Decision Making Statements (if,if-else,if-elif-else)

Statements	Syntax	Example	Definition
If	if condition: statement1 statement2	i = 10 if (i > 15): print ("10 is less than 15") print ("I am Not in if") **Output:** I am Not in if	if statement is the most simple decision making statement. It is used to decide whether a certain statement or block of statements will be executed or not

If - else	if (condition):
	statement1
	else:
	statement2
		i = 20;
		if (i < 15):
		print ("i is smaller than 15")
		print ("i'm in if Block")
		else:
		print ("i is greater than 15")
		print ("i'm in else Block")
		print ("i'm not in if and not in else Block")
		
		Output:
		i is greater than 15
		i'm in else Block
		i'm not in if and not in else Block
			We can use the else statement with if statement to execute a block of code when the condition is false.

nested-if	if (condition1):
	statement
	if (condition2):
	statement
	# if Block is end here
	# if Block is end here
		i = 10
		if (i == 10):
		if (i < 15):
		print ("i is smaller than 15")
		if (i < 12):
		print ("i is smaller than 12 too")
		else:
		print ("i is greater than 15")
		
		Output:
		i is smaller than 15
		i is smaller than 12 too
			A nested if is an if statement that is the target of another if statement. Nested if statements means an if statement inside another if statement.

if-elif-else	if (condition):
	    statement
	elif (condition):
	    statement
	    .
	    .
	else:
	    statement
		i = 20
		if (i == 10):
		  print ("i is 10")
		elif (i == 15):
		  print ("i is 15")
		elif (i == 20):
		  print ("i is 20")
		else:
		  print ("i is not present")
		
		Output:
		i is 20
			Here, a user can decide among multiple options. The if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the ladder is bypassed.

Comparison Operators (==,<,>,<=,>=)

Operator	Example	Meaning	Result
==	a == b	Equal to	True if the value of a is equal to the value of b False otherwise
!=	a != b	Not equal to	True if a is not equal to b False otherwise
<	a < b	Less than	True if a is less than b False otherwise
<=	a <= b	Less than or equal to	True if a is less than or equal to b False otherwise
>	a > b	Greater than	True if a is greater than b False otherwise
>=	a >= b	Greater than or equal to	True if a is greater than or equal to b False otherwise

Logical Operator

Operator	Example	Meaning
Not	not x	True if x is False False if x is True (Logically reverses the sense of x)
Or	x or y	True if either x or y is True False otherwise
and	x and y	True if both x and y are True

		False otherwise
not in	x not in y	x not in y, here not in results in a 1 if x is not a member of sequence y
In	x in y	x in y, here in results in a 1 if x is a member of sequence y

Bit-wise Operator

Operator	Example	Meaning	Result
<<	x << y	**bits shifted to the left**	Returns x with the bits shifted to the left by y places
>>	x >> y	**bits shifted to the right**	Returns x with the bits shifted to the right by y places
&	x & y	**bitwise and**	Each bit of the output is 1 if the corresponding bit of x AND of y is 1, otherwise it's 0
|	x | y	**bitwise or**	Each bit of the output is 0 if the corresponding bit of x AND of y is 0, otherwise it's 1
~	~ x	**complement of x**	Returns the complement of x - the number you get by switching each 1 for a 0 and each 0 for a 1
x ^ y	x ^ y	**Bitwise XOR operator**	Each bit of the output is the same as the corresponding bit in x if that bit in y is 0, and it's the complement of the bit in x if that bit in y is 1.

