Python Programming

Lesson 4

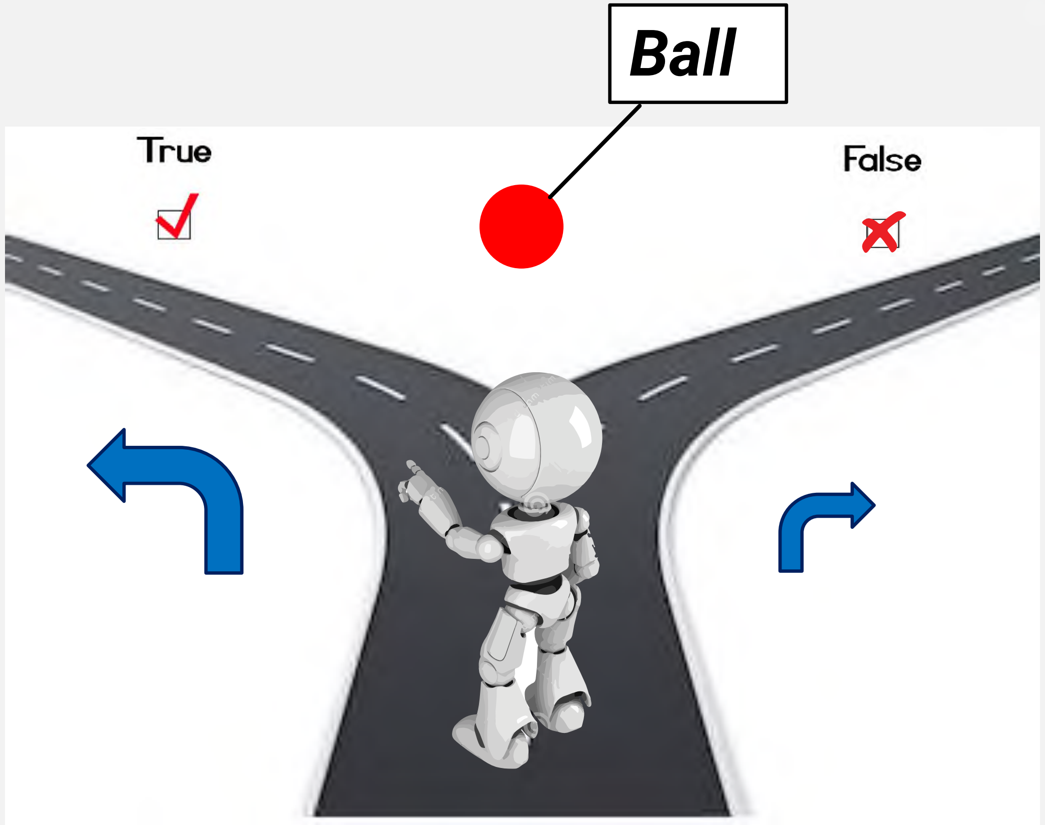
If this,Then that.

Presented by Advaspire Team

Boolean

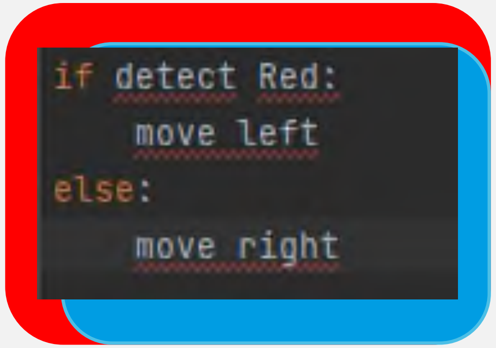
Boolean is a special type of data that can only have the values of (true) or (false).

In some language and in electronics, you may see these values represented as 0 and 1.

