JOURNEY SCRAPBOOK

Technical Training - Week 5
19-Sep-2023 to 22-Sep-2023 (Days 14 – 17)

YUVA SAHITH VARMA SANGARAJU

BATCH - 3

Day 14 – Python Functions

- Args and kwargs
- Keyword and positional parameters
- Types of arguments
- OOP and Classes in Python
- Lists and their functions
- Zip function
- Enumerate
- Lambda function
- Implementation of all the functions

In [1]:
```
def my_function(*products):
    print("The costly product in the collection is: "+products[3])
my_function("Mouse","Keyboard","Laptop","Speaker","Projector")
```

```
The costly product in the collection is: Speaker
```

In [2]:
```
def my_func(**product):
    print("The hardware product name is: "+product["productname"])
my_func(productname="Mouse",brandname="Logitech")
```

```
The hardware product name is: Mouse
```

In [3]:
```
#Keyword only arguments
def nameAge(name, age):
    print("Hi, I am: ", name)
    print("My age is: ", age)
#Following the order of arguments
nameAge(name="Gyanesh",age=37)
#Changing the order of arguments
nameAge(age=37, name="Gyanesh")
```

```
Hi, I am:  Gyanesh
My age is:  37
Hi, I am:  Gyanesh
My age is:  37
```

In [4]:
```
#Positional parameter demo
def nameAge(name, age):
    print("Hi, I am: ",name)
    print("My age is: ",age)
print("Case-1:")
nameAge("John",20)
print("Case-2:")
nameAge(20, "John")
```

```
Case-1:
Hi, I am:  John
My age is:  20
Case-2:
Hi, I am:  20
My age is:  John
```

In [5]:
```
def minus(firstnum,secondnum):
    return firstnum-secondnum
firstnum,secondnum=20,10
result1=minus(firstnum,secondnum)
print("Used positional args", result1)
result2=minus(secondnum,firstnum)
print("Used positional args", result2)
```

```
Used positional args 10
Used positional args -10
```

In [7]:
```python
#Using a class named Person
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

p1 = Person("John", 36)

print(p1.name)
print(p1.age)
print(p1)
```

```
John
36
<__main__.Person object at 0x7fe3deef14d0>
```

In [10]:
```python
#Using a class named Person
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def __str__(self):
        return f"{self.name}({self.age})"

p1 = Person("John", 36)

print(p1)
```

```
John(36)
```

In [11]:
```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def myfunc(self):
        print("Hello, my name is " + self.name+". Age is: "+str(self.age))

p1 = Person("John", 36)
p1.myfunc()
```

```
Hello, my name is John. Age is: 36
```

In [13]:
```
class Person:
    def __init__(myobj, name, age):
        myobj.name = name
        myobj.age = age

    def myfunc(abc):
        print("Hello, my name is " + abc.name+". Age is: "+str(abc.age))

p1 = Person("John", 36)
p1.myfunc()
```

```
Hello, my name is John. Age is: 36
```

In [14]:

```
thislist=["mouse","keyboard","speaker","Laptop","Desktop","Printer"]
#length
print(len(thislist))
#capturing index value
print("Capturing the item from the list")
print(thislist[2])
#Negative indexing
print("Printing the index value")
print(thislist[-1])
#Returning the range of indexes
print("Returning range of indexes values")
print(thislist[2:5])
#Leaving out start value
print("Returning data by leaving out start value.")
print(thislist[:4])
#Leaving out end value
print("Returning data by leaving out end value.")
print(thislist[2:])
#sorting and returning list items
print("Returning data after sorting")
thislist.sort()
print(thislist)
#sorting and returning list items in descending order
print("Returning data after sorting in descending order")
thislist.sort(reverse=True)
print(thislist)
#sorting and returning list items using str.lower
print("Returning data after sorting using str.lower")
thislist.sort(key=str.lower)
print(thislist)
#Adding new element at the end of the list
print("Adding new element at the end of the list.")
```

```
thislist.append("Projector")
thislist.sort(key=str.lower)
print(thislist)
#Removing any element from the list
print("Removing value from the list.")
thislist.remove("Projector")
thislist.sort(key=str.lower)
print(thislist)
```

```
6
Capturing the item from the list
speaker
Printing the index value
Printer
Returning range of indexes values
['speaker', 'Laptop', 'Desktop']
Returning data by leaving out start value.
['mouse', 'keyboard', 'speaker', 'Laptop']
Returning data by leaving out end value.
['speaker', 'Laptop', 'Desktop', 'Printer']
Returning data after sorting
['Desktop', 'Laptop', 'Printer', 'keyboard', 'mouse', 'speaker']
Returning data after sorting in descending order
['speaker', 'mouse', 'keyboard', 'Printer', 'Laptop', 'Desktop']
Returning data after sorting using str.lower
['Desktop', 'keyboard', 'Laptop', 'mouse', 'Printer', 'speaker']
Adding new element at the end of the list.
['Desktop', 'keyboard', 'Laptop', 'mouse', 'Printer', 'Projector', 'speaker']
Removing value from the list.
['Desktop', 'keyboard', 'Laptop', 'mouse', 'Printer', 'speaker']
```

In [15]:
```
#using zip() function
a=("John","Oliver","Malcolm","Barry")
b=("Doe","Queen","Merlin","Allen","XYZ")
x=zip(a,b)
print(tuple(x))
```

```
(('John', 'Doe'), ('Oliver', 'Queen'), ('Malcolm', 'Merlin'), ('Barry', 'Allen'))
```

In [16]:
```
#using enumerate() function
l1=["eat","sleep","repeat"]
s1="geek"

#creating enumerate objects
obj1=enumerate(l1)
obj2=enumerate(s1)

print("Return type:", type(obj1))
print(list(enumerate(l1)))

#changing start index to 2 from 0
print(list(enumerate(s1,2)))
```

```
Return type: <class 'enumerate'>
[(0, 'eat'), (1, 'sleep'), (2, 'repeat')]
[(2, 'g'), (3, 'e'), (4, 'e'), (5, 'k')]
```

In [1]:
```python
#lambda function - calc
calc = lambda num: "Even number" if num % 2 == 0 else "Odd number"
print(calc(20))
```

```
Even number
```

Day 15 – Advanced Python

- Formatted and Raw Strings
- File paths
- Sequences and data structures
- Sets and their features
- Dictionaries – Indexing and Manipulation
- Saving individual cells as python files in Jupyter
- Numpy and its applications
- Basic functions, Broadcasting, Clip, astype(), etc.
- Rounding, Visualizations and Image processing
- Arange, linspace and unique

Day 15 continued...

