

PYTHON LAB - 18

NUMPY FUNCTION

NAME: Keerthana K R

ID: AF0363623

QUESTIONS

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).
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.
3. Suppose you have a dataset containing customer data, and you want to split this data into two groups: one group for customers who made a purchase in the last 30 days and another group for customers who haven't made a purchase in the last 30 days.
4. Suppose you have two sets of employee data—one containing information about full-time employees and another containing information about part-time employees. You want to combine this data to create a comprehensive employee dataset for HR analysis.

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).

Code:

```
#import numpy package
import numpy as np

#given 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])

#Finding hot day using condition
hot_day = np.where(temperatures>35) [0]
print("Hot days :")
print("Day\tTemperature")
for day in hot_day:
    print(f"{day}\t{temperatures[day]}") #printing hot days

#Finding cold day using condition
cold_day = np.where(temperatures<5) [0]
print("Cold days :")
print("Day\tTemperature")
for day in cold_day:
    print(f"{day}\t{temperatures[day]}") #printing cold days
```

OUTPUT:

Hot days :

Day	Temperature
2	36.8
5	38.7
9	37.2

Cold days :

Day	Temperature
10	4.0
13	-4.0
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])`

Code:

```
#importing numpy package
import numpy as np

#input data
monthly_sales = np.array([120, 135, 148, 165, 180, 155, 168, 190,
205, 198, 210, 225])

#splitting data
Quarter_data = np.split(monthly_sales,4)
print("Monthly sales data in quarterly data")

#printing data
for quarter,sales in enumerate(Quarter_data,start=1):
    print(f"Quarter {quarter} sales : {sales}")
```

OUTPUT:

Monthly sales data in quarterly data

Quarter 1 sales : [120 135 148]

Quarter 2 sales : [165 180 155]

Quarter 3 sales : [168 190 205]

Quarter 4 sales : [198 210 225]

3. Suppose you have a dataset containing customer data, and you want to split this data into two groups: one group for customers who made a purchase in the last 30 days and another group for customers who haven't made a purchase in the last 30 days.
- Input: `customer_ids = np.array([101, 102, 103, 104, 105, 106, 107, 108, 109, 110])`
`last_purchase_days_ago = np.array([5, 15, 20, 25, 30, 35, 40, 45, 50, 55])`

Code:

```
#importing numpy package
import numpy as np

#Input data of customer ids and last purchased days
customer_ids = np.array([101, 102, 103, 104, 105, 106, 107, 108, 109, 110])
last_purchase_days_ago = np.array([5, 15, 20, 25, 30, 35, 40, 45, 50, 55])

#finding days where purchased is greater than 30 days
days = np.where(last_purchase_days_ago>30)[0][0]

#splitting ids
purchased_in_30_days, not_purchased = np.split(customer_ids, [days])

#printing data who have purchases and who have not
print("Customer ids who purchased in last 30 days")
print(purchased_in_30_days)
print("Customer ids who have not purchased in last 30 days")
print(not_purchased)
```

OUTPUT:

```
Customer ids who purchased in last 30 days
[101 102 103 104 105]
Customer ids who have not purchased in last 30 days
[106 107 108 109 110]
```

4. Suppose you have two sets of employee data—one containing information about full-time employees and another containing information about part-time employees. You want to combine this data to create a comprehensive employee dataset for HR analysis.

Input: # Employee data for full-time employees

```
full_time_employees = np.array([ [101, 'John Doe', 'Full-Time', 55000], [102, 'Jane Smith', 'Full-Time', 60000], [103, 'Mike Johnson', 'Full-Time', 52000] ])
```

Employee data for part-time employees

```
part_time_employees = np.array([ [201, 'Alice Brown', 'Part-Time', 25000], [202, 'Bob Wilson', 'Part-Time', 28000], [203, 'Emily Davis', 'Part-Time', 22000] ])
```

Code:

```
#import numpy package
import numpy as np

#Input data of full time and part time employees
full_time_employees = np.array([ [101, 'John Doe', 'Full-Time', 55000], [102, 'Jane Smith', 'Full-Time', 60000], [103, 'Mike Johnson', 'Full-Time', 52000] ])
part_time_employees = np.array([ [201, 'Alice Brown', 'Part-Time', 25000], [202, 'Bob Wilson', 'Part-Time', 28000], [203, 'Emily Davis', 'Part-Time', 22000] ])

#Joining data using vstack function
comprehensive_employee_dataset =
np.vstack((full_time_employees,part_time_employees))

#printing data
print(comprehensive_employee_dataset)
```

OUTPUT:

```
[['101' 'John Doe' 'Full-Time' '55000']
 ['102' 'Jane Smith' 'Full-Time' '60000']
 ['103' 'Mike Johnson' 'Full-Time' '52000']
 ['201' 'Alice Brown' 'Part-Time' '25000']
 ['202' 'Bob Wilson' 'Part-Time' '28000']
 ['203' 'Emily Davis' 'Part-Time' '22000']]
```