if & else

A program needs to take some decision.

```
Ex: 01 if (Boolean condition) →True
                got to next section
                current section becomes this one
        else
                go back to the beginning of current section
```

Syntax:

if (Boolean condition): → if the Boolean condition is true, whatever work you want to do, you will put a tab and anything that is written in the tab get executed. Every ‘if’ condition ends with a colon ‘:’. After the colon you put a tab (or four spaces) and whatever you write in the tab falls inside the ‘**if**’ block.

```
If (<Boolean_condition>):
        print(“hi”)
        a = 3
        b = 4
        print (a + b)
```

this is called if block. If the Boolean condition is True, the compiler would enter the if statement and enter hi. Let’s supposed the given ‘**if**’ statement is true, then everything

In the block would get executed.

Ex: 01

<u>INPUT:</u>

```
1       a = 10
2       if (a % 2 == 0):
3               print(“The no is even”)
4
5       print (“program completed”)
```

The compiler will go to the first line, a = 10. It will check if the condition given in line number 2 is True or False. If the Boolean condition is true then the compiler will go to the next line and execute the command (in our case, print statement). If the Boolean condition is false in the line number 2, then the compiler will not enter the if block. It will go to the last line (line no.5) in this program as we have not mentioned any ***elif*** or ***else***. It will execute the command in the last line (print).

So, the **<u>OUTPUT:</u>**

```
The no is even
program complete
```

Ex: 02

INPUT **OUTPUT**

```
a = 4
if (a % 2 == 0 and a > 5):
    print (“the no is even”)
print (“program completed”)
```

```
program completed
```

Since we had **and** the whole statement is false and hence the output will be “*program completed*”. If we had **or** in place of **and**:

INPUT **OUTPUT**

```
a = 4
if (a % 2 == 0 or a > 5):
    print ("the no is even")
print ("program completed")
```

```
the no is even

program completed
```

>= → greater than or equal to <= → less than or equal to

Ex:

4 <= 5 is true, because 4 is less than 5
2 >= 1 is true
1 <= 5 is true
1 <= 1 is true
5 <= 4 is false

These are called relation operators.

Ex: 03

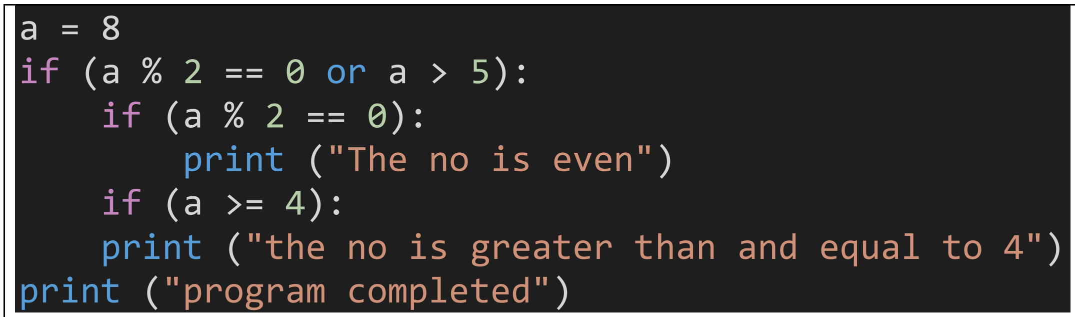
INPUT **OUTPUT**

```
a = 3
if (a % 2 == 0 or a >= 4):
    print ("the no is even")
print ("program completed")
```

```
program completed
```

Ex: 04

INPUT **OUTPUT**



```
a = 8
if (a % 2 == 0 or a > 5):
    if (a % 2 == 0):
        print ("The no is even")
    if (a >= 4):
    print ("the no is greater than and equal to 4")
print ("program completed")
```

The no is even

the no is greater than and equal to 4

program completed

In the 'if' condition, the first condition a % 2 == 0 is true, but the second condition, a > 5 is false. Since the gate is **or**;

[True or False = True**]**

The "if" statement becomes true and the compiler enters the second line. Now the second line also has the same condition as the first line-first condition. It is true as well, so the compiler will enter this "if" statement in the second line and will execute the print statement; "*The no is even*".