Others:

Data Type	Meaning
Booleans	**Boolean in Python can have two values – True or False**
Numbers	The numbers in Python are classified using the following keywords: **int, float, and complex**.
Strings	A sequence of one or more characters enclosed within either single quotes ' or double quotes " is considered as String in Python. Any letter, a number or a symbol could be a part of the sting.
Lists	Lists in Python can be declared by placing elements inside **square brackets separated by commas.**
Tuples	A tuple is a heterogeneous collection of Python objects, using enclosing parentheses ( ) having its elements separated by commas inside.
Sets	A set is an unordered collection of unique and immutable objects. Its definition starts with enclosing braces { } having its items separated by commas inside.
Dictionaries	**Python syntax for creating dictionaries use braces { } where each item appears as a pair of keys and values.**

List:

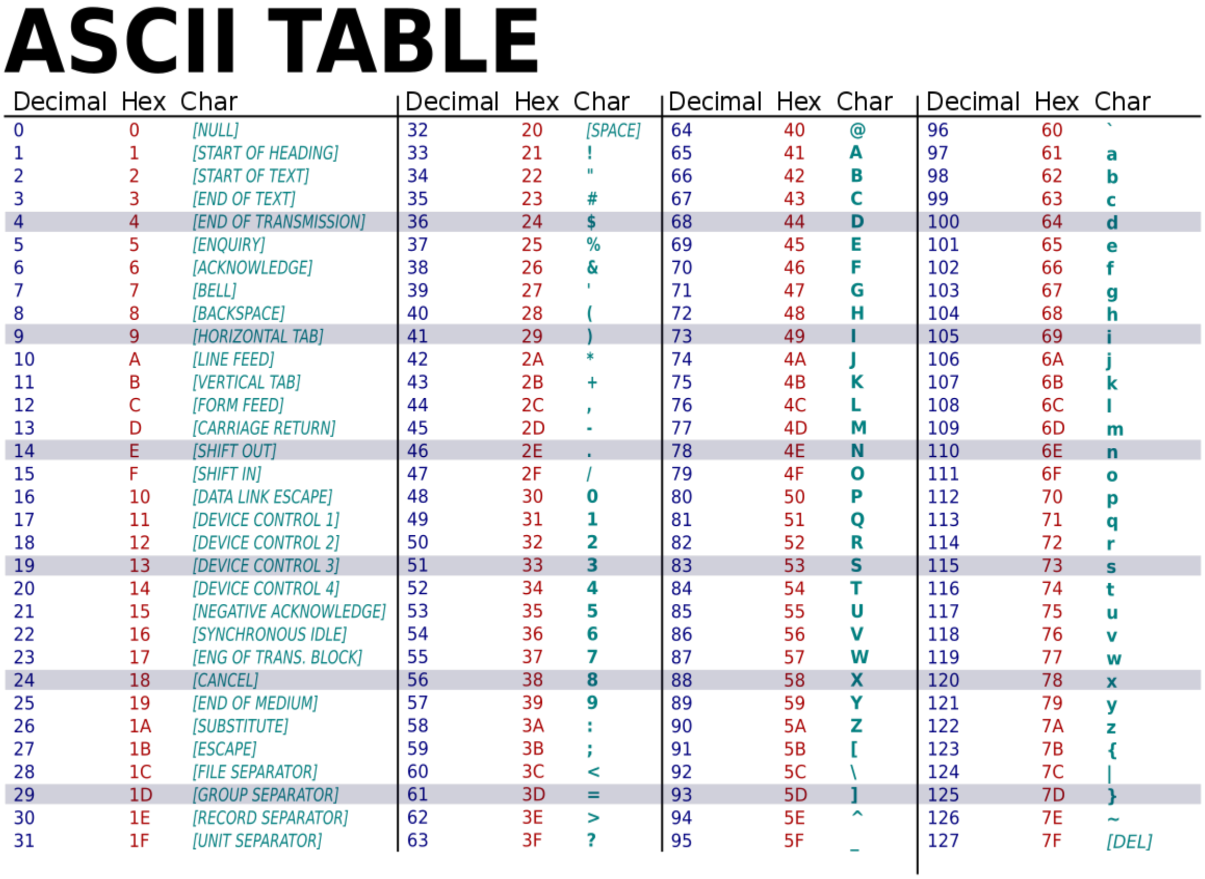
Lists are used to store multiple items in a single variable. Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage. Lists are created using square brackets:

```
Mylist= [‘apple’, ‘10’, ‘cost’, ‘120.5’]
print(Mylist)
```

Tuple:

Tuples are used to store multiple items in a single variable. A tuple is a collection which is ordered and **unchangeable**. Tuples are written with round brackets.

