📓 Lecture 5: Operators in Python

🧠 What Are Operators?

Operators are **symbols or keywords** used to perform **calculations**, compare values, assign data, and **build logic** in programs.

---

1. 🔢 Arithmetic Operators

Used for **basic math operations**.

Operator	Purpose	Example	Output
`+`	Addition	`5 + 3`	`8`
`-`	Subtraction	`10 - 2`	`8`
`*`	Multiplication	`4 * 2`	`8`
`/`	Division	`10 / 3`	`3.33` (Float)
`//`	Floor Division	`10 // 3`	`3` (No decimals)
`%`	Modulus	`10 % 3`	`1` (Remainder)
`**`	Exponentiation	`2 ** 3`	`8` (2^3)

📌 Floor Division vs. Simple Division

Type	Description	Example	Result
`/`	Simple division → returns float	`10 / 3`	`3.3333`
`//`	Floor division → removes decimals	`10 // 3`	`3`

❓ Highlighted Question:

Q: What is the difference between simple division and floor division?
A: Simple division returns a **float**, while floor division returns an **integer** by removing the decimal part.

---

2. 📝 Assignment Operators

Used to **store values** in variables or **update existing values**.

Operator	Meaning	Example	Result
`=`	Assign value	`x = 5`	`x = 5`
`+=`	Add and assign	`x += 3`	`x = x + 3`
`-=`	Subtract and assign	`x -= 2`	`x = x - 2`
`*=`	Multiply and assign	`x *= 2`	`x = x * 2`
`/=`	Divide and assign	`x /= 3`	`x = x / 3`

✅ Example:

```
python
CopyEdit
x = 5
x += 3   # x becomes 8
print(x)  # Output: 8
```

---

3. 🆚 Comparison Operators

Used to **compare two values**. Returns **Boolean** (`True` / `False`).

Operator	Meaning	Example	Result
`>`	Greater Than	`5 > 2`	`True`
`<`	Less Than	`5 < 2`	`False`
`==`	Equal To	`5 == 5`	`True`
`!=`	Not Equal To	`5 != 2`	`True`

Operator	Meaning	Example	Result
`>=`	Greater Than or Equal	`5 >= 5`	`True`
`<=`	Less Than or Equal	`3 <= 2`	`False`

💡 Key Insight:

These are **crucial for conditions** like `if`, `while`, and logic building.

---

4. 🔗 Logical Operators

Used to **combine multiple conditions**.

Operator	Meaning	Example	Result
`and`	Both must be True	`True and False`	`False`
`or`	At least one is True	`True or False`	`True`
`not`	Reverses condition	`not True`	`False`

□ Example:

```
python
CopyEdit
a = True
b = False

print(a and b)  # False
print(a or b)   # True
print(not a)    # False
```

💡 Key Insight:

Used in **multi-condition decisions**, like `if x > 5 and y < 10`.

---

5. ⚙ Practical Use Case Examples

```
python
CopyEdit
```

```
x = 10
y = 3

# Modulus & Comparison
if x % y == 1:
    print("Remainder is 1")

# Logical Operators
if x > 5 and y < 5:
    print("Both conditions are true")
```

📌 Final Summary

Operator Type	Role
Arithmetic	Do calculations
Assignment	Store/update values
Comparison	Check equality/inequality
Logical	Combine multiple conditions

💬 Difficult Terms Explained

Term	Meaning
Modulus (%)	Returns the remainder after division
Exponentiation (**)	Raises a number to a power (e.g., $2^4 = 16$)
Boolean	A data type with only two values: `True` or `False`

🤔 Review Questions

1. What is the difference between `/` and `//` in Python?
2. Which operator would you use to check if two values are not equal?
3. What will be the output of this?

```
python
CopyEdit
x = 5
x *= 2
```

```
print(x)
```

4. What does `not True` return?
