```
#                                                                    "String Manipulation"

#                                                                        Definitation :-

# "String is nothing but a combination of character a multiple character"

#  ( D,a,t,a,s,c,i,e,n,c,e ) - these are nothing but character.Here are going to extract some of the character from given string(str)

# Note :- Always use square bracket to extract the data.

#   1.Forward Index -01245678 (positive indexes moving forward postitive direction)

#   2.Backward Index- -8-7-6-5-4-3-2-1 (Negative indexes moving backward direction)

v = "datascience"

v[0]
```

```
'd'
```

```
v[5]
```

```
'c'
```

```
v[-1]
```

```
'e'
```

```
v1 = "vijay"

v1[-1]
```

```
'y'
```

```
v1[-2]
```

```
'a'
```

```
#

# 1.single couch  (' ') &

# Double couch or quote ( " " )

'this is my week 2 class assignment'
```

```
'this is my week 2 class assignment'
```

```
"this is my week 2 class assignment"
```

```
'this is my week 2 class assignment'
```

```
v2 = "this is my week 2 class assignment"

v2
```

```
'this is my week 2 class assignment'
```

```
v2[8]
```

```
'm'
```

```
#  " upper Bound " - excluding upper bound- start from 1 go till 5 but excluding upper bound so 5 willnot print as a outcome.

v2[1:5]
```

```
'his '
```

```
v2[6:30]
```

```
's my week 2 class assign'
```

```
# Here start from 6 go till 30 but but alternative one I am looking over here. I am looking for after 'S' is 1step ahead.

# Start from 6 go till 30 & try to take a jump of 2

v2[6:30:2]
```

```
'sm ek2casasg'
```

```
v = "vijaygurung"
```

```
v[0:10:2]
```

```
'vjyuu'
```

```
# In this code there is one difference, As a outcome we will get entire data complete dataset

# Here jump size is equal to 1. By default "0" zero can't be jump size because we have to move.

v[0:11:1]
```

```
'vijaygurung'
```

```
# Here we have written start from "0" zero & go till 10th positive direction & try to take a jump of -1 towards negative
# direction, because of these it create a contradiction over here.Beacuse of contradiction I am not geeting any dataset.So that
# is the region we are geeting an as outcome blank nothing.

# As a outcome we will receive blank, string so it has forward direction as well as backward direction.

v[0:10:-1]
```

```
''
```

```
v
```

```
'vijaygurung'
```

```
v[10:0:-1]
```

```
'gnurugyaji'
```

```
v[10:0:-2]
```

```
'guuyj'
```

```
# Here is I have not given starting point.only Given end point.I have also not given jump size.But still I am geeting dataset.

# Because in these scenario my jump size will be (+1). Here from "v" upto "r" only it will print 0 upto -4.

v[:-3]
```

```
'vijaygur'
```

```
v[-2:]
```

```
'ng'
```

```
v[::1]
```

```
'vijaygurung'
```

```
v[::-1]
```

```
'gnurugyajiv'
```

```
v[0:50:1]
```

```
'vijaygurung'
```

```
v2
```

```
'this is my week 2 class assignment'
```

```
v2[::-1]
```

```
'tnemngissa ssalc 2 keew ym si siht'
```

```
v3 = "metascifor"
```

```
v3[-2:-7:1]
```

```
''
```

```
v3[-7:-2:1]
```

```
'ascif'
```

```
v
```

```
'vijaygurung'
```

```
v[5:0]
```

```
''
```

```
v3[-7:0:1]
```

```
''
```

```
v
```

```
'vijaygurung'
```

```
v[:-1:-1]
```

```
''
```

```
v+1
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-67-a8113876cc72> in <cell line: 1>()
----> 1 v+1

TypeError: can only concatenate str (not "int") to str
```

```
v+"1"
```

```
'vijaygurung1'
```

```
v + str(1)
```

```
'vijaygurung1'
```

```
"vijay" + 4567894258
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-70-c8b55f1b1134> in <cell line: 1>()
----> 1 "vijay" + 4567894258

TypeError: can only concatenate str (not "int") to str
```

```
v
```

```
'vijaygurung'
```

```
#Length, Len & Function
```

```
len(v)
```

```
11
```

```
# Multiplication Operation
```

```
v*2
```

```
'vijaygurungvijaygurung'
```

```
#Function
```

```
v
```

```
'vijaygurung'
```

```
#                                                                    [ "Function" inside String ]

# We have total 43 Functions are available

# 1.Capitalize

# 2.Casefold

# 3.Center

# 4.count

# 5.Encode

# 6.endswitch

# 7.Expantabs

# 8.Find

# 9.Format

# 10.Format_Map

# 11.Index

# 12.isalnum

# 13.isalpha

# 14.isascii

# 15.isdecimal

# 16.isdigit

# 17.isidentifier

# 18.isprintable

# 19.isspace

# 20.istitle

# 21.isupper

# 22.join

# 23.Ljust

# 24.Lower

# 25.Lstrip

# 26.Maketrans

# 27.Partition

# 28.Replace

# 29.Rfind

# 30.Rindex

# 31.Rjust

# 32.Rpartition

# 33.Rsplit

# 34.RStrip

# 35.Split

# 36.SplitLines

# 37.StartSwith
```

