Python Conditions/Decision-Making Statements

In this module, we will cover :

- ✓ Python Conditional Statements
- ✓ If in Python
- ✓ If Else in Python
- ✓ If Elif Else Python
- ✓ Nested If / If Else in Python
- ✓ Shorthand If and If else in Python

Python Conditional Statements:

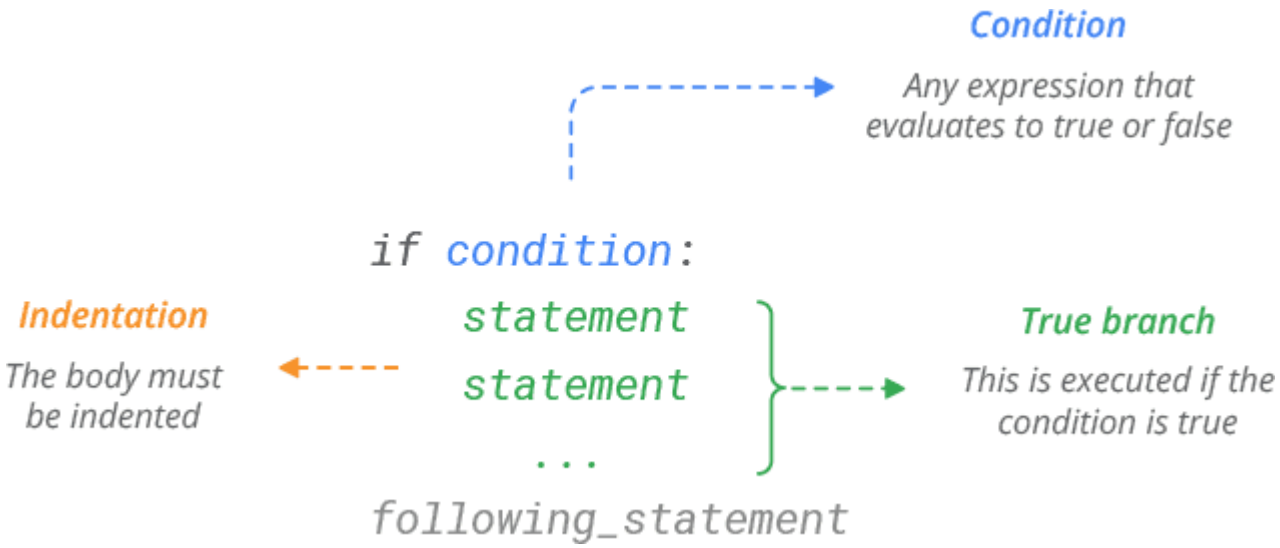
Often you need to execute some statements, only when certain condition holds. You can use following conditional statements in your code to do this.

STATEMENT	FUNCTION
If Statement	This statement is use to check whether the condition is true or not, if the condition is true then the code is executed.
If Else Statement	This statement is use to check whether the condition is true or not however if the condition doesn't meet the criteria then the else statement gets executed.
Else - If Statement	It is similar to the if else statement but it allows us the freedom of executing multiple else if blocks to check as many blocks of code we want

The if Statement

Use if statement to execute a block of Python code, if the condition is true.

Syntax



Basic Example

```
x, y = 7, 5

if x > y:

        Print ('x is greater')

# Prints x is greater
```

Likewise, you can use following comparison operators to compare two values:

Operator	Example	Operator	Example	Operator	Example
==	if x == y	>	if x > y	<	if x < y
!=	if x != y	>=	if x >= y	<=	if x <= y

More Examples

In Python, any non-zero value or nonempty container is considered **TRUE**, whereas Zero, None, and empty container is considered **FALSE**.

That's why all the below if statements are valid.

```
# any non-zero value

if -3:

    print('True')

# Prints True

# mathematical expression

x, y = 7, 5

if x + y:

    print('True')

# Prints True

# nonempty container

L = ['red','green']

if L:

        print('True')    # Prints True
```

Significance of Indentation

Indentation has a special significance in Python. It is used to **define a block of code** (often referred to as, a suite). Contiguous statements that are indented to the same level are considered as part of the same block.

if statement without indentation raises syntax error.

```
x, y = 7, 5

if x > y:

Print ('x is greater')    # Triggers SyntaxError: expected an indented block
```