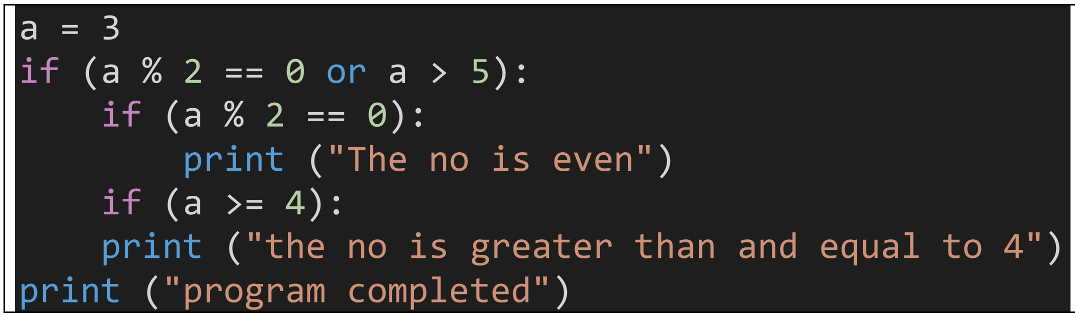
Once it has executed the print statement it will come out and go to the next line, and check the Boolean condition in the "if" command; (a >= 4). The value of 'a' is 8 which is two times 4. Since 8 is greater than 4 (8 > 4) the "if" command is true and hence the compiler will enter this "if" statement as well and execute the print command; "*The no is greater and equal to 4*.

After this, the compiler will go to the next line and will execute the print statement as "*program completed*"

This is a good example of ***"Nested-if"***.

If a = 3, what will be output?

INPUT **OUTPUT**



```
a = 3
if (a % 2 == 0 or a > 5):
    if (a % 2 == 0):
        print ("The no is even")
    if (a >= 4):
    print ("the no is greater than and equal to 4")
print ("program completed")
```

program completed

If a = 3, the if condition will become false and it will not enter the if block, skip it and will execute the print statement.

If a = 4,

INPUT

```
a = 4
if (a % 2 == 0 or a > 5):
    if (a % 2 == 0):
        print ("The no is even")
    if (a >= 4):
    print ("the no is greater than and equal to 4")
print ("program completed")
```

OUTPUT

The no is even

the no is greater than and equal to 4

program completed

Ex: 04

```
"""
Program will take user input; INPUT = 1 - OUTPUT = :)
or else, OUTPUT = :(

else statement is always after an if statement and it doesn't require
any condition.

else:
and else statement can never exist alone
"""

user_input = input ()

if (user_input == "1"):
    print (":)")
else:
    print (":(")
```

OUTPUT:

user_input = 1
:)