```
# 38.Strip

# 39.Swapcase

# 40.Title

# 41.Translate

# 42.Upper

# 43.Zfill
```

```
v
```

```
'vijaygurung'
```

```
#                                                                    4."Count"

#  ( If i will call "g" it will check inside string how many times "g" words is there inside string)
```

```
v.count("g")
```

```
2
```

```
v.count("vi")
```

```
1
```

```
v.count("vir")
```

```
0
```

```
#                                                                        35."Split"

# ( It will going to give an "List variable". So whatever will come after "g" & inbetween "g" & before "g"
#                     it will going to give that as a seperate data spliting that data from the string(str))
```

```
v.split("g")
type(v.split("g"))
```

```
list
```

```
v.split("g")
```

```
['vijay', 'urun', '']
```

Double-click (or enter) to edit

```
 #                                                                  "Single Couch" (' ')
```

```
 sw = 'Samsung Group,[3] or simply Samsung (Korean: 삼성; RR: samseong [samsʌŋ]) (stylized as SΛMSUNG), is a South Korean multinational manu

Samsung was founded by Lee Byung-chul in 1938 as a trading company. Over the next three decades, the group diversified into areas including

Notable Samsung industrial affiliates include Samsung Electronics (the world's largest information technology company, consumer electronics
```

```
  File "<ipython-input-86-5517dcf8dee3>", line 4
    sw = 'Samsung Group,[3] or simply Samsung (Korean: 삼성; RR: samseong [samsʌŋ])
(stylized as SΛMSUNG), is a South Korean multinational manufacturing conglomerate
headquartered in Samsung Town, Seoul, South Korea.[1] It comprises numerous affiliated
businesses,[1] most of them united under the Samsung brand, and is the largest South
Korean chaebol (business conglomerate). As of 2020, Samsung has the eighth highest
global brand value.[4]
         ^
```

```
#Double Couch (" ")

sw = "Samsung Group,[3] or simply Samsung (Korean: 삼성; RR: samseong [samsʌŋ]) (stylized as SΛMSUNG), is a South Korean multinational manuf

Samsung was founded by Lee Byung-chul in 1938 as a trading company. Over the next three decades, the group diversified into areas including

Notable Samsung industrial affiliates include Samsung Electronics (the world's largest information technology company, consumer electronics
```

```
  File "<ipython-input-87-f80eae889661>", line 3
    sw = "Samsung Group,[3] or simply Samsung (Korean: 삼성; RR: samseong [samsʌŋ])
(stylized as SΛMSUNG), is a South Korean multinational manufacturing conglomerate
headquartered in Samsung Town, Seoul, South Korea.[1] It comprises numerous affiliated
businesses,[1] most of them united under the Samsung brand, and is the largest South
Korean chaebol (business conglomerate). As of 2020, Samsung has the eighth highest
global brand value.[4]
         ^
```

```
sw = ("Samsung Group,[3] or simply Samsung (Korean: 삼성; RR: samseong [samsʌŋ]) (stylized as SΛMSUNG), is a South Korean multinational manu


sw.split(' ')
```

```
['Samsung',
 'Group,[3]',
 'or',
 'simply',
 'Samsung',
 '(Korean:',
 '삼성;',
 'RR:',
 'samseong',
 '[samsʌŋ])',
 '(stylized',
 'as',
 'SΛMSUNG),',
 'is',
 'a',
 'South',
 'Korean',
 'multinational',
 'manufacturing',
 'conglomerate',
 'headquartered',
 'in',
 'Samsung',
 'Town,',
 'Seoul,',
 'South',
 'Korea']
```

```
#                                                                                    42."UPPER CASES"

# ( Upper case what it will do it will convert all sentences into a "Capital Letter" )



sw
```

```
'Samsung Group,[3] or simply Samsung (Korean: 삼성; RR: samseong [samsʌŋ]) (stylized as
SΛMSUNG), is a South Korean multinational manufacturing conglomerate headquartered in S
amsung Town, Seoul, South Korea'
```

```
sw.upper()
```

```
'SAMSUNG GROUP,[3] OR SIMPLY SAMSUNG (KOREAN: 삼성; RR: SAMSEONG [SAMSΛŊ]) (STYLIZED AS
SΛMSUNG), IS A SOUTH KOREAN MULTINATIONAL MANUFACTURING CONGLOMERATE HEADQUARTERED IN S
AMSUNG TOWN, SEOUL, SOUTH KOREA'
```

```
# Upper Cases Reassignment
# If i have to write my entire string (str) that case I would like to store this entire converted string in that case
#we can do a reassignment. Because here I am going to write sw = equal is used for the reassignment operation.
