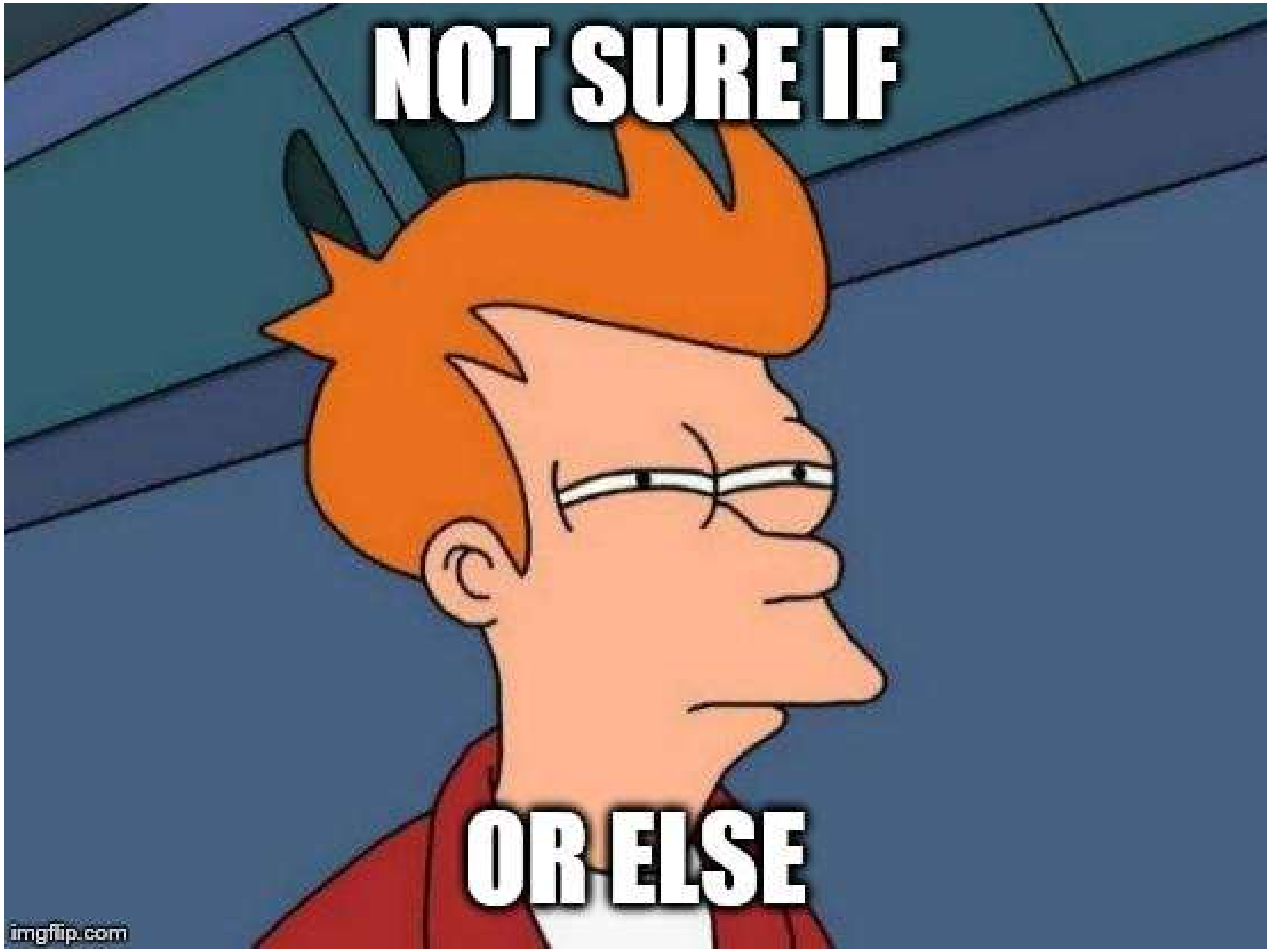
# The if elif else statements

Harry Potter is the first one to understand the python language.

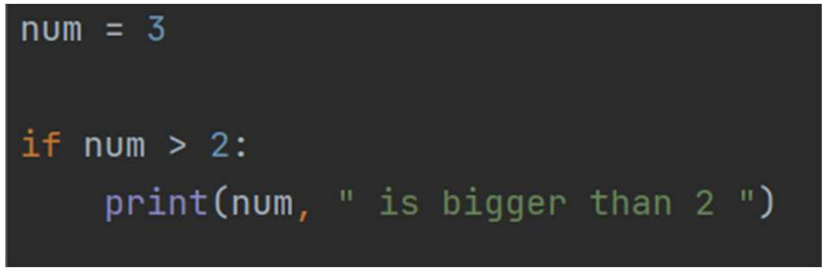
## If and else Statements

Instead of repeating a given process, computers can make decisions with if and else statements.

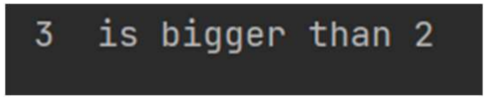
```
if (Condition):
        Process this section of the code if Condition is Ture
else:
        Process this section of the code if Condition is False
```


NOT SURE IF
OR ELSE
imgflip.com

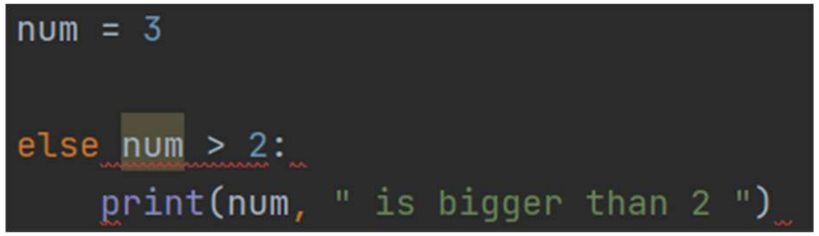
- else is optional, but else must be preceed by an if statement



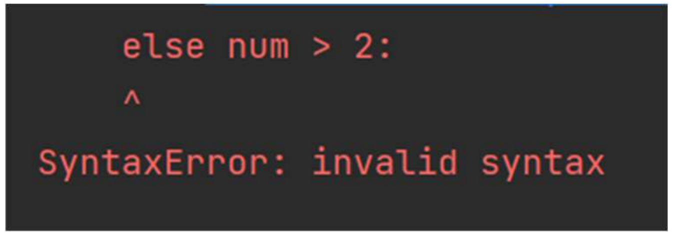
```
num = 3

if num > 2:
    print(num, " is bigger than 2 ")
```



```
 3  is bigger than 2
```

