19/09/25- math Module

The **math module** in Python provides built-in mathematical functions and constants. These functions are useful for performing advanced math operations like square root, powers, trigonometry, and factorial.

3.1 sqrt()

```
import math
print(math.sqrt(25))
```

Output:

```
5.0
```

🔎 Theory

- math.sqrt(x) returns the square root of a number x.
- The square root of a number is a value that, when multiplied by itself, gives the original number.
- Example: √25 = 5, because 5 × 5 = 25.
- Always returns a floating-point number (decimal).

3.2 ceil() and floor()

```
import math
print(math.ceil(4.3))
print(math.floor(4.8))
```

Output:

```
5
4
```

🔎 Theory

- math.ceil(x) → Returns the **smallest integer greater than or equal to x** (rounds UP).
- math.floor(x) → Returns the **largest integer less than or equal to x** (rounds DOWN).

- These are useful when handling **rounding operations** in programs like billing, statistics, or game scores.

3.3 pow() and factorial()

```
import math
print(math.pow(2, 3))
print(math.factorial(5))
```

Output:

```
8.0
120
```

🔎 **Theory:**

- math.pow(a, b) → Returns a raised to the power b (a^b).
    - Example: $2^3 = 8$.
    - Always returns a float.
- math.factorial(n) → Returns the product of all positive integers up to n.
    - Example: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$.
    - Only works for non-negative integers.

3.4 trigonometry

```
import math
print(math.sin(math.pi/2))
print(math.cos(0))
```

Output:

```
1.0
1.0
```

🔎 **Theory:**

- math.sin(x) and math.cos(x) return the sine and cosine values of an angle.

- Angles must be in radians, not degrees.
    - To convert degrees → radians, use math.radians(degree_value).
- Examples:
    - sin(π/2) = 1
    - cos(0) = 1
- These are widely used in geometry, physics, engineering, and graphics programming.

3.5 gcd() – Greatest Common Divisor:

```
import math
print(math.gcd(24, 36))
```

Output:

```
12
```

🔎 **Theory:**

- math.gcd(a, b) returns the largest integer that divides both numbers without remainder.
- Example: GCD(24, 36) = 12, because 12 is the largest number dividing both 24 and 36.
- Useful in fractions, simplifying ratios, and number theory.

3.6 log() – Logarithm:

```
import math
print(math.log(100))       # Natural log (base e)
print(math.log(100, 10))   # Logarithm base 10
```

Output:

```
4.605170185988092
2.0
```

🔎 **Theory**

- math.log(x) → Natural logarithm of x (base e ≈ 2.718).
- math.log(x, base) → Logarithm with a custom base.
- Example: $\log_{10}(100) = 2$, since $10^2 = 100$.
- Very important in data science, probability, and growth models.

3.7 degrees() and radians():

```
import math
print(math.degrees(math.pi))     # Convert radians to degrees
print(math.radians(180))         # Convert degrees to radians
```

Output:

```
180.0
3.141592653589793
```

🔎 **Theory**

- math.degrees(radian_value) → Converts radians into degrees.
- math.radians(degree_value) → Converts degrees into radians.
- Essential in trigonometry, since Python uses radians for functions like sin() and cos().

3.8 Constants: pi and e

```
import math
print(math.pi)   # Value of pi
print(math.e)    # Euler's number
```

Output:

```
3.141592653589793
2.718281828459045
```

🔎 **Theory**

- **math.pi → Value of π (ratio of circumference to diameter).**
- **math.e → Euler's number, base of natural logarithms.**

- **Used widely in geometry, probability, calculus, and statistics.**