

Understanding Conditional Statements

Instructor

Prashant Sahu
Manager (Data Science), Analytics Vidhya



Decision-Making in Python

Why Do We Need Conditional Statements

We need conditional statements when:



Making Decisions: To execute code only if a particular condition is true.

Controlling Flow: To determine the direction of program execution based on variables or inputs.

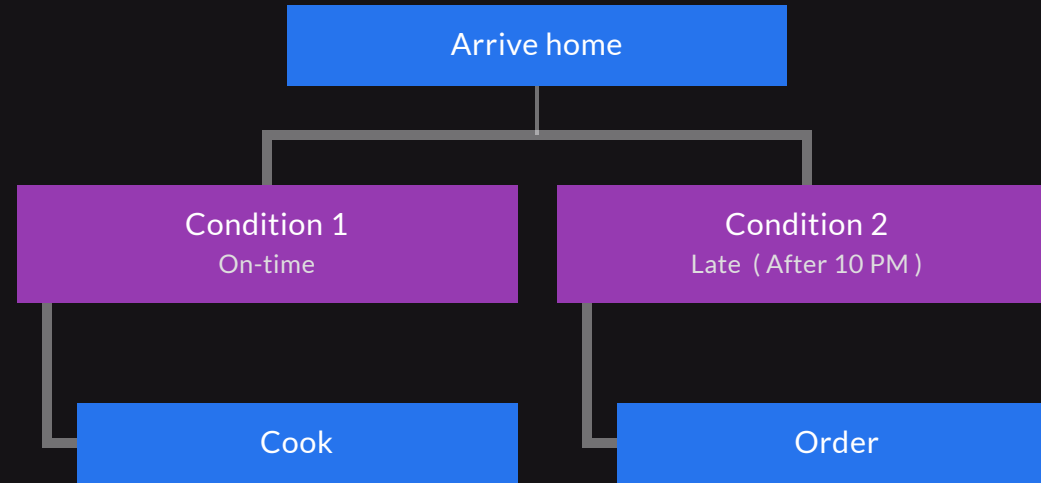
Validating Inputs: To check if user inputs or data meet certain criteria before processing.

Handling Errors: To manage exceptions or unexpected situations gracefully.

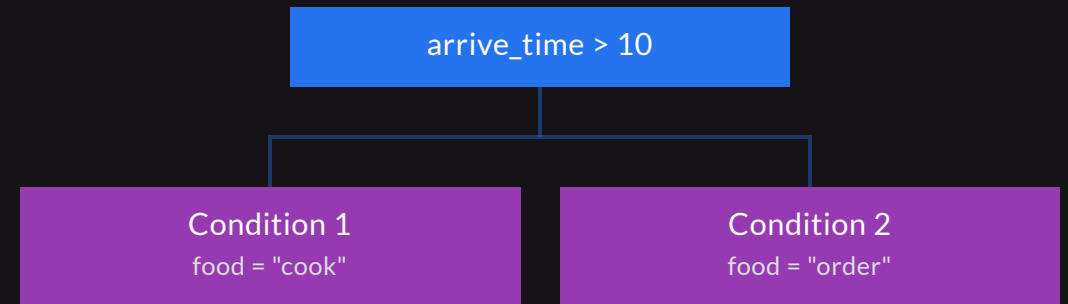
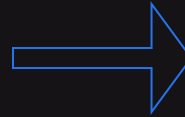
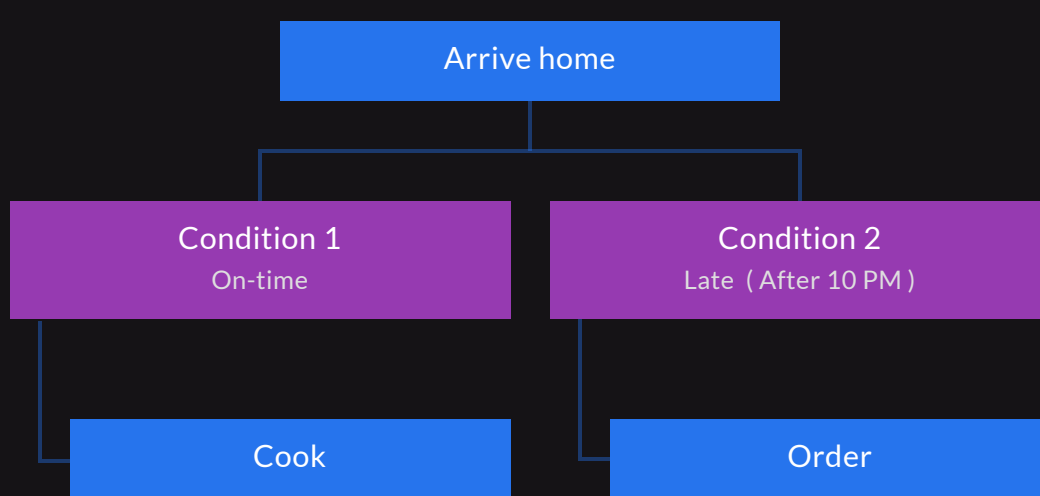
Scenario

