Conditional Statement (If-else)

Conditional statement allow you to make decisions based on the values of variables or the result of comparisons. That is why they are also called decision making statements.

In Python, there are three forms of the conditional statement.

- if statement
- if...else statement
- if...elif...else statement

If Statement

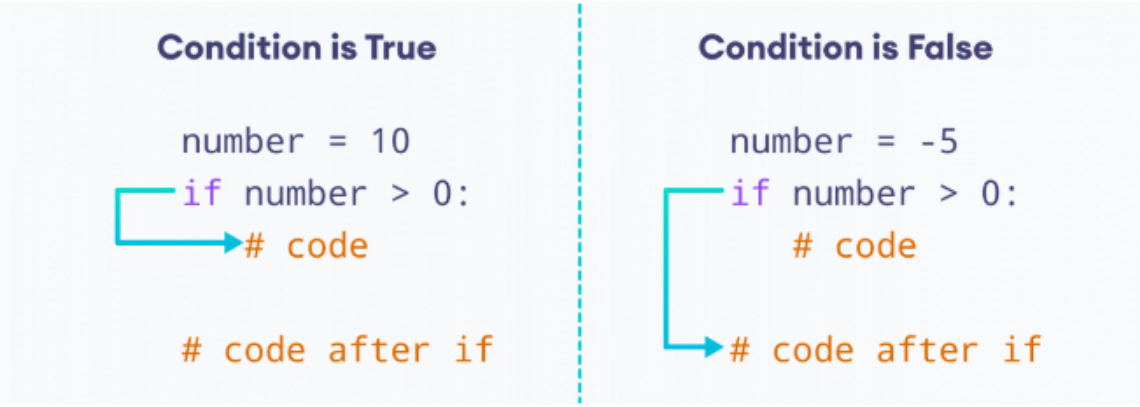
The if statement allows you to execute a block of code if a certain condition is true.

Syntax:

if condition:

```
    block of code/ statements
```

- If the condition satify, then the block of code or statements inside the body of the if execute.
- If the condition not satisfy, then the block of code or statements inside the body of the if execute.



Note: There are two most important point that need to be remember while using if statement.

1. colon(:)
2. Indentation

If any of these two points skipped, then the code will raise an error. So, while using "if statements", always use colon and a proper indentation at a required place.

In [1]:
```
num = 2          # The value of num is assigned as 2.
if num < 5: #Here, we are checking the condition. Since 2 is less than 5 which mea
```

```
    print("I am inside the code.")   # Here, this statement is written inside the 

print("I am outside the code")   # Here, this statement is written outside the if 
                                 # condition is true or not, this statement will 1
```

```
I am inside the code.
I am outside the code
```

In [2]:
```
num = 8             # The value of num is assigned as 8.
if num < 5: #Since 8 is greater than 5 which means the statement inside "if" will 
    print("I am inside the code.")   # Here, this statement is written inside the 

print("I am outside the code")   # Here, You can observe that only the code writte
                                 # writtten inside the if body was not executed be
```

```
I am outside the code
```

In [3]:
```
num = 8
if num < 5     #Here, i didn't use colon that is why this code will raise an error
    print("I am inside the code.")

print("I am outside the code")
```

```
  Cell In[3], line 2
    if num < 5     #Here, i didn't use colon that is why this code will raise an e
rror.
                   ^
SyntaxError: expected ':'
```

In [4]:
```
num = 8
if num < 5:
print("I am inside the code.")     #Here I didn't give indentation after using if 

print("I am outside the code")
```

```
  Cell In[4], line 3
    print("I am inside the code.")     #Here I didn't give indentation after using
if statement that is why it raise an error.
    ^
IndentationError: expected an indented block after 'if' statement on line 2
```

Note: The condition that we used in if statement returns either **True** or **False**. This means if the condition is TRUE, then the code written inside the if body will execute but if the condition is FALSE, then the code written inside the if body will not execute.