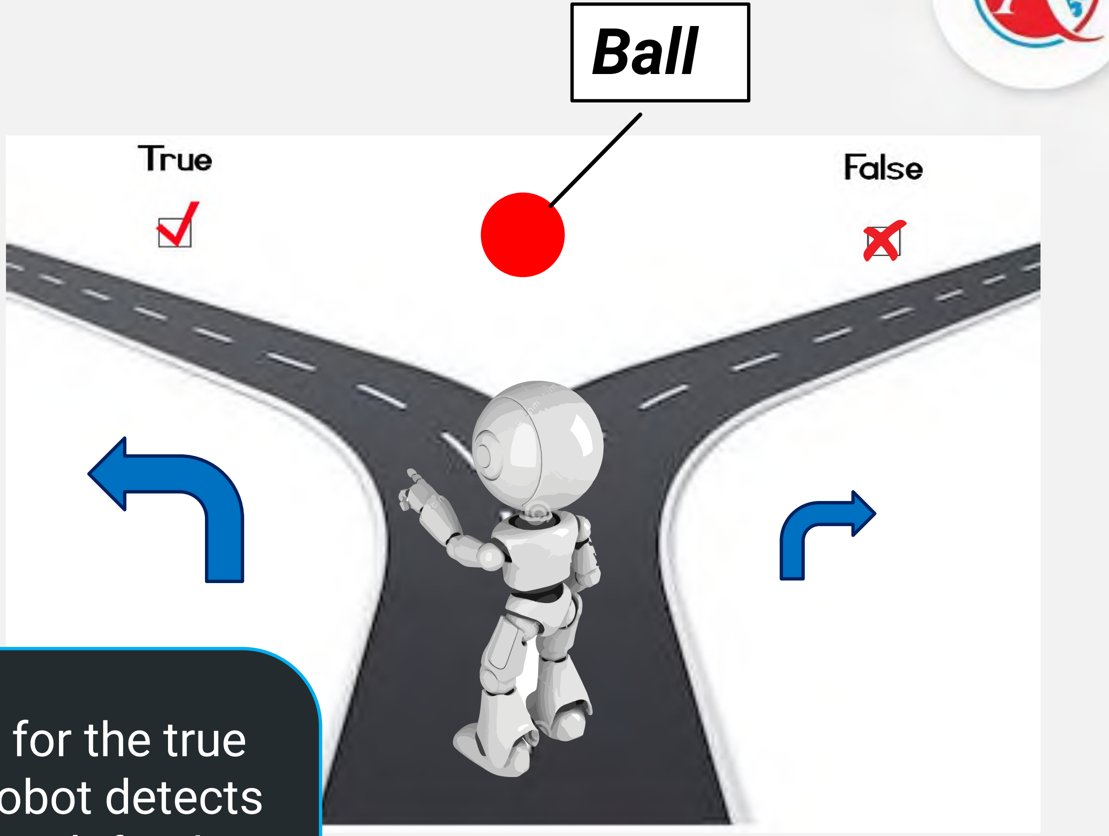


Boolean Explained

Based on the robot image, lets say we programmed it to only move left if it detects the color red. If it did not, then it will move right.



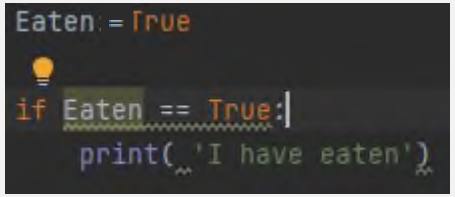
```
if detect Red:
    move left
else:
    move right
```



The if in the code is for the true condition so if the robot detects red color, then it move left. The else is for false condition. So if the robot can't detect any red color, it will move right.

How it is used.

Booleans are used by the keywords : if and while. When using an if statement, the code needs to be in indented position. Usually, we will set a variable first that will contain the condition whether its true or false.



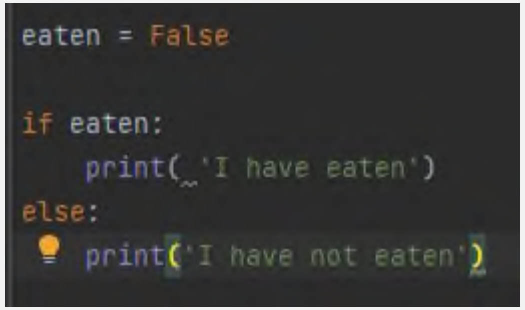
```
Eaten = True

if Eaten == True:
    print( 'I have eaten')
```

In the example above, I have set a variable called Eaten that have a true condition. Then in the if statement, if it is true then it will print the strings. However, the way that I write here is a bit redundant, because if statement will always test if the Boolean is true.

