BUILT - IN MODULES -:

Python Modules -: There are two types of python modules:

Built-in modules

User-defined modules

MATH

In [41]:
```
import math
print(math.sqrt(25))
```

```
5.0
```

1]Trigonometric functions (sin, cos, tan) work with angles in radians, so math.radians() is used to convert degrees to radians.

2]Exponential functions (exp), logarithmic functions (log, log10), power function (pow), and square root (sqrt) are demonstrated.

3]The math.pi and math.e constants are also used.

In [42]:
```
# Trigonometric functions
angle = 45
sin_value = math.sin(math.radians(angle))
cos_value = math.cos(math.radians(angle))
tan_value = math.tan(math.radians(angle))
print(f'Sin({angle} degrees): {sin_value}')
print(f'Cos({angle} degrees): {cos_value}')
print(f'Tan({angle} degrees): {tan_value}')
```

```
Sin(45 degrees): 0.7071067811865476
Cos(45 degrees): 0.7071067811865476
Tan(45 degrees): 0.9999999999999999
```

In [43]:
```
# Exponential and logarithmic functions
exp_result = math.exp(2)
log_result = math.log(10)
log10_result = math.log10(100)

print(f'Exp(2): {exp_result}')
print(f'Log(10): {log_result}')
print(f'Log10(100): {log10_result}')
```

```
Exp(2): 7.38905609893065
Log(10): 2.302585092994046
Log10(100): 2.0
```

In [44]:
```
# Power and square root
power_result = math.pow(2, 3)
sqrt_result = math.sqrt(25)

print(f'2^3: {power_result}')
print(f'Square root of 25: {sqrt_result}')
```

```
2^3: 8.0
Square root of 25: 5.0
```

In [45]:
```
# Constants
print(f'Pi: {math.pi}')
print(f'Euler\'s number (e): {math.e}')
```

```
Pi: 3.141592653589793
Euler's number (e): 2.718281828459045
```

DATE AND TIME

The date module in Python is part of the standard library and is primarily used for working with dates

In [46]:
```
from datetime import datetime, timedelta
```

In [47]:
```
print(datetime.now())  # Output: Current date and time
```

```
2024-02-02 19:58:11.999204
```

In [48]:
```
# Get current date and time
current_datetime = datetime.now()
print(f'Current date and time: {current_datetime}')
```

```
Current date and time: 2024-02-02 19:58:12.146768
```

In [49]:
```
# Extract components from a datetime object
year = current_datetime.year
month = current_datetime.month
day = current_datetime.day
hour = current_datetime.hour
minute = current_datetime.minute
second = current_datetime.second

print(f'Year: {year}, Month: {month}, Day: {day}, Hour: {hour}, Minute: {minut
```

```
Year: 2024, Month: 2, Day: 2, Hour: 19, Minute: 58, Second: 12
```

random

various examples showcasing different functionalities provided by the random module in Python:

In [50]:
```
# Generate a random float between 0 and 1
random_float = random.random()
print(random_float)
```

```
0.359121208527287
```

In [51]:
```
# Generate a random integer within a range
random_integer = random.randint(1, 10)
random_integer
```

Out[51]:
```
6
```

In [52]:
```
# Generate a random integer within a range with step
random_step_integer = random.randrange(1, 10, 2)
random_step_integer
```

Out[52]:
```
9
```

In [53]:
```
# Generate a random choice from a sequence
colors = ['red', 'green', 'blue']
random_color = random.choice(colors)
print(colors)
print(random_color)
```

```
['red', 'green', 'blue']
red
```

In [54]:
```
# Generate a random sample without replacement
random_sample = random.sample(range(1, 11), 5)
random_sample
```

Out[54]:
```
[7, 8, 6, 2, 4]
```

calendar

The calendar module in Python provides functionalities related to the calendar

In [55]:
```
import calendar
```

In [56]:
```
# Display a calendar for a given year and month
year = 2022
month = 2
cal = calendar.month(year, month)
cal
```

Out[56]:
```
'   February 2022\nMo Tu We Th Fr Sa Su\n    1  2  3  4  5  6\n 7  8  9 10 11
12 13\n14 15 16 17 18 19 20\n21 22 23 24 25 26 27\n28\n'
```

In [57]:
```
# Display a calendar for a given year and month
year = 2024
month = 2
cal = calendar.month(year, month)
print(f'Calendar for {calendar.month_name[month]} {year}:\n{cal}')
```

```
Calendar for February 2024:
   February 2024
Mo Tu We Th Fr Sa Su
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29
```