In [5]:
```
if True:
    print("This will execute as condition is TRUE.")
print("Outside the if body means this will execute.")

#Here both the statements (inside and outside) executed.
```

```
This will execute as condition is TRUE.
Outside the if body means this will execute.
```

In [6]:
```
if False:
    print('This will not execute as condition is FALSE.')
print("Outside the if body means this will execute.")
#Here only the outside statement executed.
```

```
Outside the if body means this will execute.
```

In [7]:
```
if 1:
    print("1 is equivalent to True.")
#Here 1 means True that is why inside statement executed. Note: Any number except 0
```

```
1 is equivalent to True.
```

In [8]:
```
if 0:
    print("0 is equivalent to False.")
#Here there is no output as 0 means False that is why nothing executed.
```

In [9]:
```
if 5-5:         #Here 5-5 = 0 which is equivalent to False.
    print("Nothing will execute.")
```

In [10]:
```
if 5-2:          #Here 5-2 = 3 which is a positive number but not 0.
    print("This will execute as any number except 0 means True.")
```

```
This will execute as any number except 0 means True.
```

In [11]:
```
if 2-5:         #Here 2-5 = -3 which is a negative number but not 0.
    print("This will execute as any number except 0 means True.")
```

```
This will execute as any number except 0 means True.
```

Note: We can use Comparsion operators (>, <, ==, !=, <=, >=) for condition. Also, we can merge two or more condition using logical operators (and, or, not).

Comparison operator used in Condtion:-

In [12]:
```
if 2 < 5:
    print("I used < (Less than) operator in the condition.")
    print('2 is less than 5.')
```

```
I used < (Less than) operator in the condition.
2 is less than 5.
```

In [13]:
```
if 8 > 5:
    print("I used > (Greater than) operator in the condition.")
    print('8 is greater than 5.')
```

```
I used > (Greater than) operator in the condition.
8 is greater than 5.
```

In [14]:
```
if 5 == 5:
    print("I used == (Equal to) operator in the condition.")
    print('5 is equal to 5.')
```

```
I used == (Equal to) operator in the condition.
5 is equal to 5.
```

NOTE: Equal to operator is represented by == (double equal) not = (single operator). Most of the beginner make this mistakes while equating something using equal to operator. = is an assignment operator.

In [15]:
```
if 2 != 5:
    print("I used != (Not Equal to) operator in the condition.")
    print('2 is not equal to 5.')
```

```
I used != (Not Equal to) operator in the condition.
2 is not equal to 5.
```

In [16]:
```
if 2 <= 5:
    print("I used <= (Less than or equal to) operator in the condition.")
    print('2 is less than or equal to 5.')
```

```
I used <= (Less than or equal to) operator in the condition.
2 is less than or equal to 5.
```

In [17]:
```
if 8 >= 5:
    print("I used != (Greater than or equal to) operator in the condition.")
    print('8 is greater than or equal to 5.')
```

```
I used != (Greater than or equal to) operator in the condition.
8 is greater than or equal to 5.
```

Logical operator and comparison operator used in condtion:-

In [18]:
```
if 2 < 5 and 2 > 5:
    print("I used 'and' operator which result true when both the conditions are tru

print("Since True and False is equal to False that is why the inside statement will
```

```
Since True and False is equal to False that is why the inside statement will not e
xecute..
```

In [19]:
```
if 2 < 5 or 2 > 5:
    print("I used 'or' operator which result true when any of the condition is True

print("Since True and False is equal to True that is why the inside statement will
```

```
I used 'or' operator which result true when any of the condition is True else retu
rn false.
Since True and False is equal to True that is why the inside statement will execut
e..
```

In [20]:
```
if not 5:
    print("Not 5 means False that is why inside code will not execute.")

print("This is outside the if body.")
```

```
This is outside the if body.
```

In [21]:
```
# We can use empty list or anything that is empty(zero elements) as a condition.
l1=[]
if l1:
    print('List is empty.')
print('Here the list is empty that is why it return False and inside statement will
```

```
Here the list is empty that is why it return False and inside statement will not e
xecute
```

In [22]:
```
l1=[1,2,3,4,5]
if l1:
    print('List is not empty.')
```

```
List is not empty.
```

In [23]:
```
# We can also compare datatype to check the condition.
if '2'==2:
    print("It will execute.")

print("The datatype of 's' is string whereas the datatype of 2 is Integer. The cond
```

```
The datatype of 's' is string whereas the datatype of 2 is Integer. The condition
return false.
```

In [24]:
```
if "hello":
    print("This will execute.")

# Here, "hello" is used as a condition which return True.
```

```
This will execute.
```

If..else Statement

If the condition is true, the if block code is executed and if the condition is false, the else block code is executed.

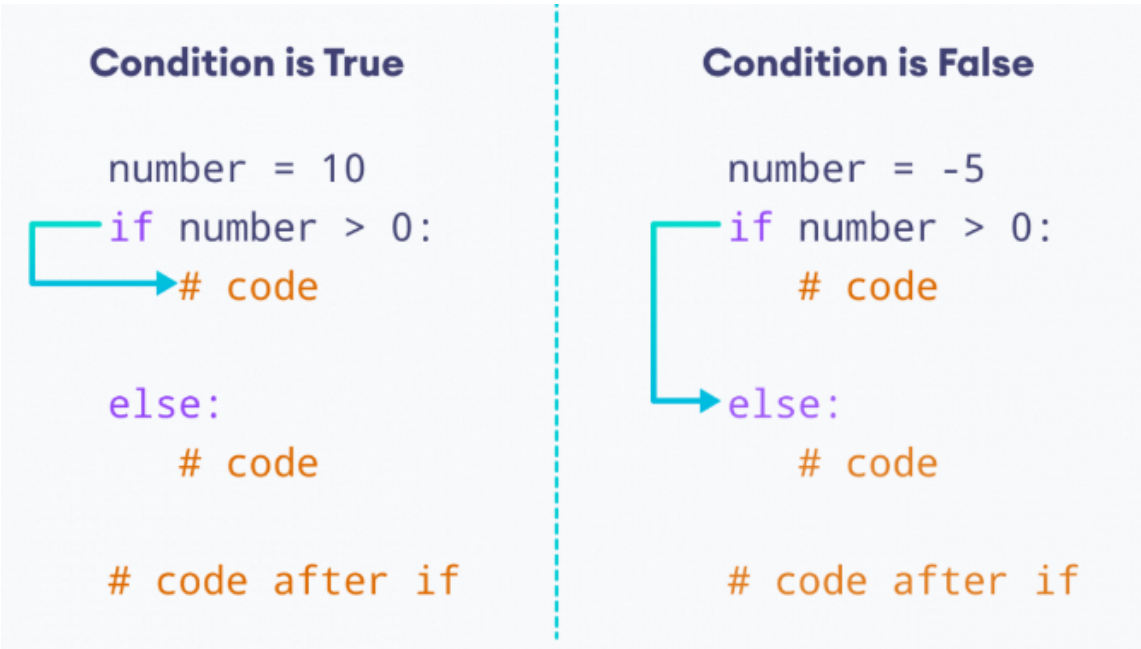
Syntax:

if condition:

```
    block of code/ statements
```

else:

```
    block of code/ statements
```



In [26]:
```
age = int(input("Enter your age: "))
if age >= 18:
    print('You can cast the vote.')
else:
    print('You cannot cast the vote.')
```

```
Enter your age: 12
You cannot cast the vote.
```

In [27]:
```
age = int(input("Enter your age: "))
if age >= 18:
    print('You can cast the vote.')
else:
    print('You cannot cast the vote.')
```

```
Enter your age: 21
You can cast the vote.
```

Note: Here you can see when i entered 12 as age, then condition return False that is why the code written inside the "if" body was not executed whereas the code written inside the

'else' body was executed. But when i entered 21 as age, then condition return True that is why the code written inside the "if" body was executed whereas the code written inside the 'else' body was notexecuted. This implies that "else" works only when the condition used in "if" return false.

In [28]:
```
num = int(input("Enter any number: "))
if num > 0:
    print('The number is positive.')
else:
    print('The number is negative.')
```

```
Enter any number: 5
The number is positive.
```

In [29]:
```
num = int(input("Enter any number: "))
if num > 0:
    print('The number is positive.')
else:
    print('The number is negative.')
```

```
Enter any number: -5
The number is negative.
```

Note: Here, don't forget to use colon and indentation else it will return error.

If..elif..else Statement

If the condition is true, the if block code is executed and if the condition is false, the else block code is executed.

