Day19_stringfunctions_modules

split : split() is used for splitting the string along into pieces by the reference of a specific character and returns a list of strings
Syntax
String.split(seperator,maxsplit)

Ex:

```
s1 = "Weekend is a long way away, always weekend is good"
s1.split(',')

#['Weekend is a long way away', ' always weekend is good']

s1 = "Weekend is a long way away, always weekend is good"
s1.split()

['Weekend',
 'is',
 'a',
 'long',
 'way',
 'away,',
 'always',
 'weekend',
 'is',
 'good']

s1.split(' ',3)

 ['Weekend', 'is', 'a', 'long way away, always weekend is good']
```

join() : Is used to joining the list of strings along with a seperator
Syntax:
Seperator.join(iterable)
Ex:

```
lst = 'Weekend is a long way away'.split()
lst
#['Weekend', 'is', 'a', 'long', 'way', 'away']

'-'.join(lst)
#'Weekend-is-a-long-way-away'
```

Working with Cases
upper() : It is used for changing the case of the string to upper case
lower(): It is used for changing the case of the string to lower case
swapcase(): Converts lower case to upper case and upper case to lowercase
title() : Converts all the words first character into upper case
casefold() : ignores the case especially used when we compare the strings

```
s2 = "Today is thursday in USA"
s2.upper()
# 'TODAY IS THURSDAY IN USA'

s2.lower()
#'today is thursday in usa'

s2.swapcase()
```

```
#'tODAY IS THURSDAY IN usa'

s2.title()
#'Today Is Thursday In Usa'


s3 = 'thursday'
s4 = "Thursday"
s3==s4
#False

s3.casefold() == s4.casefold()
#True

print(s1)
```

Note: As strings are immutable they are not changed , the above methods will work instantly

String Testing Methods

1. isalnum():Returns True if all the characters in the string are alphanumeric(A-Z,a-z,0-9), if atleast one different character we get False
   ```
   s5 = "ABCabc123"
   s5.isalnum()
   #True

   s6 = "ABCabc_123"
   s6.isalnum()
   #False
   ```
2. isalpha():Returns True if all the characters in the string are alphanumeric(A-Z,a-z) o.w False

```
"ABCabc".isalpha()
#True

s6.isalpha()
#False
```

3. isdigit()
4. islower()
5. isupper()
6. istitle()
7. isspace()

```
s7 = ' '
s7.isspace()
#True

s5.isspace()
#False
```

Formatting of the strings

format() method is used for formatting the string especially used in the output formatting

Syntax:

```
'string to be formatted with replacementfields'.format(values)
```

Without format

```
print("Sno=",sno,"Sname=",sname,"Saddress=",sadd)
#Sno= 1011 Sname= Carl Saddress= USA
```

With format

```
print("Sno={} Sname={} Saddress={}".format(sno,sname,sadd))
#Sno=1011 Sname=Carl Saddress=USA

print("Sno={}\nSname={}\nSaddress={}".format(sno,sname,sadd))
           0           1              2


print("Sno={1}\nSname={2}\nSaddress={0}".format(sno,sname,sadd))
Sno=Carl
Sname=USA
Saddress=1011

s1 = "Student No. is {num},Studentname is {name},student add {add}".format(num=sno,name=sname,add=sadd)

s1
#'Student No. is 1011,Studentname is Carl,student add USA'


eno = 1236
ename = "Monie"
esal = 52533

s1 = "Eno={:5d},name= {:10s} salary={:10.2f}".format(eno,ename,esal)
s1

#'Eno= 1236,name= Monie      salary=  52533.00'
```