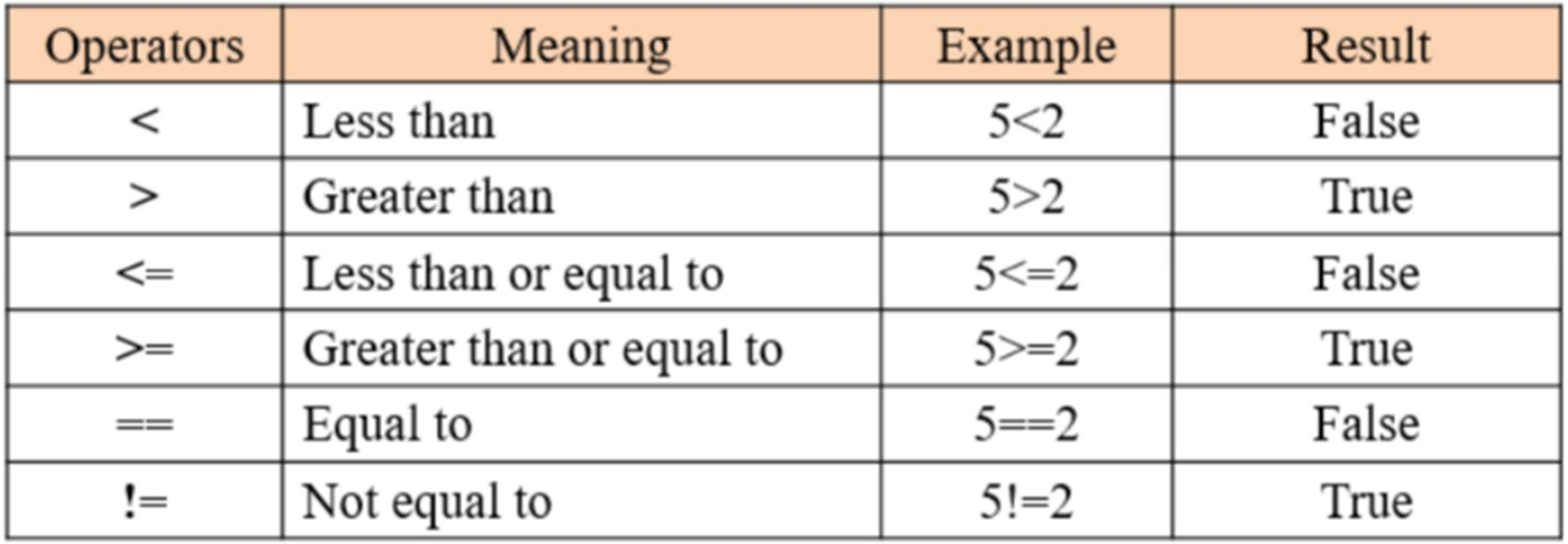


```
num = 3

else num > 2:
    print(num, " is bigger than 2 ")
```



```
      else num > 2:
      ^
SyntaxError: invalid syntax
```

## Numeric Comparison

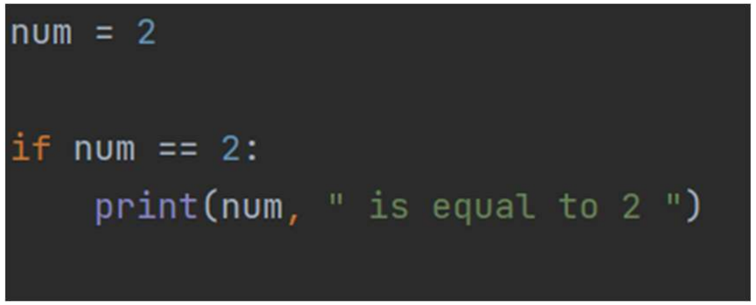


| Operators | Meaning | Example | Result |
|---|---|---|---|
| < | Less than | 5<2 | False |
| > | Greater than | 5>2 | True |
| <= | Less than or equal to | 5<=2 | False |
| >= | Greater than or equal to | 5>=2 | True |
| == | Equal to | 5==2 | False |
| != | Not equal to | 5!=2 | True |

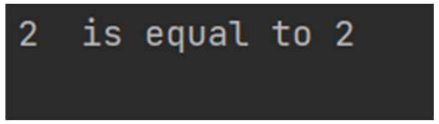
❖Note that the compound operators consisting of more than one symbol do not have spaces in between the symbols involved.

## !!!!

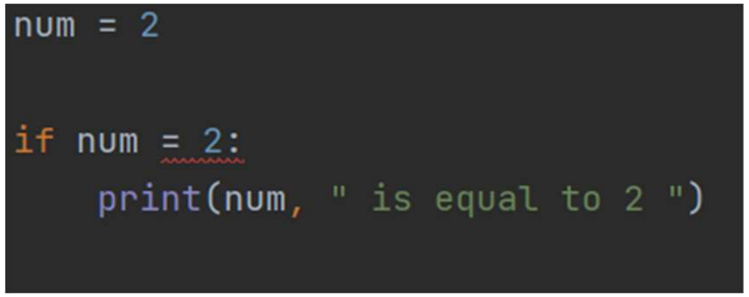
- = and == are not the same



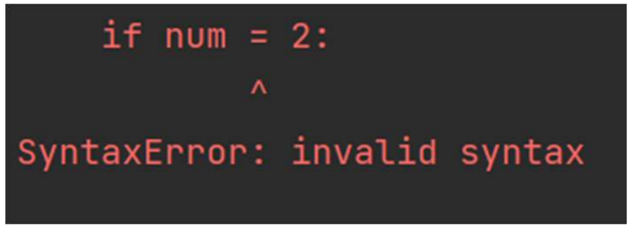
```
num = 2

if num == 2:
    print(num, " is equal to 2 ")
```



```
2  is equal to 2
```

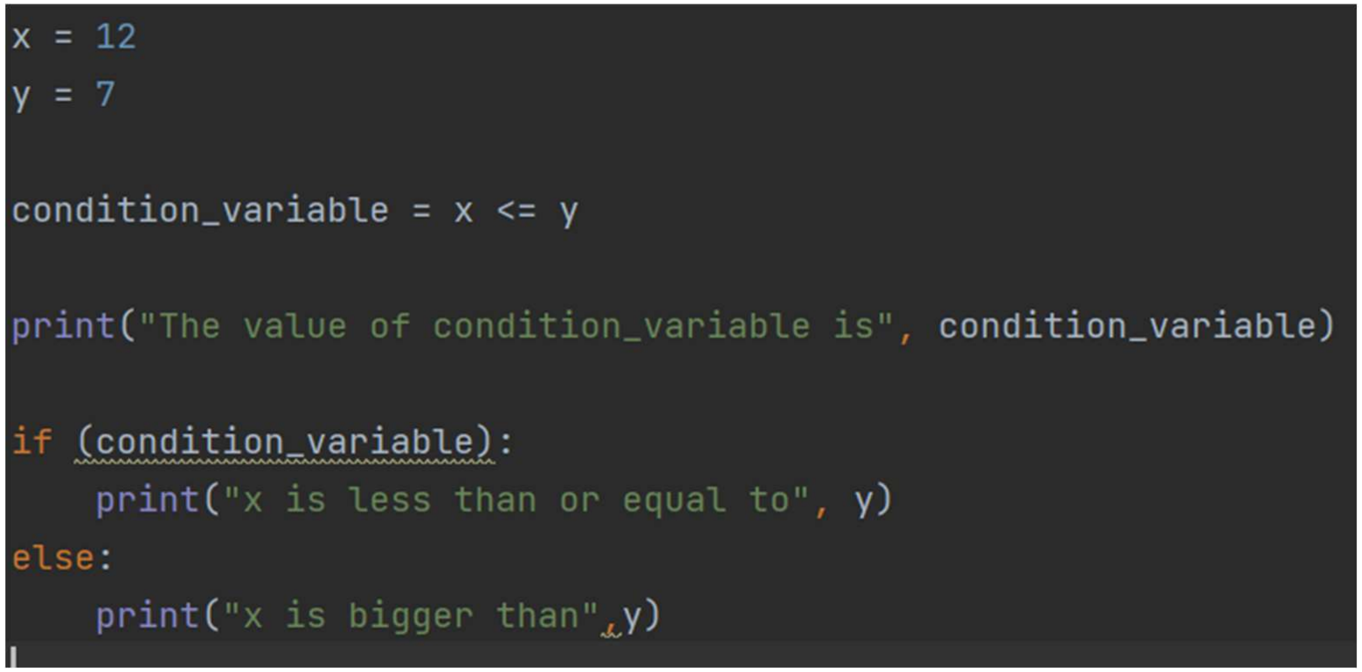


```
num = 2

if num = 2:
    print(num, " is equal to 2 ")
```

