Data Science & AI
&
NIC - Param

Python-For Data Science

Language Fundamentals

Lecture No.- 02

By- Pankaj Sharma Sir

Recap of Previous Lecture

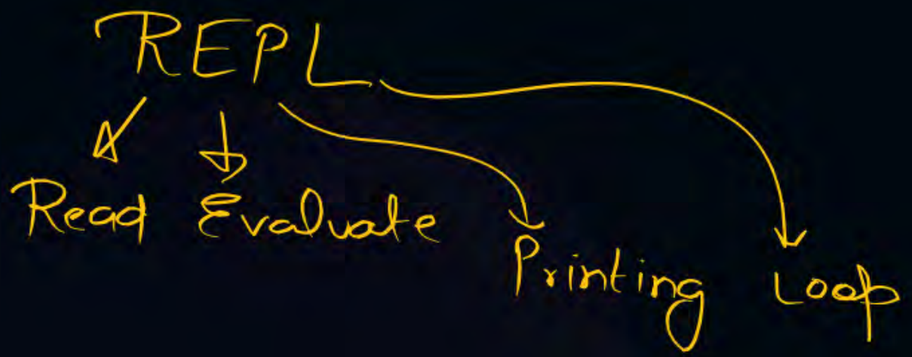
Topic Language Fundamentals - 01

Topics to be Covered

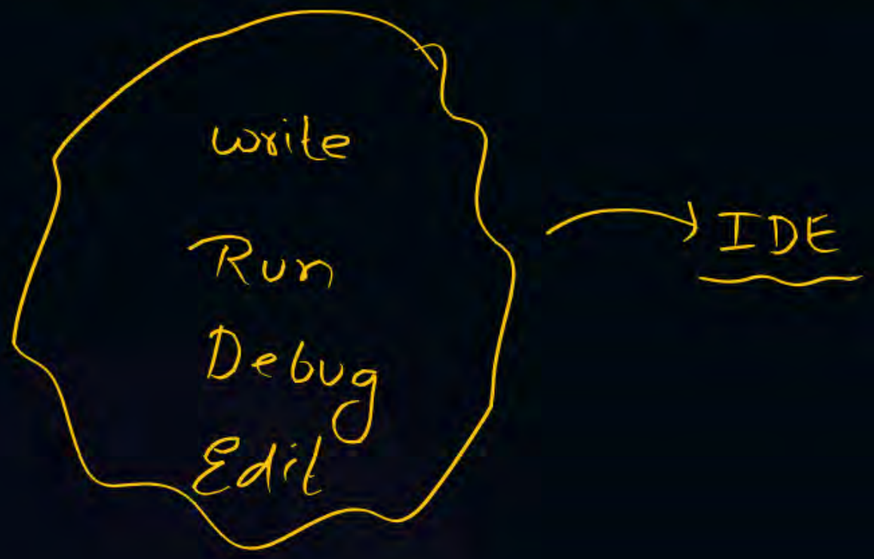
Topic Language Fundamentals - 02

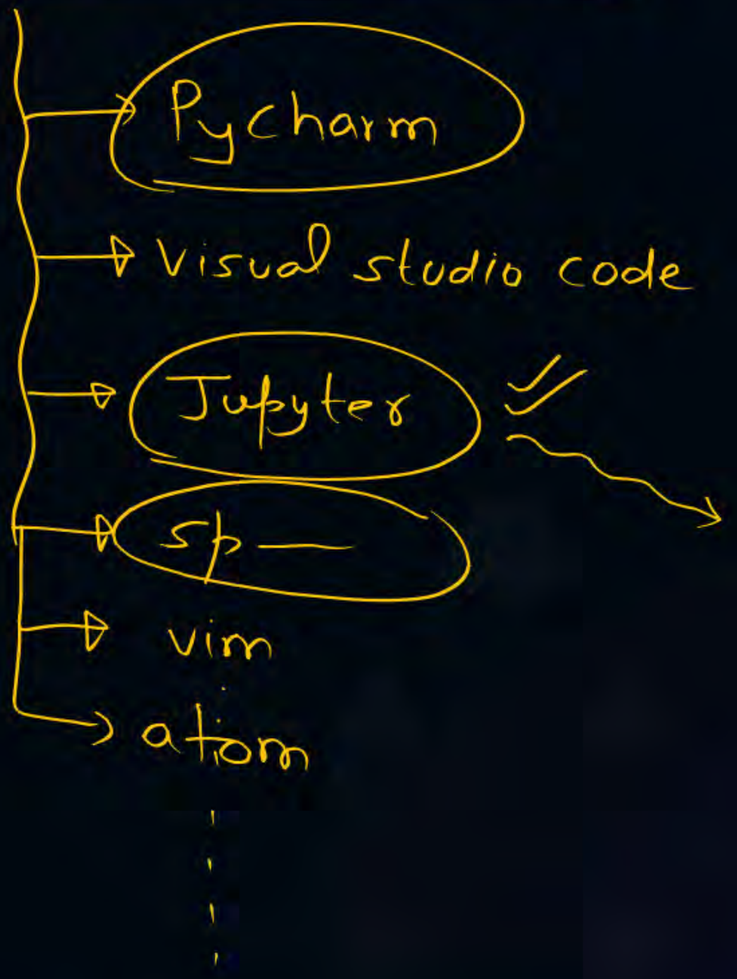
Topic : Installation of Python

Python.org