Nested if Statement

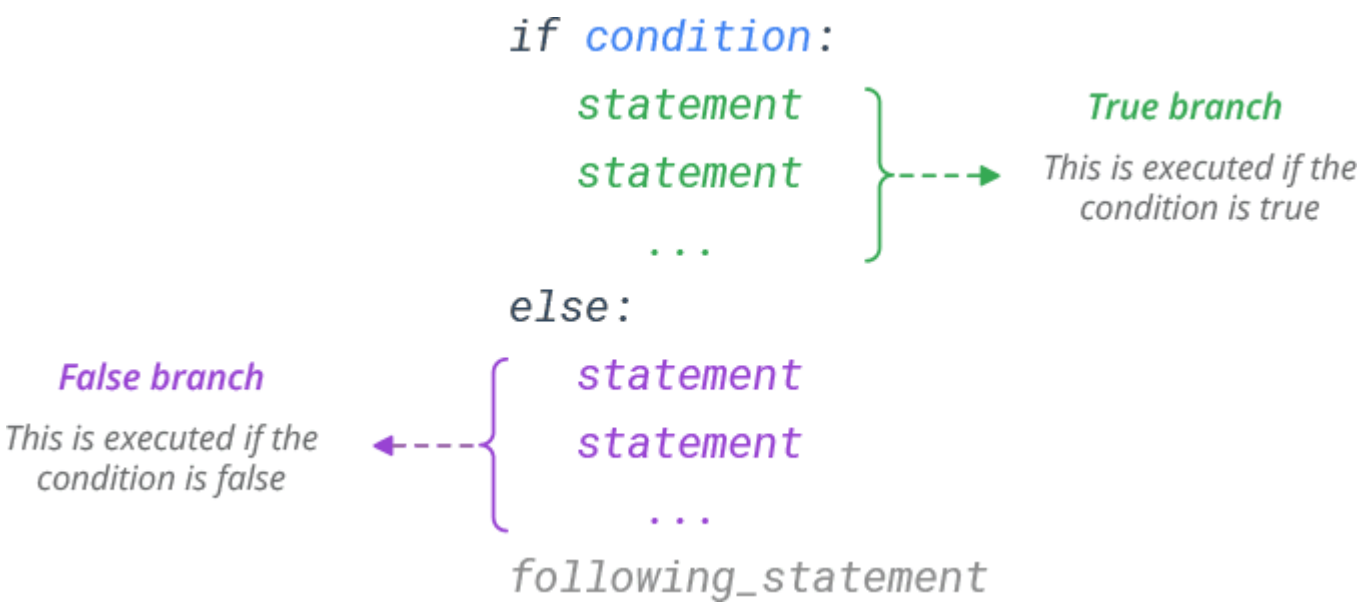
You can nest statements within a code block to begin a new code block, as long as they follow their respective indentations.

```
x, y, z = 7, 4, 2

if x > y:

    print("x is greater than y")

    if x > z:

        print("x is greater than y and z")

# Prints x is greater than y

# Prints x is greater than y and z
```

The else Statement

Use else statement to execute a block of Python code, if the condition is false.

Syntax



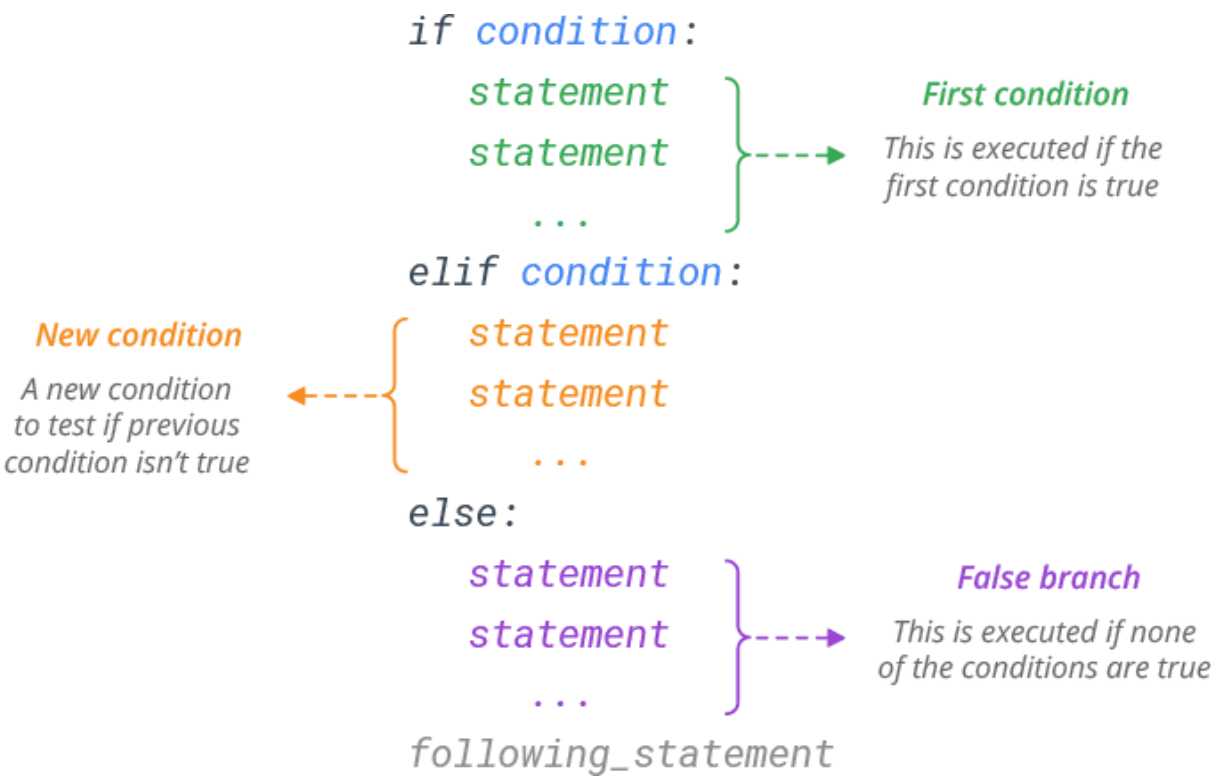
Basic Example

```
x, y = 7, 5

if x < y:

    print('y is greater')

else:

    print('x is greater')

# Prints x is greater
```