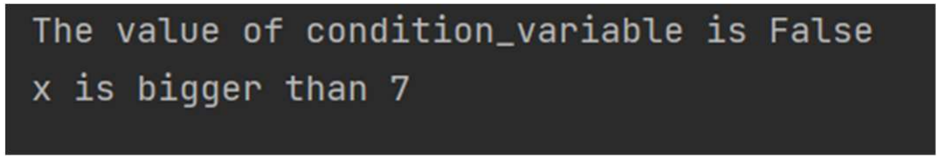


```
    if num = 2:
           ^
SyntaxError: invalid syntax
```

## The result of the comparison expression can be stored as True or False



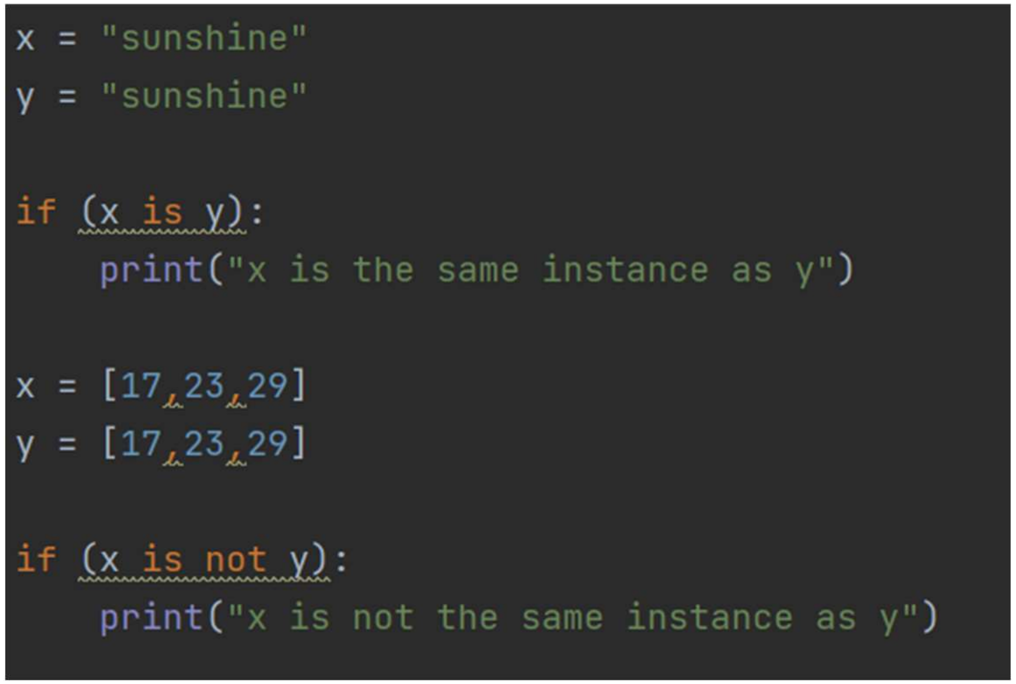
```
x = 12
y = 7

condition_variable = x <= y

print("The value of condition_variable is", condition_variable)

if (condition_variable):
    print("x is less than or equal to", y)
else:
    print("x is bigger than",y)
```



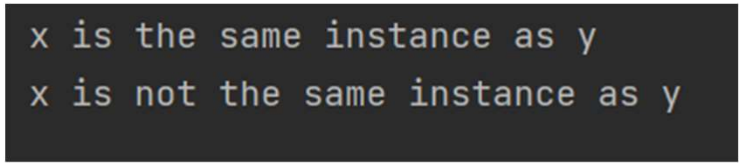
```
The value of condition_variable is False
x is bigger than 7
```

## is Operator

- Two data type instance can be compared to each other to check whether the two are the same instances in the program



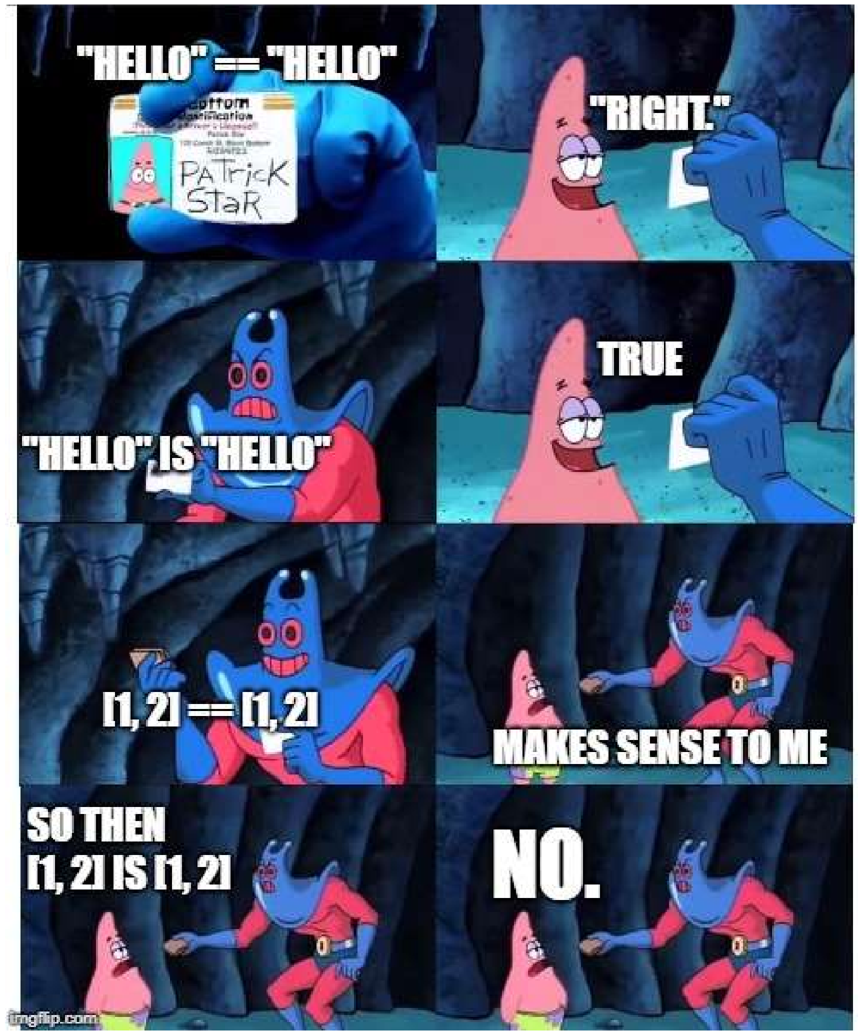
```python
x = "sunshine"
y = "sunshine"

if (x is y):
    print("x is the same instance as y")

x = [17,23,29]
y = [17,23,29]

if (x is not y):
    print("x is not the same instance as y")
```



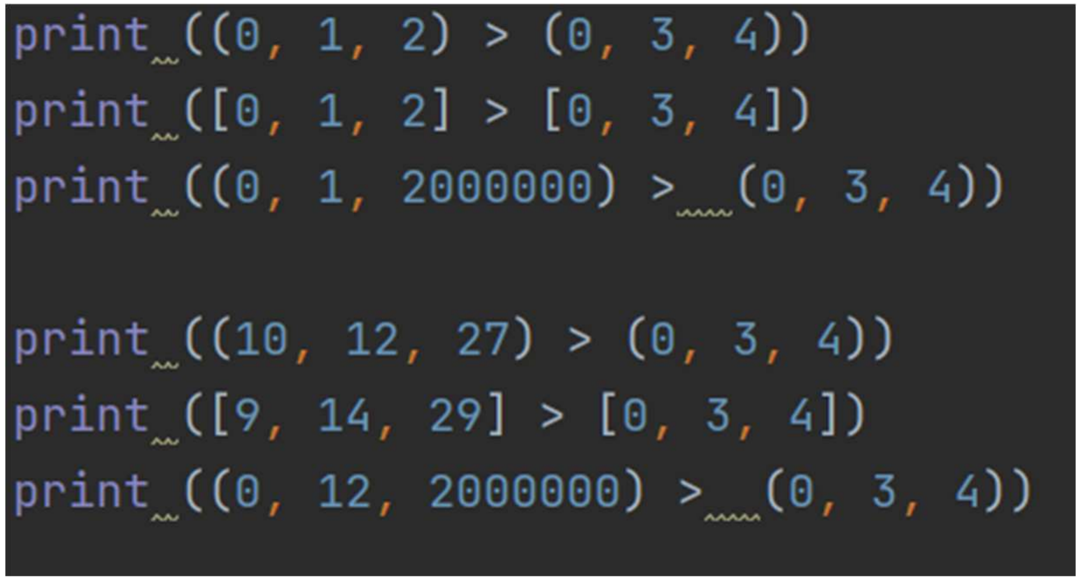
```
x is the same instance as y
x is not the same instance as y
```

- Even though lists contains the same elements they are different list instances

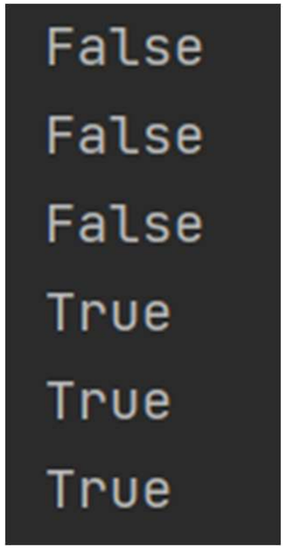
- Strings are mutable data structures so, if two strings have the same sequence of characters they are treated as a single instance


"HELLO" == "HELLO"
PATrick StaR
"RIGHT."
"HELLO" IS "HELLO"
TRUE
[1, 2] == [1, 2]
MAKES SENSE TO ME
SO THEN
[1, 2] IS [1, 2]
NO.
imgflip.com

- Two tuples or lists can be compared using the comparison operators where the result is based on the comparison of respectiive items in the tuples



```
print ((0, 1, 2) > (0, 3, 4))
print ([0, 1, 2] > [0, 3, 4])
print ((0, 1, 2000000) > (0, 3, 4))

print ((10, 12, 27) > (0, 3, 4))
print ([9, 14, 29] > [0, 3, 4])
print ((0, 12, 2000000) > (0, 3, 4))
```



```
False
False
False
True
True
True
```

ELSE
ELIF
IF
What the hell is this?
imgflip.com

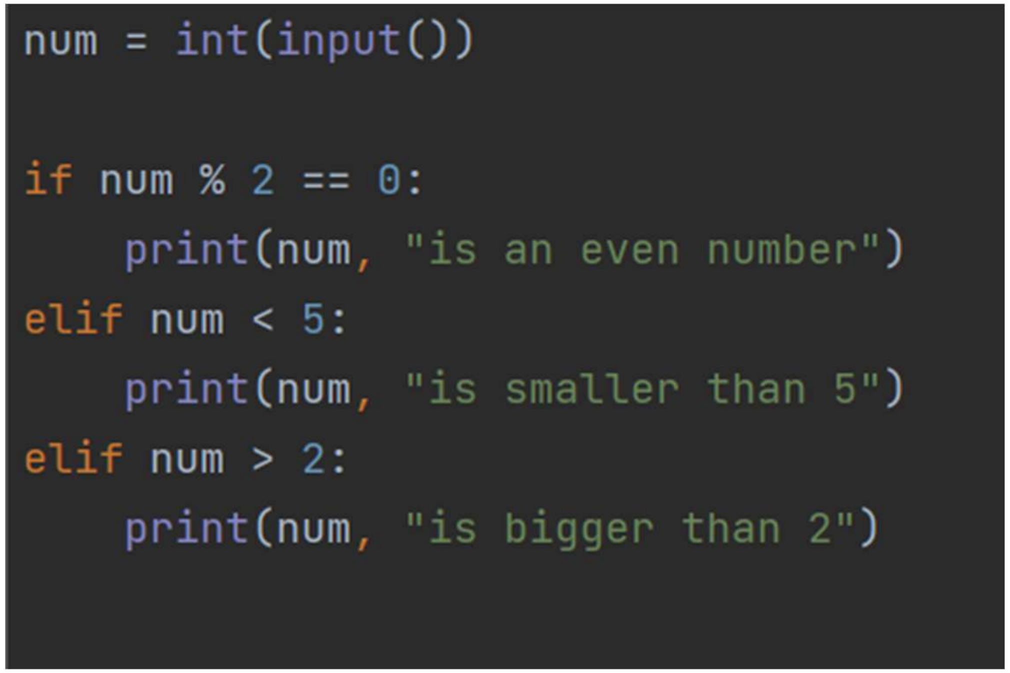
## elif

- When there are more than two condition elif statements can be us
- elif is optional
- elif must be preceded by a single if
- There can be as many elif as necessary
- The else cannot precede any elif in the same conditional control sequence

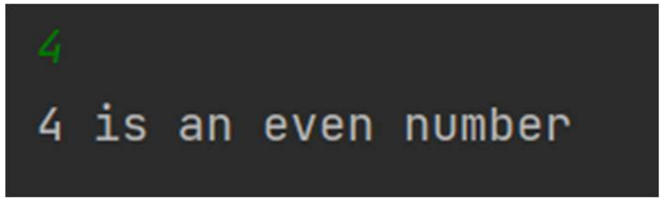
## !!!!