- Data types and random array generation
- Matplotlib – Graphs, Capabilities
- Customizations with plt functions
- Pandas - Series and Dataframes
- Basic df functions
- Five-point summary
- Dealing with null values
- iloc, drop, inplace and axis
- unique, value_counts and plot functions
- Filtering data, sort, groupBy and aggregation
- ML classification

In [4]:
```
a = int(input("Enter first number:"))
b = int(input("Enter second number:"))
print("Sum of a and b is:", a+b)
print("Difference of a and b is:", a-b)
```

```
First number:5
Second number:2
Sum is: 7
Difference is: 3
```

In [5]:
```
print(f"Sum of {a} and {b} is",a+b)
print(f"Difference of {a} and {b} is",a-b)
```

```
Sum of 5 and 2 is 7
```

In [7]:
```
# if condition
# min age for voting
age = int(input("Enter the age:"))
if age >= 18:
    print("Eligible to vote")
else:
    print("Not eligible to vote!")
#    print("Eligible to vote after",18-age," years")
    print(f"Come back after {18-age} years")
```

```
Enter the age:15
Not eligible to vote!
Come back after 3 years
```

In [8]:
```
# Raw string
print(r"C:/programfiles/backup/newfolder/abcd/xyz")
```

```
C:/programfiles/backup/newfolder/abcd/xyz
```

In [ ]:
```
# Sequences / Data structures in Python
# Lists
# Tuples
# Set
# Dictionary
```

In [ ]:
```
a = [1,2,"abcd",True,2.04,2+3j] #List
type(a)
b = (1,2,"abcd",True,2.04,2+3j) #Tuple
type(b)
```

In [ ]:
```
# Sets
```

In [25]:
```
s = {2,4,3,9,1,10,7}
type(s)
```

Out[25]:
```
set
```

In [26]:
```
s.add(5)
```

In [27]:
```
s
```

Out[27]:
```
{1, 2, 3, 4, 5, 7, 9, 10}
```

In [28]:
```
print(s)
```

```
{1, 2, 3, 4, 5, 7, 9, 10}
```

In [29]:
```
s.discard(5)
s
```

Out[29]:
```
{1, 2, 3, 4, 7, 9, 10}
```

In [30]:
```
s.remove(3)
```

Out[30]: {1, 2, 4, 7, 9, 10}

In [31]:
```
s.pop()
s
```

Out[31]: {2, 4, 7, 9, 10}

In [32]:
```
a = {1,2,3,4,5}
a
```

Out[32]: {1, 2, 3, 4, 5}

In [33]:
```
s | a # union
```

Out[33]: {1, 2, 3, 4, 5, 7, 9, 10}

In [34]:
```
s & a # intersection
```

Out[34]: {2, 4}

In [35]:
```
s - a # difference    s - (s&a)
```

Out[35]: {7, 9, 10}

In [36]:
```
s ^ a # symmetric difference   (union of uncommon elements)
```

Out[36]: {1, 3, 5, 7, 9, 10}

In [37]:
```
# Dictionary
# d = {key:value, key:value, key:value} #syntax
d = {1:"C",2:"Java",3:"Python",4:"C++",5:"Javascript"}
d
```

Out[37]: {1: 'C', 2: 'Java', 3: 'Python', 4: 'C++', 5: 'Javascript'}

```
In [39]: d1 = {3:"C",4:"Java",6:"Python",8:"C++",10:"Javascript"}
         d1[3]

Out[39]: 'C'

In [40]: type(d1)

Out[40]: dict

In [41]: d1[1] = "SQL"
         d1

Out[41]: {3: 'C', 4: 'Java', 6: 'Python', 8: 'C++', 10: 'Javascript', 1: 'SQL'}

In [43]: d1.get(5, "Key not found")

In [44]: d1[5]

---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[44], line 1
----> 1 d1[5]

KeyError: 5

In [45]: d1.get(4, "Key not found")
```

In [47]:
```
del d1[3]
```

In [48]:
```
d1
```

Out[48]:
```
{4: 'Java', 6: 'Python', 8: 'C++', 10: 'Javascript', 1: 'SQL'}
```

In [50]:
```
ida = {"Batch 1":"30 participants","Batch 2":"44 participants","Batch 3":"25 participants","Batch 4":"28 participan
ida
```

Out[50]:
```
{1: '30 participants',
 2: '44 participants',
 3: '25 participants',
 4: '28 participants',
 5: '17 participants'}
```

In [51]:
```
editors = {'localhost': 'Jupyter Notebook',
           'offline':['Notepad++','IDLE','Pycharm','VSCode','Atom','Spyder'],
           'online': {'google':'colaboratory','kaggle':'kaggle notebook','aws':'Sagemaker','azure':'azure ml studio
```

In [52]:
```
editors['offline'][2]
```

Out[52]:
```
['Notepad++', 'IDLE', 'Pycharm', 'VSCode', 'Atom', 'Spyder']
```

In [53]:
```
editors['localhost']
```

Out[53]:
```
'Jupyter Notebook'
```

In [54]:
```
editors['online']['aws']
```

Out[54]:
```
'Sagemaker'
```

In [55]:
```
editors.get('online')
```

Out[55]:
```
{'google': 'colaboratory',
 'kaggle': 'kaggle notebook',
 'aws': 'Sagemaker',
 'azure': 'azure ml studio'}
```

In [56]:
```
%%writefile editors.py
editors = {'localhost': 'Jupyter Notebook',
           'offline':['Notepad++','IDLE','Pycharm','VSCode','Atom','Spyder'],
           'online': {'google':'colaboratory','kaggle':'kaggle notebook','aws':'Sagemaker','azure':'azure ml studio
editors['localhost']
editors['offline'][2]
editors['online']['aws']
editors.get('online')
```

```
Writing editors.py
```

In [57]:
```
%%writefile abc.java
efbweifbwo fieoqwfqoi
```

```
Writing abc.java
```

In [58]:
```
l1 = ['p1','p2','p3','p4']
l2 = ['pizza','burger','pasta','french fries']
d2 = dict(zip(l1,l2))
d2
```

Out[58]:
```
{'p1': 'pizza', 'p2': 'burger', 'p3': 'pasta', 'p4': 'french fries'}
```

In [ ]:
```
#NUMPY
```

In [59]:
```
import numpy as np
```

In [60]:
```
a = np.array(13)
a
```

Out[60]:
```
array(13)
```

In [61]:
```
a.ndim
```

Out[61]:
```
0
```

In [62]:
```
np.array(-0.56)
a
```

Out[62]:
```
array(13)
```

In [63]:
```
a.ndim
```

Out[63]:
```
0
```

In [64]:
```
a = np.array([1,2,3,4,5])
a
```

Out[64]:
```
array([1, 2, 3, 4, 5])
```

In [65]:
```
type(a)
```

Out[65]:
```
numpy.ndarray
```

In [66]:
```
a.ndim
```

Out[66]:
```
1
```

In [68]:
```
a = np.array([[1,2,3,4,5],[6,7,8,9,10]])
a
```

Out[68]:
```
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10]])
```

In [69]:
```
a.shape
```

Out[69]:
```
(2, 5)
```

In [73]:
```
a = np.array([[[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15]]])
```

```
In [74]: a.size
Out[74]: 15

In [75]: a.dtype
Out[75]: dtype('int64')

In [79]: a = np.array([3,1,2,4.6,5,"aws"])
         a.dtype
Out[79]: dtype('<U32')

In [82]: a = np.array(["aa","bb","ccc"])
         a.dtype
Out[82]: dtype('<U3')

In [83]: a = np.array([3,1,2,4,5,"aws"])
         a.dtype
Out[83]: dtype('<U21')

In [84]: a[0:3]
Out[84]: array(['3', '1', '2'], dtype='<U21')

In [86]: a = np.array([3,1,2,4,5])

In [87]: a[0]+a[2] #adding elements wrt index
Out[87]: 5

In [89]: b=np.array([1,2,3,4,5])
         a+b
Out[89]: array([ 4,  3,  5,  8, 10])
```

```
In [90]: np.sort(b)[::-1]
Out[90]: array([5, 4, 3, 2, 1])

In [91]: np.mean(b)
Out[91]: 3.0

In [92]: a1 = np.array([1.1,2.5,3.4,0.2])
         a1
Out[92]: array([1.1, 2.5, 3. , 4. , 0.2])

In [93]: a2 = a1.astype(int)
         a2
Out[93]: array([1, 2, 3, 4, 0])

In [94]: a = np.append(a,8)
         a
Out[94]: array([3, 1, 2, 4, 5, 8])

In [ ]: #Clip

In [95]: np.concatenate((a,b))
Out[95]: array([3, 1, 2, 4, 5, 8, 1, 2, 3, 4, 5])

In [96]: np.round(np.tan(b),2)
Out[96]: array([ 1.56, -2.19, -0.14,  1.16, -3.38])
```

