8.6. Conditional Execution: Binary Selection

In order to write useful programs, we almost always need the ability to check conditions and change the behavior of the program accordingly. **Selection statements**, sometimes also referred to as **conditional statements**, give us this ability. The simplest form of selection is the **if statement**. This is sometimes referred to as **binary selection** since there are two possible paths of execution.

Save & Run 5/9/2021, 11:04:18 AM - 2 of 2 Show in CodeLens

```
1 x = 15
2
3 if x % 2 == 0:
4     print(x, "is even")
5 else:
6     print(x, "is odd")
7
```

```
15 is odd
```

Activity: 1 -- ActiveCode (ac7_6_1)

The syntax for an `if` statement looks like this:

```
if BOOLEAN EXPRESSION:
    STATEMENTS_1        # executed if condition evaluates to True
else:
    STATEMENTS_2        # executed if condition evaluates to False
```

The boolean expression after the `if` statement is called the **condition**. If it is true, then the indented statements get executed. If not, then the statements indented under the `else` clause get executed.

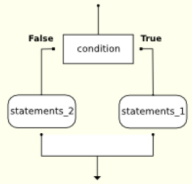
As with the function definition from the last chapter and other compound statements like `for`, the `if` statement consists of a header line and a body. The header line begins with the keyword `if` followed by a *boolean expression* and ends with a colon (:).

The indented statements that follow are called a **block**. The first unindented statement marks the end of the block.

Each of the statements inside the first block of statements is executed in order if the boolean expression evaluates to `True`. The entire first block of statements is skipped if the boolean expression evaluates to `False`, and instead all the statements under the `else` clause are executed.

There is no limit on the number of statements that can appear under the two clauses of an `if` statement, but there has to be at least one statement in each block.

Flowchart of a if statement with an else



Check your understanding

condition-6-1: How many lines of code can appear in the indented code block below the if and else lines in a conditional?

- A. Just one.
- B. Zero or more.
- C. One or more.
- D. One or more, and each must contain the same number.

Check me Compare me

✔ Yes, a block must contain at least one statement and can have many statements.

Activity: 2 -- Multiple Choice (question7_6_1)

condition-6-2: What does the following code print? (choose from output a, b, c or nothing)

```
if (4 + 5 == 10):
    print("TRUE")
else:
    print("FALSE")
```

- A. TRUE
- B. FALSE
- C. TRUE on one line and FALSE on the next
- D. Nothing will be printed

Check me | Compare me

✔ Since 4+5==10 evaluates to False, Python will skip over the if block and execute the statement in the else block.

Activity: 3 -- Multiple Choice (question7_6_2)

condition-6-3: What does the following code print?

```
if (4 + 5 == 10):
    print("TRUE")
else:
    print("FALSE")
print("TRUE")
```

```
a. TRUE

b.
   TRUE
   FALSE

c.
   FALSE
   TRUE
d.
   TRUE
   FALSE
   TRUE
```

- A. Output a
- B. Output b
- C. Output c
- D. Output d

Check me | Compare me

✔ Python will print FALSE from within the else-block (because 5+4 does not equal 10), and then print TRUE after the if-else statement completes.

Activity: 4 -- Multiple Choice (question7_6_3)

Write code to assign the string `"You can apply to SI!"` to `output` *if* the string `"SI 106"` is in the list `courses`. If it is not in `courses`, assign the value `"Take SI 106!"` to the variable `output`.

Save & Run | 5/13/2021, 5:41:16 PM - 4 of 4 | Show in CodeLens

```
courses = ["ENGR 101", "SI 110", "ENG 125", "SI 106", "CHEM 130"]

if "SI 106" in courses:
    output = "You can apply to SI!"
else:
    output = "Take SI 106!"
```

Activity: 5 -- ActiveCode (ac7_6_2)

Result	Actual Value	Expected Value	Notes	
Pass	'You c...o SI!'	'You c...o SI!'	Testing that output has the correct value, given the courses list provided	Expand Differences
Pass	'if'	'cours...106!'"	Testing output (Don't worry about actual and expected values).	Expand Differences

You passed: 100.0% of the tests

Create a variable, `b`, and assign it the value of `15`. Then, write code to see if the value `b` is greater than that of `a`. If it is, `a`'s value should be multiplied by 2. If the value of `b` is less than or equal to `a`, nothing should happen. Finally, create variable `c` and assign it the value of the sum of `a` and `b`.

Save & Run 5/13/2021, 5:42:00 PM - 4 of 4 Show in CodeLens

```
a = 20
b = 15
if a < b:
    a = a * 2
    print(a)
else:
    print(a)
c = a + b

```

```
20
```

Activity: 6 -- ActiveCode (ac7_6_4)

Result	Actual Value	Expected Value	Notes
Pass	20	20	Testing that a has the correct value.
Pass	35	35	Testing that c has the correct value.

You passed: 100.0% of the tests

You have attempted 8 of 7 activities on this page

8.5. Precedence of Operators">
cedence of Operators">
8.7. Omitting the else Clause: Unary Selection">
Completed. Well Done!

| Back to top