Proper Way to Use the If Statement

It is better if you write the code as in this example. So what it will do is that the program will first check the if statement. Does it match the Boolean in the variable? If it match, it will proceed to print out the strings.



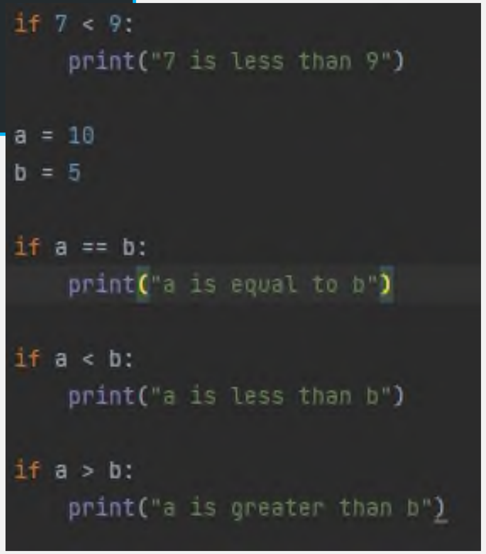
```
eaten = False

if eaten:
    print( 'I have eaten')
else:
    print('I have not eaten')
```

But if it doesn’t match, it will go for the else clause if you have it. If you don’t put an else clause, it will not print anything. So anything that is false will usually be in the else clause.

Comparison Operator

Comparison operators take two numbers, strings or other variables, compare them, and then return a Boolean from them.



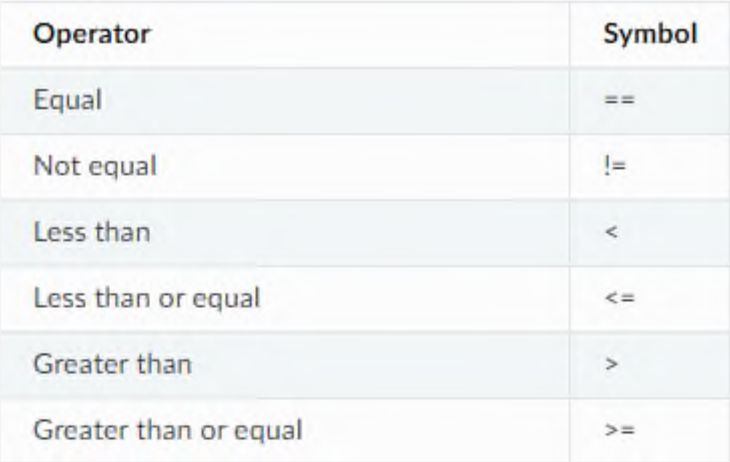
```
if 7 < 9:
    print("7 is less than 9")

a = 10
b = 5

if a == b:
    print("a is equal to b")

if a < b:
    print("a is less than b")

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



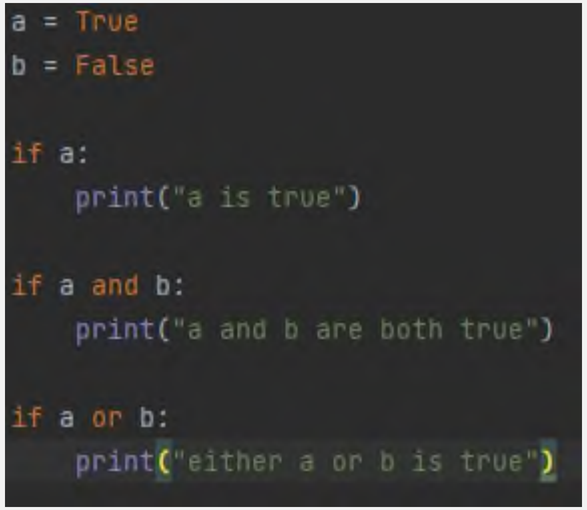
Operator	Symbol
Equal	==
Not equal	!=
Less than	<
Less than or equal	<=
Greater than	>
Greater than or equal	>=

In the example here, there are 2 variable with a number inside. All the statements may or may not print itself, depending on whether it matches the Boolean. When using the == or !=, you can use normal string and not only numbers.

Boolean Logic

Boolean logic involves the use of these operators : and, or, not. Each will have different outcome.

Try out this example first. Once you have tested it out, switch the value for a and b. The operators will be explained after this.