Topic : Integrated Development Environment (IDE)






Text Editor
Text

Topic : Anaconda Navigator

Anaconda

Topic : Jupyter Notebook

data science

Heading

ch-1

* IDE
* Jupyter
* Anaconda

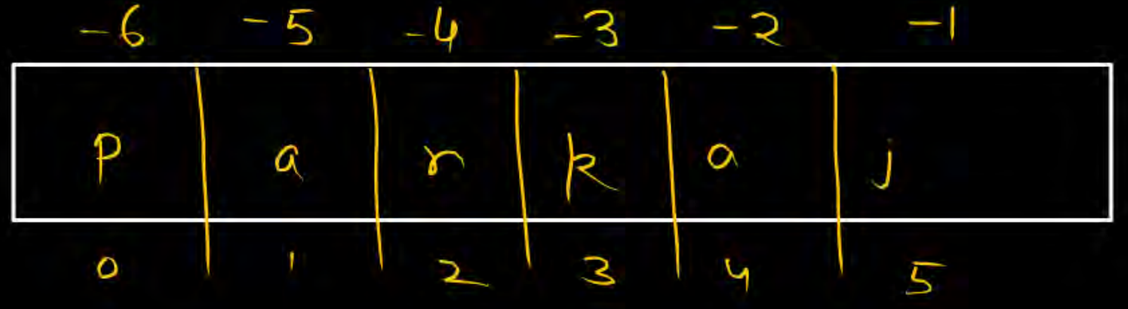
VA-2

Essay


Company
Code
Single line comment

String

s = "Pankaj"



index → +ve
→ -ve

s[0] → P
s[-6] → P

s[1] → a
s[-5] → a

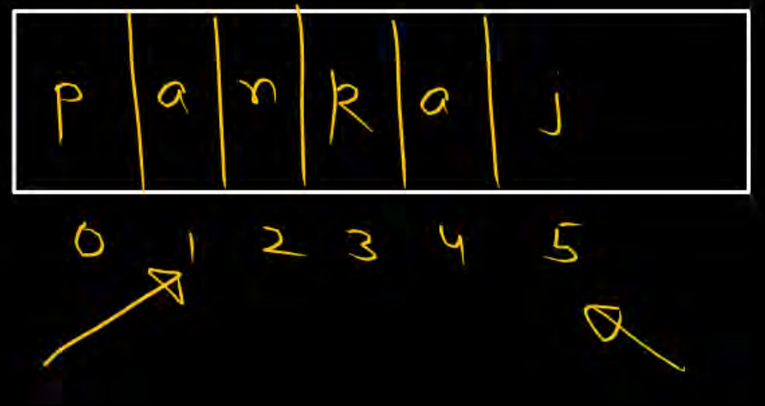
Slicing [ ]

