NumPy: Display all the dates for the month of March, 2026

1. March 2017 Dates

Write a NumPy program to display all the dates for the month of March,

```
import numpy as np

A = (np.arange('2026-03' , '2026-04', dtype='datetime64[D]'))
A
```

```
array(['2026-03-01', '2026-03-02', '2026-03-03', '2026-03-04',
       '2026-03-05', '2026-03-06', '2026-03-07', '2026-03-08',
       '2026-03-09', '2026-03-10', '2026-03-11', '2026-03-12',
       '2026-03-13', '2026-03-14', '2026-03-15', '2026-03-16',
       '2026-03-17', '2026-03-18', '2026-03-19', '2026-03-20',
       '2026-03-21', '2026-03-22', '2026-03-23', '2026-03-24',
       '2026-03-25', '2026-03-26', '2026-03-27', '2026-03-28',
       '2026-03-29', '2026-03-30', '2026-03-31'],
dtype='datetime64[D]')
```

1. Yesterday, Today, Tomorrow Dates

Write a NumPy program to get the dates of yesterday, today and tomorrow.

```
today = np.datetime64('today')
print(f'today Date:' ,today)
```

```
today Date: 2025-08-04
```

```
yesterday = np.datetime64('today') - np.timedelta64(1, 'D')

print(f'Yesterday Date: {yesterday}')
```

```
Yesterday Date: 2025-08-03
```

```
tomorrow = np.datetime64('today') + np.timedelta64(1, 'D')

print(f'Tomorrow Date: {tomorrow}')
```

```
Tomorrow Date: 2025-08-05
```

1. Count Days in a Specific Month

Write a NumPy program to count the number of days of specific month.

```
print(np.datetime64("2026-03-02") - np.datetime64("2027-03-02"))
```

```
-365 days
```

1. 24 Hour Datetime Objects

Write a NumPy program to create 24 python datetime.datetime objects (single object for every hour), and then put it in a numpy array.

```
import datetime

start = datetime.datetime(2000 ,1,1)
start

datetime.datetime(2000, 1, 1, 0, 0)

d_array = np.array([start + datetime.timedelta(hours = i)for i in
range(24)])
d_array

array([datetime.datetime(2000, 1, 1, 0, 0),
       datetime.datetime(2000, 1, 1, 1, 0),
       datetime.datetime(2000, 1, 1, 2, 0),
       datetime.datetime(2000, 1, 1, 3, 0),
       datetime.datetime(2000, 1, 1, 4, 0),
       datetime.datetime(2000, 1, 1, 5, 0),
       datetime.datetime(2000, 1, 1, 6, 0),
       datetime.datetime(2000, 1, 1, 7, 0),
       datetime.datetime(2000, 1, 1, 8, 0),
       datetime.datetime(2000, 1, 1, 9, 0),
       datetime.datetime(2000, 1, 1, 10, 0),
       datetime.datetime(2000, 1, 1, 11, 0),
       datetime.datetime(2000, 1, 1, 12, 0),
       datetime.datetime(2000, 1, 1, 13, 0),
       datetime.datetime(2000, 1, 1, 14, 0),
       datetime.datetime(2000, 1, 1, 15, 0),
       datetime.datetime(2000, 1, 1, 16, 0),
       datetime.datetime(2000, 1, 1, 17, 0),
       datetime.datetime(2000, 1, 1, 18, 0),
       datetime.datetime(2000, 1, 1, 19, 0),
       datetime.datetime(2000, 1, 1, 20, 0),
       datetime.datetime(2000, 1, 1, 21, 0),
       datetime.datetime(2000, 1, 1, 22, 0),
       datetime.datetime(2000, 1, 1, 23, 0)], dtype=object)
```

1. First Monday in May 2026

Write a NumPy program to find the first Monday in May

```
monday = (np.busday_offset('2026-05' , 0 , roll = 'forward' , weekmask
= 'Mon'))

monday

numpy.datetime64('2026-05-04')
```

1. Number of Weekdays in March 2017

Write a NumPy program to find the number of weekdays in March

```
print("Number of weekdays in March 2017:")
print(np.busday_count('2017-03', '2017-04'))

Number of weekdays in March 2017:
23
```

1. Convert np.datetime64 to Timestamp

Write a NumPy program to convert numpy datetime64 to Timestamp.

Sample output: Current date: 2017-04-01 08:01:12.722055 Timestamp: 1491033672.72 UTC from Timestamp: 2017-04-01 08:01:12.722055

```
# Importing the required libraries
import numpy as np
from datetime import datetime

# Getting the current date and time in UTC
dt = datetime.utcnow()

# Displaying the current date and time
print("Current date:")
print(dt)

# Converting the datetime object to a NumPy datetime64 object
dt64 = np.datetime64(dt)

# Calculating the timestamp from the datetime64 object
# The timestamp is calculated by subtracting the datetime '1970-01-01T00:00:00Z' from the current datetime
# Then it's divided by the number of seconds in one unit of timedelta64 to get the timestamp
ts = (dt64 - np.datetime64('1970-01-01T00:00:00Z')) / np.timedelta64(1, 's')

# Displaying the timestamp
print("Timestamp:")
print(ts)

# Converting the timestamp back to UTC datetime and displaying it
print("UTC from Timestamp:")
print(datetime.utcfromtimestamp(ts))

Current date:
2025-08-04 06:31:53.170670
Timestamp:
1754289113.17067
UTC from Timestamp:
2025-08-04 06:31:53.170670
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_13436\2351231006.py:6:
DeprecationWarning: datetime.datetime.utcnow() is deprecated and
scheduled for removal in a future version. Use timezone-aware objects
to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  dt = datetime.utcnow()
C:\Users\ASUS\AppData\Local\Temp\ipykernel_13436\2351231006.py:18:
DeprecationWarning: parsing timezone aware datetimes is deprecated;
this will raise an error in the future
  ts = (dt64 - np.datetime64('1970-01-01T00:00:00Z')) /
np.timedelta64(1, 's')
C:\Users\ASUS\AppData\Local\Temp\ipykernel_13436\2351231006.py:26:
DeprecationWarning: datetime.datetime.utcfromtimestamp() is deprecated
and scheduled for removal in a future version. Use timezone-aware
objects to represent datetimes in UTC:
datetime.datetime.fromtimestamp(timestamp, datetime.UTC).
  print(datetime.utcfromtimestamp(ts))
```