

Lab Assignment 18

Student Name : Aryan Verma

Student Id: AF0417100

Topic: Numpy Functions

KEY POINTS OF NUMPY:

- ****NUMPY** IS A POWERFUL PYTHON LIBRARY DESIGNED FOR NUMERICAL COMPUTING, PROVIDING ROBUST SUPPORT FOR LARGE, MULTI-DIMENSIONAL ARRAYS AND MATRICES.**
- **IT INCLUDES AN EXTENSIVE ARRAY OF MATHEMATICAL FUNCTIONS THAT FACILITATE EFFICIENT OPERATIONS ON DATA STRUCTURES.**
- **NUMPY ARRAYS ARE STORED IN CONTIGUOUS BLOCKS OF MEMORY, RESULTING IN FASTER PERFORMANCE COMPARED TO TRADITIONAL PYTHON LISTS.**
- **THE LIBRARY SUPPORTS **BROADCASTING**, ENABLING OPERATIONS BETWEEN ARRAYS OF DIFFERENT SHAPES WITHOUT THE NEED FOR EXPLICIT LOOPS.**
- **IT OFFERS **VECTORIZED OPERATIONS**, WHICH STREAMLINE ELEMENT-WISE CALCULATIONS BY ELIMINATING THE NEED FOR MANUAL LOOPING.**
- **NUMPY PROVIDES COMPREHENSIVE TOOLS FOR **LINEAR ALGEBRA**, **RANDOM NUMBER GENERATION**, AND VARIOUS MATRIX OPERATIONS.**

- THE LIBRARY INTEGRATES SMOOTHLY WITH OTHER POPULAR LIBRARIES LIKE ****PANDAS****, ****MATPLOTLIB****, AND ****SCIKIT-LEARN****, ENHANCING DATA ANALYSIS AND SCIENTIFIC COMPUTING CAPABILITIES.

- NUMPY ALLOWS FOR FLEXIBLE ****SHAPE MANIPULATION**** (E.G., RESHAPING, FLATTENING, TRANSPOSING) TO ACCOMMODATE DIVERSE DATA HANDLING REQUIREMENTS.

- IT IS ESSENTIAL FOR PERFORMING NUMERICAL TASKS IN FIELDS SUCH AS DATA SCIENCE, MACHINE LEARNING, AND ENGINEERING APPLICATIONS.

SOME FUNCTIONS USED IN NUMPY:

- **NUMPY.ARRAY()** : CREATES AN ARRAY FROM LISTS OR TUPLES, FORMING THE FOUNDATION FOR NUMERICAL OPERATIONS.

- **NUMPY.ZEROS()** : GENERATES AN ARRAY FILLED WITH ZEROS, USEFUL FOR INITIALIZING MATRICES AND DATA STRUCTURES.

- **NUMPY.ONES()** : PRODUCES AN ARRAY FILLED WITH ONES, COMMONLY USED IN MATHEMATICAL COMPUTATIONS AND ALGORITHMS.

- **NUMPY.ARANGE()** : CREATES AN ARRAY WITH A SPECIFIED RANGE OF VALUES, FACILITATING ITERATION IN LOOPS.

- **NUMPY.Linspace()** : GENERATES EVENLY SPACED VALUES OVER A SPECIFIED INTERVAL, BENEFICIAL FOR PLOTTING AND FUNCTION EVALUATIONS.