Conditional Statements

Construct showcasing actions based on the conditions



Conditional Statement



Types of Conditional Statement in Python

```
if Statement  
if...else Statement  
if...elif...else Statement  
Nested if Statement
```

Conditional Statement: Pseudo Code



Pseudo-Code

```
check if arrive_time > 10  
then food = "order"  
else food = "cook"
```

Code

```
if arrive_time > 10:  
    food = "order"  
else:  
    food = "cook"
```

Conditional Statement: Syntax (Single Condition)

Pseudo-Code

```
check if arrive_time > 10  
then food = "order"  
else food = "cook"
```

Syntax

```
if condition:  
    statement 1  
else:  
    statement 2
```


Conditional Statement: Syntax (Multiple Conditions)

If - elif - else statements: Multiple Conditions

Example: Assume a variable x, print "positive" if x is greater than 0, "Zero" if x is equal to 0 or "negative" if x is less than 0.

Pseudo-Code

```
check if x > 0
if yes then print ("positive")

otherwise check if x == 0
if yes then print ("zero")

For every situation just print ("negative")
```

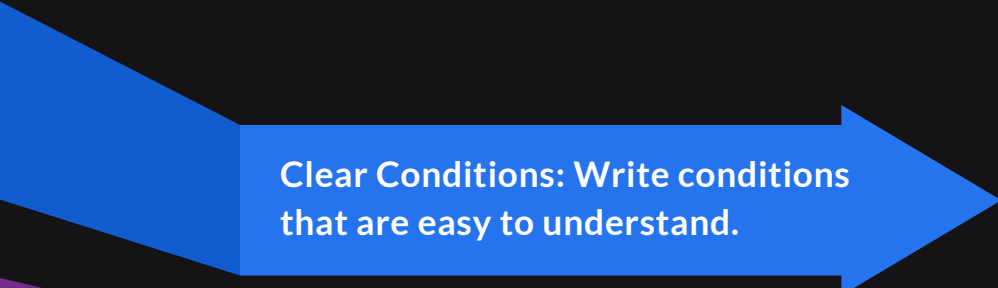
Syntax

```
if x > 0:
    print("Positive")
elif x==0:
    print("Zero")
else:
    print("Negative")
```

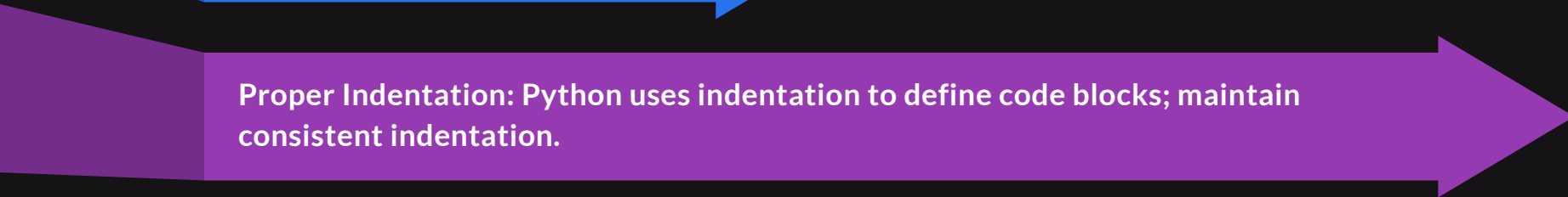
Hands on

Implementing Conditional Statements in Python

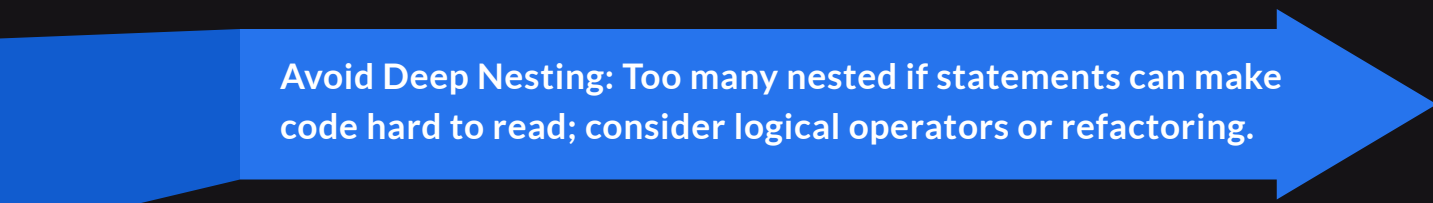
Best Practices with Conditional Statement



Clear Conditions: Write conditions that are easy to understand.



Proper Indentation: Python uses indentation to define code blocks; maintain consistent indentation.



Avoid Deep Nesting: Too many nested if statements can make code hard to read; consider logical operators or refactoring.



Use Comments: Explain complex conditions with comments for better readability.

Thank You
