In [ ]:

```
string1="Python and Data Science"

String is collection of characters enclosed in ' ' , " " , ''' ''' , """ """

# positive indexing  >> starts with 0 from left to right
                        max index = len-1

# negative indexing >>  starts with -1 from right to left
                        max index = -len

```

In [4]:

```
string1="Python and Data Science"
print(string1[3])
print(string1[7])
print(string1[-3])
print(string1[-10])
```

```
h
a
n
t
```

In [5]:

```
len(string1)
```

Out[5]:

```
23
```

In [6]:

```
string1[23]
```

```
---------------------------------------------------------------------------
--
IndexError                                Traceback (most recent call las
t)
Cell In[6], line 1
----> 1 string1[23]

IndexError: string index out of range
```

In [7]:

```
string1[22]
```

Out[7]:

```
'e'
```

In [8]:

```
1 string1[-23]
```

Out[8]:

```
'P'
```

In [9]:

```
1 string1[-24]
```

```
---------------------------------------------------------------------------
--
IndexError                                Traceback (most recent call las
t)
Cell In[9], line 1
----> 1 string1[-24]

IndexError: string index out of range
```

String Slicing

```
1 1. Slicing with positive Indexing
2 2. Slicing with negative Indexing
```

1. String Slicing with positive Indexing

In [ ]:

```
1 Slicing means we will cut the string in diff parts
2 string[start_index : end_index]
3
4 start_index >> default 0   >> included
5 end_index   >> default len of string   >> excluded
```

In [11]:

```
1 s1="Python Training"
2 s1[2:5]   #  2 : 4    >> tho
3
```

Out[11]:

```
'tho'
```

In [13]:

```
1 s1[7:12]  # 7 : 11
```

Out[13]:

```
'Train'
```

In [14]:

```
1 len(s1)
```

Out[14]:

```
15
```

In [15]:

```
1 s1[4:16] # 4 : 15
```

Out[15]:

```
'on Training'
```

In [16]:

```
1 s1[14]
```

Out[16]:

```
'g'
```

In [17]:

```
1 s1[15]
```

```
---------------------------------------------------------------------------
--
IndexError                                Traceback (most recent call las
t)
Cell In[17], line 1
----> 1 s1[15]

IndexError: string index out of range
```

In [18]:

```
1 s1[4:100] # 4 : 99
```

Out[18]:

```
'on Training'
```

In [19]:

```
s2="Python"
s2[:3]  # 0:2 >> Pyt
```

Out[19]:

```
'Pyt'
```

In [20]:

```
s2="Python"
s2[2:]  # 2:6(excluded) >> thon
```

Out[20]:

```
'thon'
```

In [23]:

```
s3="Dolly - Python Developer"
len(s3)
print(s3[0:14])
print(s3[:14]) # same as s3[0:14]
```

```
Dolly - Python
Dolly - Python
```

In [24]:

```
s3="Dolly - Python Developer"
len(s3)
print(s3[10:24])
print(s3[10:]) # same as s3[10:24]
```

```
thon Developer
thon Developer
```

2. String Slicing with negative Indexing

In [39]:

```
s4="Python"
#   P      y      t     h     o     n
#  -6     -5     -4    -3    -2    -1  Negative index
#   0      1      2     3     4     5  Positive Index

print("Slicing with positive index")
print(s4[2:5])   # tho  actual 2:4
print(s4[:5])    # 0:5 >>
print(s4[2:])    # 2:end of string

print("*"*50)
print("Slicing with negative index")
s4="Python"
#   P      y      t     h     o     n
#  -6     -5     -4    -3    -2    -1  Negative index
#   0      1      2     3     4     5  Positive Index

print(s4[-4 : -2])    # th   actual -4 to -3
print(s4[-5 :-1])     # ytho  actual -5  to -2
print(s4[:-4])        #  Py     actual  -6 to -5
print(s4[-5:])        #  ython     actual  -5 to end of the string

```

```
Slicing with positive index
tho
Pytho
thon
**************************************************
Slicing with negative index
th
ytho
Py
ython
```

String Slicing with Step Size

1. Step size slicing with Positive index

In [ ]:

```
#string[start_ index : end_index  : step_size]   >> Default Default 1
s4="Python and Machine Learning"
print(s4[2:12])   # 2 to 11
print(s4[2:14:1])   # 2 to 13 >> positive   step_size-1  1 -1 = 0
print(s4[2:14:2])  # to n M     >> 2-1 = 1


```

In [50]:

```
str1="Python"
print(str1[1:4])
print(str1[1:4:])  # defualt step size 1  (1-1 = 0 )
print(str1[1:4:1])
print(str1[1:4:2])  # step size 2  ( 2 -1 =1 )
```

```
yth
yth
yth
yh
```

In [54]:

```
str1="Python & Data Science"
print(str1[4:15])
print(str1[4:15:])  # defualt step size 1  (1-1 = 0 )
print(str1[4:15:1])
print(str1[4:15:2])  # step size 2  ( 2 -1 =1 )
```

```
on & Data S
on & Data S
on & Data S
o  aaS
```

In [59]:

```
str1="Python and Data Science"

print(str1[:])  # string[start_ind : end_ind]
print(str1[::])  # string[start_ind : end_ind : step_size ]
print(str1[::2])
print(str1[::3])  # 3-1= 2 characters neglect
```

```
Python and Data Science
Python and Data Science
Pto n aaSine
Ph da ic
```

2. Step size slicing with Negative index

In [64]:

```
str1="Python and Data Science"
print(str1, len(str1))
print(str1[-20:-6])
print(str1[-20:-6:1])
print(str1[-20:-6:2])
```

```
Python and Data Science 23
hon and Data S
hon and Data S
hnadDt
```

Negative Step Size

In [69]:

```
string="Python and Data Science"
print(string[:])
print(string[::-1])  # Step Size negative  (-1  +1  = 0) # reverse string
print(string[::-2])  # -2+1 = -1 >> reverse  string
```

```
Python and Data Science
ecneicS ataD dna nohtyP
eniSaa n otP
```

In [72]:

```
print(string[10:0:-1])
```

```
 dna nohty
```

In [86]:

```
# string="Python and Data Science"
# s1=str(reversed(string))
# s1
# # list1=[6,7,2,5,6,9]
# # print(list1)
# # list(reversed(list1))

```

Out[86]:

```
'<reversed object at 0x000001E765C8B250>'
```

In [88]:

```
str2="Machine Learning"
str2[: : -1]
```

Out[88]:

```
'gninraeL enihcaM'
```

In [ ]:

```

```