In [58]:
```
# Display a calendar for a given year
year = 2022
cal = calendar.TextCalendar().formatyear(year)
print(f'Calendar for {year}:\n{cal}')
```

```
Calendar for 2022:
                                  2022

      January               February              March
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
                1  2      1  2  3  4  5  6      1  2  3  4  5  6
 3  4  5  6  7  8  9   7  8  9 10 11 12 13   7  8  9 10 11 12 13
10 11 12 13 14 15 16  14 15 16 17 18 19 20  14 15 16 17 18 19 20
17 18 19 20 21 22 23  21 22 23 24 25 26 27  21 22 23 24 25 26 27
24 25 26 27 28 29 30  28                    28 29 30 31
31

       April                 May                   June
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
             1  2  3                    1           1  2  3  4  5
 4  5  6  7  8  9 10   2  3  4  5  6  7  8   6  7  8  9 10 11 12
11 12 13 14 15 16 17   9 10 11 12 13 14 15  13 14 15 16 17 18 19
18 19 20 21 22 23 24  16 17 18 19 20 21 22  20 21 22 23 24 25 26
25 26 27 28 29 30     23 24 25 26 27 28 29  27 28 29 30
                      30 31

        July                 August              September
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
             1  2  3   1  2  3  4  5  6  7            1  2  3  4
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   5  6  7  8  9 10 11
11 12 13 14 15 16 17  15 16 17 18 19 20 21  12 13 14 15 16 17 18
18 19 20 21 22 23 24  22 23 24 25 26 27 28  19 20 21 22 23 24 25
25 26 27 28 29 30 31  29 30 31              26 27 28 29 30

      October              November              December
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
                1  2      1  2  3  4  5  6            1  2  3  4
 3  4  5  6  7  8  9   7  8  9 10 11 12 13   5  6  7  8  9 10 11
10 11 12 13 14 15 16  14 15 16 17 18 19 20  12 13 14 15 16 17 18
17 18 19 20 21 22 23  21 22 23 24 25 26 27  19 20 21 22 23 24 25
24 25 26 27 28 29 30  28 29 30              26 27 28 29 30 31
31
```

OS

The os module in Python provides a way to interact with the operating system. Here are some common functionalities demonstrated using the os module:

In [59]:
```
import os
print(os.getcwd())  # Output: Current working directory
```

```
C:\Users\arnak\DATA SCIENCE(NIT)\1 to 100
```

In [60]:
```
print(os.getlogin())
```

```
arnak
```

sys

The sys module in Python provides access to some variables used or maintained by the interpreter, as well as to functions that interact strongly with the interpreter.

In [61]:
```
import sys
print(sys.version)  # Output: Python version information
```

```
3.11.3 | packaged by Anaconda, Inc. | (main, Apr 19 2023, 23:46:34) [MSC v.19
16 64 bit (AMD64)]
```

Python has a rich standard library that includes a wide variety of modules to handle different functionalities. Here are some commonly used Python modules:

```
1] math: Mathematical functions (e.g., sqrt, sin, cos).
2]random: Random number generation and related functions.
3]datetime: Date and time manipulation.
4]os: Interactions with the operating system (e.g., file operations,
directory handling).
5]sys: Access to interpreter variables and functions.
6]json: JSON encoding and decoding.
7]requests: HTTP library for sending HTTP requests.
8]re: Regular expressions for pattern matching.
9]sqlite3: SQLite database interface.
10]csv: CSV file reading and writing.
11]PIL (Pillow): Image processing.
12]numpy: Numerical operations and array manipulation.
13]pandas: Data manipulation and analysis.
14]matplotlib: Plotting and data visualization.
15]scikit-learn: Machine learning algorithms and tools.
16]socket: Networking and socket programming.
17]threading: Thread-based parallelism.
18]argparse: Command-line argument parsing.
19]logging: Flexible logging framework.
20]hashlib: Secure hash and message digest algorithms.
21]xml.etree.ElementTree: XML parsing and manipulation.
22]collections: Additional data structures (e.g., Counter, defaultdict).
23]itertools: Tools for working with iterators and combinations.
24]gzip: Gzip compression and decompression.
25]shutil: High-level file operations (e.g., file copying, moving).
```

```
26]platform: Access to underlying platform’s identifying data.
27]getpass: Secure password input.
28]urllib: URL handling module.
29]hashlib: Cryptographic hash functions.
30]datetime: Basic date and time types.
```

In [ ]: