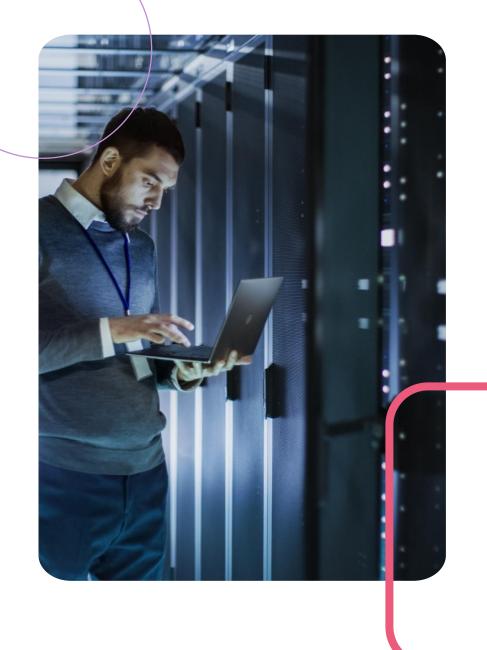
Diploma in

### Python Programming

**Decision Making** 



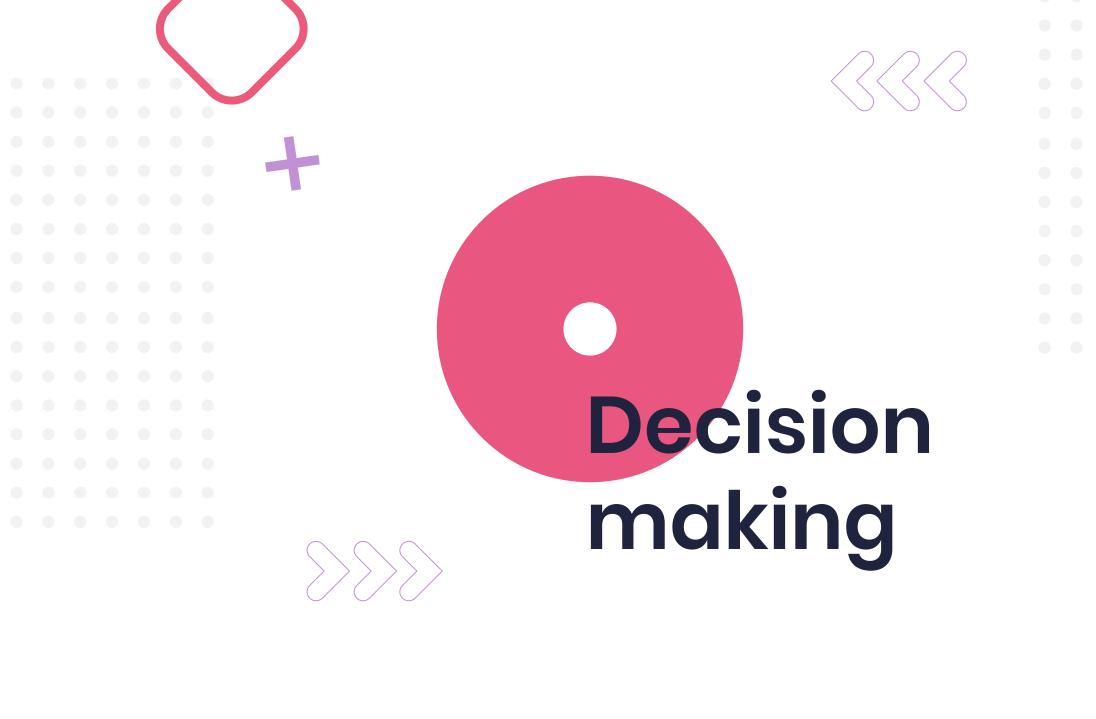


The process of decision making <

Python nested *if* statements

Python decision-making code <

**Objectives** 



### DID YOU

#### **Python**

Decision making is required when we want to execute a code only if a certain condition is satisfied.



>>>
Python *if*statement



#### **If** statement

- A true Boolean datatype, the program will be executed.
- A false Boolean datatype, the program will be not executed, it will be skipped.

if condition:
 statement(s)



#### **If** statement

- The condition expression in the if statement is replaced by a true Boolean data type.
- The statement(s) inside the code block will be executed by the Python interpreter.

if True:
 statement(s)

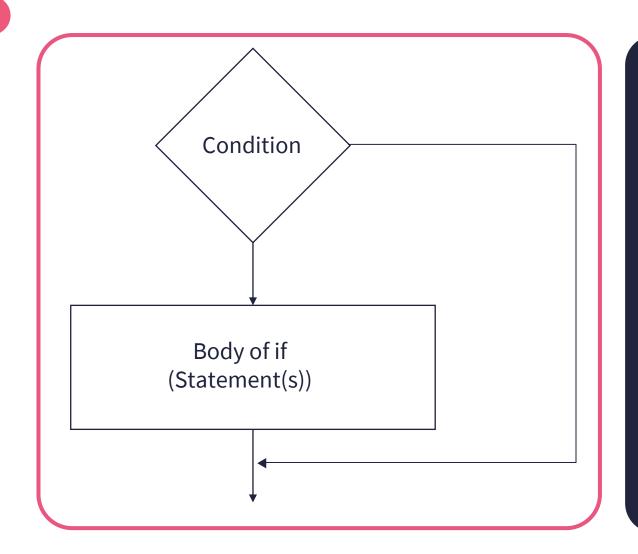


#### **If** statement

The statement(s) inside the code block will not be executed by the Python interpreter.

if False:





Python If statement

#### Code displaying If statement

- Created a variable named *lucky\_number*.
- Followed by an *if* statement.
- Body statement print.
- Print statement will be executed.

```
lucky_number=5
if lucky_number>0:
    print('Happy')
```

```
r.get('/register', function(req, res, next) {
 er.post('/register', function(req, res, next) {
f (req.body.email &&
 reg.body.name &&
 req.body.favoriteBook &&
 reg.body.confirmPassword) {
    if (reg.body.password !== reg.body.confirmPassword) {
       var err = new Error('Passwords do not match.');
      var userData = {
         email: req.body.email,
          favoriteBook: req.body.favoriteBook,
                  ate(userData, function (error, user) {
                 error) {
                                                                                                    LF UTF-8 JavaScript 🖹 0 files
```

# Python comparison operator

- 4 > 2 which makes this a True Boolean data type.
- 4 > 7 this proves to be a false Boolean data type.

```
lucky_number=5
if lucky_number>0:
    print('Happy')
Happy
```



- If the variable *lucky\_number* is assigned a value less than 0.
- The body print statement will not be executed.

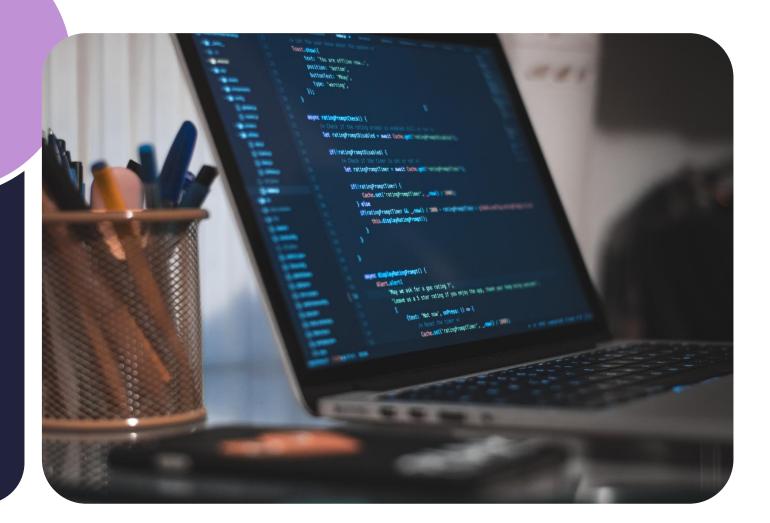
The *if* statement is False, which means the print statement (*print*(<sup>0</sup> *Happy*<sup>0</sup>)) will not be executed.

```
if lucky_number=-2
print('Happy')
```

### DID YOU KNOW?

#### **If** statement

An *if* statement in Python is the simplest **decision-making** statement you can use.





## Python *if, elif* and *else* Statements

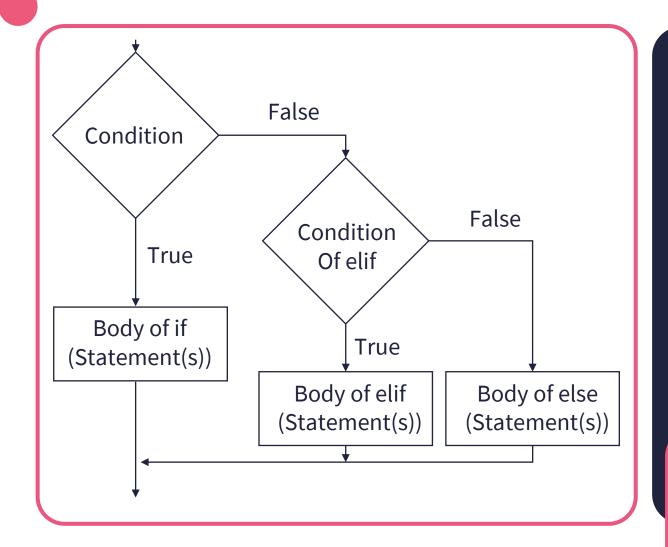


#### *else* statement

- When the *if* statement Boolean expression (condition) is False or 0.
- When the *if* statement Boolean expression is True, the *else* statement is not executed.

```
if condition:
    first_statement(s)
else:
    second_statement(s)
```