- Else does not specify any condition
- Inside the same conditional sequence, if every condition before the else evaluate to False, else block is executed

The statements are considered in order, if one of the statement is evaluated to True rest of the sequences are skipped over, even though they could also be evaluated to True

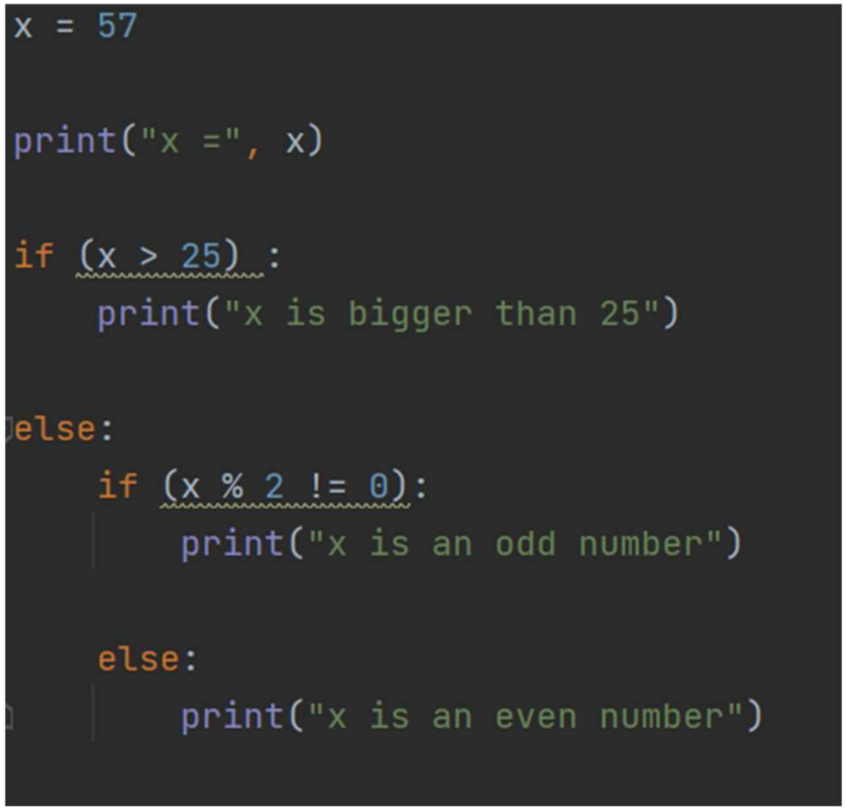


```
num = int(input())

if num % 2 == 0:
    print(num, "is an even number")
elif num < 5:
    print(num, "is smaller than 5")
elif num > 2:
    print(num, "is bigger than 2")
```

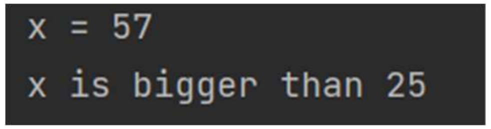


```
4
4 is an even number
```

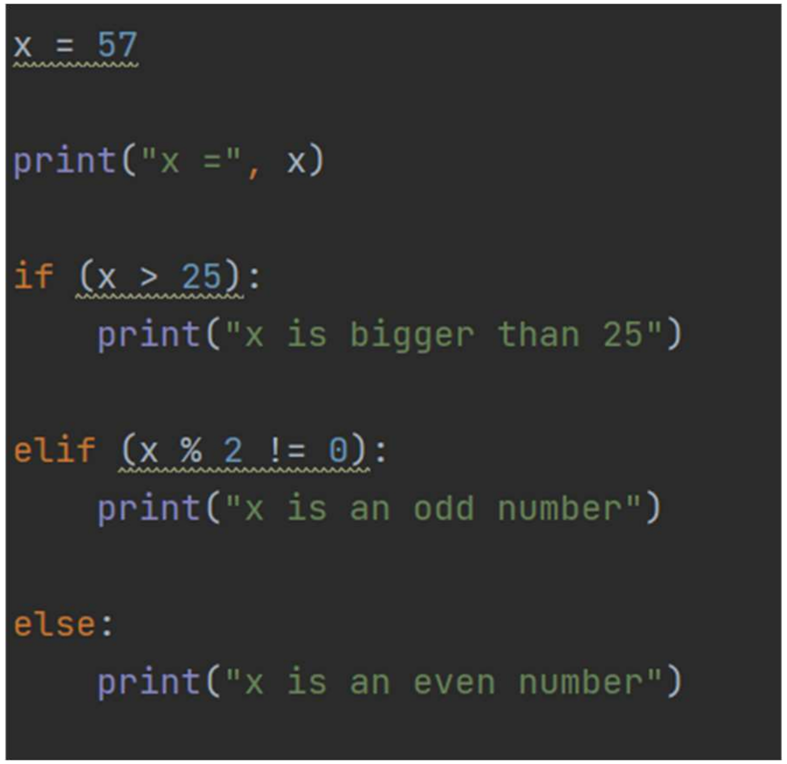
## elif represent the case check as else-if (two codes are functionally equivalent



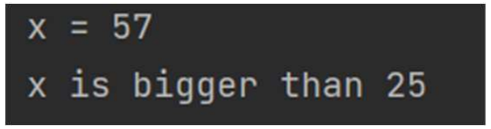
```
x = 57

print("x =", x)

if (x > 25) :
    print("x is bigger than 25")

else:
    if (x % 2 != 0):
        print("x is an odd number")

    else:
        print("x is an even number")
```



```
x = 57
x is bigger than 25
```

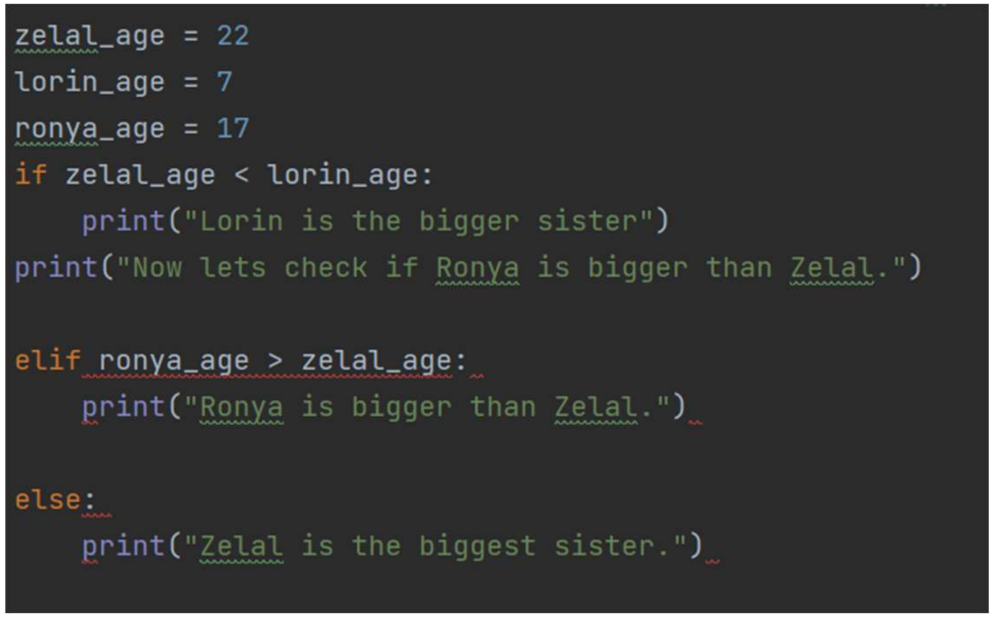


```
x = 57

print("x =", x)

if (x > 25):
    print("x is bigger than 25")

elif (x % 2 != 0):
    print("x is an odd number")

else:
    print("x is an even number")
```

