

Capitalize() -The capitalize() method returns a string where the first character is upper case, and the rest is lower case.

```
In [2]: #Upper case the first letter in this sentence
txt = "hello, and welcome to my world."

x = txt.capitalize()

print (x)
```

Hello, and welcome to my world.

Casefold()- Converts string into lower case

```
In [3]: #Make the string lower case
txt = "Hello, And Welcome To My World!"

x = txt.casefold()

print(x)
```

hello, and welcome to my world!

Center()- Returns a centered string

```
In [7]: #Print the word "banana", taking up the space of 20 characters, with "banana" in the middle
txt = " banana "
x = txt.center(20)
print(x)
```

banana

count()-Returns the number of times a specified value occurs in a string

```
In [6]: #Return the number of times the value "apple" appears in the string
txt = "I love apples, apple are my favorite fruit"

x = txt.count("apple")
```

```
print(x)
```

```
2
```

encode()-Returns an encoded version of the string

```
In [8]: txt = "My name is Ståle"
```

```
x = txt.encode()
```

```
print(x)
```

```
b'My name is St\xc3\xa5le'
```

endswith()-Returns true if the string ends with the specified value

```
In [10]: #Check if the string ends with a punctuation sign (.):
```

```
txt = "Hello, welcome to my world."
```

```
x = txt.endswith(",")
```

```
print(x)
```

```
False
```

expandtabs() Sets the tab size of the string

```
In [11]: #Set the tab size to 2 whitespaces:
```

```
txt = "H\te\tl\tl\to"
```

```
x = txt.expandtabs(2)
```

```
print(x)
```

```
H e l l o
```

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

```
In [12]: #Where in the text is the word "welcome"?:
```

```
txt = "Hello, welcome to my world."

x = txt.find("welcome")

print(x)
```

7

format() Formats specified values in a string

```
In [13]: #Insert the price inside the placeholder, the price should be in fixed point, two-decimal format
```

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

For only 49.00 dollars!

format signs

```
In [ ]: :<      Left aligns the result (within the available space)
        :>      Right aligns the result (within the available space)
        :^      Center aligns the result (within the available space)
        :=      Places the sign to the left most position
        :+      Use a plus sign to indicate if the result is positive or negative
        :-      Use a minus sign for negative values only
        :       Use a space to insert an extra space before positive numbers (and a minus sign before negative numbers)
        :,      Use a comma as a thousand separator
        :_      Use a underscore as a thousand separator
        :b      Binary format
        :c      Converts the value into the corresponding unicode character
        :d      Decimal format
        :e      Scientific format, with a lower case e
        :E      Scientific format, with an upper case E
        :f      Fix point number format
        :F      Fix point number format, in uppercase format (show inf and nan as INF and NAN)
        :g      General format
        :G      General format (using a upper case E for scientific notations)
        :o      Octal format
        :x      Hex format, lower case
        :X      Hex format, upper case
        :n      Number format
        :%      Percentage format
```

format_map() Formats specified values in a string

```
In [14]: # input stored in variable a.
a = {'x':'John', 'y':'Wick'}

# Use of format_map() function
print("{x}'s last name is {y}".format_map(a))

John's last name is Wick
```

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

```
In [17]: txt = "Hello , welcome to my world."

x = txt.index("welcome")

print(x)

8
```

isalnum() Returns True if all characters in the string are alphanumeric

```
In [18]: #Check if all the characters in the text are alphanumeric:

txt = "Company12"

x = txt.isalnum()

print(x)

True
```

isalpha() Returns True if all characters in the string are in the alphabet

```
In [19]: #Check if all the characters in the text are letters:

txt = "CompanyX"
```

```
x = txt.isalpha()
```

```
print(x)
```

True

isascii() Returns True if all characters in the string are ascii characters

In [22]: *#Check if all the characters in the text are ascii characters:*

```
txt = "Company123"
```

```
x = txt.isascii()
```

```
print(x)
```

True

isdecimal() Returns True if all characters in the string are decimals

In [24]: *#Check if all the characters in a string are decimals (0-9):*

```
txt = "12410"
```

```
x = txt.isdecimal()
```

```
print(x)
```

True

isdigit() Returns True if all characters in the string are digits

In [26]: *#Check if all the characters in the text are digits:*

```
txt = "50800"
```

```
x = txt.isdigit()
```

```
print(x)
```

True

isidentifier() Returns True if the string is an identifier

```
In [27]: txt = "Demo"

x = txt.isidentifier()

print(x)

True
```

islower() Returns True if all characters in the string are lower case

```
In [28]: #Check if all the characters in the text are in lower case:

txt = "hello world!"

x = txt.islower()

print(x)

True
```

isnumeric() Returns True if all characters in the string are numeric

```
In [30]: #Check if all the characters in the text are numeric:

txt = "565543"

x = txt.isnumeric()

print(x)

True
```

isprintable() Returns True if all characters in the string are printable

```
In [31]: #Check if all the characters in the text are printable:

txt = "Hello! Are you #1?"
```

```
x = txt.isprintable()
```

```
print(x)
```

True

isspace() Returns True if all characters in the string are whitespaces

In [32]: *#Check if all the characters in the text are whitespaces:*

```
txt = "    "
```

```
x = txt.isspace()
```

```
print(x)
```

True

istitle() Returns True if the string follows the rules of a title

In [34]: *#Check if each word start with an upper case letter:*

```
txt = "Hello, And Welcome To My World!"
```

```
x = txt.istitle()
```

```
print(x)
```

True

isupper() Returns True if all characters in the string are upper case

In [35]: *#Check if all the characters in the text are in upper case:*

```
txt = "THIS IS NOW!"
```

```
x = txt.isupper()
```

```
print(x)
```

True

join() Converts the elements of an iterable into a string

In [36]: *#Join all items in a tuple into a string, using a hash character as separator:*

```
myTuple = ("John", "Peter", "Vicky")  
  
x = "#".join(myTuple)  
  
print(x)
```

John#Peter#Vicky

ljust() Returns a left justified version of the string

In [37]: *#Return a 20 characters long, left justified version of the word "banana":*

```
txt = "banana"  
  
x = txt.ljust(20)  
  
print(x, "is my favorite fruit.")
```

banana is my favorite fruit.

lower() Converts a string into lower case

In [39]: *#Lower case the string:*

```
txt = "Hello my FRIENDS"  
  
x = txt.lower()  
  
print(x)
```

hello my friends

lstrip() Returns a left trim version of the string

In [59]: *#Remove spaces to the left of the string:*


```
txt = "        banana"

x = txt.lstrip()

print("of all fruits", x, "is my favorite")

of all fruits banana        is my favorite
```

maketrans() Returns a translation table to be used in translations

In [41]: *#Create a mapping table, and use it in the translate() method to replace any "S" characters with a "P" character:*

```
txt = "Hello Sam!"
mytable = str.maketrans("S", "P")
print(txt.translate(mytable))

Hello Pam!
```

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

In [43]: *#Search for the word "bananas", and return a tuple with three elements:*

```
#1 - everything before the "match"
#2 - the "match"
#3 - everything after the "match"

txt = "I could eat bananas all day"

x = txt.partition("bananas")

print(x)

('I could eat ', 'bananas', ' all day')
```

replace() Returns a string where a specified value is replaced with a specified value

In [44]: *#Replace the word "bananas":*

```
txt = "I like bananas"

x = txt.replace("bananas", "apples")
```

```
print(x)
```

I like apples

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

In [45]: *#Where in the text is the last occurrence of the string "casa"?:*

```
txt = "Mi casa, su casa."
```

```
x = txt.rfind("casa")
```

```
print(x)
```

12

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

In [46]: *#Where in the text is the last occurrence of the string "casa"?:*

```
txt = "Mi casa, su casa."
```

```
x = txt.rindex("casa")
```

```
print(x)
```

12

rjust() Returns a right justified version of the string

In [47]: *#Return a 20 characters long, right justified version of the word "banana":*

```
txt = "banana"
```

```
x = txt.rjust(20)
```

```
print(x, "is my favorite fruit.")
```

banana is my favorite fruit.

rpartition() Returns a tuple where the string is parted into three parts

In [48]: *#Search for the last occurrence of the word "bananas", and return a tuple with three elements:*

```
#1 - everything before the "match"  
#2 - the "match"  
#3 - everything after the "match"  
  
txt = "I could eat bananas all day, bananas are my favorite fruit"  
  
x = txt.rpartition("bananas")  
  
print(x)  
  
('I could eat bananas all day, ', 'bananas', ' are my favorite fruit')
```

rsplit() Splits the string at the specified separator, and returns a list

In [49]: *#Split a string into a list, using comma, followed by a space (,) as the separator:*

```
txt = "apple, banana, cherry"  
  
x = txt.rsplit(", ")  
  
print(x)  
  
['apple', 'banana', 'cherry']
```

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

In [50]: *#Split a string into a list where each word is a list item:*

```
txt = "welcome to the jungle"  
  
x = txt.split()  
  
print(x)  
  
['welcome', 'to', 'the', 'jungle']
```

splitlines() Splits the string at line breaks and returns a list

In [51]: *#Split a string into a list where each line is a list item:*

```
txt = "Thank you for the music\nWelcome to the jungle"
x = txt.splitlines()
print(x)

['Thank you for the music', 'Welcome to the jungle']
```

startswith() Returns true if the string starts with the specified value

In [52]: *#Check if the string starts with "Hello":*

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

True
```

strip() Returns a trimmed version of the string

In [53]: *#Remove spaces at the beginning and at the end of the string:*

```
txt = "    banana    "
x = txt.strip()
print("of all fruits", x, "is my favorite")

of all fruits banana is my favorite
```

swapcase() Swaps cases, lower case becomes upper case and vice versa

In [54]: *#Make the lower case letters upper case and the upper case letters lower case:*

```
txt = "Hello My Name Is PETER"

x = txt.swapcase()

print(x)

hELLO mY nAME iS peter
```

title() Converts the first character of each word to upper case

In [55]: *#Make the first letter in each word upper case:*

```
txt = "Welcome to my world"

x = txt.title()

print(x)

Welcome To My World
```

translate() Returns a translated string

In [56]: *#Replace any "S" characters with a "P" character:*

```
#use a dictionary with ascii codes to replace 83 (S) with 80 (P):
mydict = {83: 80}
txt = "Hello Sam!"
print(txt.translate(mydict))

Hello Pam!
```

upper() Converts a string into upper case

In [57]: *#Upper case the string:*

```
txt = "Hello my friends"

x = txt.upper()

print(x)

HELLO MY FRIENDS
```

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

In [58]: *#Fill the string with zeros until it is 10 characters long:*

```
txt = "50"  
  
x = txt.zfill(10)  
  
print(x)
```

0000000050

In []: