

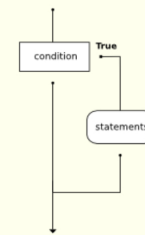


8.7. Omitting the `else` Clause: Unary Selection



Another form of the `if` statement is one in which the `else` clause is omitted entirely. This creates what is sometimes called **unary selection**. In this case, when the condition evaluates to `True`, the statements are executed. Otherwise the flow of execution continues to the statement after the body of the `if`.

Flowchart of an if with no else



Save & Run

5/9/2021, 11:39:21 AM - 2 of 2

Show in CodeLens

```
1 x = 10
2 if x < 0:
3     print("The negative number ", x, " is not valid here.")
4 print("This is always printed")
5
```

This is always printed

Activity: 1 -- ActiveCode (ac7_7_1)

What would be printed if the value of `x` is negative? Try it.

Check your understanding

condition-7-1: What does the following code print?

```
x = -10
if x < 0:
    print("The negative number ", x, " is not valid here.")
print("This is always printed")
```

- a.
This **is** always printed
- b.
The negative number -10 **is not** valid here
This **is** always printed
- c.
The negative number -10 **is not** valid here

- ☐ A. Output a
- ☒ B. Output b
- ☐ C. Output c
- ☐ D. It will cause an error because every if must have an else clause.

Check me

Compare me

✓ Python executes the body of the if-block as well as the statement that follows the if-block.

condition-7-2: Will the following code cause an error?

```
x = -10
if x < 0:
    print("The negative number ", x, " is not valid here.")
else:
    print(x, " is a positive number")
else:
    print("This is always printed")
```

- ☐ A. No
☒ B. Yes

[Check me](#)[Compare me](#)

✔ This will cause an error because the second else-block is not attached to a corresponding if-block.

Activity: 3 -- Multiple Choice (question7_7_2)

You have attempted 4 of 4 activities on this page

✔ Completed. Well Done!

[8.6. Conditional Execution: Binary Selection">](#)[Conditional Execution: Binary Selection">](#)[8.8. Nested conditionals">](#)