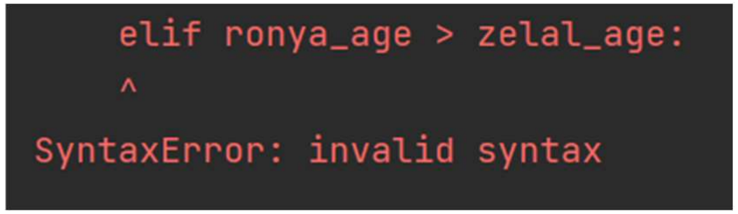


```
x = 57
x is bigger than 25
```

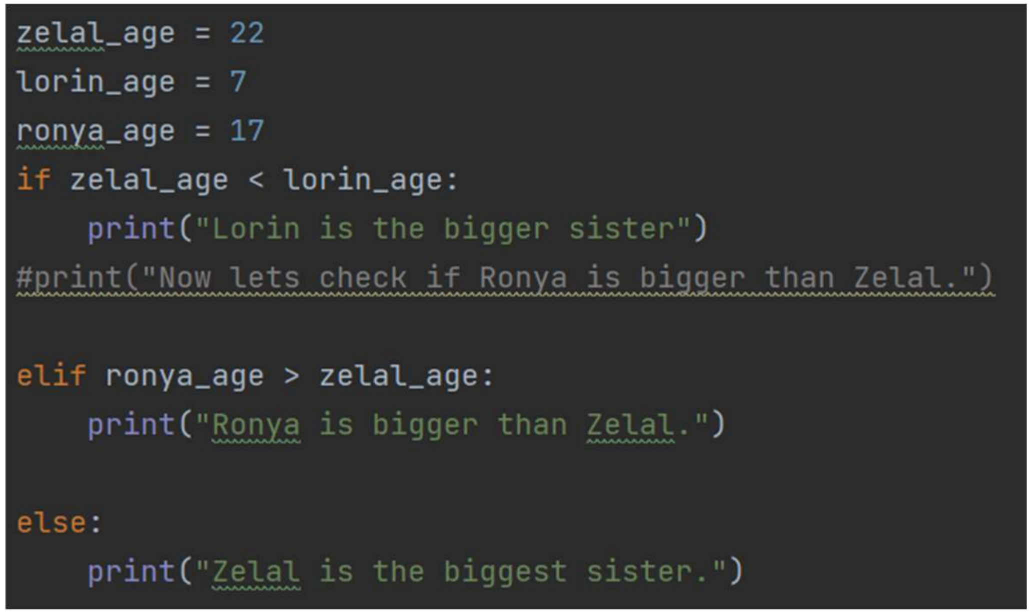
## If a statement is added between the if-elif-else sequence the statements will be detached from each other



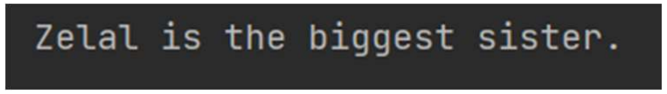
```
zelal_age = 22
lorin_age = 7
ronya_age = 17
if zelal_age < lorin_age:
    print("Lorin is the bigger sister")
print("Now lets check if Ronya is bigger than Zelal.")

elif ronya_age > zelal_age:
    print("Ronya is bigger than Zelal.")

else:
    print("Zelal is the biggest sister.")
```



```
    elif ronya_age > zelal_age:
    ^
SyntaxError: invalid syntax
```



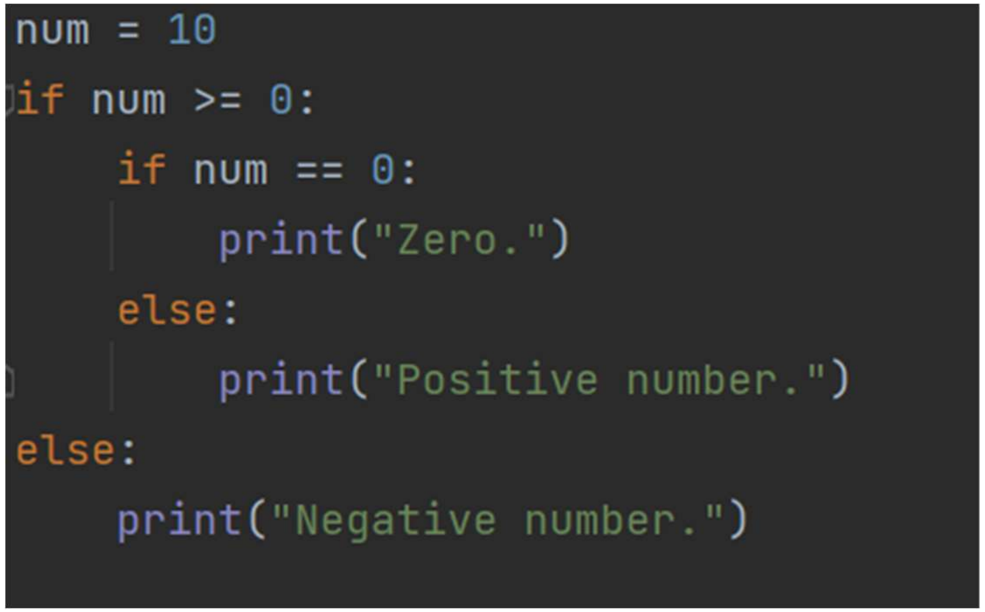
```
zelal_age = 22
lorin_age = 7
ronya_age = 17
if zelal_age < lorin_age:
    print("Lorin is the bigger sister")
#print("Now lets check if Ronya is bigger than Zelal.")

elif ronya_age > zelal_age:
    print("Ronya is bigger than Zelal.")

else:
    print("Zelal is the biggest sister.")
```



```
Zelal is the biggest sister.
```

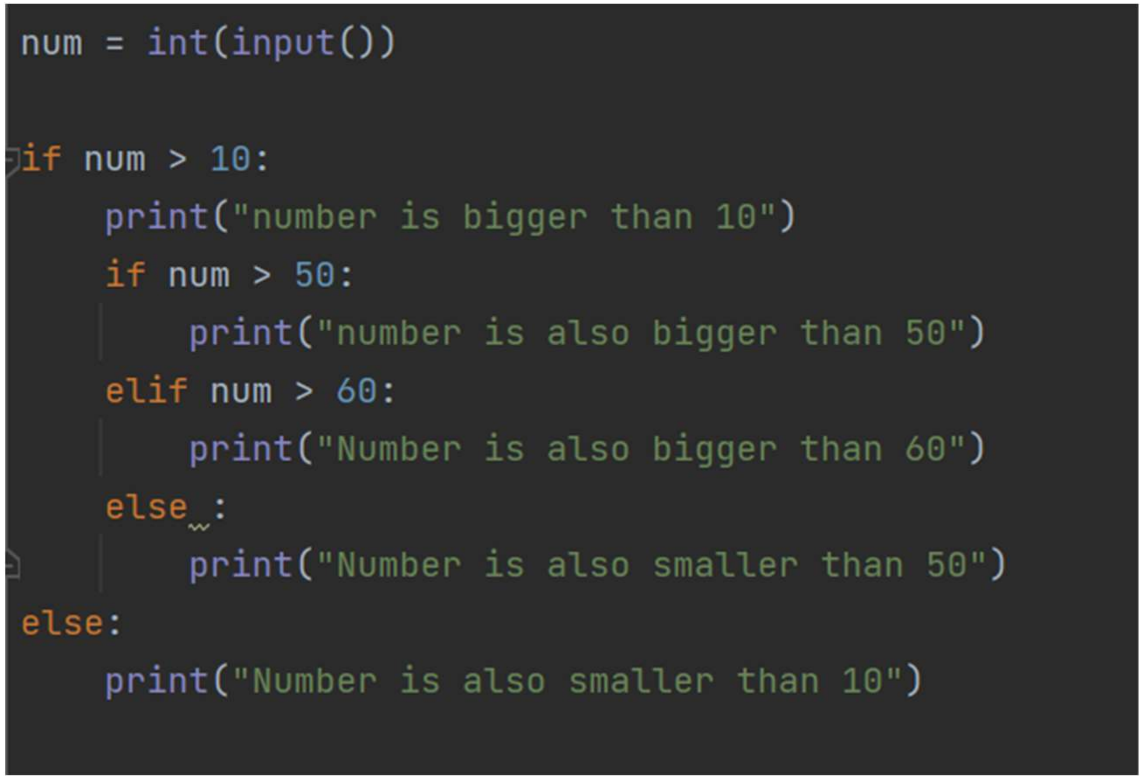
## Nested if Statements

- Conditional statements can be nested inside one another.

