Set 7: Conditionals Part 1

Skill 7.1: Predict the output of an *if* statement
Skill 7.2: Write a Python *if* statement
Skill 7.3: Predict the output of an *if-else* statement
Skill 7.4: Write a Python *if-else* statement

Skill 7.1: Predict the output of an *if* statement

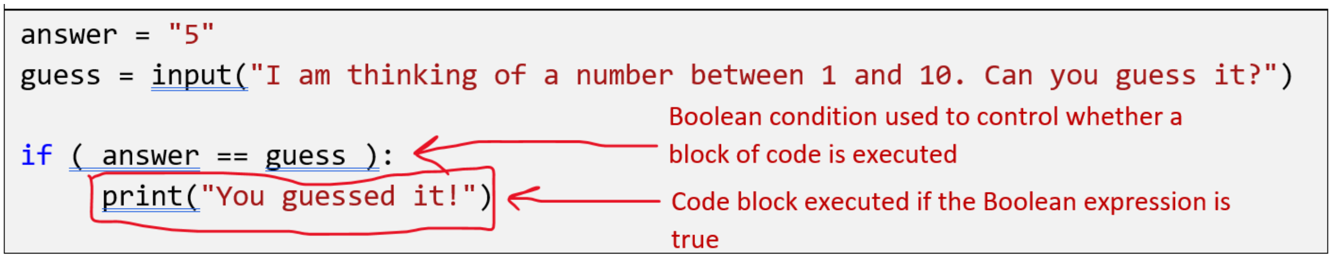
Skill 7.1 Concepts

When we consider our real-time scenario every day, we make some decisions, and based on the decisions made we will take further actions. Hence all our daily life activities depend on the decisions we make.

A similar situation arises in the programming language as well where we must make some decisions and based on that the program will execute.

Python provides four conditional statements, the most basic being the *if* statement. The *if* statement decides whether certain statements need to be executed or not. It checks for a given condition, if the condition is true, then the set of code present inside the *if* block will be executed otherwise not.

The if condition evaluates a Boolean expression and executes the block of code only when the Boolean expression becomes TRUE. Below is an example,



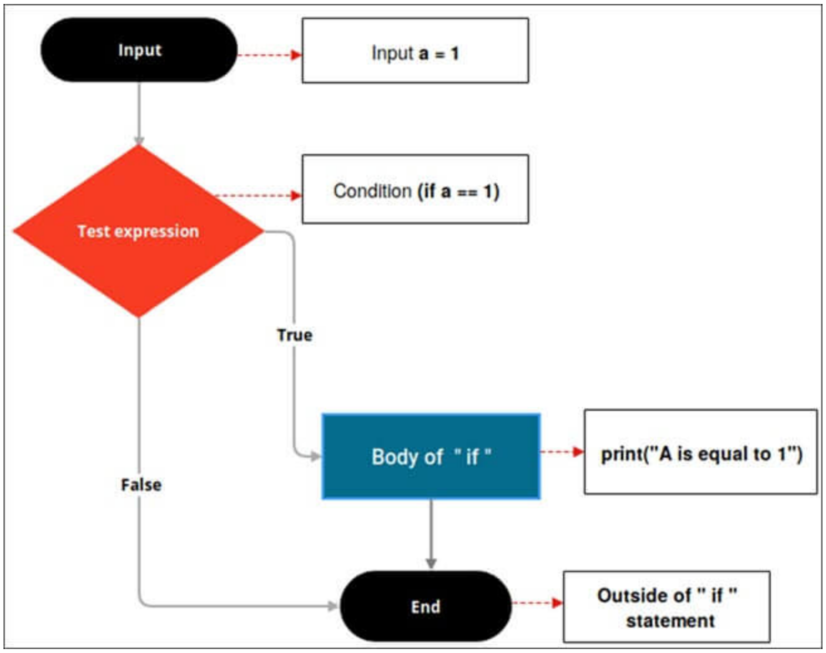
```
answer = "5"
guess = input("I am thinking of a number between 1 and 10. Can you guess it?")

if ( answer == guess ):
    print("You guessed it!")
```

Boolean condition used to control whether a block of code is executed

Code block executed if the Boolean expression is true

Let’s see how this looks with a flow chart,



If you observe the above flow-chart, first the controller will come to an if condition and evaluate the condition if it is true, then the statements will be executed, otherwise the code present outside the block will be executed.

Skill 7.1 Exercise 1

Skill 7.2: Write a Python *if* stateemnt

Skill 7.2 Concepts

Notice in the example below, the conditional (b > a) is followed by a colon (:). Whatever code you write after the colon operator will be part of the "if block". Also, notice that the code following the colon is indented. Python relies on indentation (whitespace at the beginning of a line) to define scope in the code. Other programming languages often use curly-brackets for this purpose.

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

An *if* statement, without indentation (will raise an error):

```
a = 33
b = 200
if b > a:
print("b is greater than a")
```

Skill 7.2 Exercise 1

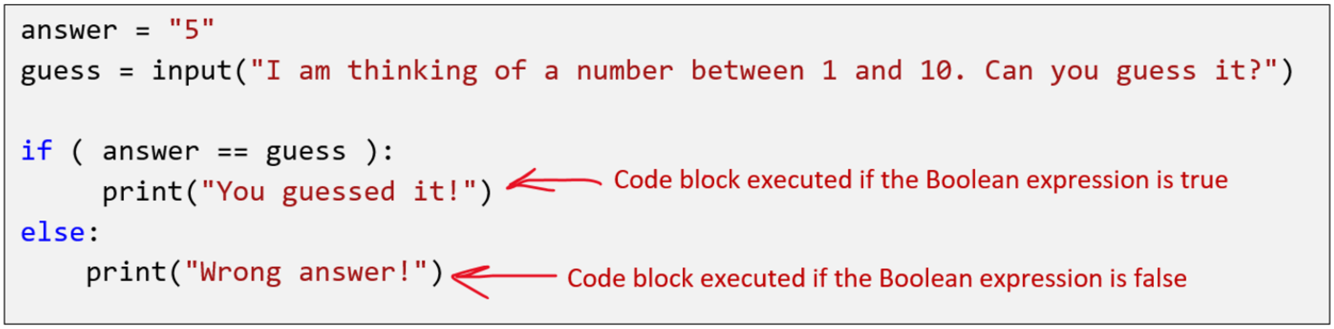
© Pluska

Skill 7.3: Predict the output of an *if-else* statement

Skill 7.3 Concepts

An *if-else* statement says if a given condition is true then execute the statements present inside the “if block” and if the condition is false then execute the “else” block.

An *if-else* statement evaluates the Boolean expression. If the condition is TRUE then, the code present in the “if” block will be executed otherwise the code of the “else“ block will be executed



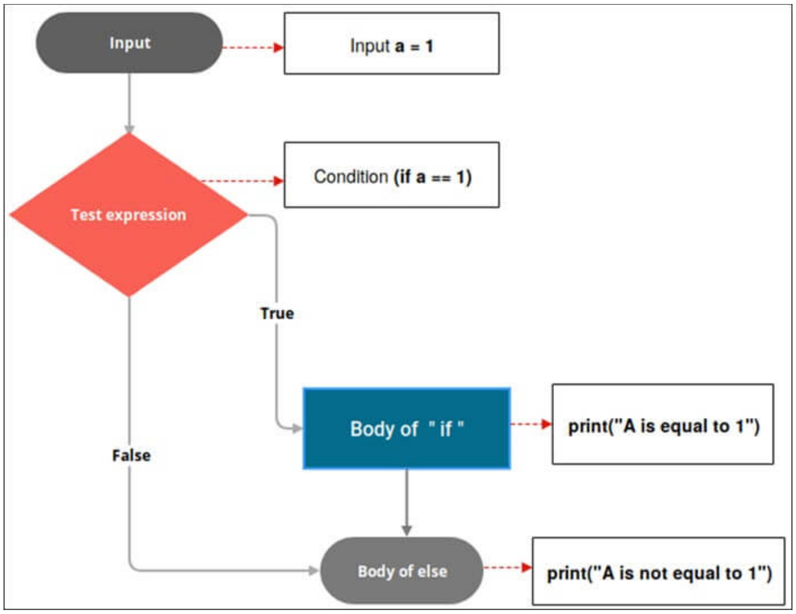
```
answer = "5"
guess = input("I am thinking of a number between 1 and 10. Can you guess it?")

if ( answer == guess ):
     print("You guessed it!")
else:
    print("Wrong answer!")
```

Code block executed if the Boolean expression is true

Code block executed if the Boolean expression is false

Let’s see the flowchart of if-else.



In the above code, the controller will come to the *if* condition and evaluate the condition. If it is true the *if* block will be executed otherwise the *else* block will be executed.

Skill 7.3 Exercise 1

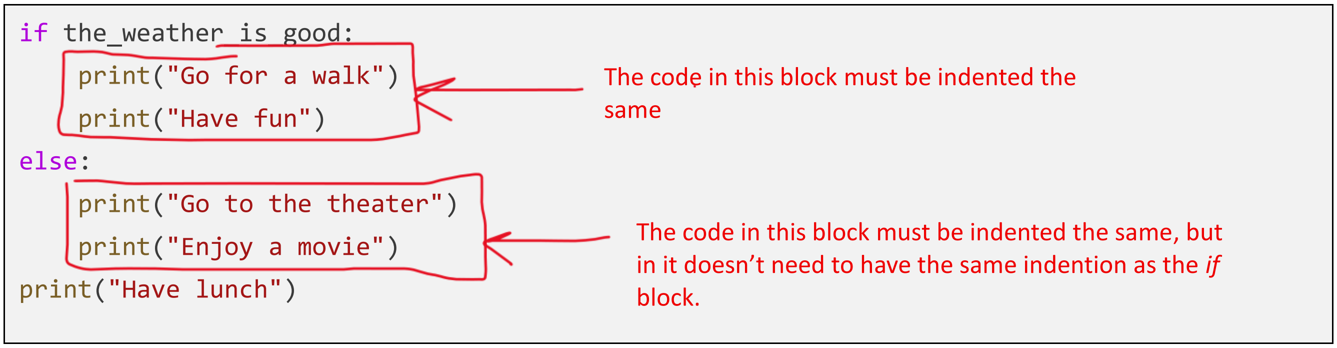
© Pluska

Skill 7.4: Write a Python *if-else* statement

Skill 7.4 Concepts

Just as we saw before with simple if statements, the conditional in all the examples above is followed by a colon (:). Also, all the code written after the colon operator is indented and is part of the "if block".   Python recognizes that the if block is done once each reaches the else statement, which is not indented. The *else* is followed by a colon and all the code associated with the else block is indented.   Some things to be aware of

- All the code within a block must be indented the same
- The code within the if and else blocks do not need to use the same indention. Although, doing so makes your code more readable



```
if the_weather_is_good:
    print("Go for a walk")
    print("Have fun")
else:
    print("Go to the theater")
    print("Enjoy a movie")
print("Have lunch")
```

The code in this block must be indented the same

The code in this block must be indented the same, but in it doesn't need to have the same indention as the *if* block.

Skill 7.4 Exercise 1

© Pluska