#If I have to assign my orginal variable with a new variable which is a upper case that case I would like to call.

                                                    # sw = sw.upper()


sw = sw.upper()


sw
```

```
'SAMSUNG GROUP,[3] OR SIMPLY SAMSUNG (KOREAN: 삼성; RR: SAMSEONG [SAMSʌŋ]) (STYLIZED AS
SΛMSUNG), IS A SOUTH KOREAN MULTINATIONAL MANUFACTURING CONGLOMERATE HEADQUARTERED IN S
ΔMSUNG TOWN, SEOUL, SOUTH KOREA'
```

```
#                                                                    " 24.Lower Cases "

# If I would like to convert everything into a lower case or Lower sentences there is a function call "Lower"

sw.lower()
```

```
'samsung group,[3] or simply samsung (korean: 삼성; rr: samseong [samsʌŋ]) (stylized as
sλmsung), is a south korean multinational manufacturing conglomerate headquartered in s
amsung town, seoul, south korea'
```

```
#                                                                      40." Title "

#If we use "Title Function" then our 1st letter of sentences will be in capital letter or upper case

#Example: "vijay" then if we call "Title Function" then my very 1st letter "V" will be Capital letter & rest letter will be smal

#So its convert into # "Vijay"


v = "vijay"


v.title()
```

```
'Vijay'
```

```
 #                                                                     " 1.Capitalize "

# Capital Function - Capital function if we call then we will get same output as it is in "Title function".

# whenever we call "Capitalize Function" my very 1st letter is going to be a upper case or Capital letter.

# Note :- "TITLE & CAPITALIZE FUNCTION" Gives us same outcome as a very 1st letter in Capital letter.


v =  "vijay"


v.capitalize()
```

```
'Vijay'
```

```
 #                                                                         " 39.Swapcase "

#Swapcase FUnction :- Upper cases or Capital Letters converted into a Lower Cases or Small Letter character / &
# Lower Cases or Small Letter Convert into Upper Cases/Capital Letter.


v =  "Vijay guRunG"


v.swapcase()
```

```
'vIJAY GUrUNg'
```

```
v = "Vijay guRuNG"


reversed(v)
```

```
<reversed at 0x7e2c6173ed10>
```

```
' '.join(reversed(v))
```

```
'G N u R u g   y a j i V'
```

```
                                                                    # " 38.Strip "

# Strip Functon :- Here in these code, we can see inside string code a space before "vijay" string & after space of "Vijay".

# If you want to remove that space between "Vijay" after & before space that case we will use "Strip Function".

# Strip always try to strip of space which is available before & after string.

# "Strip Function" - IF we have space on left hand side as well as right hand side. If we want to remove both then, I will
#Just call Strip()


v =  "   vijay  "


v.strip()
```

```
'vijay'
```

```
v = "   Vi  jay  "


v.strip()
```

```
'Vi  jay'
```

```
 #                                                                  " 25.Lstrip & 34.RStrip "

# 1.LStrip Function :-
#If i a going to call "Lstrip" Function. we can that from left hand side whatever space we have. It will going to strip that
#part. But the right Hand Side space it will be not remove it remain same as it is.
#"LStrip" only remove space of left side not right side.

# 2.Rstrip Fuction :-

# If I am going to call a "Rstrip" then it will going to remove space from right hand side.It will not going to remove
#left Hand side , it will only remove right hand side.


v
```

```
'   Vi  jay  '
```

```
v.lstrip()
```

```
'Vi  jay  '
```

```
v.rstrip()
```

```
'   Vi  jay'
```

```
#                                                                  " Join Function "


# If we have to join something, so with the help of [ space ("  ").join ] & I can try to call"vijaygurung" string.
# Here a outcome we can see that joined that "vijaygurung" & it joined this particular space. ("  ").With each &
#ever Character.

# Lets suppose I would like to put some character over here instead of space.This time we put letter "a"
# As a outcome we can see over here is that, after every character it has joined letter "a"


v = "Vijaygurung"


"  ".join("vijaygurung")
```

```
'v  i  j  a  y  g  u  r  u  n  g'
```

```
v =  "vijaygurung"


"a".join("vijaygurung")
```

```
'vaiajaaayagauarauanag'
```

```
#                                                                    " 3.Center Function ""


# If i a having a string v = "vijay" If am going call "center Function".It take 2 parameter.

# 1. Width
# 2. Character

#1. Width (z) :-
# If I am going to give 20. 20 is nothing but reserved space.
#So it will going to reserved 20 space & then I can try to call basically reversed all of these with "Z"

# It will going to reserved 20 number of space. In center it will try to put particular string "Vijay".

#2. Charater (# ) :-

# Here also whatever data, I am going to place it is going to fill all these space with that particular character.


v =  "vijay"


v.center(20,"z")
```

```
'zzzzzzzvijayzzzzzzzz'
```

```
v = "vijay"


v.center(20, "#")
```

```
'#######vijay########'
```

```
#                                                                 " Is - Function "

#With respect to string(str) there is many function which is available in the string (Str)

#Here whatever function available with "IS" they are going to return  "True or False".
