control flow statements

PYTHON KA CHILLA

Control flow statements in Python determine the order in which statements or blocks of code are executed.

- They allow you to control the decision-making process in a program, letting the program take different actions depending on the conditions.

Types of Control Flow Statements in Python:

1. `if`, `elif`, `else` **Statements** (Conditional statements)
2. `for` **and** `while` **Loops** (Iterative statements)
3. `break`, `continue`, **and** `pass` **Statements**
4. `try`, `except`, `finally` **Statements** (Exception handling)

control flow statements

PYTHON KA CHII

Control Flow Statement	Real-Life Scenario	When to Use	Explanation
if, elif, else	Choosing a transportation mode based on the weather	When making decisions	Use this to select an action based on different conditions. For example, if it's raining, take a car; if it's sunny, walk.
	Example		python weather = "rainy" if weather == "rainy": print("Take a car.") elif weather == "sunny": print("Walk.") else: print("Take an umbrella!")
for loop	Sending invitations for a wedding	Repeating tasks	Use a for loop to send invitations to a list of people. It helps automate repetitive tasks where each iteration works on one item from a sequence.
	Example		python guests = ["Ali", "Sara", "Ahmed"] for guest in guests: print(f"Sending invitation to {guest}")
while loop	Filling bottles with water in a factory until the tank is empty	Continuous actions	Use a while loop when the number of iterations is not known in advance and depends on a condition being true, such as filling bottles until the tank is empty.
	Example		python water_tank = 100 while water_tank > 0: print("Filling bottle...") water_tank -= 10

If, else, and elif

PYTHON KA CHILLA

Apni Shaten manwana

`if`, `else`, and `elif` Conditions in Python

In Python, conditional statements allow you to execute code based on specific conditions. The most common conditional statements are `if`, `else`, and `elif`. These allow you to make decisions in your code by evaluating expressions that result in either `True` or `False`.

- `if`: Executes the code block if the condition is `True`.
- `elif`: Checks additional conditions if the previous `if`/`elif` conditions are `False`.
- `else`: Executes if none of the preceding `if` or `elif` conditions are `True`.

If, else, and elif

PYTHON KA CHILLA

Syntax

```
if condition:
    # Code block that executes if the condition is True
elif another_condition:
    # Code block that executes if the first condition is False and this condition is True
else:
    # Code block that executes if all the above conditions are False
```

If, else, and elif

PYTHON KA CHILI

1. Basic `if` Statement

An `if` statement checks if a condition is `True`. If the condition is `True`, the indented code block under `if` will be executed.

Example:

python Copy code

```python
age = 18

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

```python
age = 17
if age >= 18:
    print("You are eligible to vote.")
else:
    print("Abhi chotay ho ap.")
```

✓ 0.0s

```
Abhi chotay ho ap.
```

```python
age = 16
if age >= 18:
    print("You are eligible to vote.")
elif age==17:
    print("Aik saal sabar kar lo.")
else:
    print("Abhi chotay ho ap.")
```

✓ 0.0s

```
Abhi chotay ho ap.
```

If, else, and elif with relational operators

PYTHON KA CHI

Operator	Operator name	Example	Returns
==	Equal to	a == b	True if the value of a is equal to the value of b else False
!=	Not equal to	a != b	True if the value of a is not equal to the value of b else False
<	Less than	a < b	True if the value of a is less than b else False
>	More than	a > b	True if the value of a is more than the value of b else False
<=	Less than or equal to	a <= b	True if the value of a is less than or equal to b else False
>=	More than or equal to	a >= b	True if the value of a is more than or equal to b else False

If, else, and elif with relational operators

PYTHO

Relational Operators

Python	Mathematics	Meaning
<	<	Less than
<=	≤	Less than or equal to
==	=	Equal to
>=	≥	Greater than or equal to
>	>	Greater than
!=	≠	Not equal to

Example:

python — Copy code

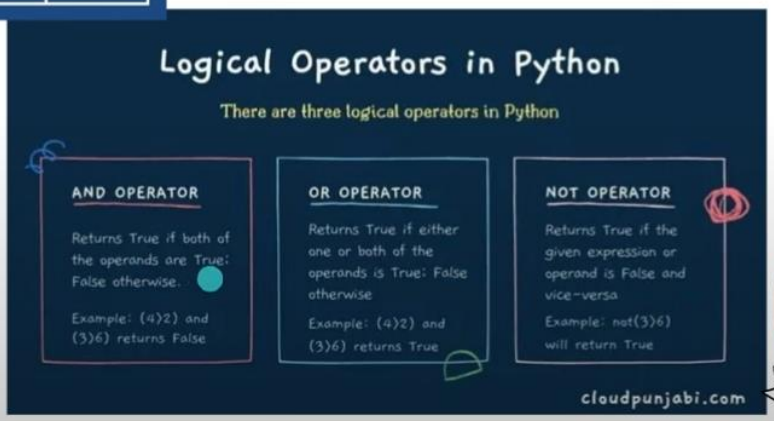
```python
score = 85

if score >= 90:
    print("Grade: A")
elif score >= 80:
    print("Grade: B")
elif score >= 70:
    print("Grade: C")
else:
    print("Grade: F")
```

If, else, and elif
with logical operators

PYTHON KA

Operator	Description	Example
AND	Returns True if both operands are True	A and B
OR	Returns True if either of the operands are True	A or B
NOT	Retruns True if the operand in False	not A



with logical operators

AND

```
age = 25
citizen = True

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

Here and mean it is necessary both conditions is true

```
You are eligible to vote.
```

Example with `or`:

```python
day = "Sunday"

if day == "Saturday" or day == "Sunday":
    print("It's the weekend!")
else:
    print("It's a weekday.")
```

Output:

```rust
It's the weekend!
```

In this example, only one condition needs to be `True` for the code in the `if` block to execute. Since `day == "Sunday"` is `True`, the code inside the `if` block runs.

with logical operators

Example with `not`:

```python
is_raining = False

if not is_raining:
    print("You can go outside.")
```

no

Output:

```go
You can go outside.
```

Here, the `not` operator negates the condition. Since `is_raining` is `False`, `not is_raining` evaluates to `True`, and the code block inside the `if` statement runs.

Example:

Nested statement

```python
age = 20
license = True

if age >= 18:
    if license:
        print("You can drive.")
    else:
        print("You need a license to drive.")
else:
    print("You are too young to drive.")
```

Output:

```
You can drive.
```

nested (if

```python
temp = 35

if temp >= 30:
    print("It's a hot day. Thanda pani pee lo")
elif temp <= 10:
    print("It's a cold day. Garam chai pee lo.")
else:
    print("It's neither hot nor cold. Enjoy the weather.")
```

✓ 0.0s

```
It's a cold day. Garam chai pee lo.
```

```python
occassion = "Eid-ul_Fitr"

if occassion == "Eid-ul_Fitr":
    print("Let's prepare some Sheer Khurma.")
elif occassion == "Eid-ul_Adha":
    print("Let's prepare some Kebab.")
else:
    print("It's a normal day. let's have Aalo Gobhi.")
```