Syntax:

if condition:

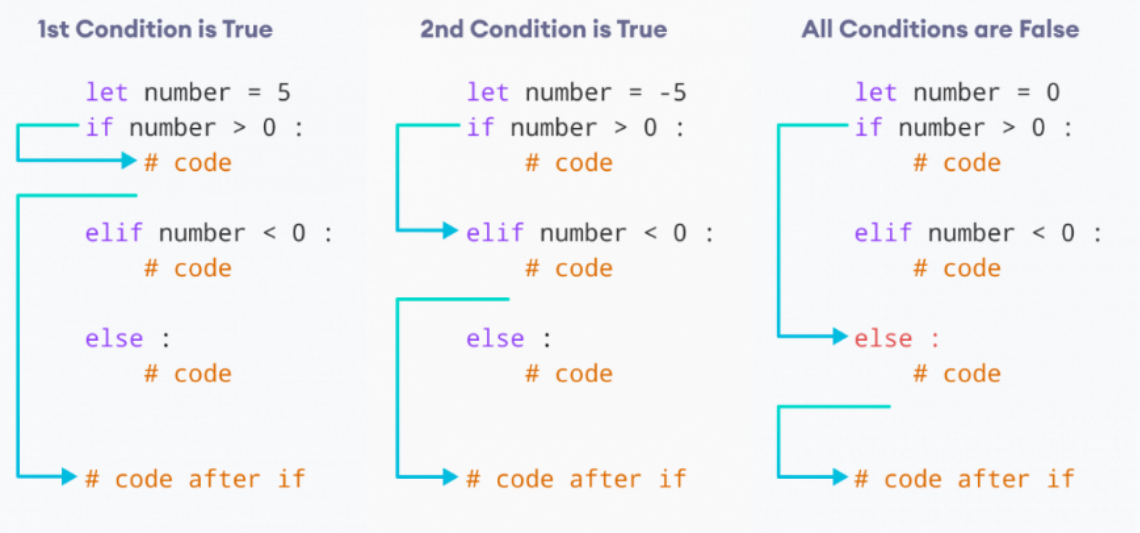
```
    block of code/ statements
```

elif condition:

```
    block of code/ statements
```

else:

```
    block of code/ statements
```



Note: In elif we again use condtion but in else we didn't use any condtion which means the condition used in "if" return False, then the python interpretor goes on "elif condition" and check this condition, if condition return True then the statement written inside the 'elif' will execute but if the condition used in 'elif' return False, then the python interpretor go on else and here there is no condition which means any statement written inside else will definitely execute.

In [30]:
```
num = int(input("Enter any number: "))
if num > 0:
    print('The number is positive.')
elif num < 0:
    print('The number is negative')
else:
    print('The number is 0.')

# Here the number 5 is greater than 0 that is why this number is positive and the s
```

```
Enter any number: 5
The number is positive.
```

In [31]:
```
num = int(input("Enter any number: "))
if num > 0:
    print('The number is positive.')
elif num < 0:
    print('The number is negative')
else:
    print('The number is 0.')

# Here the number -5 is less than 0 that is why this number is negatove and the sta
```

```
Enter any number: -5
The number is negative
```

In [32]:
```
num = int(input("Enter any number: "))
if num > 0:
    print('The number is positive.')
elif num < 0:
    print('The number is negative')
else:
    print('The number is 0.')
# Here the number 0 does not satisfy the condition used in either 'if' or 'elif', t
```

```
Enter any number: 0
The number is 0.
```

Note: We can use **any number**_ of '**elif**' in between '**if**' and '**else**', but there can be **only one or zero** '**else**' for a particular '**if**' condition.

Nested if statements

When we use if, if..else, if..elif..else inside other if, if..else, if..elif..else, then this situation is called nested if statements.