In [97]:
```
np.argmax(a)
```

Out[97]:
```
5
```

In [98]:
```
np.delete(b,3)
```

Out[98]:
```
array([1, 2, 3, 5])
```

In [99]:
```
a = np.arange(1,11,1)
a
```

Out[99]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [101]:
```
a=np.arange(11)
a
```

Out[101]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [102]:
```
a = np.arange(1,11,2)
```

In [103]:
```
np.abs(a)
```

Out[103]:
```
array([1, 3, 5, 7, 9])
```

In [104]:
```
np.exp(a)
```

Out[104]:
```
array([2.71828183e+00, 2.00855369e+01, 1.48413159e+02, 1.09663316e+03,
       8.10308393e+03])
```

In [106]:
```
np.linspace(0,11,5) # 5 equally spaced values from 0 to 11
```

Out[106]:
```
array([ 0.  ,  2.75,  5.5 ,  8.25, 11.  ])
```

In [107]:
```
u = np.array([1,1,1,1,3,3,2,4,2,4,4,4,3,3,3,7,5,8])
```

In [108]:
```
np.unique(u,return_counts=True)
```

Out[108]:
```
(array([1, 2, 3, 4, 5, 7, 8]), array([4, 2, 5, 4, 1, 1, 1]))
```

In [109]:
```
np.ones(10)
```

Out[109]:
```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

In [110]:
```
a = np.ones(shape=[10,10], dtype='int')
a
```

Out[110]:
```
array([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]])
```

In [111]:
```
np.zeros(5)
```

Out[111]:
```
array([0., 0., 0., 0., 0.])
```

In [113]:
```
np.random.randint(0,10,5)
```

Out[113]:
```
array([4, 2, 0, 8, 7])
```

In [114]:
```
print(np.random.randint(0,10,6))
```

```
[5 7 5 0 5 3]
```

In [115]:
```
np.random.choice([10,5,0],size=[3,3])
```

Out[115]:
```
array([[ 0,  5,  0],
       [ 5,  0, 10],
       [ 5, 10, 10]])
```

In [117]:
```
np.random.seed(100)
np.random.randint(0,10,5)
```

Out[117]:
```
array([8, 8, 3, 7, 7])
```

In [ ]:
```
# Matplotlib - DAta Visualization Library
```

In [118]:
```
import matplotlib.pyplot as plt
import numpy as np
```

```
Matplotlib is building the font cache; this may take a moment.
```

In [119]:
```
a = np.array([[[0,0,0]]])
plt.imshow(a)
```

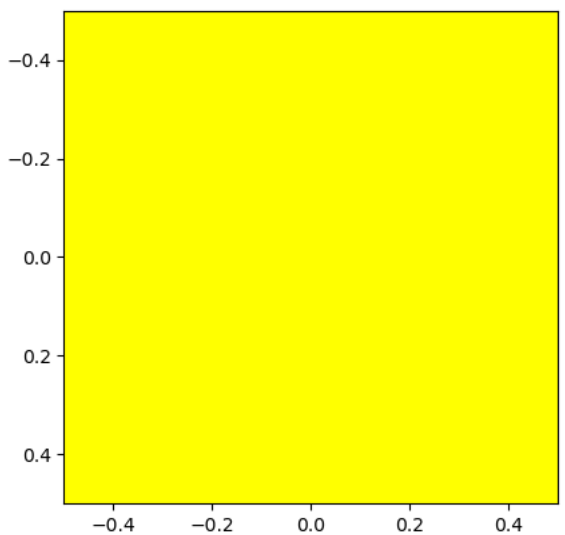
...

In [120]:
```
a = np.array([[[255,0,0]]])
plt.imshow(a)
```

...

In [121]:
```
a = np.array([[[255,255,0]]])
plt.imshow(a)
```

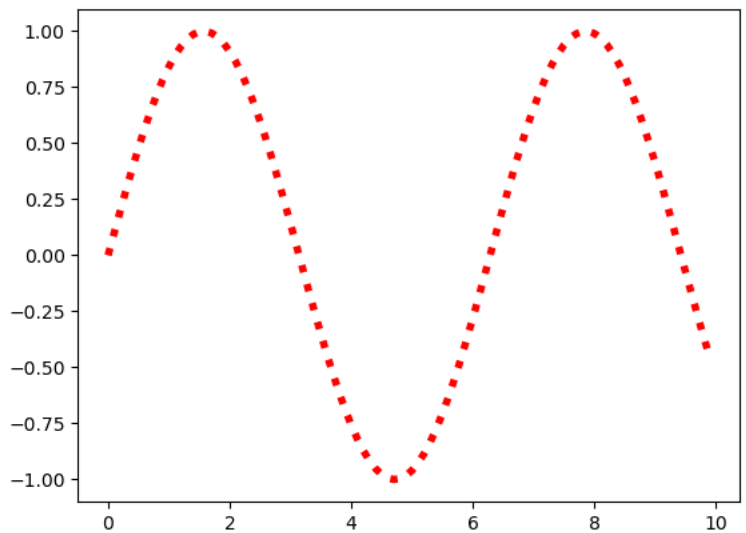
Out[121]:
```
<matplotlib.image.AxesImage at 0x7f407e49b190>
```



In [123]:
```
x = np.arange(0,10,0.1)
x
```

Out[123]:
```
array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1, 1.2,
       1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. , 2.1, 2.2, 2.3, 2.4, 2.5,
       2.6, 2.7, 2.8, 2.9, 3. , 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8,
       3.9, 4. , 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5. , 5.1,
       5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6. , 6.1, 6.2, 6.3, 6.4,
       6.5, 6.6, 6.7, 6.8, 6.9, 7. , 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7,
       7.8, 7.9, 8. , 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 9. ,
       9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.8, 9.9])
```

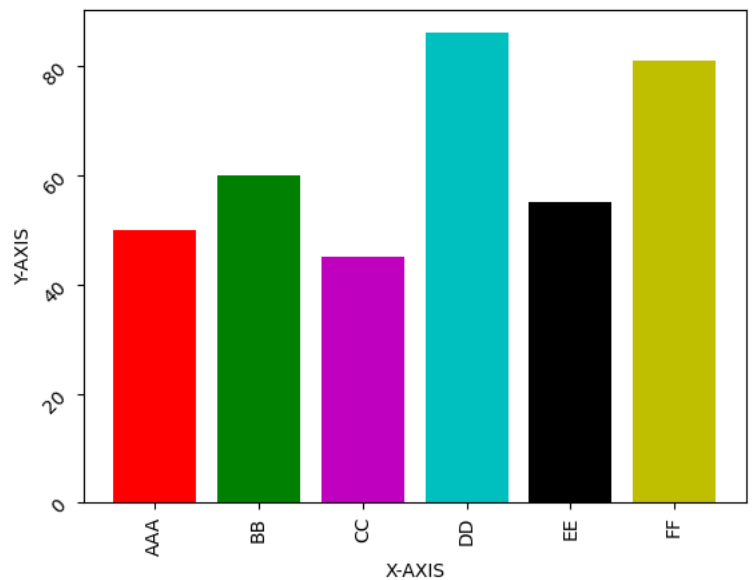
In [125]:
```
y = np.sin(x)
plt.plot(x,y,color='red',linewidth=4,linestyle='dotted')
plt.show()
```



In [126]:
```
x = np.arange(0,10*np.pi,0.1)
y = np.cos(x)
plt.plot(x,y,color='red')
plt.fill_between(x,y,color='red',alpha=0.9)
plt.show()
```

In [129]:
```
# BAR PLOT
x = ["AAA","BB","CC","DD","EE","FF"]
y = [50,60,45,86,55,81] #weight in kgs