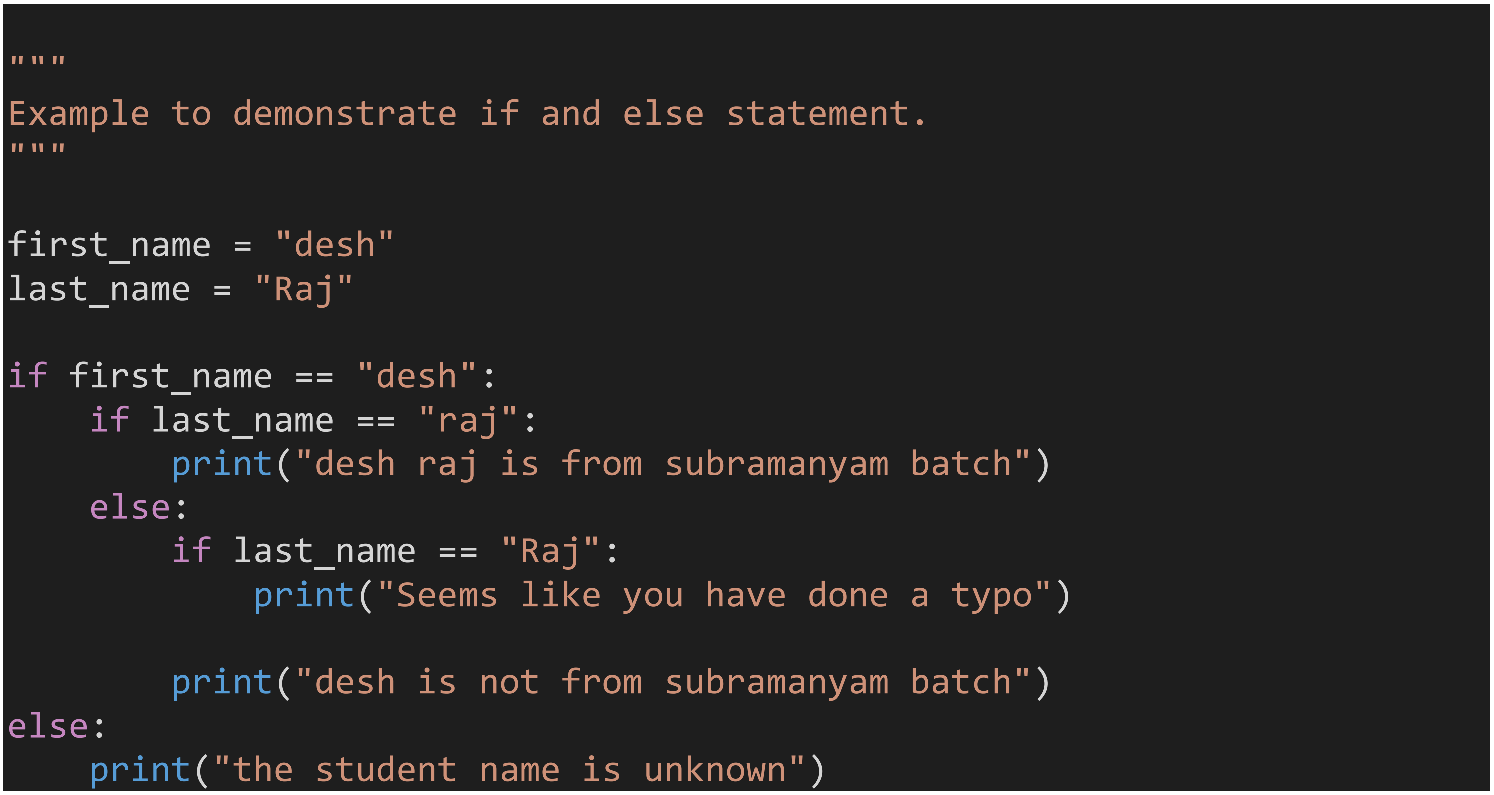
user_input = a
: (

for any input other than 1, the OUTPUT will be a sad emoji.

NOTE:

An “if” condition can exist without “else”, but without “if” condition you cannot have an “else” condition.

Ex: 05

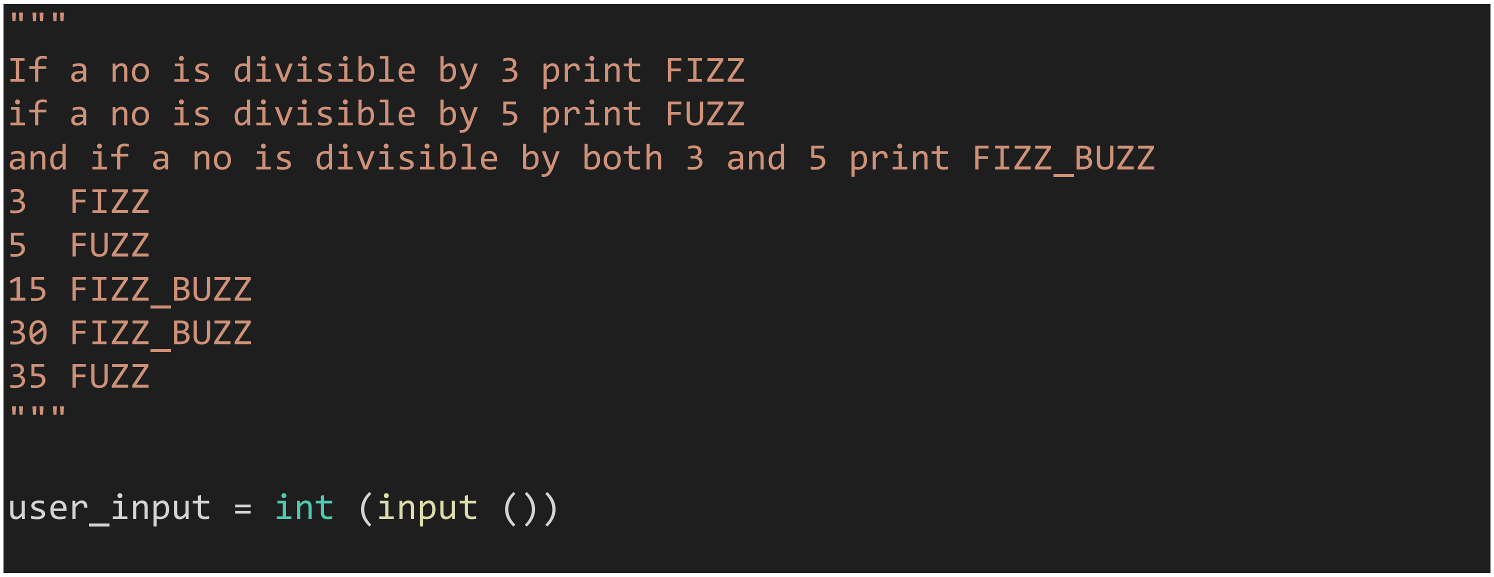


```
"""
Example to demonstrate if and else statement.
"""

first_name = "desh"
last_name = "Raj"

if first_name == "desh":
    if last_name == "raj":
        print("desh raj is from subramanyam batch")
    else:
        if last_name == "Raj":
            print("Seems like you have done a typo")

        print("desh is not from subramanyam batch")
else:
    print("the student name is unknown")
```