#So, It is going to check it is available in Upper case or Lower case Alphabetic, numeric all shot of things it will going
# to check.

# "IS" is the function which is going to start with "IS" bydefault it is going to return whether "True or False".

# We can say a boolean Value at any point of the time.


#                                                              IS - Function

# 1.IsAlnum

# 2.IsAlpha

# 3.IsAscii

# 4.IsDecimal

# 5.IsDigit

# 6.IsIdentifier

# 7.IsLower

# 8.IsNumeric

# 9.IsPrintable

# 10.IsSpace

# 11.IsTitle

# 12.IsUpper


# This is my "String"      -

v = "Vijaygurung"
```

```
v.isupper()
```

```
False
```

```
v1 = "GURUNG"
```

```
v1.isupper()
```

```
True
```

```
#                                                                    1.IsAlnum Function | 2.IsAlpha

# 1.IsAlnum / "AlphaNumeric" Function  & 2. IsAlpha - "Alpha"

# If I am going to call Alnum or AlphaNumeric means combination of both "Alpha & Numeric".

# "Alpha Numeric" always check for either "Alpha or Numeric".It always apply "or Condition".

# In these code " Vijaygurung" we can see that "Alpha" is available but numeric is not available, but anyone is available
# then  outcome will be "True"

#   V7 = "345345sgfdg" Here we have "345345sgfdg" string we have taken, If I am going to check V7.Isalpha()

# My outcome will be "False" Because, we are just checking for "Alpha" "V7.Isalpha" but in our string we have both

# Appha as well as Numeric character.That case my outcome will be "False"
```

```
v =  "Vijaygurung"
```

```
v.isalnum()
```

```
True
```

```
v = "Vijaygurung"
```

```
v.isalpha()
```

```
True
```

```
v4 = "345345sgfdg"
```

```
v4.isalpha()
```

```
False
```

```
#                                                         "IS Function"  ( "3.IsAscii ")

#ISAscii -  ( Return "True" if Characters in the string are "ASCII", False Otherwise)
#ASCII -  characters have code points in the range U+ 0000-U+007F.

#Empty string is ASCII too.

# Whatever we are trying to type its a "Ascii" for better undersatand Google it.

#Every single string or character which is available which we can see in our keyboard,that will be going to be a "ASCII".
```

#5.IsDigit

```
# ISDigit -  Return "True" if the string is a digit, False otherwise.

# If my string is "Vijaygurung"  if I am calling "Isdigit" my outcome will be "False".

#Because, it is not available in digit form. So, That is why my outcome will be the "False"

#If I am giving "integer" in the digit "3456" form instead of "string",then my outcome will be "True".

# If we have both "Numeric as well as digit "pqr3456". My outcome will be 'False', Because it is not on the complete digit
#not a complete String.

#So, "Digit" always going to check for the digit one.
```

```
v = "Vijaygurung"
```

```
v.isdigit()
```

```
False
```

```
v5 = "3456"
```

```
v5.isdigit()
```

```
True
```

```
v6 =  "Pqr3456"
```

```
v6.isdigit()
```

```
False
```

```
 #                                                                    " 6.endswitch Function "

# Here I am trying to check "U". Here I am trying to check "endwith" with "U".

#If we execute it we get "False" because my string Str "Vijaygurung" endswith letter "g". But I am saying endswith "u".
#That is the region geeting an Error.

# same case this time if I say endswith with "g". That is "True" so I am geeting "True" outcome.
```

```
v = "vijaygurung"
```

```
v.endswith("u")
```

```
False
```

```
v.endswith("g")
```

```
True
```

```
#                                                "IS Function"  - ( "5.IsDigit" ) |  ( "8.IsNumeric")

#ISDigit & IsNumeric Difference between.

#IsDigit :- Return "True" if the string is a digit string, False otherwise.

# -> A string is a digit string if all character in the string are digits & there is at least one digits & there is atleast
# one character in the string.

#IsDigit :- Retrn "True" if the string is a Numeric string, False Otherwise.

#-> A string is Numeric if all character in the string are Numeric & there is atleast one character in the string.
```

```
v = "vijaygurung"
```

```
v.isdigit()
```

```
False
```

```
v = "453443"
```