colors1 = ['r','#349beb','m','#00FF1F','k','y']
plt.bar(x,y,color=colors1)
plt.xlabel("X-AXIS")
plt.ylabel("Y-AXIS")
plt.xticks(rotation=90)
plt.yticks(rotation=45)
plt.show()
```



In [ ]:
```
# Pandas
```

In [131]:
```
import pandas as pd
```

In [ ]:
```
# series - 1 dimensional
# dataframe - 2 dimensional
```

In [132]:
```
s = [100,200,300,400,500,600,700]
#s = np.array([100,200,300,400,500,600,700])
#s = (100,200,300,400,500,600,700)
sr = pd.Series(s)
sr
```

Out[132]:
```
0    100
1    200
2    300
3    400
4    500
5    600
6    700
dtype: int64
```

In [133]:
```
type(sr)
```

Out[133]:
```
pandas.core.series.Series
```

In [134]:
```
d = {'col1':[100,200,300,400,500],'col2':[200,300,400,500,600]}
s2 = pd.Series(d)
s2
```

Out[134]:
```
col1    [100, 200, 300, 400, 500]
col2    [200, 300, 400, 500, 600]
dtype: object
```

In [135]:
```
df = pd.DataFrame(d)
df
```

In [137]:
```
df = pd.read_csv("/home/labuser/Downloads/insurance.csv")
df
```

Out[137]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [138]:
```
df.shape
```

Out[138]:
```
(1338, 7)
```

In [139]:
```
df.size
```

Out[139]:
```
9366
```

In [140]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
```

In [141]:
```
df.describe()
```

Out[141]:

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

In [143]:
```
df.describe(include='all').T
```

Out[143]:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
age	1338.0	NaN	NaN	NaN	39.207025	14.04996	18.0	27.0	39.0	51.0	64.0
sex	1338	2	male	676	NaN	NaN	NaN	NaN	NaN	NaN	NaN
bmi	1338.0	NaN	NaN	NaN	30.663397	6.098187	15.96	26.29625	30.4	34.69375	53.13
children	1338.0	NaN	NaN	NaN	1.094918	1.205493	0.0	0.0	1.0	2.0	5.0
smoker	1338	2	no	1064	NaN	NaN	NaN	NaN	NaN	NaN	NaN
region	1338	4	southeast	364	NaN	NaN	NaN	NaN	NaN	NaN	NaN
charges	1338.0	NaN	NaN	NaN	13270.422265	12110.011237	1121.8739	4740.28715	9382.033	16639.912515	63770.42801

In [144]:
```
df.isnull().sum()
```

Out[144]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [145]:
```
#extract a col and create series
df['age']
#df.age
```

Out[145]:
```
0       19
1       18
2       28
3       33
4       32
        ..
1333    50
1334    18
1335    18
1336    21
1337    61
Name: age, Length: 1338, dtype: int64
```

In [146]:
```
#extract a col and create dataframe
df.iloc[:,0:1]
#df.iloc[:,[0]]
```

Out[146]:

	age
0	19
1	18
2	28
3	33
4	32
...	...
1333	50
1334	18
1335	18
1336	21
1337	61

1338 rows × 1 columns

In [147]:
```
df.iloc[:,0:3]
```

In [151]:
```
x = df.iloc[[100,200,300,250],:]
x
```

Out[151]:

	age	sex	bmi	children	smoker	region	charges
100	41	female	31.60	0	no	southwest	6186.1270
200	19	female	32.11	0	no	northwest	2130.6759
300	36	male	27.55	3	no	northeast	6746.7425
250	18	male	17.29	2	yes	northeast	12829.4551

In [152]:
```
x = df.drop(columns='smoker',axis=1) #inplace=True
x
```

Out[152]:

	age	sex	bmi	children	region	charges
0	19	female	27.900	0	southwest	16884.92400
1	18	male	33.770	1	southeast	1725.55230
2	28	male	33.000	3	southeast	4449.46200
3	33	male	22.705	0	northwest	21984.47061
4	32	male	28.880	0	northwest	3866.85520
...	...	...	...	...	...	...
1333	50	male	30.970	3	northwest	10600.54830
1334	18	female	31.920	0	northeast	2205.98080
1335	18	female	36.850	0	southeast	1629.83350
1336	21	female	25.800	0	southwest	2007.94500
1337	61	female	29.070	0	northwest	29141.36030

1338 rows × 6 columns

In [153]:
```
df['region'].nunique()
```

Out[153]:
```
4
```

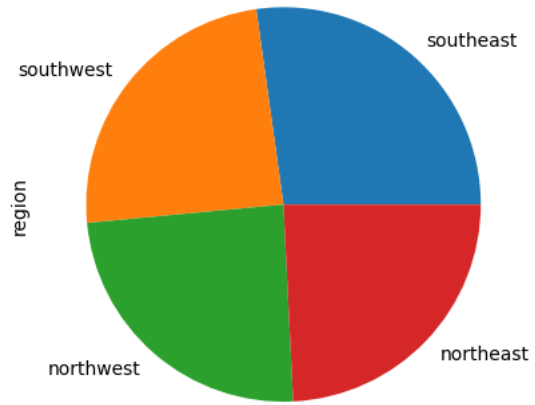
In [154]:
```
df['region'].unique()
```

Out[154]:
```
array(['southwest', 'southeast', 'northwest', 'northeast'], dtype=object)
```

In [155]:
```
df['region'].value_counts()
```

In [159]:
```
df['region'].value_counts().plot(kind='pie')
```

Out[159]: <Axes: ylabel='region'>



In [160]:
```
df.head()
```

Out[160]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

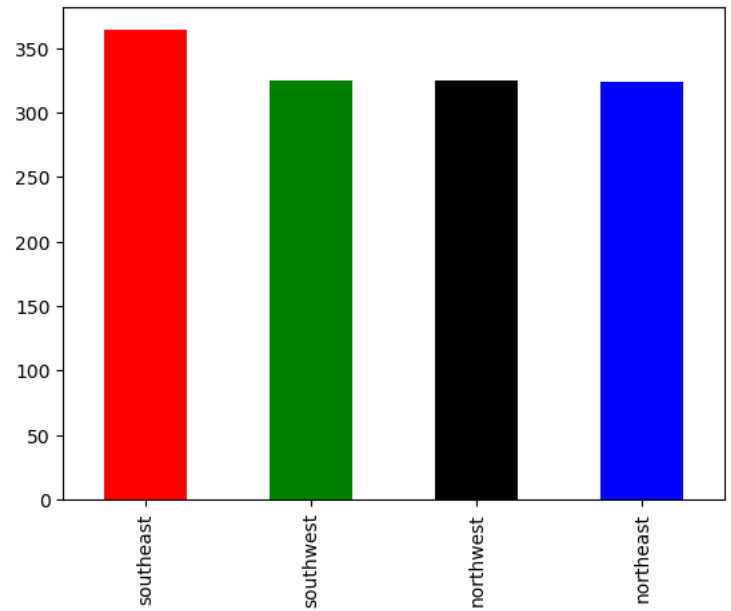
In [161]:
```
df.select_dtypes(include=['int64','float64'])
```