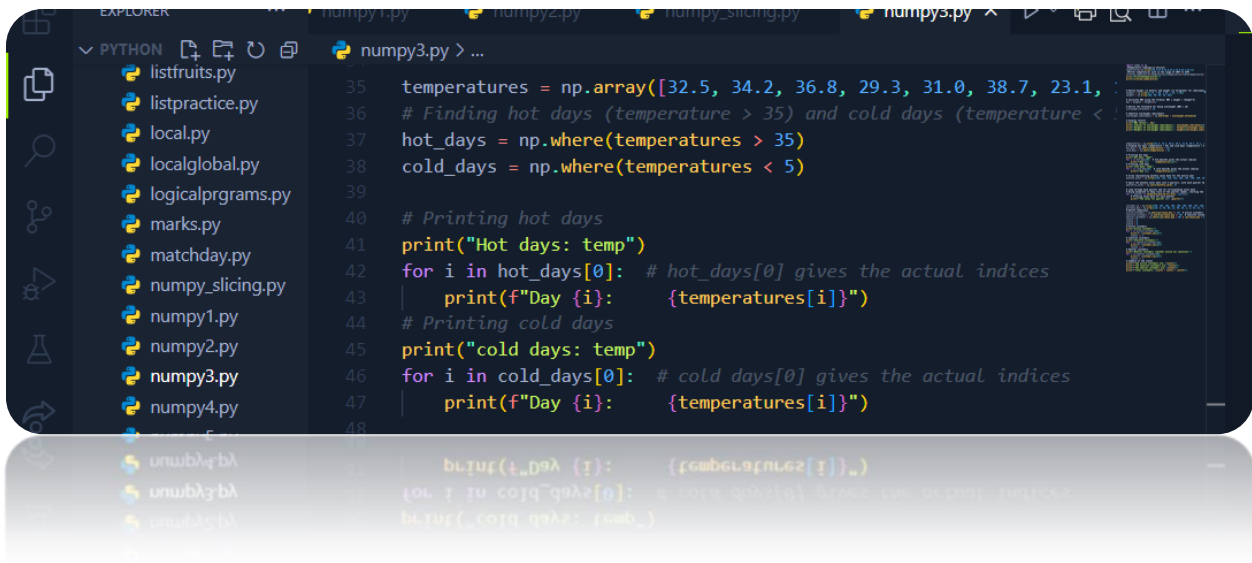
- **`NUMPY.RESHAPE()` : CHANGES THE SHAPE OF AN ARRAY WITHOUT MODIFYING ITS DATA, AIDING IN DATA MANIPULATION AND PREPARATION.**
- **`NUMPY.TRANSPOSE()` : SWITCHES THE AXES OF AN ARRAY, FREQUENTLY EMPLOYED IN LINEAR ALGEBRA OPERATIONS.**
- **`NUMPY.SPLIT()` : DIVIDES AN ARRAY INTO MULTIPLE SUB-ARRAYS, ENABLING ORGANIZED DATA HANDLING AND ANALYSIS.**
- **`NUMPY.SUM()` : COMPUTES THE SUM OF ARRAY ELEMENTS, COMMONLY USED IN AGGREGATING DATA FOR ANALYSIS.**
- **`NUMPY.MEAN()` : CALCULATES THE AVERAGE VALUE OF AN ARRAY, CRUCIAL FOR STATISTICAL ANALYSIS AND DATA INTERPRETATION.**

1. Suppose you have a dataset containing daily temperature readings for a city, and you want to identify days with extreme temperature conditions. Find days where the temperature either exceeded 35 degrees Celsius (hot day) or dropped below 5 degrees Celsius (cold day).

Input:

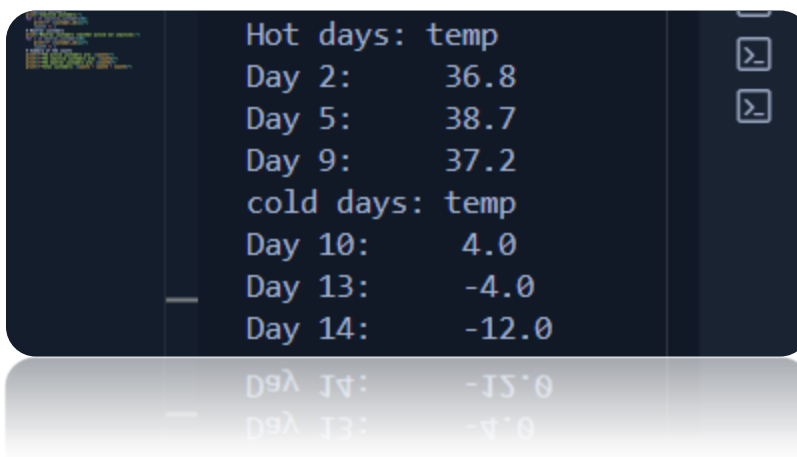
```
temperatures = np.array([32.5, 34.2, 36.8, 29.3, 31.0, 38.7,  
23.1, 18.5, 22.8, 37.2,4,25,12,-4,-12])
```