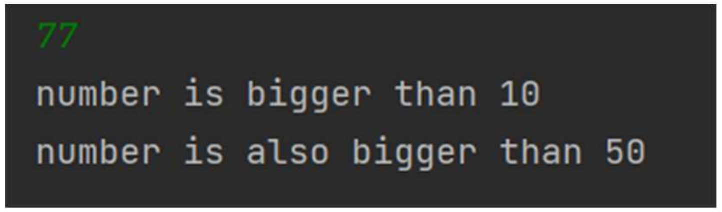


```python
num = 10
if num >= 0:
    if num == 0:
        print("Zero.")
    else:
        print("Positive number.")
else:
    print("Negative number.")
```

Conditional blocks can be nested inside other conditional blocks as well. It is possible to add other if-elif-else blocks inside other if, elif, or else blocks.



```python
num = int(input())

if num > 10:
    print("number is bigger than 10")
    if num > 50:
        print("number is also bigger than 50")
    elif num > 60:
        print("Number is also bigger than 60")
    else :
        print("Number is also smaller than 50")
else:
    print("Number is also smaller than 10")
```

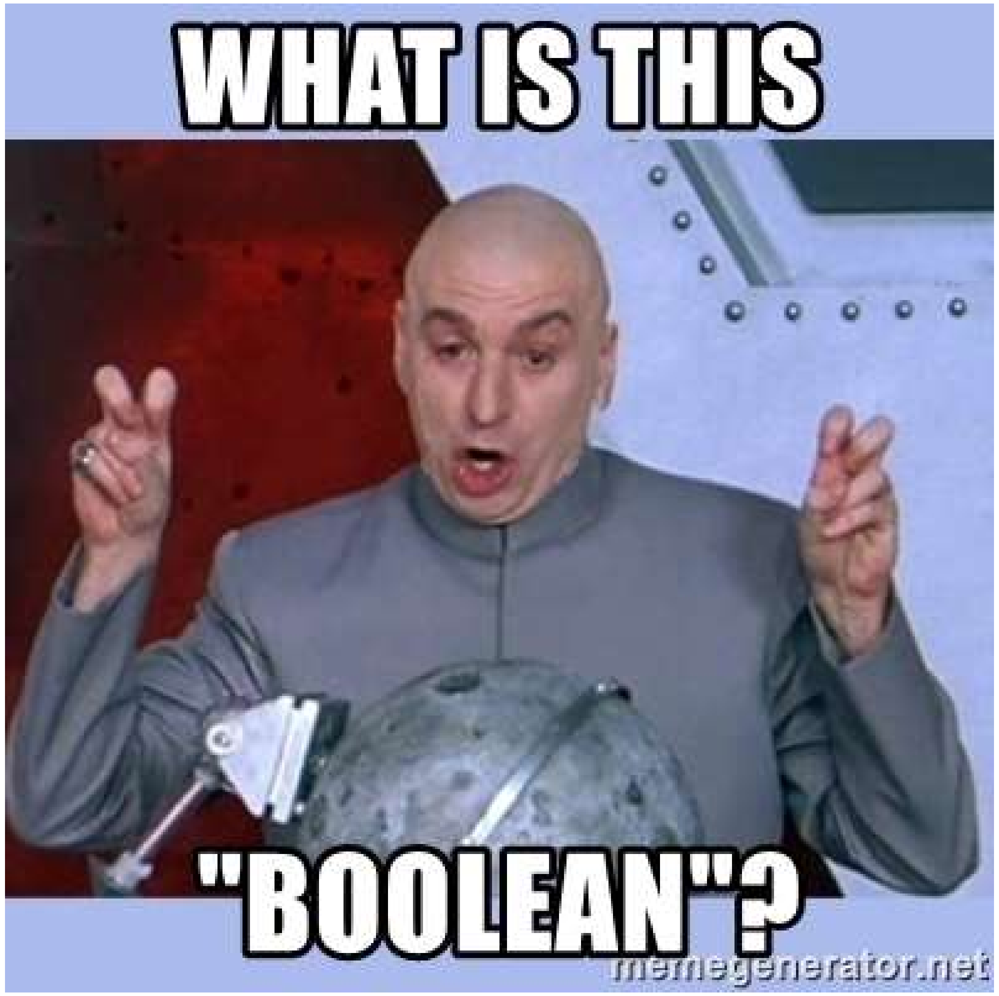


```
77
number is bigger than 10
number is also bigger than 50
```

CONDITIONAL STATMENTS
CONDITIONAL STATEMENTS EVERYWHERE
...negenerator.net

YO DAWG, I HEARD YOU LIKE IF/ELSE STATEMENTS
SO I PUT AN IF/ELSE INTO AN IF/ELSE WRAPPED IN AN IF/ELSE.
memegenerator.net

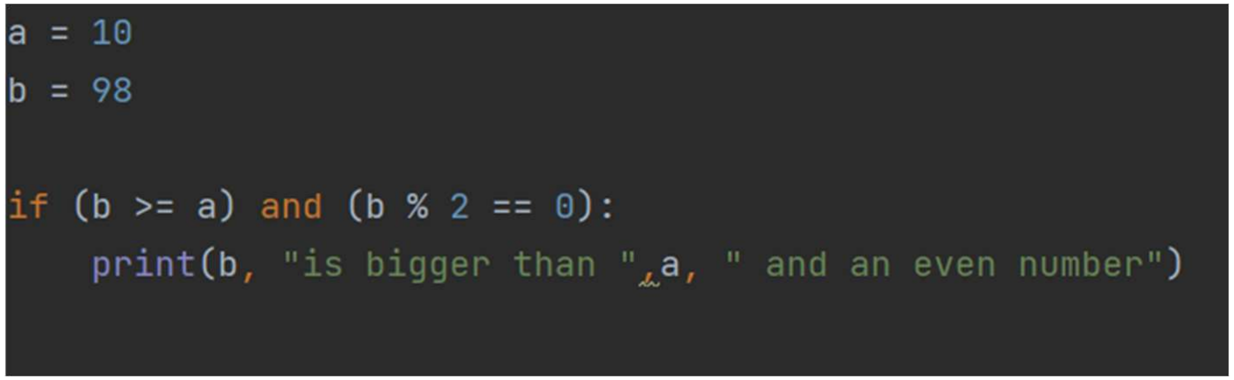
```
if (Condition1):
    execute this code block if Condition1 evaluates to True
    if(Condition1_1_0):
        execute this code block if Condition1 and Condition1_1_0 evaluate to True
    elif(Condition1_1_1):
        execute this code block if Condition1 evaluates to True, Condition1_1_0 evaluates to False, and Condition1_1_1 evaluates to True
    elif(Condition1_1_2):
        execute this code block if Condition1 evaluates to True, Condition1_1_0 and Condition1_1_1 evaluate to False, and Condition1_1_2 evaluates to True
    # not stating an else statement in this section

    # starting a new nested if condition sequence independent of the one started as if(Condition1_1_0):
    if(Condition1_2_0):
        execute this code block if Condition1 and Condition1_2_0 evaluate to True
    elif(Condition1_2_1):
        execute this code block if Condition1 evaluates to True, Condition1_2_0 evaluates to False, and Condition1_2_1 evaluates to True
    else:
        execute this code block if Condition1 evaluates to True and none of the Condition1_2_0 and Condition1_2_1 evaluates to True
elif (Condition2):
    execute this code block if Condition1 evaluates to False and Condition2 evaluates to True
    if(Condition2_1_0):
        execute this code block if Condition1 evaluates to False, Condition2 and Condition2_1_0 evaluate to True
    elif(Condition2_1_1):
        execute this code block if Condition1 evaluates to False, Condition2 evaluates to True, Condition2_1_0 evaluates to False, and Condition2_1_1 evaluates to True
    else:
        execute this code block if Condition1 evaluates to False, Condition2 evaluates to True, and none of the Condition2_1_0 and Condition2_1_1 evaluate to True
elif (Condition3):
    execute this code block if Condition1 and Condition2 evaluate to False and Condition3 evaluates to True
elif (Condition4):
    execute this code block if Condition1, Condition2, and Condition3 evaluate to False and Condition4 evaluates to True
else:
    execute this code block if none of Condition1, Condition2, Condition3, and Condition4 evaluates to True
```