Out[161]:

	age	bmi	children	charges
0	19	27.900	0	16884.92400
1	18	33.770	1	1725.55230
2	28	33.000	3	4449.46200

In [167]:
```
df['region'].value_counts().plot(kind='bar',color=['red','green','black','blue'])
```

Out[167]: <Axes: >



In [168]:
```
plt.hist(df['charges'])
```

Out[168]:
```
(array([536., 398., 129.,  86.,  35.,  59.,  57.,  32.,   2.,   4.]),
 array([ 1121.8739  ,  7386.729311, 13651.584722, 19916.440133,
        26181.295544, 32446.150955, 38711.006366, 44975.861777,
        51240.717188, 57505.572599, 63770.42801 ]),
 <BarContainer object of 10 artists>)
```

In [172]:
```
df[(df['region'] == "northwest")&(df['smoker']=="no")]
```

Out[172]:

	age	sex	bmi	children	smoker	region	charges
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
7	37	female	27.740	3	no	northwest	7281.50560
9	60	female	25.840	0	no	northwest	28923.13692
24	37	male	28.025	2	no	northwest	6203.90175
...	...	...	...	...	...	...	...
1311	33	female	26.695	0	no	northwest	4571.41305
1319	39	female	26.315	2	no	northwest	7201.70085
1320	31	male	31.065	3	no	northwest	5425.02335
1324	31	male	25.935	1	no	northwest	4239.89265
1333	50	male	30.970	3	no	northwest	10600.54830

267 rows × 7 columns

In [173]:
```
df['smoker']=df['smoker'].replace(to_replace=["yes","no"],value=[1,0])
df['smoker']
```

Out[173]:
```
0       1
1       0
2       0
3       0
4       0
       ..
1333    0
1334    0
1335    0
1336    0
1337    1
Name: smoker, Length: 1338, dtype: int64
```

Day 16 – Spark

- Apache Spark architecture
- Driver, Worker and Executor nodes
- Cloud Providers and JVMs
- Spark engine and distributed computing
- Lazy evaluation and Partitioning
- Job, Stage and Task
- RDD operations: Transformations and Actions
- Narrow and Wide Transformations
- Caching and DAG
- Adaptive Query Execution (AQE)

Day 16 continued...

- rdd functions
- map, flatMap, reduceByKey, groupBy functions
- Issues while reading csv files and Read options
- User defined schema
- Dataframes and datasets
- withColumn Transformations

[1]:
```
import findspark
```

[2]:
```
# Initializing PySpark by finding the location
# findspark.init('') also works after restarting
findspark.init('/opt/anaconda3/lib/python3.11/site-packages/pyspark/')
```

[3]:
```
from pyspark.sql import SparkSession

#Initialize SparkSession
spark = SparkSession.builder.appName("WordCount").getOrCreate()
```

```
Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

23/09/25 03:52:23 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

[4]:
```
# Reading a text file
text_file = spark.sparkContext.textFile("/home/labuser/Desktop/Readme.txt")
```

[9]:
```
# Split lines into words and flatten them
words = text_file.flatMap(lambda line: line.split(" "))

# Map each word to (word, 1) to prepare for counting
word_counts = words.map(lambda word: (word, 1))

# Reduce by key to count the occurences of each word
word_count = word_counts.reduceByKey(lambda a, b: a+b)

# Collect the results
results = word_count.collect()
```

•••

[4]:
```
sc = spark.sparkContext
```

[ ]:
```
# RDD creation and functions
rdd = sc.parallelize([1,23,4,5])
```

[ ]:
```
rdd.collect()
```

[ ]:
```
result = rdd.map(lambda x: x*2)
result.collect()
```

```
[ ]: rdd1=sc.parallelize([1,2,3,4,5])
```

```
[ ]: resultrdd1 = rdd1.flatMap(lambda x:(x, x*2))
resultrdd1.collect()
```

```
[ ]: rdd2=sc.parallelize([1,2,3,4,5,6,7])
resultrdd2 = rdd.filter(lambda x: x*2 ==0)
resultrdd2.collect()
```

```
[ ]: rdd4=sc.parallelize([(1,2),(3,4),(1,6),(2,3)])
res4 = rdd4.reduceByKey(lambda x,y : x+y)
res4.collect()
```

```
[ ]: rdd5=sc.parallelize([(1,2),(3,4),(1,6),(2,3)])
res5 = rdd5.groupByKey()
res5.collect()
```

```
[ ]: for key,values in res5.collect():
    print(f"Key: {key}, Values: {list(values)}")
```

```
[ ]: # Counting words in a list using rdd
words_list = ["this","is", "a", "sample", "text", "document", "for", "word", "count", "example", "word", "count"]
```

```
[ ]: rdd = sc.parallelize(words_list)
```

```
[ ]: word_count = rdd.map(lambda x: (x,1))
```

```
[ ]: word_count.collect()
```

```
[ ]: count = word_count.reduceByKey(lambda x,y: x+y)
```

```
[ ]: count.collect()
```

```
[ ]: results = count.collect()
for word, count in results:
    print(f"{word}: {count}")
```

[11]:
```
# Reading and loading files
purchaserdd = sc.textFile("/home/labuser/Downloads/purchases.csv")
```

[12]:
```
purchaserdd.collect()
```

•••

[13]:
```
purchasedf = spark.read.csv("/home/labuser/Downloads/purchases.csv")
```

[14]:
```
purchasedf.show()
```

•••

[15]:
```
# Overcoming schema and header issues while reading csv files
purchased = spark.read.option("inferSchema",True).option("header",True).csv("/home/labuser/Downloads/purchases.csv")
purchased.show()
```

•••

[16]:
```
purchased.printSchema()
```

•••

[5]:
```
# Reading csv in df
import pandas as pd
df = pd.read_csv("/home/labuser/Downloads/purchases.csv")
df
```

[5]:

	Unnamed: 0	apples	oranges
0	June	3	0
1	Robert	2	3
2	Lily	0	7
3	David	1	2

Day 17 – PySpark

- Saving partitioned files in different formats after repartitioning
- Understanding SparkUI
- Spark session and context
- Spark SQL
- More transformations such as sort and select
- Last updated timestamp and literal function
- Cast and selectExpr
- dropDuplicates, dropna and fillna
- When-Otherwise conditions
- Python functions in PySpark

Day 17 continued...

- Caching a df
- Retail dataset implementation
- Joining tables and chaining
- Regex and write modes
- Parquet vs csv
- Coalesce

[18]:
```
# User defined schema
from pyspark.sql.types import StructType, StructField, IntegerType, StringType