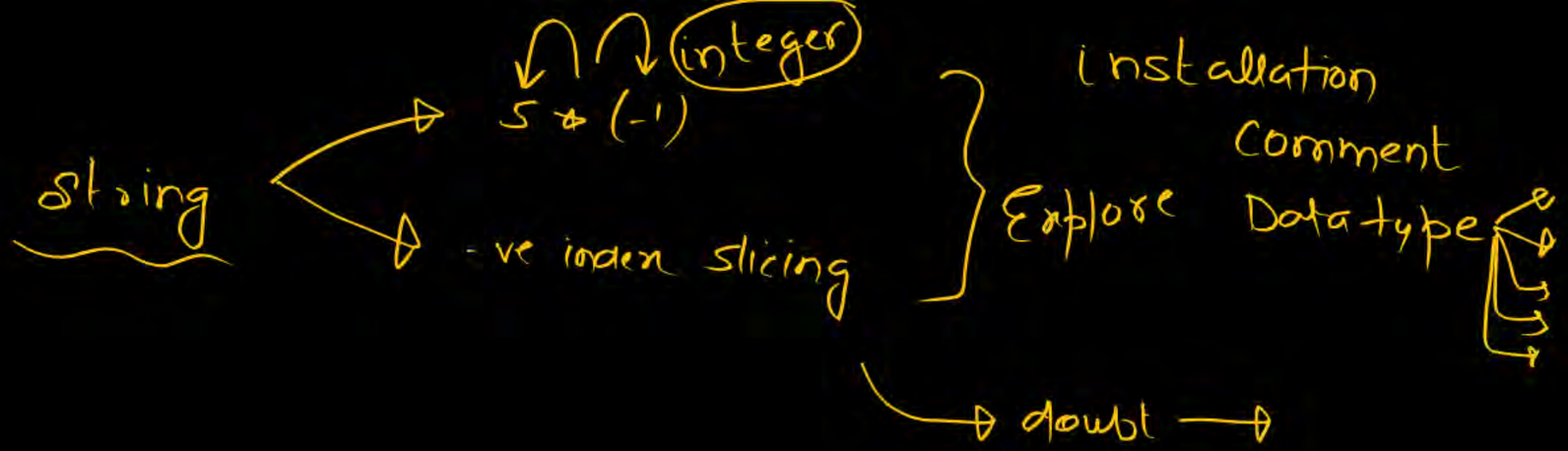
a piece

nic ⇔ oops (——)

```
s = 'Pankaj'

s[1:5]
```





t.me/PWbankajsirP

bytes
bytearray
] x
list
set
frozenset
dictionary
tuple
None
range

s = "Python"

s[-1:2]

Python → Data structure

OOPS

Day 2 : installation and Jupyter

Second class of python

In [3]:
```
print("This class is very good but the faculty is very bad")
```

```
This class is very good but the faculty is very bad
```

In [4]:
```
#this is a comment
print("Comment krna sikhlo beta")
```

```
Comment krna sikhlo beta
```

In [5]:
```
print("Comment krne ka ek aur tarika")#this is also a comment
```

```
Comment krne ka ek aur tarika
```

In [6]:
```
#this is comment line 1
#this is comment line 2
print("Ye rawan faculty hai")
```

```
Ye rawan faculty hai
```

In [7]:
```
'''This is multi line comment
aap yaha kuch bhi likh skte hain
but meaningful likhe'''
print("Twinkle twinkle little star")
```

```
Twinkle twinkle little star
```

In [8]:
```
"""Ye bhi multiline comment
ka tarika hai
mast raho aur sikho"""
print("Data science walo")
```

```
Data science walo
```

In [9]:
```
s="pankaj"
print(s[0])
```

```
p
```

In [10]:
```
print(s[-6])
```

```
p
```

In [11]:
```
print(s[1])
print(s[-5])
```

```
a
a
```

In [12]:
```
s[6]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[12], line 1
----> 1 s[6]

IndexError: string index out of range
```