```
a = True
b = False

if a:
    print("a is true")

if a and b:
    print("a and b are both true")

if a or b:
    print("either a or b is true")
```

Or Operator

Boolean operators combine two truth values together. The or operator is true if either of its data is true.

The program here checks whether people are allowed to ride the rollercoaster. The condition to ride is only people older than 12 or taller than 150 cm are allowed. Try make the code shorter by using the or.

```
print("How old are you?")
age = int(input())
print("How tall are you?")
height = int(input())
if age > 12:
    print("You can ride")
elif height > 150:
    print("You can ride")
else:
    print("YOU MAY NOT RIDE, GO AWAY!")
```

Teachers Note

Let the student try out the previous code and figure out how to shorten it up. The code on the right will be the answer.

```
print("How old are you?")
age = int(input())
print("How tall are you?")
height = int(input())
if age > 12 or height > 150:
    print("You can ride")
else:
    print("YOU MAY NOT RIDE, GO AWAY!")
```

And Operator

The and operator is true if both of its data is true.

Try out this code. Then change the value of b to true and run it again.

```
a = True
b = False

print (a and b)
```

Rollercoaster

```
print("How old are you?")
age = int(input())
print("How tall are you?")
height = int(input())
if age > 12 or height > 150:
    print("You can ride")
else:
    print("YOU MAY NOT RIDE, GO AWAY!")
```

Using the code from the Or exercise, we are going to add another condition. The rollercoaster is only allowed to run on days when the temperature is less than 30 degrees. Make the code so it will print out the You can ride only when both condition is true.

Solution + Explanation

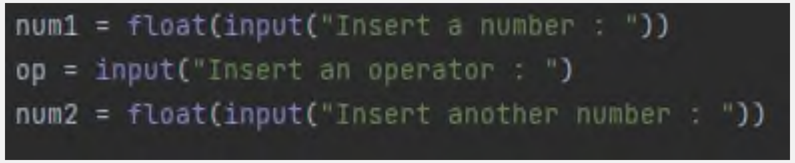
```
print("How old are you?")
age = int(input())
print("How tall are you?")
height = int(input())
print("What is the temperature?")
temp = int(input())
if (age > 12 or height > 150) and temp < 30:
    print("You can ride")
else:
    print("YOU MAY NOT RIDE, GO AWAY!")
```

Note that we have put brackets around the or expression. This ensures it is calculated first and the result of that calculation is then used in the and expression. So when both condition is true, only then the rollercoaster can run.

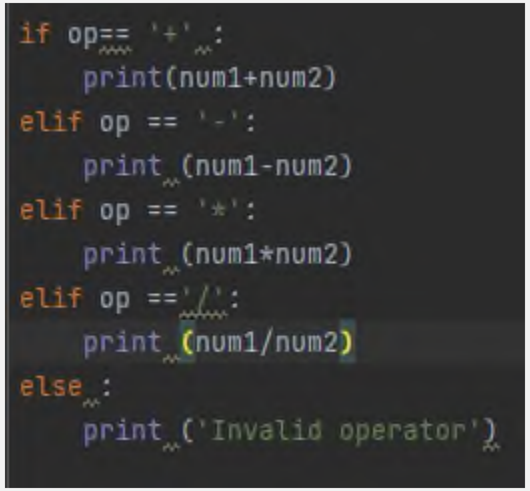
Create an Upgraded Calculator

Now that we have learned how to use the If statement, we are going to create another calculator, but this time it can accept more than one operator without changing the code.

We will need to create 3 variable which will accept user inputs. Then we need to create several if True statements that will have the math operator.



```
num1 = float(input("Insert a number : "))
op = input("Insert an operator : ")
num2 = float(input("Insert another number : "))
```



```
if op== '+' :
    print(num1+num2)
elif op == '-':
    print (num1-num2)
elif op == '*':
    print (num1*num2)
elif op =='/':
    print (num1/num2)
else :
    print ('Invalid operator')
```

You can direct message your teacher and ask your question through **Slack Robotene Community** or arrange a **One-to-One Consultation** with your teacher.

Q&A Any Questions?

Thank you ;)