CoGrammar

Conditional Statements

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Lecture Objectives

1. Define operators and explain their usage in programming.

1. Explore Conditional Statements and control flow

2. Demonstrate and apply basic conditional statement and operator usage in your programs.

Software Engineering Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: **Open Class Questions**

Software Engineering Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Poll:

Assessment

Booleans

- ★ Booleans can only be stored as one of two things: True or False.
- ★ Mainly used for conditional checks.
- ★ Booleans should be declared in Python with capitals. Using lowercase for booleans will return an error in Python.

Comparison Operators

OPERATOR	OPERATION	EXAMPLE
== Equal to	True if **x** has the same value as **y**	x == y # True
!= Not equal to	True if **x** does **NOT** have the same value as **y**	x != y # False
>= greater than or equal to	True if **x** is greater than or equal to **y**	x >= y # True
<= Less than or equal to	True if **x** is less than or equal to **y**	x <= y # True

Logical Operators

OPERATOR	OPERATION	EXAMPLE
and	True if both x AND y are true (logical conjunction)	If x and y: print(z)
or	True if either x OR y are true (**logical disjunction**)	If x or y print(z)
not	True if the opposite of x is true (**logical negation**)	If not x print(y)

and Operator

- ★ Returns as True when both conditions specified are met.
- ★ Example:

```
if 10 < 50 and 500 > 100:
    print("This is a conjunction")
else:
    print("Not a conjunction")
```

or Operator

- ★ Returns True if either of the specified conditions are met.
- ★ Example:

```
if 10 < 50 or 500 > 100:
    print("This is a disjunction")
else:
    print("Not a disjunction")
```

not Operator

- ★ Changes the condition from True to False and vice versa.
- ★ Example:

```
if not 100 < 500:
    print("This is negation")
else:
    print("Not negation")
```

Question:

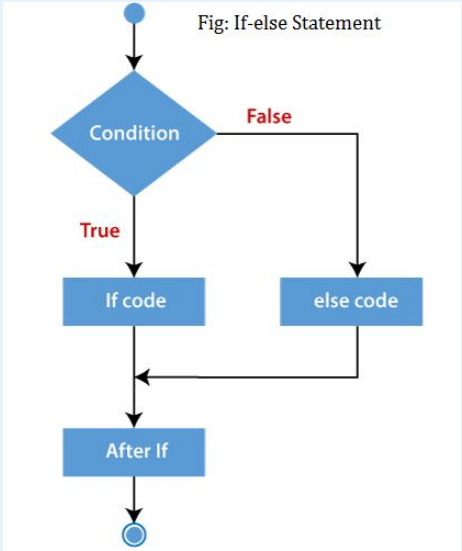
How are comparison operators different from logical operators?

Poll:

Assessment

Control Structures

- ★ Control structures are code that will analyse variables and then choose a direction(control flow) to follow based on the input provided.
- ★ Think of it as a form of branching: depending on the provided input, your program will have one of x branches to follow.
- ★ e.g. “If I finish my work early, I will go to bed. Else, I will have to work through the night”.



if Statements

- ★ An If statement is a fundamental control structure used to make decisions in your code.
- ★ It allows you to execute a block of code only if a certain condition is true.
- ★ Note: Remember to carefully structure and format your if statements according to Python's syntax rules to avoid errors and ensure the intended logic of your code.

Elif Statements

- ★ What if there is a situation where we could have multiple statements that are True?
- ★ This is where elif comes into play: Else if → elif
- ★ Elif statements are mainly used to handle the case when multiple True statements are present.
- ★ Note that you can have multiple elif statements in an if-else block.

Else Statements

- Else is a keyword that is often used in conjunction with the if statement to define a block of code that should be executed when the specified condition in the if statement is not true.
- It allows your program to take different paths or actions based on whether a certain condition is met or not.
- its corresponding code block is executed only when none of the previous conditions is true.

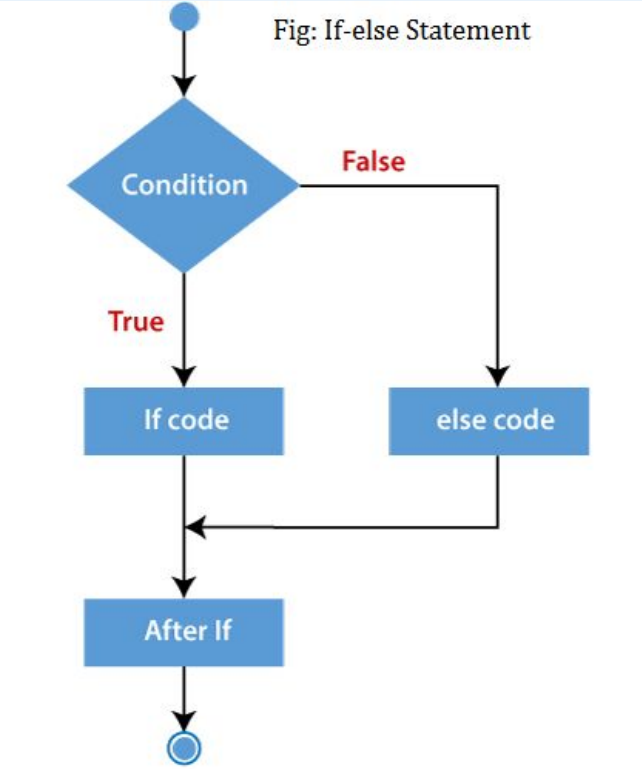


Fig: If-else Statement
Condition
False
True
If code
else code
After If

Few Things To Note

- ★ There is no limit to the number of elif statements one could have in an if-else block.
- ★ Only one final else statement is allowed.
- ★ Each condition is checked in order.
- ★ If one condition is True, that branch executes, and the statement ends.
- ★ Even if there are multiple True conditions, only the first True branch will execute.

Question:

Can you have more than one “else” statement in a conditional structure ?

Poll:

Assessment

Best Practices

Wrapping Up

Control Structures

Programming Constructs that manage the flow of execution in a programme

If-else statements

Allows the execution of a block of code based on specified condition

CoGrammar

Progression Criteria

✅ Criterion 1: Initial Requirements

- Complete 15 hours of Guided Learning Hours and the first four tasks within two weeks.

✅ Criterion 2: Mid-Course Progress

- Software Engineering: Finish 14 tasks by week 8.
- Data Science: Finish 13 tasks by week 8.

✅ Criterion 3: Post-Course Progress

- Complete all mandatory tasks by 24th March 2024.
- Record an Invitation to Interview within 4 weeks of course completion, or by 30th March 2024.
- Achieve 112 GLH by 24th March 2024.

✅ Criterion 4: Employability

- Record a Final Job Outcome within 12 weeks of graduation, or by 23rd September 2024.

CoGrammar

Thank you for joining