

# Control Flow in Python (Conditionals)

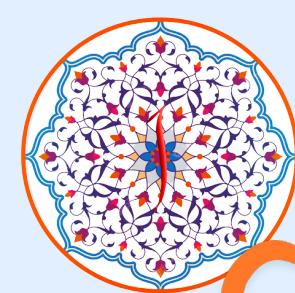
**Control flow in Python allows you to execute specific blocks of code based on certain conditions. This is typically achieved using if, elif, and else statements, which allow your program to make decisions and execute code conditionally.**

## if Statement

**The if statement is used to test a condition. If the condition evaluates to True, the code inside the if block is executed.**

```
python

if condition:
    # code to execute if the condition is True
```

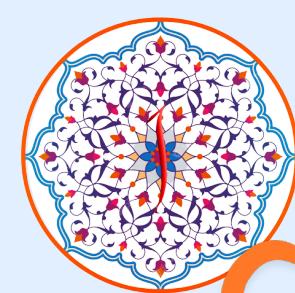


ikSaan.com

# Control Flow in Python (Conditionals)

```
python

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



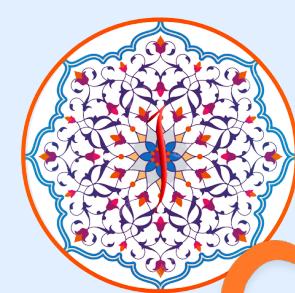
# Control Flow in Python (Conditionals)

## else Statement

The **else statement follows an if statement and is executed if the condition in the if statement is False.**

```
python

if condition:
    # code to execute if condition is True
else:
    # code to execute if condition is False
```



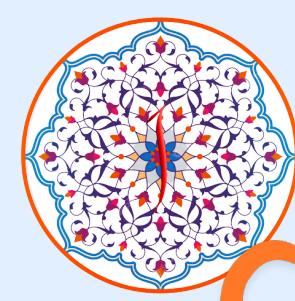
ikSaan.com

# Control Flow in Python (Conditionals)

```
python

x = 3

if x > 5:
    print("x is greater than 5")
else:
    print("x is not greater than 5")
# Output: x is not greater than 5
```



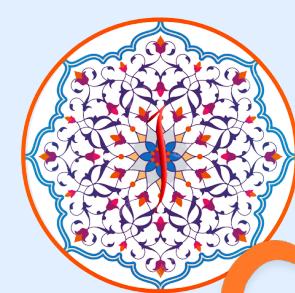
# Control Flow in Python (Conditionals)

## elif Statement

The **elif** (short for "else if") allows you to check multiple conditions. If the first condition is False, it moves to the next elif condition. If none of the conditions are True, the else block is executed.

```
python

if condition1:
    # code to execute if condition1 is True
elif condition2:
    # code to execute if condition2 is True
else:
    # code to execute if none of the above conditions are True
```

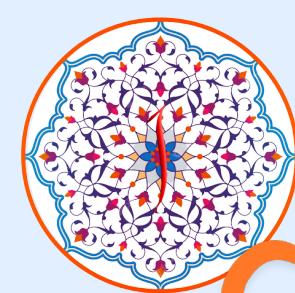


# Control Flow in Python (Conditionals)

```
python

x = 5

if x > 5:
    print("x is greater than 5")
elif x == 5:
    print("x is equal to 5")
else:
    print("x is less than 5")
# Output: x is equal to 5
```



# Control Flow in Python (Conditionals)

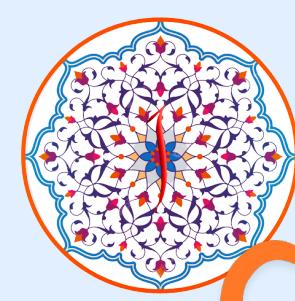
## Nested if Statements

You can nest if statements within other if statements to create more complex decision structures.