The elif (else if) Statement

Use elif statement to specify a new condition to test, if the first condition is false.

Syntax



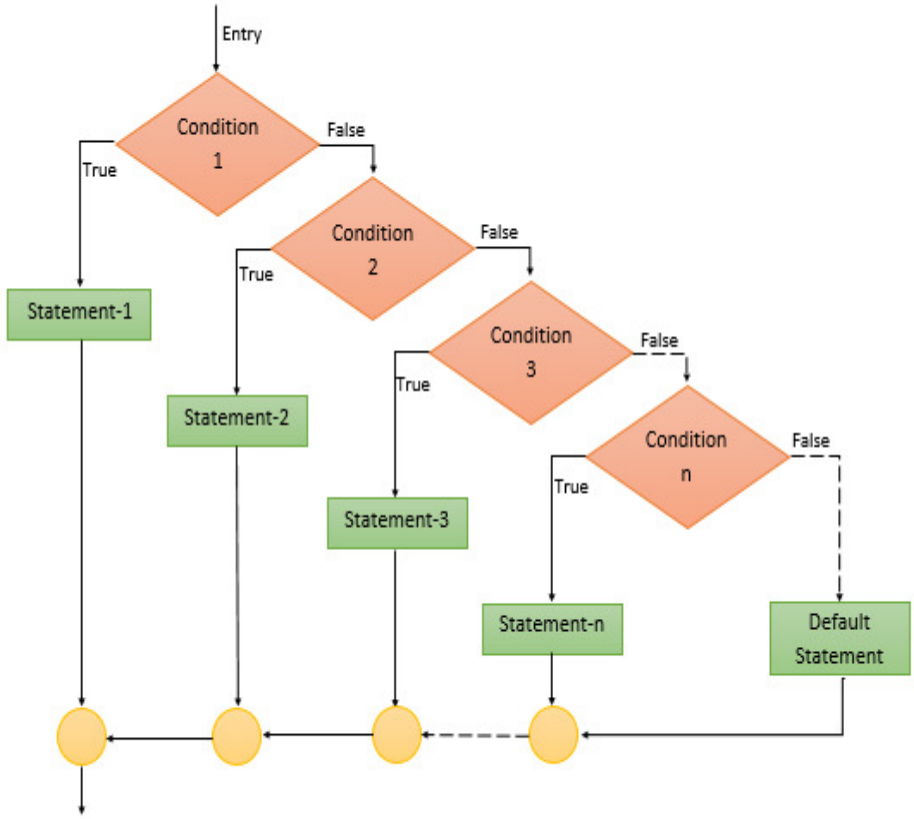
Basic Example

```
x, y = 5, 5

if x > y :

    print('x is greater')

elif x < y :

    print('y is greater')

else:

        print('x and y are equal')    # Prints x and y are equal
```

Substitute for Switch Case - elif ladder - Chained conditionals

Unlike other programming languages, Python does not have a '**switch**' statement. You can use if…elif…elif sequence as a substitute which is refered as chained conditionals

```
if choice == 1:

        print('case 1')

elif choice == 2:

        print('case 2')

elif choice == 3:

        print('case 3')

elif choice == 4:

        print('case 4')

else:

        print('default case')
```



Note : elif is an abbreviation of else if.There is no limit of the number of elif statements but only a single (and optional) final else statement is allowed and it must be the last branch in the statement.

