

CONDITIONS AND BRANCHING

If statements provide a way to execute a chosen block of code based on the run-time evaluation of one or more Boolean expressions.

if response:	Equivalent to	if response != '':
True if non empty False if empty		

Sr.No.	Syntax/command	Output	Remarks
IF and ELIF and ELSE and OR and AND conditions			
1	a = 10 if a == 10 : print("This is my first condition")	This is my first condition	Double == is an comparison operation whereas single = is a assignment operation and shall print the output only if the condition is true.
2	a = 10 if a > 5:	Error	If command without condition will give error
3	a = 10 if a > 5: pass	(blank)	In case you don't want to give condition and still execute the code then just put "pass" at the end.
4	a = 10 if a != 11 : print("my condition is false") else : print("print this")	my condition is false	First condition is 'a' not equal to 11 and shall print it if condition is true. If false then it will print condition 2.
5	a = 10 if a == 12 : print("here pare 12 with 10") elif a == 11: print('we mpare 11 with 10') else : print("if ail then do this")	if ail then do this	In case you want to provide more than 2 conditions the use 'elif' command in between.
6	price = 500 if price > 500 : print("may be able to buy a jacket") elif price <= 500 : print ("i will be to buy a jacket") else : print ('i dont know')	i will be to buy a jacket	Else command will not have any condition and will automatically include false condition. Even if there a multiple true condition in the code then system will execute only the first true statement.
7	if 1 in l : print("try to perform add") elif 2 in l: print('try to perform sub') elif 3 in l : print('try to perform multi') if 4 in l: print("this is new statement")	try to perform add this is new statement	If you use If' twice in the code then both of them will act as different blocs and both will be executed as separate conditional blocs.
8	b = input()	'1234'	To provide input from the customer side and the same will always be in string format.
9	b = int(input())	1234	Type casting opeartion :To get input in integer format because default input will always be in str format.

10	<pre>marks = int(input()) if marks >= 600 : print("You are qualified for GRE") else : print("You are not qualified for GRE") if marks < 500 : print("You can go for TOFFEL") if marks < 400 : print("try something else")</pre>		In case you enter if command inside if then both conditions will be executed. But if you enter if command separately then both will act as separate blocs.
11	<pre>a = 10 if a < 10 or a == 10: print("try to do something")</pre>	try to do something	In 'or' command if any one of the condition satisfies then it will give output for print.
12	<pre>a = "sudh" if a == 'sudh' and a == 'sudh': print("this is sudh")</pre>	this is sudh	But in 'and ' command all the conditions need to be satisfied only then it will give output for print.
LOOPS			

The for loop enables you to execute a code block multiple times. it is used for a controlled flow of repetition

Index-Based For Loops can be achieved using a built-in class named range that generates integer sequences In which case, identifier j is not an element of the data—it is an integer. data[j] can be used to retrieve the respective element.

while loop exists as a tool for repeated execution based on a condition. The code block will keep being executed until the given logical condition returns a False boolean value.

Continue command will stop the current loop body iteration and will start from the loop condition afresh.

Break command is used when you don't want the iteration after a particular condition is met.

Pass command will do nothing but just continue the loop.

FOR LOOP			
14	<pre>l = [1,2,3,4,'sudh', 223.3] for i in l: print(i)</pre>	<pre>1 2 3 4 sudh 223.3</pre>	It will assign i identifier to each and every element of list and print the elements of the list till the last element. 'For' command is used to extract all elements in list one by one in loop instead of extracting each element separately.
15	<pre>for i in l: if type(i) == int or type(i) == float: print(i+1) elif type(i) == str: print(i+str(1))</pre>	<pre>2 3 4 5 sudh1 224.3</pre>	To add a element to string that element has to be in 'str' format.
16	<pre>l1 = [] for i in l: if type(i) == int: l1.append(i)</pre>	l1 = [1, 2, 3, 4]	Initially l1 is a empty list. You assign each and every value of l to i and provided condition if int then add those elements to l1 list.
17	<pre>for i in l: pass</pre>		When command is given but you don't expect any kind of outcome.
18	<pre>for i in l: print(i) else : print("Once loop then it execute else")</pre>	<pre>1 2 3 4 sudh kumar 45.56 Once loop then it execute else</pre>	Like 'if + else' you can also use 'for + else' wherein else command will be execute only when loop is complete.

19	<pre>for i in l: if i == 4: break print(i) else: print("this will going complete itself")</pre>	<pre>1 2 3</pre>	As there is loop break at element 4, the loop command will stop at that point and will not be completed. Therefore else command also will not be executed. Only when loop execution is complete to every element else command will initiate.
20	<pre>t = (1,2,3,4,"str") for i in t: print(i)</pre>	<pre>1 2 3 4 str</pre>	"For" command can be used with tuples too.
21	<pre>b = {2, 4, 5, 6, 7, 8, 9, 'adit'} for i in b: print(i)</pre>	<pre>2 4 5 6 7 8 9 adit</pre>	"For" command can be used with sets too.
22	<pre>d = {'key1':'adit', 'cnt':'1626', "mail":"gmail.com"} for i in d: print(i)</pre>	<pre>key1 cnt mail</pre>	"For" command can be used with dict too but it will only print keys.
23	<pre>for i in d: print(d[i])</pre>	<pre>adit 1626 gmail.com</pre>	It will print dict values under each key.
24	<pre>for i in d: print(i,d[i])</pre>	<pre>key1 adit cnt 1626 mail gmail.com</pre>	It will print both key as well as values.
25	<pre>for i in d: print(i,":",d[i])</pre>	<pre>key1 : adit cnt : 1626 mail : gmail.com</pre>	
26	range(9)	range(0, 9)	It will produce data considering start default 0 to 8 excluding 9
27	list(range(9))	[0, 1, 2, 3, 4, 5, 6, 7, 8]	
28	<pre>for i in range(9): print(i)</pre>	<pre>0 1 2 3 4 5 6 7 8</pre>	in order to print data in range function you can also use for loop command.
29	<pre>n = 4 for i in range(4): for j in range(0,i+1): print("sudh", end=" ") print("\n")</pre>	<pre>sudh sudh sudh sudh sudh sudh sudh sudh sudh sudh</pre>	\n command is to get a new line whereas end = " " command is to get space in between. n = 4 means there are 4 rows

