Perfect ✅ Deepanshu!
Let's create **interview-ready, easy-to-understand, handwritten-style notes** on **Conditional Statements in Python** — from **basic to advanced** (especially useful for AI/ML engineers).

🧠 PYTHON CONDITIONAL STATEMENTS — COMPLETE NOTES

🌟 1. What are Conditional Statements?

Conditional statements are used to **make decisions** in code.
They allow your program to **execute different code blocks** based on whether a condition is **True** or **False**.

👉 Example:
If the temperature is above 30°C, print "Hot Day", else print "Cool Day".

🧩 2. Types of Conditional Statements in Python

Statement	Purpose
`if`	Used to check a condition
`elif`	Used to check multiple conditions
`else`	Runs when all conditions are false
`nested if`	Using if inside another if
`short-hand if`	Single-line condition
`match-case`	Used like switch-case (Python 3.10+)

🧠 3. Basic Syntax

```
if condition:
    # code to run when condition is True
elif another_condition:
    # code if previous condition was False but this one is True
else:
    # code if all conditions are False
```

✅ 4. Examples (Simple to Advanced)

Example 1: Basic `if` statement

```
x = 10
if x > 5:
    print("x is greater than 5")
```

Output:

```
x is greater than 5
```

Example 2: `if-else`

```
age = 18
if age >= 18:
    print("You can vote")
else:
    print("You cannot vote")
```

Output:

```
You can vote
```

Example 3: `if-elif-else`

```
marks = 85
```

```
if marks >= 90:
    print("Grade A")
elif marks >= 75:
    print("Grade B")
elif marks >= 50:
    print("Grade C")
else:
    print("Fail")
```

Output:

Grade B

Example 4: Nested `if`

```
x = 15

if x > 10:
    if x < 20:
        print("x is between 10 and 20")
```

Output:

x is between 10 and 20

Example 5: Short-Hand `if`

```
a = 5
b = 10
print("a is greater") if a > b else print("b is greater")
```

Output:

b is greater

Example 6: `match-case` (Python 3.10+)

```
day = "Sunday"

match day:
    case "Monday":
        print("Start of week")
    case "Sunday":
        print("Weekend!")
    case _:
        print("Normal day")
```

Output:

Weekend!

💡 5. Important Operators Used in Conditions

Operator	Description	Example	Result
`==`	Equal to	`5 == 5`	True
`!=`	Not equal to	`5 != 3`	True
`>`	Greater than	`10 > 8`	True
`<`	Less than	`7 < 3`	False
`>=`	Greater or equal	`4 >= 4`	True
`<=`	Less or equal	`3 <= 5`	True

⚙️ 6. Logical Operators (Often used in AI/ML)

Operator	Meaning	Example	Result
`and`	Both conditions True	`(x>5 and y<10)`	True
`or`	At least one True	`(x>5 or y<5)`	True

`not`	Negates the condition	`not(x>5)`	False

Example (AI/ML style)

```
accuracy = 0.92
loss = 0.1

if accuracy > 0.9 and loss < 0.2:
    print("Model is performing well ✅")
else:
    print("Need to improve model ⚙️")
```

Output:

Model is performing well ✅

📊 7. Conditional Statements in AI/ML

Conditional logic is used in **almost every AI/ML project**, examples:

Use Case	Example
Model evaluation	Check accuracy or loss threshold
Data preprocessing	Handle missing or invalid data
Feature selection	Include features if correlation > 0.5
Model choice	Choose different models based on dataset size
Decision-making	Predict class and check confidence score

Example in ML context:

```
accuracy = 0.87
if accuracy >= 0.9:
    print("Excellent Model")
```

```
elif accuracy >= 0.8:
    print("Good Model")
else:
    print("Needs Improvement")
```

🎞️ 8. Using Conditional Statements with Loops

Very common in AI tasks!

Example:

```
scores = [0.91, 0.67, 0.88, 0.95]

for s in scores:
    if s >= 0.9:
        print(f"{s} → Excellent")
    elif s >= 0.8:
        print(f"{s} → Good")
    else:
        print(f"{s} → Poor")
```

Output:

```
0.91 → Excellent
0.67 → Poor
0.88 → Good
0.95 → Excellent
```

⚡ 9. Advanced Concepts (For Interview)

🔷 Conditional Expressions (Ternary Operator)

Used for **one-line decision making**:

```
result = "Pass" if marks >= 40 else "Fail"
```

🔷 Conditional Comprehension

Used inside **list comprehension**:

```
numbers = [1, 2, 3, 4, 5, 6]
even_numbers = [x for x in numbers if x % 2 == 0]
print(even_numbers)
```

Output:

```
[2, 4, 6]
```

🧠 10. Common Mistakes to Avoid

Mistake	Correct Way
Using `=` instead of `==`	Use `==` for comparison
Indentation error	Always use 4 spaces or tab
Forgetting `:` after if	`if x > 5:` not `if x > 5`
Mixing up `and` / `or`	Understand logic properly

💬 11. Interview Tips (Very Common Questions)

1. **Difference between `if` and `elif`?**
   → `if` starts a condition block, `elif` checks next conditions if previous ones are False.

2. **Can we use multiple `if` without `elif`?**
   → Yes, but each `if` is checked independently.

3. **Can `else` be used without `if`?**
   → No.

4. **What is a ternary operator?**
   → One-line if-else: `x if condition else y`.

5. **What's `match-case` used for?**
   → Like `switch` in other languages; available in Python 3.10+.

🧩 12. Practice Task (Try Yourself)

```
temperature = float(input("Enter temperature: "))

if temperature > 30:
    print("It's Hot 🌞")
elif temperature >= 20:
    print("It's Warm 🌤️")
else:
    print("It's Cold ❄️")
```

🔥 BONUS (AI/ML Realistic Example)

```
accuracy = 0.88
loss = 0.12
epochs = 50

if accuracy > 0.9:
    print("Training Complete! Save the model.")
elif loss > 0.3:
    print("High loss! Try tuning hyperparameters.")
elif epochs < 100:
    print("Continue training...")
else:
    print("Review dataset quality.")
```

🎯 Summary

Concept	Description
`if`	Checks a condition
`elif`	Additional condition

`else`	Default action
Logical Operators	Combine multiple conditions
Ternary	One-line if-else
match-case	Switch-like structure
Usage in AI/ML	Data cleaning, model evaluation, etc.

Would you like me to create a **PDF version of these notes (in handwritten exam format)** so you can **print or revise before interview**?
I can include both **theory + code examples + short Q&A section**.