In [33]:
```
n=int(input("Enter any number: "))
if n % 2 == 0:
    if n < 10:            # here we are using nested if..else
        print('The number is even and less than 10.')
    elif n == 10:
        print('The number is even and equal to 10.')
    else:
        print('The number is even and greater than 10.')
else:
    print(f'The number {n} is not even.')
```

```
Enter any number: 12
The number is even and greater than 10.
```

In [34]:
```
n=int(input("Enter any number less than 10: "))
if n % 2 == 0:
    if n % 3 ==0:
        if n % 4 ==0:
            if n % 9 == 0:
                print(f'The number {n} is divisible by 2, 3, 4, 6 and 9.')
            elif n % 8 == 0:
                print(f'The number {n} is divisible by 2, 3, 4, 6 and 8.')
            else:
                print(f'The number {n} is divisible by 2, 3, 4 and 6.')
        elif n % 9 ==0:
            print(f'The number {n} is divisible by 2, 3, 6 and 9.')
        else:
            print(f'The number {n} is divisible by 2, 3, and 6.')
    elif n % 4 == 0:
        if n % 8 == 0:
            if n % 5 == 0:
                print(f'The number {n} is divisible by 2, 4, 5 and 8.')
            else:
                print(f'The number {n} is divisible by 2, 4 and 8.')
        else:
            print(f'The number {n} is divisible by 2 and 4.')
    elif n % 5 == 0:
        print(f'The number {n} is divisible by 2 and 5.')
    else:
        print(f'The number {n} is divisible by 2.')
elif n % 3 == 0:
    if n % 5 == 0:
        if n % 9 == 0:
            print(f'The number {n} is divisible by 3, 5 and 9.')
        else:
            print(f'The number {n} is divisible by 3 and 5.')
    elif n % 7 == 0:
        print(f'The number {n} is divisible by 3 and 7.')
    elif n % 9 == 0:
        print(f'The number {n} is divisible by 3 and 9.')
    else:
        print(f'The number {n} is divisible by 3.')
```

```
elif n % 5 == 0:
    if n % 7 ==0:
        print(f'The number {n} is divisible by 5 and 7.')
    else:
        print(f'The number {n} is divisible by 5.')
elif n % 7 == 0:
    print(f'The number {n} is divisible by 7.')
else:
    print(f'The number {n} is a prime number.')
```

```
Enter any number less than 10: 48
The number 48 is divisible by 2, 3, 4, 6 and 8.
```

Note: Please take care of 'if' and its corresponding 'elif' and 'else', otherwise you can face problem.

if else using membership operator (in, not in) with List and Tuple.

In [35]:
```
l=[1,2,3,4,5,6]
n=int(input("Enter any number: "))
if n in l:
    print(f'The number {n} is in the list.')
    print(f"The number {n} is at index {l.index(n)}.")
else:
    print(f'The number {n} is not in the list.')
```

```
Enter any number: 4
The number 4 is in the list.
The number 4 is at index 3.
```

In [36]:
```
t=(1,2,3,4,5,6)
n=int(input("Enter any number: "))
if n in t:
    print(f'The number {n} is in the tuple.')
    print(f"The number {n} is at index {l.index(n)}.")
else:
    print(f'The number {n} is not in the tuple.')
```

```
Enter any number: 9
The number 9 is not in the tuple.
```

if else using membership operator (in, not in) with Dictionary.

In [37]:
```
d={'name':'Abhi',
   'course':'DBDA',
   'location':'Ghaziabad'
  }
query = input('Enter the key: ')
if query in d:
    print(f"The key '{query}' is in the dictionary.")
    print(f"The value of key is '{d[query]}'.")
else:
    print("The key is not in the dictionary.")
```

```
Enter the key: name
The key 'name' is in the dictionary.
The value of key is 'Abhi'.
```

Some basic programs using if..else

In [38]:
```
#wap to check whether the given number is even or odd.
num = int(input("Enter the number: "))
if num%2 == 0:
    print(f"The entered {num} is even.")
else:
    print(f"The entered {num} is odd.")
```

```
Enter the number: 5
The entered 5 is odd.
```

In [39]:
```
# wap to grade the marks of students.
marks = int(input("Enter the marks? "))
if marks > 85 and marks <= 100:
    print("Congrats ! you scored grade A.")
elif marks > 60 and marks <= 85:
    print("You scored grade B +.")
elif marks > 40 and marks <= 60:
    print("You scored grade B.")
elif (marks > 30 and marks <= 40):
    print("You scored grade C.")
else:
    print("Sorry you are fail ?")
```

```
Enter the marks? 86
Congrats ! you scored grade A.
```

https://github.com/Abhishekk-Git`