

# Day 3: Conditional Statements and Logical Operators

## 1. Conditional Statements

Conditional statements allow programs to execute specific blocks of code based on conditions. The key types in Python are:

1. **if Statement**

Executes a block of code if the condition is `True`.

```
age = 18
if age >= 18:
    print("You are eligible to vote.")
```

2. **if-else Statement**

Executes one block of code if the condition is `True` and another if it's `False`.

```
age = 16
if age >= 18:
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote.")
```

3. **if-elif-else Statement**

Checks multiple conditions in sequence.

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

4. **Nested Conditional Statements**

Conditions can be nested to check multiple levels of logic.

```
age = 20
citizen = True
if age >= 18:
    if citizen:
```

```

        print("You are eligible to vote.")
    else:
        print("You must be a citizen to vote.")
else:
    print("You are not old enough to vote.")

```

## 2. Comparison Operators

Comparison operators compare two values and return a Boolean result (True or False):

Operator	Description	Example (a = 10, b = 20)
==	Equal to	a == b → False
!=	Not equal to	a != b → True
>	Greater than	a > b → False
<	Less than	a < b → True
>=	Greater than or equal to	a >= b → False
<=	Less than or equal to	a <= b → True

## 3. Logical Operators

Logical operators combine multiple conditions:

- **and**: Returns True if all conditions are True.

```

age = 25
has_id = True
if age >= 18 and has_id:
    print("You can enter the club.")

```

- **or**: Returns True if at least one condition is True.

```

temperature = 35
if temperature > 30 or temperature < 0:
    print("Extreme weather!")

```

- **not**: Reverses the Boolean value of a condition.

```

is_raining = False
if not is_raining:
    print("You don't need an umbrella.")

```

## 4. Tips for Writing Effective `if` Statements

### Do's

1. **Use Clear and Descriptive Conditions**

Ensure your conditions are easy to read and understand.

```
if temperature > 30:  
    print("It's hot outside!")
```

2. **Keep Conditions Simple**

Break down complex conditions into smaller, manageable ones to improve readability.

```
is_sunny = True  
temperature = 25  
if is_sunny and temperature > 20:  
    print("Great day for a walk!")
```

3. **Use `elif` Instead of Multiple `if` Statements**

Use `elif` for mutually exclusive conditions to improve clarity.

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

4. **Handle Edge Cases**

Anticipate and handle edge cases in your conditions.

```
age = 0  
if age <= 0:  
    print("Invalid age.")
```

5. **Combine Conditions with Logical Operators**

Use `and`, `or`, and parentheses to combine conditions effectively.

```
age = 25  
has_ticket = True  
if age >= 18 and has_ticket:  
    print("You can enter.")
```

## Don'ts

1. **Don't Overcomplicate Conditions**  
Avoid long or nested conditions that are difficult to understand.
2. **Don't Repeat Code Blocks**  
Avoid writing the same code in multiple branches.
3. **Don't Forget to Handle All Cases**  
Always account for all possible outcomes, especially in critical applications.
4. **Don't Use Hardcoded Values Directly**  
Avoid magic numbers or strings; use variables or constants instead.
5. **Don't Use `if` When It's Not Necessary**  
Sometimes, you can avoid an unnecessary `if` by using simpler logic.
6. **Don't Neglect Code Indentation**  
Proper indentation is critical in Python since it determines code structure.

## 5. Practical Exercises

1. **Check Voting Eligibility**  
Write a program to check if a person is eligible to vote based on their age:
  - If age  $\geq 18$ , print "Eligible to vote."
  - Otherwise, print "Not eligible to vote."
2. **Find the Largest Number**  
Write a program that takes three numbers as input and prints the largest of the three.
3. **Odd or Even**  
Write a program to check if a number is odd or even.
4. **Discount Calculator**  
Write a program to calculate a discount based on the purchase amount:
  - For amounts  $\geq 500$ , apply a 20% discount.
  - For amounts  $\geq 200$  but less than 500, apply a 10% discount.
  - Otherwise, no discount.
5. **Triangle Validator**  
Write a program to check if three given sides can form a valid triangle.  
(Hint: The sum of any two sides must be greater than the third side.)