In [13]:
```
s
```

Out[13]:
```
'pankaj'
```

In [14]:
```
s[1:5]
```

Out[14]:
```
'anka'
```

In [15]:
```
# start from index 1 go till index 5-1 (1,2,3,4)
s[1:100]
```

Out[15]:
```
'ankaj'
```

In [16]:
```
s[1:]
```

Out[16]:
```
'ankaj'
```

In [17]:
```
s[:4]  # 0-->3 0,1,2,3
```

Out[17]:
```
'pank'
```

In [18]:
```
s[:]
```

Out[18]:
```
'pankaj'
```

In [19]:
```
s*2
```

Out[19]:
```
'pankajpankaj'
```

In [20]:
```
s*3
```

Out[20]:
```
'pankajpankajpankaj'
```

In [21]:
```
s*5
```

Out[21]:
```
'pankajpankajpankajpankajpankaj'
```

In [22]:
```
len(s)
```

Out[22]:
```
6
```

In [23]:
```
#int int() this function is used to convert values from other type to int type
```

In [24]:
```
int(12.45)
```

Out[24]:
```
12
```

In [25]:
```
int(12 + 3j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[25], line 1
----> 1 int(12 + 3j)

TypeError: int() argument must be a string, a bytes-like object or a real number, not
'complex'
```

In [26]:
```
int(True)
```

Out[26]:
```
1
```

In [27]:
```
int(False)
```

Out[27]:
```
0
```

In [28]:
```
int("123")
```

Out[28]:
```
123
```

In [29]:
```
int("12.3")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[29], line 1
----> 1 int("12.3")

ValueError: invalid literal for int() with base 10: '12.3'
```

In [30]:
```
#float() : to convert any type to float type
```

In [31]:
```
float(12)
```

Out[31]:
```
12.0
```

In [32]:
```
float(1+2j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[32], line 1
----> 1 float(1+2j)

TypeError: float() argument must be a string or a real number, not 'complex'
```

In [33]:
```
float(True)
```

Out[33]:
```
1.0
```

In [34]:
```
float("123")
```

Out[34]:
```
123.0
```

In [35]:
```
float("123.4")
```

Out[35]:
```
123.4
```

In [36]:
```
float("one")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[36], line 1
----> 1 float("one")

ValueError: could not convert string to float: 'one'
```

In [38]:
```
#we can converet any type to float except complex
```

In [39]:
```
#complex() : other type to complex type
```

In [40]:
```
complex(2)
```

Out[40]:
```
(2+0j)
```

In [41]:
```
complex(12.3)
```

Out[41]:
```
(12.3+0j)
```

In [42]:
```
complex(True)
```

Out[42]:
```
(1+0j)
```

In [43]:
```
complex("12")
```

Out[43]:
```
(12+0j)
```

In [44]:
```
complex("12.4")
```

Out[44]:
```
(12.4+0j)
```

In [45]:
```
complex("one")#ud ke laat pdegi
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[45], line 1
----> 1 complex("one")

ValueError: complex() arg is a malformed string
```

In [46]:
```
complex(10,2)
```

Out[46]:
```
(10+2j)
```

In [47]:
```
#bool()
```

In [48]:
```
bool(0)
```

Out[48]:
```
False
```

In [49]:
```
bool(1)
```

Out[49]:
```
True
```

In [50]:
```
bool(12)
```

Out[50]:
```
True
```

In [51]:
```
bool(12.5)
```

Out[51]:
```
True
```

In [52]:
```
bool(1 + 2j)
```

Out[52]:
```
True
```

In [53]:
```
bool(0.0123)
```

Out[53]:
```
True
```

In [54]:
```
bool(0 + 1j)
```

Out[54]:
```
True
```

In [55]:
```
bool(0.0)
bool(0 + 0j)
```

Out[55]:
```
False
```

In [58]:
```
bool("ten")
bool("pankaj")
bool("1")
bool("True")
```

Out[58]:
```
True
```

In [57]:
```
bool("")
```

Out[57]:
```
False
```

In [59]:
```
#str() method
```

In [60]:
```
str(10)
```

Out[60]:
```
'10'
```

In [61]:
```
str(12.3)
```

Out[61]:
```
'12.3'
```

In [62]:
```
str(12 + 2j)
```

Out[62]:
```
'(12+2j)'
```

In [63]:
```
str(True)
```

Out[63]:
```
'True'
```

In [64]:
```
bool(" ")
```

Out[64]:
```
True
```

In [ ]:
```

```

THANK - YOU