Class02

September 6, 2024

[1]:
```
print("Welcome to NUCOT's DataScience Class")
```

```
Welcome to NUCOT's DataScience Class
```

[2]:
```
x = -5
y = -10
```

[3]:
```
# Comprehension operators
x == y
```

[3]: False

[4]:
```
x != y
```

[4]: True

[5]:
```
x > y
```

[5]: True

[6]:
```
x < y
```

[6]: False

[7]:
```
x >= y
```

[7]: True

[8]:
```
x <= y
```

[8]: False

[9]:
```
# conditional operators
print(x < 10 and y > 5)
```

```
False
```

[10]:
```
print(x < 10 or y > 5)
```

```
True
```

[11]:
```
# String literals
name = 'Alice'
age = 21
```

[12]:
```
message = f"My name is {name} and I am {age} years old"
print(message)
```

```
My name is Alice and I am 21 years old
```

[13]:
```
# Swapping of numbers

x = 5
y = 10
```

[14]:
```
print("Before Swapping")
print("x =", x)
print("y =", y)

print("After Swapping")
temp = x
x = y
y = temp
print("x =", x)
print("y =", y)
```

```
Before Swapping
x = 5
y = 10
After Swapping
x = 10
y = 5
```

[15]:
```
a = 5
b = 6
print("Before Swapping")
print("a =", a)
print("b =", b)

a,b = b,a

print("After Swapping")
print("a =", a)
print("b =", b)
```

```
Before Swapping
a = 5
```

```
b = 6
After Swapping
a = 6
b = 5
```

[16]:
```
# Formatted string literals
print(f"BEFORE SWAPPING a = {a} b = {b}")
print(f"AFTER SWAPPING a = {b} b = {a}")
```

```
BEFORE SWAPPING a = 6 b = 5
AFTER SWAPPING a = 5 b = 6
```

[17]:
```
# Agumented operators
x = x + 5
x += 5
```

[18]:
```
# Lambda functions

x = 5
y = 2
raise_to_power = lambda x,y : x ** y
print("x raised to y = ", raise_to_power(x,y))
```

```
x raised to y =  25
```

[19]:
```
x = 5
y = 2
a = x**y
print(a)
```

```
25
```

[20]:
```
# Map functions

nums = [13,30,40,50]
square_all = list(map(lambda nums : nums ** 2, nums))
print(square_all)
```

```
[169, 900, 1600, 2500]
```

[21]:
```
var = [10,30,45,80,100,250,1999]
ans = list(map(lambda var: var**2, var))
print(ans)
```

```
[100, 900, 2025, 6400, 10000, 62500, 3996001]
```

[22]:
```
deegre_temps = [0,10,20,30]
farenhit_temp = list(map(lambda x: (x * 9/5) + 32, deegre_temps))
```

```
print(farenhit_temp)
```

```
[32.0, 50.0, 68.0, 86.0]
```

[23]:
```
# filter function
# syntax : filter(function, iteration)

num = [10, 20, 30 , 40, 50 , 99, 48, 87]
even = list(filter(lambda x : x % 2 == 0, num))
print(even)
```

```
[10, 20, 30, 40, 50, 48]
```

[24]:
```
# reduce function

from functools import reduce
numbers = [1,2,3,4,5,6,8,99,34,56,78,89]
product = reduce(lambda x,y : x*y, numbers)
print(product)
```

```
7537185976320
```

[25]:
```
numbers = [1,2,3,4,5,-6,-8,99,34,56,-78,89]
product = reduce(lambda x,y : x*y, numbers)
print(product)
```

```
-7537185976320
```

[26]:
```
# Concatenation

str1 = "Hello"
str2 = "World"
res = str1 +" "+str2
res
```

[26]:
```
'Hello World'
```

[27]:
```
a = "Python for"
b = "Data Science"
text = a+ " "+b
print(text)
```

```
Python for Data Science
```

[28]:
```
# slicing

# Python = 012345
text = "Python Language"
```

```
print(text[1:4])
```

```
yth
```

