lesson.13

#### 42.zfill()

Fills the string with a specified number of 0 values at the beginning

စာရဲ့ဘယ်ဘက်မှာ zero ဖြည့်ရာတွင်သုံး

.zfill(စာလုံးအရေအတွက်)

xဟာ စာတစ်ကြောင်းဆိုပါစို့။ဘယ်ဘက်မှာ zero ငါးခုထည့်ချင်ရင် ၆ အာက်ကလိုရေးနိုင်ပါတယ်။

x.zfill(len(x)+5)

```
1 x = "Hello World"
2 y = x.zfill( len(x)+5 )
3 print(y)
```

```
00000Hello World
```

translate နဲ့ translatemap ကို dictionary သင်ပြီးမှ  
ဆက်သင်မှာဖြစ်လို့ Python string build in method  
44 ခု ပြီးပါပြီ။