udfschema = StructType ([StructField("Rank",IntegerType(),True),
                         StructField("Title",StringType(),True),
                         StructField("Genre",StringType(),True),
                         StructField("Description",StringType(),True),
                         StructField("Director",StringType(),True),
                         StructField("Actors",StringType(),True),
                         StructField("Year",StringType(),True),
                         StructField("Runtime (Minutes)",StringType(),True),
                         StructField("Rating",StringType(),True),
                         StructField("Votes",StringType(),True),
                         ])
```

[6]:
```
# Movie dataset
movies = spark.read.option("inferSchema",True).option("header",True).csv("/home/labuser/Downloads/IMDB-Movie-Data.csv")
movies.printSchema()
```

•••

[ ]:
```
#movies = spark.read.schema(udfschema).option("header",True).csv("/home/labuser/Downloads/IMDB-Movie-Data.csv")
#movies.printSchema()
```

[7]:
```
movies.show()
```

•••

[4]:
```
# Transformations on df
from pyspark.sql.functions import *
```

[26]:
```
movie = movies.withColumn("rev_new",col("Revenue (Millions)")*100).withColumn('batch',lit("Batch_03"))
movie.show()
```

•••

[ ]:
```
# Initiating Spark Session
#spark = SparkSession.builder\.appName("MySparkSession")\
#         .master("spark://<spark-master-host:<spark-master-port>")\.getOrCreate()
# sc = SparkContext("local",...)
```

[9]:
```
# Partitioning
movies.rdd.getNumPartitions()
```

•••

[11]:
```
parti = movies.repartition(1)
parti.rdd.getNumPartitions()
```

[11]:
```
1
```

[13]:
```
# Saving a csv file
parti.write.csv("/home/labuser/out.csv")
```

[14]:
```
parti = movies.repartition(10)
parti.rdd.getNumPartitions()
```

[14]:
```
10
```

[15]:
```
# Saving csv partitions
parti.write.csv("/home/labuser/test")
```

```
[Stage 12:=============================================>           (8 + 2) / 10]
```

[20]:
```
# Spark UI
spark_ui_url = f'{spark._jsc.sc().uiWebUrl().get()}/'
print("Spark UI URL:", spark_ui_url)
```

```
Spark UI URL: http://ip-172-31-14-120.ap-south-1.compute.internal:4040/
```

[21]:
```
sc.uiWebUrl
```

[21]:
```
'http://ip-172-31-14-120.ap-south-1.compute.internal:4040'
```

[22]:
```
# Spark SQL
movies.createOrReplaceTempView("Imdb")
```

[24]:
```
result = spark.sql("select * from Imdb")
type(result)
```

[24]:
```
pyspark.sql.dataframe.DataFrame
```

[25]:
```
result.show()
```

```
...
```

[26]:
```
res = spark.sql("select Rank, Title from Imdb")
res.show()
```

```
...
```

[30]:
```
# Sorting based on columns
testsort = movies.sort(col("Title").desc())
testsort.show()
```

•••

[31]:
```
# Select transformation to extract columns
movies.select('Title','Rank')
```

[31]:
```
DataFrame[Title: string, Rank: int]
```

[36]:
```
# Creating new last_updated_time column
from datetime import datetime
df = movies.withColumn('last_updated_ts',lit(datetime.now()))
df.show()
```

•••

[37]:
```
# CAST
dftest = df.selectExpr('CAST(last_updated_ts AS DATE) AS test','Title')
dftest.show()
```

•••

[39]:
```
# Column alias
dftest.select(col("Title").alias("MovieName")).show()
```

•••

[40]:
```
# Dropping duplicates based on certain columns
dropped = movies.dropDuplicates(["Director"])
dropped.count()
```

[40]:
```
646
```

[41]:
```
movies.count()
```

[41]:
```
1000
```

[46]:
```
# When-Otherwise in Spark
# Create a new column based on Rating
rate = movies.withColumn("rating_value", when((col("Rating")>=8) & (col("Rating")<=10),"Best").when(col("Rating")>=5,"Good").when(col("Rating")>=0,"Aver
rate.show()
```

•••

[9]:
```
def concat_shell(column):
    return column + "_shell"
```

[10]:
```
my_udf = udf(concat_shell,StringType())
```

[11]:
```
concat_shell("test")
```

[11]:
```
'test_shell'
```

[12]:
```
#from pyspark.sql.functions import *
test = movies.withColumn("new_col",my_udf(col("Title")))
```

[13]:
```
test.show()
```

•••

[15]:
```
movies.show(2)
```

•••

[16]:
```
new = movies.cache()
```

[18]:
```
new.show()
```

•••

[20]:
```
# Reading the retail dataset
cust = spark.read.option("inferSchema",True).option("header",True).option("sep", '\t').csv("/home/labuser/Downloads/Retail_Dataset/customer.csv")
cust.printSchema()
```

•••

[21]:
```
line = spark.read.option("inferSchema",True).option("header",True).option("sep", '\t').csv("/home/labuser/Downloads/Retail_Dataset/lineitem.csv")
line.printSchema()
```

•••

[22]:
```
nation = spark.read.option("inferSchema",True).option("header",True).option("sep", '\t').csv("/home/labuser/Downloads/Retail_Dataset/nation.csv")
nation.printSchema()
```

•••

[23]:
```
orders = spark.read.option("inferSchema",True).option("header",True).option("sep", '\t').csv("/home/labuser/Downloads/Retail_Dataset/orders.csv")
orders.printSchema()
```

•••

[28]:
```
cusord = cust.join(orders,cust.C_CUSTKEY == orders.O_CUSTKEY, how = "inner")
```

[29]:
```
cusord.show()
```

•••

[30]:
```
grpdf = orders.groupBy("O_CLERK").sum("O_TOTALPRICE")
grpdf.show()
```

•••

[35]:
```
cusord.rdd.getNumPartitions()
```

[35]:
```
1
```

[32]:
```
cusord.repartition(8)
```

[32]:
```
MapPartitionsRDD[158] at coalesce at NativeMethodAccessorImpl.java:0
```

[33]:
```
cusord.write.mode("append").csv("/home/labuser/Downloads/output")
```

[34]:
```
cusord.write.mode("append").parquet("/home/labuser/Downloads/output1")
```

[36]:
```
test = cusord.repartition(8)
test.rdd.getNumPartitions()
```

•••

[38]:
```
test.write.mode("append").parquet("/home/labuser/Downloads/output2/")
```

[40]:
```
test.coalesce(1).write.mode("append").parquet("/home/labuser/Downloads/output3/")
```

[19]:
```
joindf = cust.join(orders,cust.C_CUSTKEY == orders.O_CUSTKEY, how = "inner").\
        join(line, orders.O_ORDERKEY == line.L_ORDERKEY, how = "inner").\
        select(cust["*"],orders["O_TOTALPRICE"].alias("orderprice"),orders["O_ORDERSTATUS"],line["L_LINENUMBER"],line["L_LINESTATUS"])
joindf.show()
```

```
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+----------+-------------+--
--------+------------+

|C_CUSTKEY|            C_NAME|           C_ADDRESS|C_NATIONKEY|        C_PHONE|C_ACCTBAL|C_MKTSEGMENT|           C_COMMENT|orderprice|O_ORDERSTATUS|L_
NENUMBER|L_LINESTATUS|