### *If* and *else* statements

Flow Diagram

```
if True:
    first_statement(s)
else:
    second_statement(s)
```

#### else statement

- When the *if condition* is False the first\_statement will not be executed.
- The else (*second\_statement*(*s*)) will be executed.

```
if False:
    first_statement(s)
else:
    second_statement(s)
```



#### If and else statement combined

```
lucky_number=5
if lucky_number>0:
    print('Great')
else:
    print('Not Great')
Great
```

```
))){TOF(I-0+0++
      (){b.data(this,e,m))
    =r?!0:"false"===r?!!!
    e?(n=(n||"fx")+
    ===i&&(i=n.shift()
   ta(e,n)})})}),b.fm.
  we:function(e){
     r, i=1, o=b.Deferred()
**tarea|button|object
    ,b.attr,e,t,
  .each(function(){
    [a],r=1===n.node
    this.each(fun
0,a=0,s=b(this),
    Name=this.clas
```

#### else statement

- The *if* statement will yield a true Boolean data type.
- Body *print*(<sup>0</sup> *Great*<sup>0</sup>) statement will be executed.
- The *else* statement will be ignored.

```
lucky_number=-2
if lucky_number>0:
    print('Great')
else:
    print('Not Great')
Not Great
```







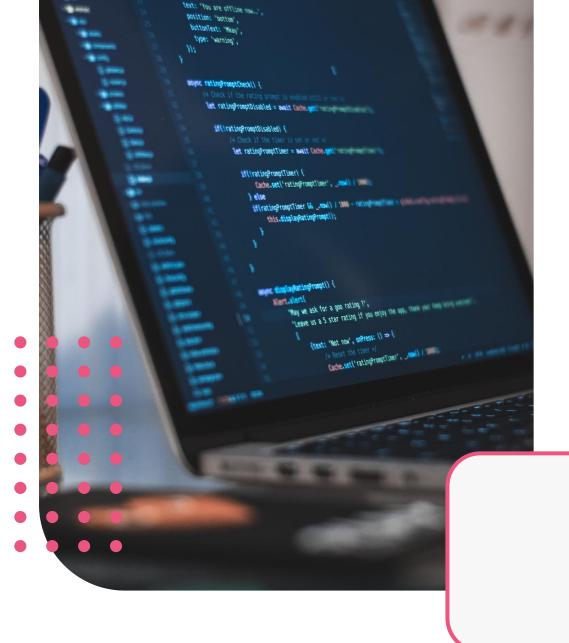
- -2 is not greater than 0.
- The body of the *else* statement (*print*(<sup>0</sup>*Not Great*<sup>0</sup>)) will be executed since the *if* statement is false.

The output is a string data type.



Python *if, elif* and *else* Statements





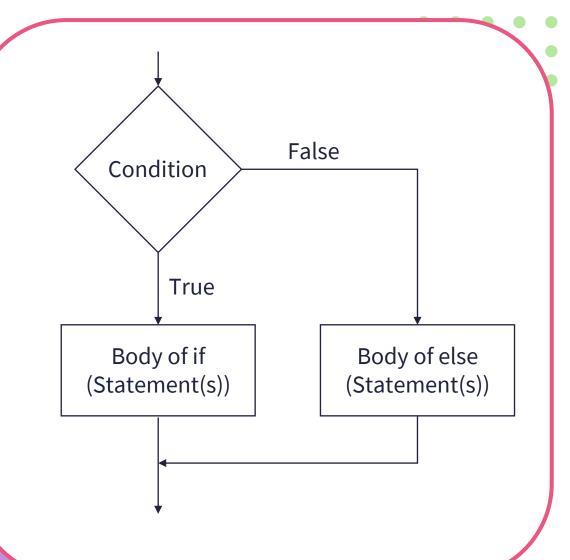
### Python *if, elif* and *else* Statement

• The *elif* statement which is the combination of the *if* and *else* statements.

```
if condition_1:
    first_statement(s)
elif condition_2:
    second_statement(s)
else:
    last_statement(s)
```

