Numerical and mathematical functions in Python

1. `round()`

Purpose: Rounds a number to a specified number of decimal places.

```
print(round(3.14159, 2))  # Output: 3.14
```

Use this for:

- Displaying numbers neatly
- Limiting decimals for output or storage
- Calculations with limited precision

---

2. `abs()`

Purpose: Returns the **absolute value** (i.e., removes the minus sign).

```
print(abs(-9))  # Output: 9
```

Useful in:

- Distance calculations
- Comparing magnitude

---

3. `pow()`

Purpose: Returns a number raised to the power of another number.

```
print(pow(2, 3))  # Output: 8
```

Equivalent to:

```
print(2 ** 3)
```

---

4. `divmod()`

Purpose: Returns both the **quotient and remainder** of a division.

```
print(divmod(10, 3))  # Output: (3, 1)
```

Useful when:

- You want to divide and keep both results
- Used in formatting, time, and date math

---

5. `math` Module Usage

Import this built-in module to access more math functions.

```
import math

print(math.sqrt(16))    # Output: 4.0
print(math.ceil(4.2))   # Output: 5
print(math.floor(4.9))  # Output: 4
print(math.pi)          # Output: 3.141592653589793
```

Also includes:

- `math.sin()`, `math.cos()`, `math.log()`, etc.
- Used for scientific/engineering apps

---

6. Decimal Formatting (2 Decimal Places)

Control how numbers **look** when displayed.

```
x = 3.14159
print(f"{x:.2f}")           # Output: 3.14
print(format(x, ".2f"))     # Output: 3.14
```

Used for:

- Clean console or file output
- Presenting currency/financial values

---

7. Float Precision Handling

Floating point numbers can behave inaccurately:

```
print(0.1 + 0.2)  # Output: 0.30000000000000004
```

Use the `decimal` module to handle such cases:

```
from decimal import Decimal

a = Decimal("0.1")
b = Decimal("0.2")
print(a + b)  # Output: 0.3
```

---

8. Type Conversion: `int`, `float`, `Decimal`

```
x = "3.14"
float_x = float(x)       # Convert string to float
int_x = int(float_x)     # Convert float to int
decimal_x = Decimal(x)   # Convert to high-precision Decimal
```

Summary Table to round a number to 2 decimal

Method	Output Type	Use Case
`round(x, 2)`	Number	Simple rounding
`format(x, ".2f")`	String	String formatting
`f"{x:.2f}"`	String	Recommended for printing
`Decimal().quantize()`	Decimal	Financial apps, high precision