Nested If-Else in Python

Some programs may have a code block under an “if” clause. And it could have subsequent conditional blocks.

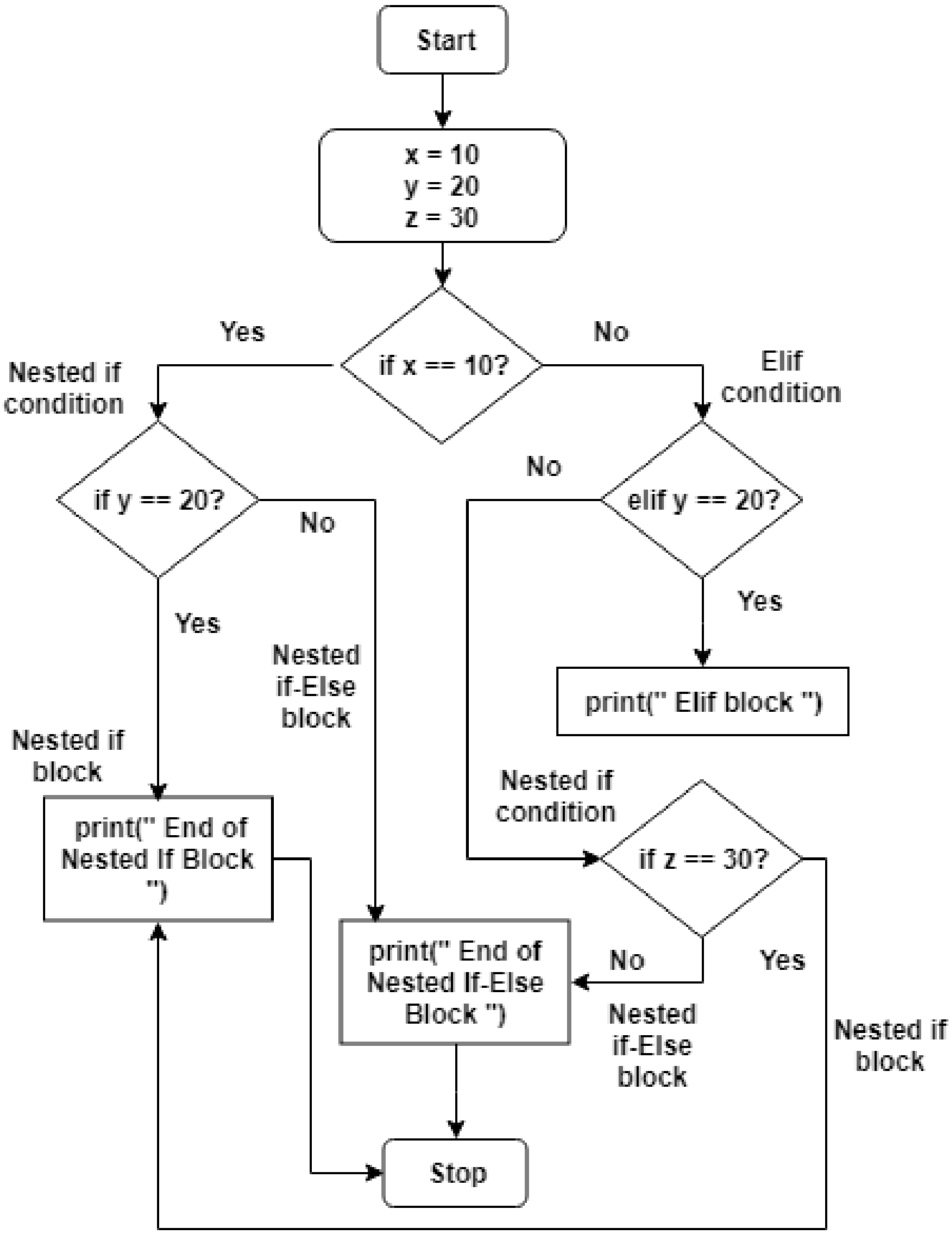
In such a case, Python allows nesting of an if-else or if-elif-else inside another conditional clause.

Python doesn’t limit the level of nested conditions in a program. Given below is the syntax of a multi-level nested if-elif-else statement.