WHAT IS THIS
"BOOLEAN"?
memegenerator.net

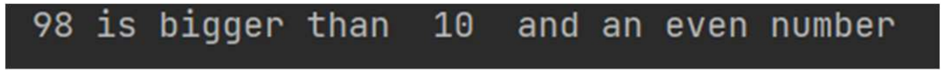
## Boolean Logic

- Conditional statements can be formed in a compound structure by joining multiple statements using or and and keywords. The statements may use different variables.
- The result of a conditional statement can be negated by using the not keyword

## When we are comparing two things at the same time, we use Boolen logic



```python
a = 10
b = 98

if (b >= a) and (b % 2 == 0):
    print(b, "is bigger than ",a, " and an even number")
```



```
98 is bigger than  10  and an even number
```

The order of precedence in boolean statements higher to lower precedence is as follows: "()", "not", "and", and "or".

| **a** | **b** | **a and b** | **a or b** | **not a or b** | **not(a or b)** |
|---|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | TRUE | TRUE |
| FALSE | TRUE | FALSE | TRUE | TRUE | FALSE |
| TRUE | FALSE | FALSE | TRUE | FALSE | FALSE |
| TRUE | TRUE | TRUE | TRUE | TRUE | FALSE |

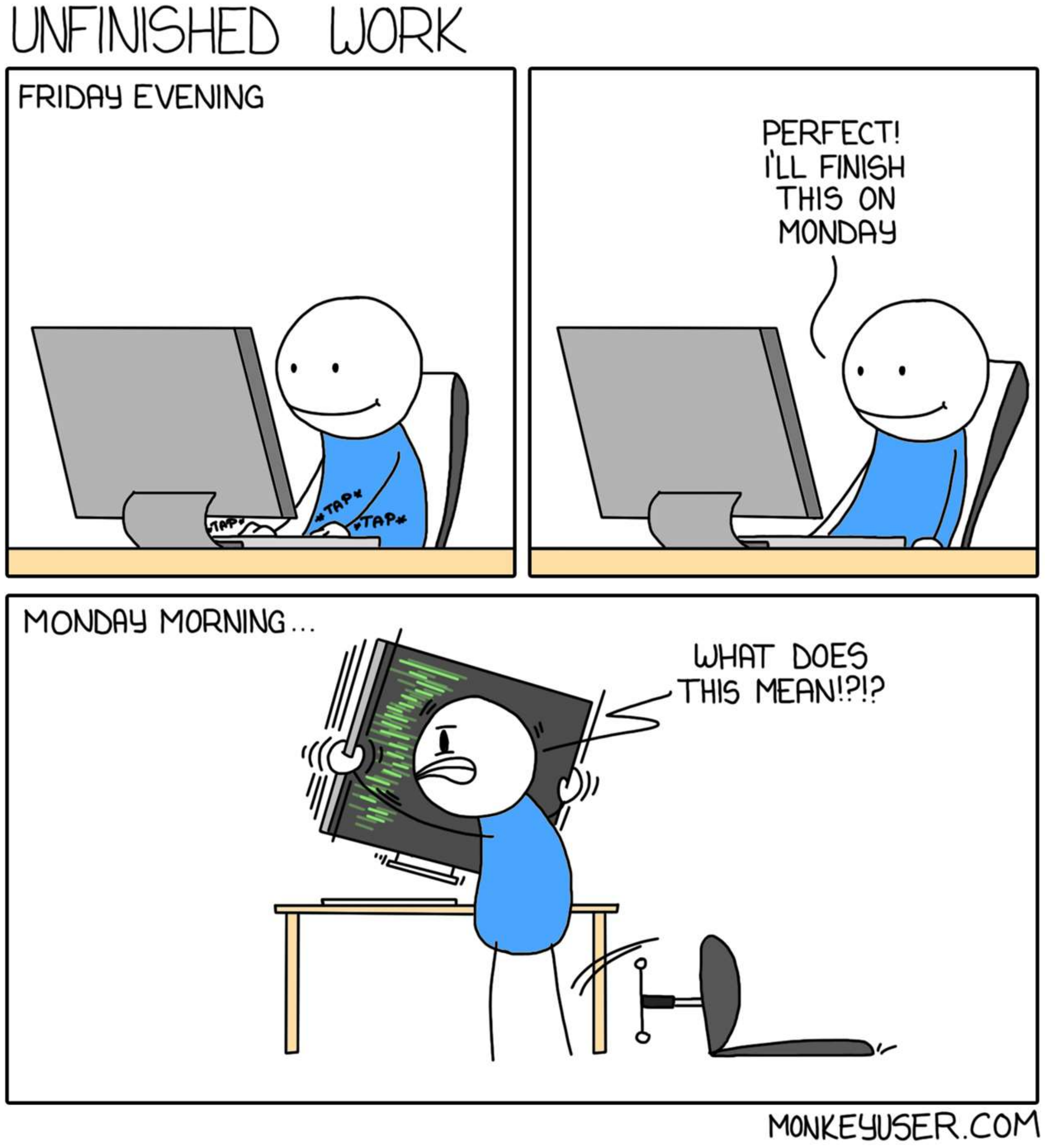
## DeMorgan’s Law

When not is applied to the logical statement DeMorgan’s law applies as

not (a or b) is equal to : not a and not b

and

not (a and b) is equal to : not a or not b


UNFINISHED WORK
FRIDAY EVENING
*TAP*
*TAP*
*TAP*
PERFECT! I'LL FINISH THIS ON MONDAY
MONDAY MORNING...
WHAT DOES THIS MEAN!?!?
MONKEYUSER.COM

## References

- https://www.geeksforgeeks.org/recursive-functions/
- https://bouncmpe150.github.io/python-slides/lab4.html