[29]:
```
# trying to search for "is" in the text
text = "Python is awesome"
print(text.find('is'))
```

```
7
```

[30]:
```
# replace function

new_text = text.replace("awesome", "great")
print(new_text)
```

```
Python is great
```

[31]:
```
# Breaking a string of substrings

text = "Python is fun"
words = text.split()
print(words)
```

```
['Python', 'is', 'fun']
```

[32]:
```
text = "Python is fun"
words = text.split('is')
print(words)
```

```
['Python ', ' fun']
```

[33]:
```
# numpy is numerical python
import numpy as np
m = int(input("Enter value for m"))
n = int(input("Enter value for n"))
matrix = np.random.rand(m,n)
print(matrix)
```

```
Enter value for m 5
Enter value for n 3
[[0.81839487 0.05212173 0.16798653]
 [0.02409657 0.88624696 0.5724023 ]
 [0.56863129 0.9738209  0.00183716]
 [0.53220684 0.94230291 0.77353329]
 [0.30018593 0.01184925 0.17403108]]
```

[68]:
```
import pandas as pd
data = [1,2,3,4,5]
series = pd.Series(data)
series
```

[68]:
```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

[70]:
```
data = {'Name': ["John", "Sahana", "Syeda"], 'City': ["California", "New York",␣
 ↪"London"], "Degree": ['MS', 'MCA', 'MBA']}
df = pd.DataFrame(data)
df
```

[70]:

	Name	City	Degree
0	John	California	MS
1	Sahana	New York	MCA
2	Syeda	London	MBA

[72]:
```
arr1 = np.zeros((3,3))
arr2 = np.ones((3,3))
result = arr1 + arr2
result
```

[72]:
```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

[74]:
```
total = np.sum(arr2)
total
```

[74]:
```
9.0
```

[76]:
```
arr1 = np.array([[1,2,3],[4,5,6]])
arr2 = np.array([[7,8,9],[10,11,12]])
hstack = np.hstack((arr1,arr2))
hstack
```

[76]:
```
array([[ 1,  2,  3,  7,  8,  9],
       [ 4,  5,  6, 10, 11, 12]])
```

[78]:
```
vstack = np.vstack((arr1,arr2))
print(vstack)
```

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

[80]:
```
# groupby functions
data = {'Students': ['John', 'Alice', 'Bob', 'John','Alice','Bob'], 'Subject':
 ↪['Math','Math','Math','Science','Science','Science'], 'Score':␣
 ↪[85,90,88,78,66,35]}
data = pd.DataFrame(data)
print(data)
```

```
  Students  Subject  Score
0     John     Math     85
1    Alice     Math     90
2      Bob     Math     88
3     John  Science     78
4    Alice  Science     66
5      Bob  Science     35
```

[82]:
```
group = data.groupby("Students")['Score'].sum()
group
```

[82]:
```
Students
Alice    156
Bob      123
John     163
Name: Score, dtype: int64
```

[84]:
```
group = data.groupby("Students")['Score'].mean()
group
```

[84]:
```
Students
Alice    78.0
Bob      61.5
John     81.5
Name: Score, dtype: float64
```

[86]:
```
group = data.groupby("Subject")['Score'].sum()
group
```

[86]:
```
Subject
Math       263
Science    179
Name: Score, dtype: int64
```

[103]:
```
data = {'Date':␣
 ↪['2022-01-01','2022-01-02','2022-01-01','2022-01-02','2022-01-01'],␣
 ↪'Product':['A','B','A','B','A'], 'Sales':[100,150,200,120,180]}
data = pd.DataFrame(data)
data
```

[103]:

	Date	Product	Sales
0	2022-01-01	A	100
1	2022-01-02	B	150
2	2022-01-01	A	200
3	2022-01-02	B	120
4	2022-01-01	A	180

[105]:
```
pivottable = data.pivot_table(index = 'Date', columns = 'Product', values =␣
 ↪'Sales', aggfunc = 'sum')
pivottable
```

[105]:

Product	A	B
Date		
2022-01-01	480.0	NaN
2022-01-02	NaN	270.0

[ ]:
```

```