### Python *if, elif* and *else* Statements

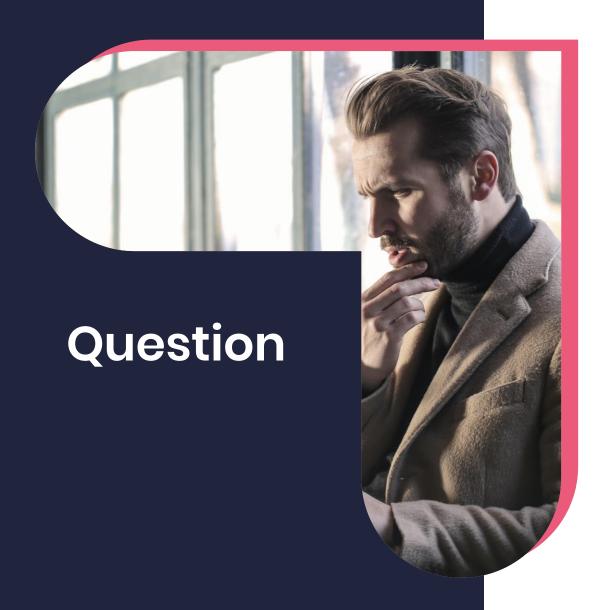
- If the first condition (*condition*\_1), the *if* statement is false, then the next condition (*condition*\_2) which is the *elif* statement, will be checked.
- *The elif* condition is also false, then the next condition will be checked.
- If all the conditions are false, then the last statement, which is the *else* statement will be executed.



### Python *if, elif* and *else* Statements

- The number is positive, is printed.
- The number is zero, will be printed.
- The number is negative from the *else* body statement will be printed.

```
lucky_number=5
if lucky_number>0:
    print("The number is positive")
elif lucky_number==0:
    print("The number is zero")
else:
    print("The number is negative")
The number is positive
```



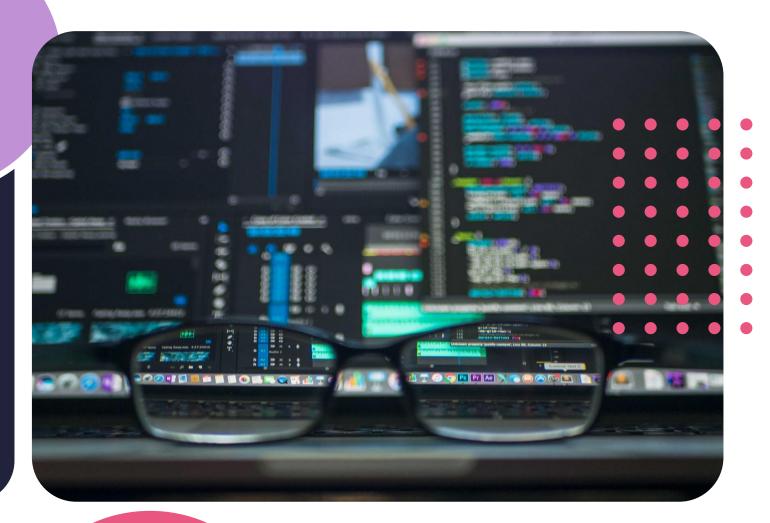


Have you noticed that the previous code has two equal signs (==) consecutively?

### DID YOU KNOW?

#### **Python**

It is very easy to implement a program in Python that has multiple decisions, that lead to different outcomes.





### Multiple cascading conditions

- We then place another condition statement inside the code block.
- We use the nested if statements.
- *if, elif* and *else* statements inside another *if, elif* and *else* statements.

```
if condition 1:
    first statement(s)
    if condition 2:
        second statement(s)
    elif condition 3:
        third statement(s)
    elif condition 4:
        fourth statement(s)
    else:
        fifth statement(s)
else:
    sixth statement(s)
```

```
if condition 1:
    first statement(s)
   if condition 2:
        second statement(s)
    elif condition 3:
        third statement(s)
    elif condition 4:
        fourth statement(s)
    else:
        fifth statement(s)
else:
    sixth statement(s)
```



#### Multiple cascading conditions

```
num=8
if num<13:
    if num==11:
        print('The number is equal to 11')
    elif num==10:
        print('The number is equal to 10')
    elif 5 <= num <= 9:
        print('The number is equal to 9 or 5, or is between 9 and 5')
    else:
        print('The number is less than 5')
else:
    print('The number is bigger than 13')</pre>
The number is equal to 9 or 5, or is between 9 and 5
```

### DID YOU

#### Nesting

Nesting allows your program to pass multiple conditions before it runs.





### Challenge >>>

Create a python program that prompts the user the question: "is an if statement the most simple decision making statement?" Yes or no. Based on the answer the user inputs, the program should print either correct or incorrect.

#AudioIsAwesome