```
python

x = 10
y = 5

if x > 5:
    if y < 10:
        print("x is greater than 5 and y is less than 10")
# Output: x is greater than 5 and y is less than 10
```

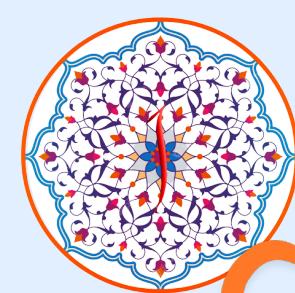


# Control Flow in Python (Conditionals)

## Logical Operators in Conditionals

Python provides logical operators (`and`, `or`, `not`) to combine multiple conditions in a single if statement.

- **and:** All conditions must be True.
- **or:** At least one condition must be True.
- **not:** Negates the result of a condition.



ikSaan.com

# Control Flow in Python (Conditionals)

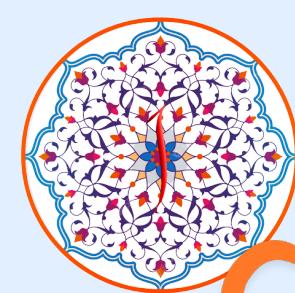
```
python

x = 10
y = 5

# Using 'and' operator
if x > 5 and y < 10:
    print("Both conditions are True")

# Using 'or' operator
if x < 5 or y < 10:
    print("At least one condition is True")

# Using 'not' operator
if not (x < 5):
    print("x is not less than 5")
```



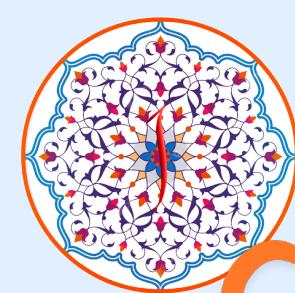
# Control Flow in Python (Conditionals)

## Conditional Expressions (Ternary Operator)

Python supports a concise way to write an if-else condition in a single line, known as the ternary operator.

```
python
```

```
value_if_true if condition else value_if_false
```

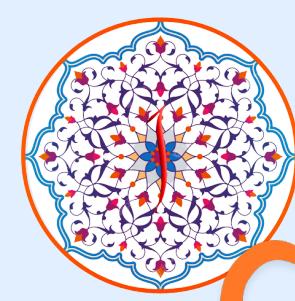


ikSaan.com

# Control Flow in Python (Conditionals)

```
python

age = 18
status = "Adult" if age >= 18 else "Minor"
print(status) # Output: Adult
```

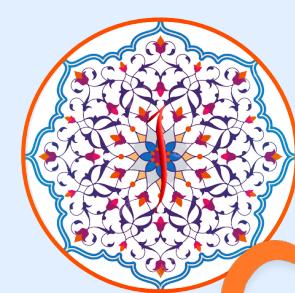


# Control Flow in Python (Conditionals)

## Comparison Operators in Conditionals

**Comparison operators are used to compare values. These return True or False and are commonly used in if statements.**

- **`==`: Equal to**
- **`!=`: Not equal to**
- **`>`: Greater than**
- **`<`: Less than**
- **`>=`: Greater than or equal to**
- **`<=`: Less than or equal to**



# Control Flow in Python (Conditionals)

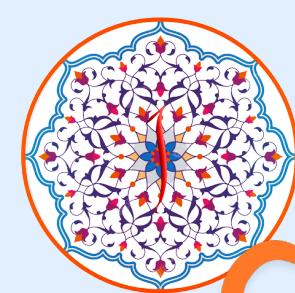
## Comparison Operators in Conditionals

```
python

x = 10
if x == 10:
    print("x is equal to 10")

if x != 5:
    print("x is not equal to 5")

if x >= 5:
    print("x is greater than or equal to 5")
```



# Control Flow in Python (Conditionals)

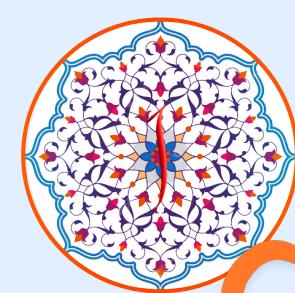
## Complete Example: Grading System

```
python

score = 85

if score >= 90:
    grade = 'A'
elif score >= 80:
    grade = 'B'
elif score >= 70:
    grade = 'C'
elif score >= 60:
    grade = 'D'
else:
    grade = 'F'

print(f"Your grade is: {grade}")
# Output: Your grade is: B
```



# Control Flow in Python (Conditionals)

## pass Statement

In Python, the **pass statement** is a placeholder used when a block of code is required syntactically but you don't want to execute any code.

```
python

x = 10
if x > 5:
    pass # You can fill in code later
```