Ans:



```
35 temperatures = np.array([32.5, 34.2, 36.8, 29.3, 31.0, 38.7, 23.1,
36 # Finding hot days (temperature > 35) and cold days (temperature < 5)
37 hot_days = np.where(temperatures > 35)
38 cold_days = np.where(temperatures < 5)
39
40 # Printing hot days
41 print("Hot days: temp")
42 for i in hot_days[0]: # hot_days[0] gives the actual indices
43     print(f"Day {i}: {temperatures[i]}")
44 # Printing cold days
45 print("cold days: temp")
46 for i in cold_days[0]: # cold_days[0] gives the actual indices
47     print(f"Day {i}: {temperatures[i]}")
48
```

Output:



```
Hot days: temp
Day 2: 36.8
Day 5: 38.7
Day 9: 37.2
cold days: temp
Day 10: 4.0
Day 13: -4.0
Day 14: -12.0
```

2. Suppose you have a dataset containing monthly sales data for a company, and you want to split this data into quarterly reports for analysis and reporting purposes.

Input: `monthly_sales = np.array([120, 135, 148, 165, 180, 155, 168, 190, 205, 198, 210, 225])`

Ans:

```
49 # Array representing monthly sales data for the entire year
50 monthly_sales = np.array([120, 135, 148, 165, 180, 155, 168, 190, 205, 198, 210, 225])
51
52 # Split the monthly sales data into 4 quarters, with each quarter having 3 months
53 quarterly_sales = np.split(monthly_sales, 4)
54
55 # Loop through each quarter and its corresponding sales data
56 # Using enumerate to keep track of the quarter number, starting the index from 1
57 for i, quarter in enumerate(quarterly_sales, start=1):
58     # Printing sales data for each quarter
59     print(f"The sales for quarter {i}: {quarter}")
60
```

output:

```
the sales quarter wise
1: [120 135 148]
the sales quarter wise
2: [165 180 155]
the sales quarter wise
3: [168 190 205]
the sales quarter wise
4: [198 210 225]
```

Sample program

Ans:

```
python
numpy1.py numpy2.py numpy_slicing.py numpy3.py x numpy4.py
numpy3.py > ...
65 active_customers = np.where(purchase_day <= 10) # Active customers (purchase_day <= 10)
66 inactive_customers = np.where(purchase_day >= 30) # Inactive customers (purchase_day >= 30)
67 neutral_customers = np.where((purchase_day > 10) & (purchase_day < 30)) # Neutral customers (pur
68 count1 = 0
69 count2 = 0
70 count3 = 0
71 # Active customers
72 print("Active customers:")
73 for i in active_customers[0]:
74     print(f" {customer_id[i]}")
75     count1 += 1
76 # Inactive customers
77 print("Inactive customers:")
78 for i in inactive_customers[0]:
79     print(f" {customer_id[i]}")
80     count2 += 1
81 # Neutral customers
82 print("Neutral customers (neither active nor inactive):")
83 for i in neutral_customers[0]:
84     print(f" {customer_id[i]}")
85     count3 += 1
86 # Summary of the counts
87 print(f"The active customers are: {count1}")
88 print(f"The inactive customers are: {count2}")
89 print(f"The neutral customers are: {count3}")
90 print(f"Total customers: {count1 + count2 + count3}")
Ln 45, Col 25 Spaces: 4 UTF-8 CRLF {} Python 3.12.4 64-bit Go Live Prettier
```

output:

225]

Active customers:

100

101

107

108

111

112

Inactive customers:

102

103

105

106

109

110

Neutral customers (neither active
nor inactive):

104

The active customers are: 6

The inactive customers are: 6

The neutral customers are: 1

Total customers: 13

PS D:\python>

PS D:\python>

Total customers: 13

The neutral customers are: 1

The inactive customers are: 6

The active customers are: 6

PS D:\python>