30	<pre>n = 4 for i in range(n): for j in range(i,n-1): print(" " * 4, end = " ") for j in range(i + 1): print("sudh", end = " ") for j in range(i): print("sudh" , end = " ") print("\n")</pre>	<pre>sudh sudh sudh sudh sudh sudh sudh sudh sudh sudh sudh sudh sudh sudh sudh sudh sudh</pre>	
31	<pre>s ="ineuron" for i in range(len(s)): print(i, s[i])</pre>	<pre>0 i 1 n 2 e 3 u 4 r 5 o 6 n</pre>	list(range(len(s))) will give list = [0,1,2,3,4,5,6,7] where range will be range(0,8)
32	<pre>l = ['sudh', 'kumar', 2324234, "adityabale@gmail.com" , 34 , 55435] l1 = [] for i in l: if type(i) == int: l1.append(i)</pre>	l1 = [2324234, 34, 55435]	It will filter out numbers from list l into list l1 using append function.
33	<pre>for i in d: print(i)</pre>		Will print all keys in the dict
WHILE LOOP			
34	<pre>a = 1 b = 10 while a <= b: print(a) a = a+1</pre>	<pre>1 2 3 4 5 6 7 8 9 10</pre>	it will first check whether the condition satisfies and will run in loop. The value of "a" will increment by 1 and will run through the loop until the condition satisfies.
35	<pre>a = 1 b = 10 while a <= b: print(a) a = a + 2 else: print("print this else block")</pre>	<pre>1 3 5 7 9 print this else block</pre>	It will go through the loop using while command and once loop is complete else command will be executed.
36	<pre>a = 1 b = 10 while a < b: print(a) if a == 5: break a = a + 1 else: print("print this")</pre>	<pre>1 2 3 4 5</pre>	It will break the loop once condition a = 5 is satisfied.
37	<pre>a = 1 b = 10 while a < b: print(a) if a == 3: continue a = a + 1</pre>	<pre>1 2 3 3 3 3 3</pre>	It will go into infinite loop once a = 3 and will print it indefinitely. "continue" command will not allow execution of program after it and will continue in loop.

38	<pre> a = 1 b = 10 while a < b: a = a + 1 if a == 3: continue print(a) </pre>	<pre> 2 4 5 6 7 8 9 10 </pre>	The programme will run in while loop and print the values. But when a = 3 due to continue command it will skip the print part and run in loop. Therefore 3 is not printed
39	<pre> a = 1 b = 10 while a < b: a = a + 1 if a == 3: pass print(a) </pre>	<pre> 2 3 4 5 6 7 8 9 10 </pre>	Pass command will do nothing but just continue the loop.
40	<pre> a = 0 while a < len(l): print(l[a]) a = a + 1 </pre>	<pre> 1 2 3 4 6 6 77 67 </pre>	To print each and every element present in a list. It is different from for loop as here elements are printed based on their indexes.
41	<pre> t = (1,2,3,4,5,6,7,8,9) a = 0 while a < len(t): if t[a] == 6 or t[a] == 7: print(a) a = a + 1 </pre>	<pre> 5 6 </pre>	to print indexes of 6 and 7
42	<pre> a = 0 while a < 1: if t[a] == 6 or t[a] == 7: print(a) a = a - 1 if a < -len(t): break </pre>	<pre> -3 -4 </pre>	to print indexes of 6 and 7 in reverse order
43	<pre> a = -1 while a >= -len(t): if t[a] == 6 or t[a] == 7: print("Index of" ,t[a], 'is', a) a = a-1 </pre>	<pre> -3 -4 </pre>	
44	<pre> a = -1 while a >= -len(t): if t[a] == 6 or t[a] == 7: print("Index of" ,t[a], 'is', a) a = a-1 </pre>	<pre> Index of 7 is -3 Index of 6 is -4 </pre>	
45	<pre> d = {"k1" : "sudh", "k2" : [1,2,3,4], "k3" : (2,3,4,5)} a = 0 b = list(d.keys()) while a < len(d): print(d[b[a]]) a = a + 1 </pre>	<pre> sudh [1, 2, 3, 4] (2, 3, 4, 5) </pre>	Extracting values from dict using while loop.

46	<pre>a = 0 while a < len(d.items()): l = list(d.items()) print(l[a][1]) a = a+1</pre>		
47	<pre>i = 0 v = 'aeiou' s = s.lower() while i < len(s): if s[i] in v: print(s[i]) i = i+1</pre>		It will print all the vowels from a sent/para. As python is case sensitive convert the string to lower case and then filter out.