```
In [1]: #Creating string
        a="Hello, my name is Ram"
```

```
In [2]: print(a)

        Hello, my name is Ram
```

```
In [3]: type(a)

Out[3]: str
```

Slicing string

```
In [4]: a[4]

Out[4]: 'o'
```

```
In [5]: a[5]

Out[5]: ','
```

```
In [6]: a[6]

Out[6]: ' '
```

```
In [7]: #so space and commas also counts.
```

```
In [8]: #Range of slicing in strings
        a[0:4]

Out[8]: 'Hell'
```

```
In [9]: #Negative indexing (Similar to lists)
        a[-1]

Out[9]: 'm'
```

```
In [11]: #Range of slicing (Negative index)
         a[-5:-1]                # from s to m, but m does not get included due to indexing rules.

Out[11]: 's Ra'
```

Few String Methods: upper, lower, Remove whitespace or strip, Replace

```
In [12]: a="nirajan"
         a.upper()

Out[12]: 'NIRAJAN'
```

```
In [13]: b='SHYAM'
         b.lower()

Out[13]: 'shyam'
```

```
In [18]: c="Hello, world " #there is space at the end, remove space from begining or from end
         c.strip()

Out[18]: 'Hello, world'
```

```
In [19]: #Replace
         c.replace("H","F")

Out[19]: 'Fello, world '
```

String Concatenation

```
In [20]: a="OM"
         b="Shakti"
         print(a+b)

         OMShakti
```

```
In [21]: #if needed space?
         print(a+ " " +b)

         OM Shakti
```

String Formating

```
In [23]: #There are 4 ways: let see every example
```

In [23]:

```
#There are 4 ways: let see every example
# First: +
name=input("What is your name: ")
age=int(input("Your age: "))
print("Hello"+ name,"your age is" + str(age))
```

```
What is your name: shivam
Your age: 21
Helloshivam your age is21
```

In [24]:

```
#in above we are seeing hello and shivam together and is and 21 together because we have not put space there
#lets put space too:
name=input("What is your name: ")
age=int(input("Your age: "))
print("Hello "+ name,"your age is " + str(age))
```

```
What is your name: Hari
Your age: 22
Hello Hari your age is 22
```

In [25]:

```
#Second formating: f'
name=input("What is your name: ")
age=int(input("Your age: "))
print(f'Hello {name}, your age is {age}')
```

```
What is your name: ram
Your age: 25
Hello ram, your age is 25
```

In [29]:

```
#third formatting is : format
name=input("What is your name: ")
age=int(input("Your age: "))
print('Hello {}, your age is {}'.format(name,age))
```

```
What is your name: god
Your age: 1000
Hello god, your age is 1000
```

In [30]:

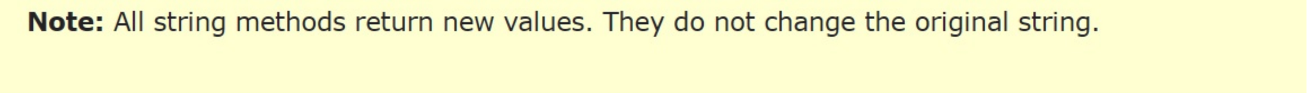
```
#%s for string and %d for int formating
name=input("What is your name: ")
age=int(input("Your age: "))
print('Hello %s, your age is %d'%(name,age))
```

```
What is your name: kkk
Your age: 45
Hello kkk, your age is 45
```