+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+----------+-------------+--
--------+------------+

|      370|Customer#000000370|          oyAPndV IN|         12|22-524-280-8721|  8982.79|   FURNITURE|ges. final packag...| 172799.49|            O|
6|          O|

|      370|Customer#000000370|          oyAPndV IN|         12|22-524-280-8721|  8982.79|   FURNITURE|ges. final packag...| 172799.49|            O|
5|          O|

|      370|Customer#000000370|          oyAPndV IN|         12|22-524-280-8721|  8982.79|   FURNITURE|ges. final packag...| 172799.49|            O|
4|          O|

|      370|Customer#000000370|          oyAPndV IN|         12|22-524-280-8721|  8982.79|   FURNITURE|ges. final packag...| 172799.49|            O|
3|          O|

|      370|Customer#000000370|          oyAPndV IN|         12|22-524-280-8721|  8982.79|   FURNITURE|ges. final packag...| 172799.49|            O|
2|          O|

|      370|Customer#000000370|          oyAPndV IN|         12|22-524-280-8721|  8982.79|   FURNITURE|ges. final packag...| 172799.49|            O|
1|          O|

|      781|Customer#000000781|FQCAkyfV0 kL3,FNA...|         18|28-478-388-5881|  6403.62|   MACHINERY|ake blithely blit...|  38426.09|            O|
1|          O|

|     1234|Customer#000001234|B3OhbH0MRJE,F0Lc7...|          1|11-742-434-6436|  -982.32|   FURNITURE|y ironic instruct...|  205654.3|            F|
6|          F|

|     1234|Customer#000001234|B3OhbH0MRJE,F0Lc7...|          1|11-742-434-6436|  -982.32|   FURNITURE|y ironic instruct...|  205654.3|            F|
5|          F|

|     1234|Customer#000001234|B3OhbH0MRJE,F0Lc7...|          1|11-742-434-6436|  -982.32|   FURNITURE|y ironic instruct...|  205654.3|            F|
4|          F|

|     1234|Customer#000001234|B3OhbH0MRJE,F0Lc7...|          1|11-742-434-6436|  -982.32|   FURNITURE|y ironic instruct...|  205654.3|            F|
3|          F|

|     1234|Customer#000001234|B3OhbH0MRJE,F0Lc7...|          1|11-742-434-6436|  -982.32|   FURNITURE|y ironic instruct...|  205654.3|            F|
2|          F|