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```



ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

1. Predict the output:

   ```
   i=-4
   num=-4
   if(num<i):
      print(num*num)
   else:
      print(num)
   ```

2. Predict the output:

   ```
   i=-8
   num=-4
   if(num<i):
      print(num*num)
   print(num+num)
   ```

3.
   ```
   a = 10
   b = 12
   ans = a | b
   print(ans)
   ```

4.
   ```
   a = 5
   b = 9
   ans = a & b
   print(ans)
   ```

5.
   ```
   a= 11
   ans= ~a
   print(ans)
   ```

6. Predict the output:

   ```
   i = 20;
   if (i < 21):
      print ("i am greater than 20")
      print ("i'm in if Block")
   elif(i<22):
      print ("i is also greater than 20")
      print ("i'm in elif Block")
   else:
      print ("i'm not in if and not in else Block")
   ```

7. Predict the output:

   ```
   i = 20;
   ```

```
if (i < 14):
   print ("i is smaller than 14")
   print ("i'm in if Block")
else:
   print ("i is greater than 15")
   print ("i'm in else Block")
print ("i'm not in if and not in else Block")
```

8. Predict the output:

```
num = 1122
if 9 < num < 99:
   print("Two digit number")
elif 99 < num < 999:
   print("Three digit number")
elif 999 < num < 9999:
   print("Four digit number")
else:
   print("number is <= 9 or >= 9999")
```

9. Predict the output:

```
num = -99
if num > 0:
   print("Positive Number")
else:
   print("Negative Number")
   #nested if
   if -99<=num:
      print("Two digit Negative Number")
```

10. num=65

```
num2= ord('A')
if (num == num2):
   print("Equal Number")
else:
   print("not equal")
```

11. num=77

```
num2= ord('m')
if (num == num2):
   print("Equal Number")
```

```
else:
   print("not equal")
```

12. a=int(5)

```
b=float(5)
if(a==b):
   print("Equal number")
else:
   print("Not equal")
```

13. Predict the outputs:

```
assorted_list = [True, False, 1, 1.1, 1+2j, "Learn", "b","Python"]
first_element = assorted_list[0]
print(first_element)
first_element = assorted_list[3]
print(first_element)
first_element = assorted_list[3]
print(first_element)
print(assorted_list[5])
print(assorted_list)
```

14. Predict the outputs:

```
first_tuple = (3, 5, 7, 9)
print(type(first_tuple))
print(first_tuple)
```

15. Predict the outputs:

```
another_set = {"red", "green", "black"}
print(type(another_set))
print(another_set)
```

16. The fine charged in a library is as follow: if a student returns a book after 5 days of due date the fine is 50 rupees, if it is in between 6-10 days the fine is 100 rupees, if it is overdue by more than 10 days the fine is 500 rupees. Also, if the book is overdue by 1 semester then membership will be canceled. Now derive a solution for the whole scenario involving if else statements where number of days will be provided by the user.

17. You are given three sides of triangle; now check whether the triangle is isosceles, equilateral, scalene.

18. A newspaper is published using Cyan-Magenta-Yellow-Black (CMYK) color model. Usually these color formats are inherited by Red-Green-Blue (RGB) color model. The value of CMYK is varied on a real scale from 0-1. Provide a solution for RGB to CMYK color model conversion. The formulas are given:
White = Max(Red/255, Green/255, Blue/255)
Cyan = ((White - Red/255)/White)
Magenta = ((White – Green/255)/White)
Yellow = ((White – Blue/255)/White)
Black = 1 – white

19. The ratio of person's weight (kg) and square of height (m) is known as Body Mass Index (BMI). Derive a solution which will receive weight and height and provides the BMI.

BMI Category	**BMI**
Starvation	<15
Anorexic	15.1 to 17.5
Underweight	17.6 to 18.5
Ideal	18.6 to 24.9
Overweight	25 to 25.9
Obese	30 to 30.9
Morbidly Obese	>=40

20. Any character/ special symbol is entered, now check whether is it capital letter, small letter, a digit, or any special symbol.