OUTPUT:

```
Seems like you have done a typo

desh is not from subramanyam batch
```

Ex: 06



```
"""
If a no is divisible by 3 print FIZZ
if a no is divisible by 5 print FUZZ
and if a no is divisible by both 3 and 5 print FIZZ_BUZZ
3  FIZZ
5  FUZZ
15 FIZZ_BUZZ
30 FIZZ_BUZZ
35 FUZZ
"""

user_input = int (input ())
```

```
if (user_input % 5 == 0 and user_input % 3 == 0):
  print ("FIZZ_BUZZ")
else:
  if (user_input % 5 == 0):
    print ("FUZZ")
  if (user_input % 3 == 0):
    print ("FIZZ")
```

if, elif & else

```
if ( ) :
        pass
elif ( ) :
        pass
else:
        pass
```

In an “if” condition, if we have to make more than 2 or more options then we can add ***elif***. So, after the “if” command, we can give elif condition to check. There can be multiple elif between if and else.

Now, when the “if” condition is true, you will not go to the elif. But, when the “if” condition is false, then it will go to the elif and check the condition.

Ex:

```
weather = "sunny"
if weather == "raining":
    print ("I will wear raincoat")
elif weather == "sunny":
    print ("I will wear sunscreen")
elif weather == "stormy":
    print ("I will stay home")
else:
    print ("I will sleep at home")
```

UNDERSTANDING:

In the above example, if the “if” statement is true, then the program will enter the if statement giving us an output as “*I will take umbrella*”.

If the “if” statement is false, then the program will not enter the if statement and will go the first elif and check it’s condition. If the it is true then the program will enter the elif statement giving us an output as “*I will wear sun glasses*”

If the first elif is false, then then the program will not enter the first elif statement and will go the second elif and check its condition. If the it is true then the program will enter the second elif statement giving us an output as "*I will party*"

When all the three conditions become false (if, first_elif, second_elif), then the program will enter the else command and will give us an output as "*I will sleepy*"

```
if weather == "raining":
    print ("I will wear raincoat")

if weather == "sunny":
    print ("I will wear sunscreen")

if weather == "stormy":
    print ("I will stay home")
else:
    print ("I will sleep at home")
```

NOTE: In the above program, when we replace elif with if, then the program will check every "if" condition irrespective of the condition being true or false. It will check all the 3 if conditions. But if we give elif, then the program will check for if condition and if it true then there is no need for it to check the remaining 2 elif's. But, if we replace the elif's with if, the even though the first if is true, the program will check the remaining two if conditions as well.

Ex:07

```
no1 = 5
no2 = 20


if no2 == 10 * 2:
    if no2 % 2 == 0:
        print("the no is even")
    elif no2 == 20:
        print("the no is 20")
else:
    print("the no is not 20")
```

<u>OUTPUT</u>

```
the no is even
```

Ex: 08

```python
no1 = 100
no2 = 200

if no2 == 2 * 300:
    print("Hey 1")
else:

    if no2 == 200:
        print("Hey 2")

    if no1 == 100:
        print("Hey 3")
    else:
        print("Hey 4")
```

OUTPUT

```
Hey 2

Hey 3
```