String Method

s= "Hello world ! single line string

```
s= "Hello world ! single line string"
print(s,type(s),id(s))
s=s.upper()
print(s)
#__method__
```

```
Hello world ! single line string <class 'str'> 134150706938960
HELLO WORLD ! SINGLE LINE STRING
```

Python is a dynamic languages that it can understanf integers float on its own, No need to initialise it

```
class Person(object):
  def __sir__(self):
    return "an object of class Person"
p1=Person()
print(p1)
#print(p1.__str__())
```

```
<__main__.Person object at 0x7dd8b88b3a00>
```

```
print("Hello",end="\n") # this says dont end at the next line
print("World")
```

```
Hello
World
```

```
s="Hello World"
us=s.upper()
ls=s.lower()
ss=s.swapcase()
print(f"Original : , {s}")
print(f"Upper Case : , {us}")
print(f"Lower Case : , {ls}")
print(f"Swap Case : , {ss}")
```

```
Original : , Hello World
Upper Case : , HELLO WORLD
Lower Case : , hello world
Swap Case : , hELLO wORLD
```

```
line1="Dear Sir,"
line2=" \n\t\t Sir how are you, I want to inform you that....\n"
print(line1, end="")
print(line2, end="")
```

```
Dear Sir,
                    Sir how are you, I want to inform you that....
```

```
#Right Justification
name =["Raj","Aayush Gupta", "Meet"]
print(f"Hello {name[0].rjust(20)}, Welcome to python")
print(f"Hello {name[1]:>20}, Welcome to python")
print(f"Hello {name[2]:>20}, Welcome to python")
```

```
Hello                  Raj, Welcome to python
Hello         Aayush Gupta, Welcome to python
Hello                 Meet, Welcome to python
```

```
#Left Justification
name =["Raj","Aayush Gupta", "Meet"]
print(f"Hello {name[0].ljust(20)}, Welcome to python")
print(f"Hello {name[1]:<20}, Welcome to python")
print(f"Hello {name[2]:<20}, Welcome to python")
```

```
Hello Raj                 , Welcome to python
Hello Aayush Gupta        , Welcome to python
Hello Meet                , Welcome to python
```

```
#Center Justification
name =["Raj","Aayush Gupta", "Meet"]
print(f"Hello {name[0].center(20)}, Welcome to python")
print(f"Hello {name[1]:^20}, Welcome to python")
print(f"Hello {name[2]:^20}, Welcome to python")
```

```
Hello         Raj         , Welcome to python
Hello     Aayush Gupta    , Welcome to python
Hello         Meet        , Welcome to python
```

```
# to fill the empty spaces on a string
n1="1001"
n2="01011001"
n3="10011101001"
print(n1.zfill(8))
print(n2.zfill(10))
print(n3.zfill(16))
```

```
00001001
0001011001
0000010011101001
```

```
#position
app="""To
        The headmaster
        {}
Dear sir,
        Due to {} I am not able to attedn classes for Next f{} day.
        Please grant me leave.
Yours Student
{}
"""

name=input("Enter your name")
school_name=input("Enter your school")
reason=input("ENter the reason")
days=input("Enter the number of days")
output=app.format(school_name,reason,days,name)
print(output)
```

```
Enter your nameAayush
Enter your schoolST karens high school
ENter the reasonto have sex
Enter the number of days2
To
        The headmaster
        ST karens high school
Dear sir,
        Due to to have sex I am not able to attedn classes for Next f2 day.
        Please grant me leave.
Yours Student
Aayush
```

List method

Add Method

list.append(item) -> item will be added to at the end of the list

list.insert(pos,item) -> item will be inserted at the pos, other item will be rightr shifted

list.extend(seq) -> add all item from the sequene at the end of list

```
l=["Aayush","Kumar","Gupta",]
print(l)
l.append("Krishna")
print(l)
l.insert(2,"Brahma")
print(l)
l[1]="Shiv"
print(l)
```

```
['Aayush', 'Kumar', 'Gupta']
['Aayush', 'Kumar', 'Gupta', 'Krishna']
['Aayush', 'Kumar', 'Brahma', 'Gupta', 'Krishna']
['Aayush', 'Shiv', 'Brahma', 'Gupta', 'Krishna']
```

```
###Remove
    list.pop() -> delete and return last element by default
    list.pop(index) -> delete and return element at index from list
    list.remove(item) -> Will search and delete first of occurence o
```

Remove

```
list.pop() -> delete and return last element by default
list.pop(index) -> delete and return element at index from list
list.remove(item) -> Will search and delete first of occurence of item
```