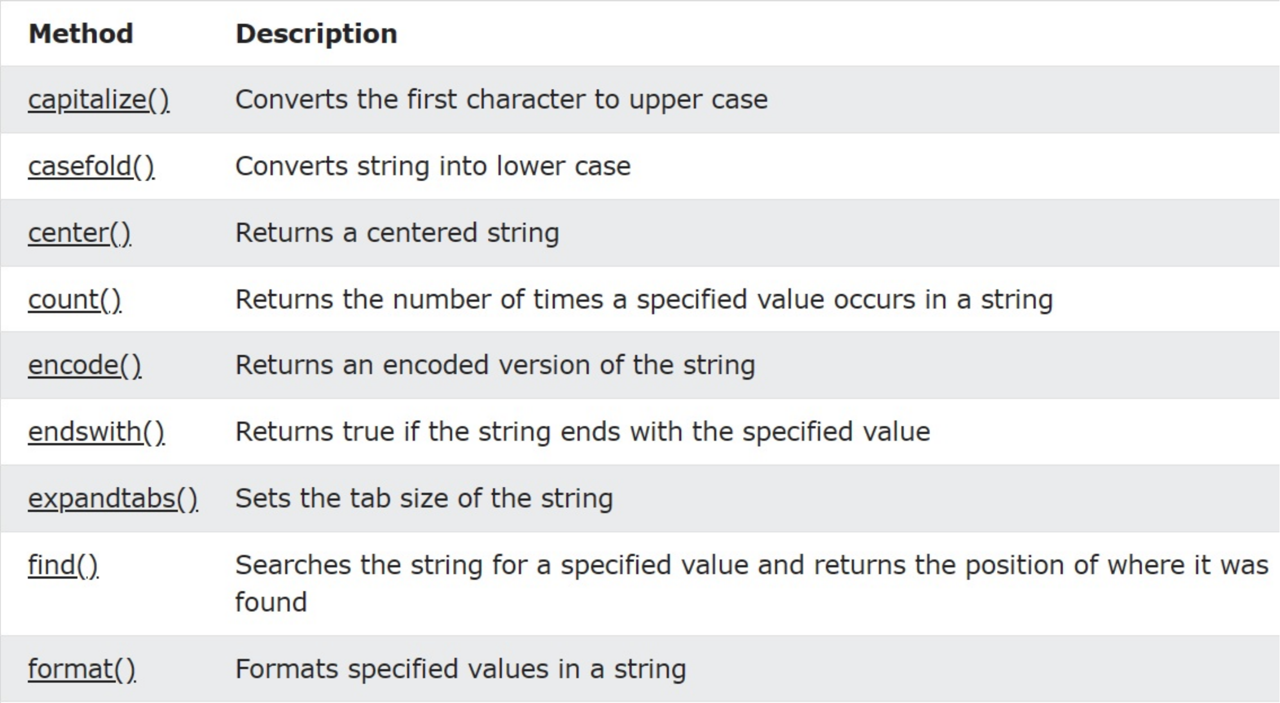
In [31]:

```
#so you can see below 3 format except + formating, other are advance formating, where we do not have to think m
#spacing and formattings.
```

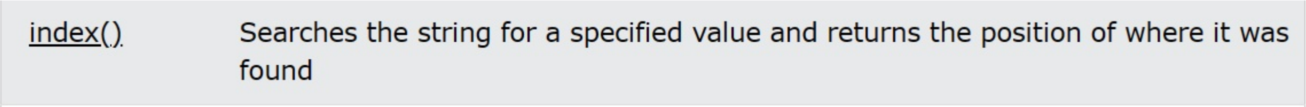
String methods



Note: All string methods return new values. They do not change the original string.



Method	Description
capitalize()	Converts the first character to upper case
casefold()	Converts string into lower case
center()	Returns a centered string
count()	Returns the number of times a specified value occurs in a string
encode()	Returns an encoded version of the string
endswith()	Returns true if the string ends with the specified value
expandtabs()	Sets the tab size of the string
find()	Searches the string for a specified value and returns the position of where it was found
format()	Formats specified values in a string



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

isalnum()	Returns True if all characters in the string are alphanumeric
isalpha()	Returns True if all characters in the string are in the alphabet
isdecimal()	Returns True if all characters in the string are decimals
isdigit()	Returns True if all characters in the string are digits
isidentifier()	Returns True if the string is an identifier
islower()	Returns True if all characters in the string are lower case
isnumeric()	Returns True if all characters in the string are numeric
isprintable()	Returns True if all characters in the string are printable
isspace()	Returns True if all characters in the string are whitespaces
istitle()	Returns True if the string follows the rules of a title
isupper()	Returns True if all characters in the string are upper case
join()	Joins the elements of an iterable to the end of the string
ljust()	Returns a left justified version of the string
lower()	Converts a string into lower case
lstrip()	Returns a left trim version of the string
maketrans()	Returns a translation table to be used in translations
partition()	Returns a tuple where the string is parted into three parts
replace()	Returns a string where a specified value is replaced with a specified value
rfind()	Searches the string for a specified value and returns the last position of where it was found
rindex()	Searches the string for a specified value and returns the last position of where it was found
rjust()	Returns a right justified version of the string
rpartition()	Returns a tuple where the string is parted into three parts
rsplit()	Splits the string at the specified separator, and returns a list

In [32]:

```
#we will try to use it in some practice exercise later on.here lets try isdigit() with if else statement
a="Ram1"
if a.isdigit():
    print("All letter is digit")
else:
    print("All are not digits")
```

```
All are not digits
```

In [33]:

```
b="1234"
if b.isdigit():
    print("All letter is digit")
else:
    print("All are not digits")
```

```
All letter is digit
```

In [ ]: