**Python Core Concepts: Operators, Input, and Math Module**

**1. Python Operators**

Operators are special symbols in Python that carry out arithmetic or logical computation.

1.1 Arithmetic Operators

These are used to perform mathematical operations.

* **+ (Addition):** Adds two operands. x + y
* **- (Subtraction):** Subtracts the right operand from the left. x - y
* **\* (Multiplication):** Multiplies two operands. x \* y
* **/ (Division):** Divides the left operand by the right, resulting in a floating-point number. x / y
* **% (Modulus):** Returns the remainder of the division. x % y
* **// (Floor Division):** Divides and returns the largest whole number (integer). x // y
* **\*\* (Exponent):** Raises the left operand to the power of the right. x \*\* y

1.2 Assignment Operators

These are used to assign values to variables.

* **=:** Assigns the value from the right to the left operand. x = 5
* **+=:** Adds the right operand to the left and assigns the result back to the left. x += 3 is equivalent to x = x + 3
* **-=:** Subtracts the right operand from the left and assigns the result. x -= 3 is equivalent to x = x - 3
* **\*=:** Multiplies and assigns the result. x \*= 3 is equivalent to x = x \* 3

1.3 Comparison Operators

These are used to compare two values and return a Boolean result (True or False).

* **== (Equal to):** x == y
* **!= (Not equal to):** x != y
* **> (Greater than):** x > y
* **< (Less than):** x < y
* **>= (Greater than or equal to):** x >= y
* **<= (Less than or equal to):** x <= y

1.4 Logical Operators

These are used to combine conditional statements.

* **and:** Returns True if both statements are true.
* **or:** Returns True if one of the statements is true.
* **not:** Reverses the result, returns False if the result is true.

**2. Bitwise Operators**

Bitwise operators act on operands as if they were strings of binary digits.

* **& (AND):** Sets each bit to 1 if both bits are 1.
  + 5 & 3 (0101 & 0011) results in 1 (0001)
* **| (OR):** Sets each bit to 1 if one of two bits is 1.
  + 5 | 3 (0101 | 0011) results in 7 (0111)
* **~ (NOT):** Inverts all the bits.
  + ~5 (~0101) results in -6
* **^ (XOR):** Sets each bit to 1 if only one of two bits is 1.
  + 5 ^ 3 (0101 ^ 0011) results in 6 (0110)
* **>> (Right Shift):** Shifts bits to the right by a specified number, pushing copies of the leftmost bit in from the left.
  + 5 >> 1 (0101 >> 1) results in 2 (0010)
* **<< (Left Shift):** Shifts bits to the left, pushing zeros in from the right.
  + 5 << 1 (0101 << 1) results in 10 (1010)

**3. Taking User Input with input()**

Python has a built-in function input() to get user input from the keyboard.

* The function pauses program execution and waits for the user to type something and press Enter.
* **Important:** The input() function always returns the user's input as a **string**, even if the user enters a number.

**Example:**

Python

name = input("Enter your name: ")

print("Hello, " + name)

Type Casting Input:

To use the input as a number, you must convert it using functions like int() or float().

**Example:**

Python

age\_str = input("Enter your age: ")

age\_int = int(age\_str) # Convert the string to an integer

birth\_year = 2025 - age\_int

print("You were likely born in", birth\_year)

**4. The math Module**

The math module is a built-in module that provides access to many common mathematical functions and constants. To use it, you must first import it.

import math

**4.1 Common Functions**

* math.sqrt(x): Returns the square root of x.
* math.pow(x, y): Returns x raised to the power of y.
* math.ceil(x): Rounds a number **up** to the nearest integer.
* math.floor(x): Rounds a number **down** to the nearest integer.
* math.factorial(x): Returns the factorial of x.
* math.sin(x), math.cos(x), math.tan(x): Return the sine, cosine, and tangent of x (assumes x is in radians).
* math.degrees(x): Converts angle x from radians to degrees.
* math.radians(x): Converts angle x from degrees to radians.

**4.2 Constants**

* math.pi: The mathematical constant Pi (approximately 3.14159).
* math.e: The mathematical constant e (approximately 2.71828).

**Example Usage:**

Python

import math

radius = 10

area = math.pi \* math.pow(radius, 2)

print("The area of the circle is:", area) # Output: 314.159...

print("The ceiling of 4.2 is:", math.ceil(4.2)) # Output: 5

print("The floor of 4.2 is:", math.floor(4.2)) # Output: 4