Example :

```
#Nested If Else For Finding Greatest Number Among Three Numbers

x=int(input("Please Enter First number : "))

y=int(input("Please Enter Second number : "))

z=int(input("Please Enter Third number : "))

if(x>b):

    if(x>c):

        print("x is gretest number”)

    else:

        print(“z is gretest numbe”)

elif(y>z):

    print("y is gretest number")

else:

    print("z is gretest number")
```

Flow Chart :


Start
x = 10
y = 20
z = 30
if x == 10?
Yes
No
Nested if condition
Elif condition
if y == 20?
elif y == 20?
No
No
Yes
Yes
Nested if-Else block
print(" Elif block ")
Nested if block
Nested if condition
print(" End of Nested If Block ")
if z == 30?
print(" End of Nested If-Else Block ")
No
Yes
Nested if-Else block
Nested if block
Stop

Multiple Conditions - Usage of Logical Operators

To join two or more conditions into a single if statement,
use logicaloperators viz. and, or and not.

✓ **and expression is True, if all the conditions are true.**

```
x, y, z = 7, 4, 2

if x > y and x > z:

    Print ('x is greater')

# Prints x is greater
```

✓ **or expression is True, if at least one of the conditions is True.**

```
x, y, z = 7, 4, 9

if x > y or x > z:

    print('x is greater than y or z')

# Prints x is greater than y or z
```

✓ **not expression is True, if the condition is false.**

```
x, y = 7, 5

if not x < y:

    print('x is greater')

# Prints x is greater
```

One Line if Statement - Short Hand If

Python allows us to write an entire if statement on one line.

```
# Short Hand If - single statement

x, y = 7, 5

if x > y : print('x is greater')

# Prints x is greater
```

You can even keep several lines of code on just one line, simply by separating them with a semicolon ; .

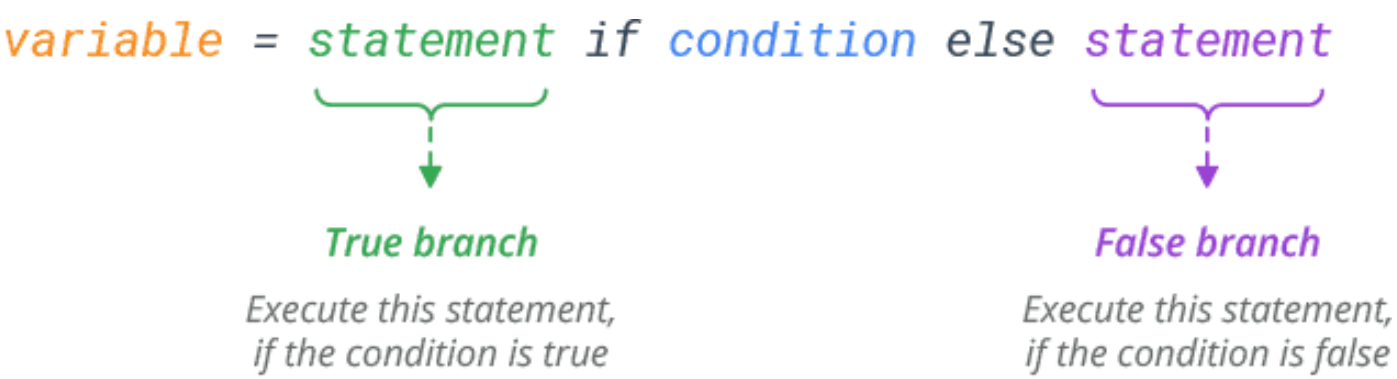
```
# Short Hand If - multiple statements

x, y = 7, 5

if x > y: print('x is greater');    print('y is smaller');    print('x and y are not equal')

# Prints x is greater

 # Prints y is smaller

# Prints x and y are not equal
```

Conditional Expressions (ternary operator) - Short Hand If-Else

Conditional expression (sometimes referred to as 'ternary operator') allows us to select one of two statements depending on the specified condition.

The syntax of the conditional expression is :

Syntax



Examples

```
x, y = 7, 5

Print ('x is greater') if x > y else print ('y is greater')

# Prints x is greater
```

You can also use it to select **variable assignment**.

```
x, y = 7, 5

max = x    if x > y else y

Print (max)

 # Prints 7
```

You can also have multiple else statements on the same line:

Example

One line if else statement, with 3 conditions:

```
a = 330
b = 330
print("A") if a > b else print("=") if a == b else print("B")
```

Check If Item Present in a Sequence - Usage of Membership Operator

The in operator is used to check if a value is present in a sequence (list, tuple, string etc.).

```
# list

L = ['red', 'green', 'blue']if 'red' in L:

    print('yes')# Prints yes

# tuple

T = ('red', 'green', 'blue')if 'red' in T:

    print('yes')# Prints yes

# string

S = 'Hello, World!'

if 'Hello' in S:

    print('Yes')# Prints yes
```