|     1234|Customer#000001234|B3OhbH0MRJE,F0Lc7...|          1|11-742-434-6436|  -982.32|   FURNITURE|y ironic instruct...|  205654.3|            F|
1|          F|
```

Spark Jobs (?)

User: labuser
Total Uptime: 54 min
Scheduling Mode: FIFO
Completed Jobs: 11

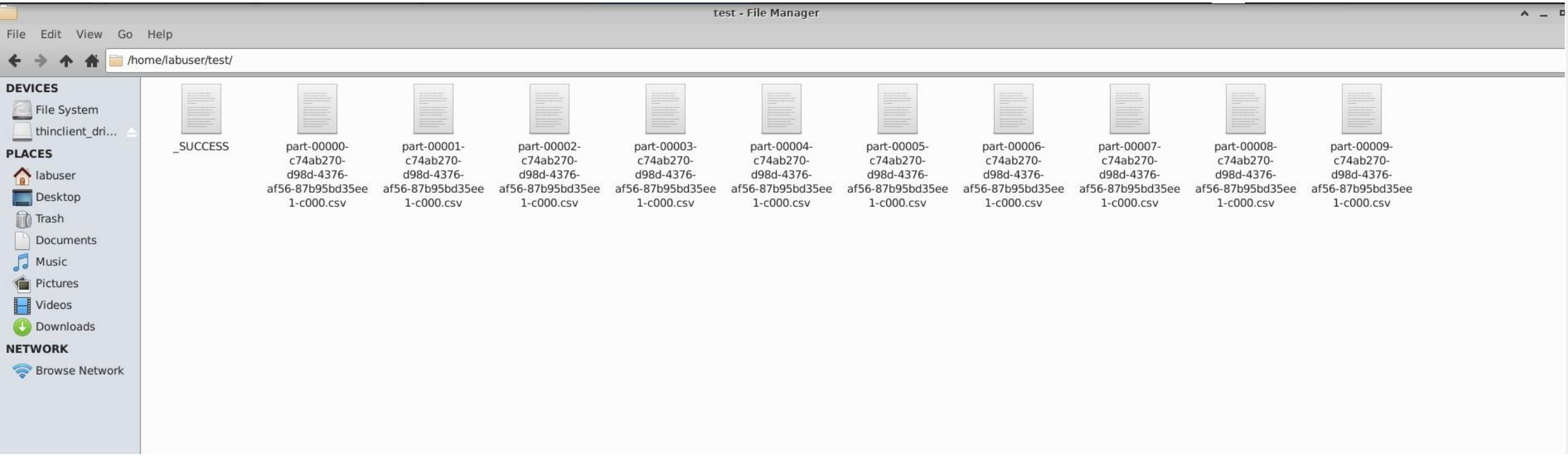
▸ Event Timeline

▾ Completed Jobs (11)

Page: 1

1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
10	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/09/22 04:13:45	0.7 s	1/1 (1 skipped)	10/10 (1 skipped)
9	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/09/22 04:13:45	0.1 s	1/1	1/1
8	javaToPython at NativeMethodAccessorImpl.java:0 javaToPython at NativeMethodAccessorImpl.java:0	2023/09/22 04:13:22	0.1 s	1/1	1/1
7	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/09/22 04:12:48	0.5 s	1/1 (1 skipped)	1/1 (1 skipped)
6	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/09/22 04:12:47	0.1 s	1/1	1/1
5	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/09/22 04:12:24	0.1 s	1/1	1/1
4	javaToPython at NativeMethodAccessorImpl.java:0 javaToPython at NativeMethodAccessorImpl.java:0	2023/09/22 04:10:38	0.1 s	1/1	1/1
3	javaToPython at NativeMethodAccessorImpl.java:0 javaToPython at NativeMethodAccessorImpl.java:0	2023/09/22 04:01:39	0.4 s	1/1	1/1
2	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2023/09/22 03:45:48	0.5 s	1/1	1/1
1	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/09/22 03:45:38	0.4 s	1/1	1/1
0	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2023/09/22 03:45:37	1 s	1/1	1/1


test - File Manager
File Edit View Go Help
/home/labuser/test/
DEVICES
File System
thinclient_dri...
PLACES
labuser
Desktop
Trash
Documents
Music
Pictures
Videos
Downloads
NETWORK
Browse Network
_SUCCESS
part-00000-c74ab270-d98d-4376-af56-87b95bd35ee1-c000.csv
part-00001-c74ab270-d98d-4376-af56-87b95bd35ee1-c000.csv
part-00002-c74ab270-d98d-4376-af56-87b95bd35ee1-c000.csv
part-00003-c74ab270-d98d-4376-af56-87b95bd35ee1-c000.csv
part-00004-c74ab270-d98d-4376-af56-87b95bd35ee1-c000.csv
part-00005-c74ab270-d98d-4376-af56-87b95bd35ee1-c000.csv
part-00006-c74ab270-d98d-4376-af56-87b95bd35ee1-c000.csv
part-00007-c74ab270-d98d-4376-af56-87b95bd35ee1-c000.csv
part-00008-c74ab270-d98d-4376-af56-87b95bd35ee1-c000.csv
part-00009-c74ab270-d98d-4376-af56-87b95bd35ee1-c000.csv

Details for Job 10

Status: SUCCEEDED
Submitted: 2023/09/22 04:13:45
Duration: 0.7 s
Associated SQL Query: 4
Completed Stages: 1
Skipped Stages: 1

- Event Timeline
- DAG Visualization

Completed Stages (1)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
12	csv at NativeMethodAccessorImpl.java:0 +details	2023/09/22 04:13:45	0.6 s	10/10		303.5 KiB	287.3 KiB	

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Skipped Stages (1)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
11	csv at NativeMethodAccessorImpl.java:0 +details	Unknown	Unknown	0/1				

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Details for Stage 4 (Attempt 0)

Resource Profile Id: 0
Total Time Across All Tasks: 83 ms
Locality Level Summary: Process local: 1
Input Size / Records: 302.5 KiB / 1000
Shuffle Write Size / Records: 279.6 KiB / 1000
Associated Job Ids: 4

- DAG Visualization
- Show Additional Metrics
- Event Timeline

Summary Metrics for 1 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	83.0 ms	83.0 ms	83.0 ms	83.0 ms	83.0 ms
GC Time	0.0 ms	0.0 ms	0.0 ms	0.0 ms	0.0 ms
Input Size / Records	302.5 KiB / 1000	302.5 KiB / 1000	302.5 KiB / 1000	302.5 KiB / 1000	302.5 KiB / 1000
Shuffle Write Size / Records	279.6 KiB / 1000	279.6 KiB / 1000	279.6 KiB / 1000	279.6 KiB / 1000	279.6 KiB / 1000

Aggregated Metrics by Executor

Tasks (1)

Show 20 entries Search:

Index	Task ID	Attempt	Status	Locality level	Executor ID	Host	Logs	Launch Time	Duration	GC Time	Input Size / Records	Shuffle Write Size / Records	Errors
0	4	0	SUCCESS	PROCESS_LOCAL	driver	ip-172-31-14-120.ap-south-1.compute.internal		2023-09-22 04:10:38	83.0 ms		302.5 KiB / 1000	279.6 KiB / 1000	

Showing 1 to 1 of 1 entries

Previous 1 Next

Environment

Runtime Information

Name	Value
Java Home	/usr/lib/jvm/java-8-openjdk-amd64/jre
Java Version	1.8.0_352 (Private Build)
Scala Version	version 2.12.17

Spark Properties

Name	Value
spark.app.id	local-1695354288242
spark.app.name	WordCount
spark.app.startTime	1695354283778
spark.app.submitTime	1695354283149
spark.driver.extraJavaOptions	-Djava.net.preferIPv6Addresses=false -XX:+IgnoreUnrecognizedVMOptions --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.invoke=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.net=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.base/java.util.concurrent.atomic=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.base/sun.nio.cs=ALL-UNNAMED --add-opens=java.base/sun.security.action=ALL-UNNAMED --add-opens=java.base/sun.util.calendar=ALL-UNNAMED --add-opens=java.security.jgss/sun.security.krb5=ALL-UNNAMED -Djdk.reflect.useDirectMethodHandle=false
spark.driver.host	ip-172-31-14-120.ap-south-1.compute.internal
spark.driver.port	40947
spark.executor.extraJavaOptions	-Djava.net.preferIPv6Addresses=false -XX:+IgnoreUnrecognizedVMOptions --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.invoke=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.net=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.base/java.util.concurrent.atomic=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.base/sun.nio.cs=ALL-UNNAMED --add-opens=java.base/sun.security.action=ALL-UNNAMED --add-opens=java.base/sun.util.calendar=ALL-UNNAMED --add-opens=java.security.jgss/sun.security.krb5=ALL-UNNAMED -Djdk.reflect.useDirectMethodHandle=false
spark.executor.id	driver
spark.master	local[*]
spark.rdd.compress	True

Executors

Show Additional Metrics

- [ ] Select All
- [ ] On Heap Memory
- [ ] Off Heap Memory
- [ ] Peak JVM Memory OnHeap / OffHeap
- [ ] Peak Execution Memory OnHeap / OffHeap
- [ ] Peak Storage Memory OnHeap / OffHeap
- [ ] Peak Pool Memory Direct / Mapped
- [ ] Resources
- [ ] Resource Profile Id
- [ ] Exec Loss Reason

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(1)	0	135.4 KiB / 366.3 MiB	0.0 B	2	0	0	20	20	57 min (1 s)	2.2 MiB	566.9 KiB	1.7 MiB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(1)	0	135.4 KiB / 366.3 MiB	0.0 B	2	0	0	20	20	57 min (1 s)	2.2 MiB	566.9 KiB	1.7 MiB	0

Executors

Show 20 entries Search:

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Thread Dump
driver	ip-172-31-14-120.ap-south-1.compute.internal:43757	Active	0	135.4 KiB / 366.3 MiB	0.0 B	2	0	0	20	20	57 min (1 s)	2.2 MiB	566.9 KiB	1.7 MiB	Thread Dump

Showing 1 to 1 of 1 entries

Previous 1 Next

SQL / DataFrame

Completed Queries: 4
Failed Queries: 1

Completed Queries (4)

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

ID ▾	Description	Submitted	Duration	Job IDs
4	csv at NativeMethodAccessorImpl.java:0 +details	2023/09/22 04:13:45	1.0 s	[9][10]
3	csv at NativeMethodAccessorImpl.java:0 +details	2023/09/22 04:12:47	0.8 s	[6][7]
1	showString at NativeMethodAccessorImpl.java:0 +details	2023/09/22 03:45:47	0.9 s	[2]
0	csv at NativeMethodAccessorImpl.java:0 +details	2023/09/22 03:45:36	3 s	[0]

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

Failed Queries (1)

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

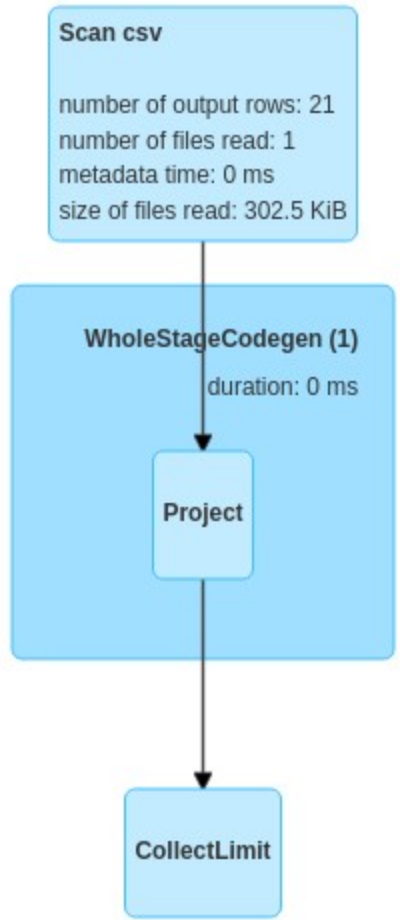
ID ▾	Description	Submitted	Duration	Succeeded Job IDs	Failed Job IDs
2	csv at NativeMethodAccessorImpl.java:0 +details	2023/09/22 04:12:24	0.3 s	[5]	

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

Details for Query 1

Submitted Time: 2023/09/22 03:45:47
Duration: 0.9 s
Succeeded Jobs: 2

☐ Show the Stage ID and Task ID that corresponds to the max metric



▸ Details