

| | |
|----------|--|
| Page No. | |
| Date | |

College Name :

DELHI GLOBAL INSTITUTE OF TECHNOLOGY

Name : BAZGHA RAZI

Course Code : PCC-CSE-207 G

Subject : Python Programming.

Session : 2019 - 2023

Ans 1 a) Python variable. : It is a name that is used to refer to memory location. Variable names can be a group of both letters and digits, but they have to begin with a letter or an underscore.

Rules for declaring the variable in python:

- 1) All the characters except the first may be an alphabet of lower case (a to z), upper case (A to Z), digit (0-9) or underscore (_).
 - 2) The first character of the variable must be an alphabet or underscore.
 - 3) It must not contain any whitespace or any special characters (@, %, #, etc)
 - 4) It must not be similar to any keyword defined in the python.
 - 5) It is case sensitive.

Example : 81819, -8, -19, etc. are the valid variables.

%abc, 198, & b19, etc are the invalid variables.

WAP for displaying reversal of a number.

```
a = int(input("Enter the number: "))
```

$c = b = 0$;
while ($a > 0$):

$$b = a \% / 10$$

```
print('b, end='')
```

$$a = \text{int}(a / 10)$$

Output:

Enter the number: 1819

9181

Ans 1 e) Negative indexing in list

Negative indexing is counted from right. The last element of the list or the first element from the right has the index (-1), its adjacent left element is present at the index (-2), next to that element from right is present at the index (-3) and so on.

Example: $L = [1, 2, 3, 4, 5]$

Negative indexing of this list L is as follows:

`print(L[-1])`

`print(L[-4:-1])`

Output: $[5] \quad -5 \ -4 \ -3 \ -2 \ -1 \rightarrow$ Negative indexing
 $[2, 3, 4]$ of list L

WAP to insert element in the list.

$L = [1, 2, 3, 4, 5]$

`L.insert(3, "Thanks")`

`print(L)`

Output: $[1, 2, 3, \text{Thanks}, 4, 5]$

WAP to update element in the list.

$L = [1, 2, 3, 4, 5]$

`print(L)`

$L[0] = 50$

`print(L)`

Output: $[1, 2, 3, 4, 5]$
 $[50, 2, 3, 4, 5]$

→ Old list

→ Updated list

Ans 2(b) break is a keyword in python which is used to bring the program control out of the loop. The break statement breaks the loop one by one. The break is used to break the current execution of the program and control goes to the next line after the loop.

Syntax : loop statements
break;

Example: WAP to use break in the loop

```
s='python'  
count = 0;  
for i in s:  
    count
```

```
s='python'  
count = 0;  
for i in s:  
    count = count + 1  
    if i == 'h':  
        break;  
    print(i)
```

Output : pyt

Continue keyword is used to bring the program control to the beginning of the loop.

The continue statement skips the remaining lines of code inside the loop and start with next iteration.

Syntax: loop statement

continue

the code to be skipped

Example: WAP to use continue in the loop.

$s = "Python"$

for i in s:

if (i == 'a'):

continue

print(i)

Output: Python

WAP to find the power of a user-defined number.

a = int(input("Enter the number a:"))

b = int(input("Enter the number b:"))

c = a**b

print(c)

Output: Enter the number a: 3

Enter the number b: 2

(Ans 3a) Slicing operator is used in string to create substring. Slicing creates a new substring from the source string and original string remains unchanged.
Syntax : slice [start, stop, step].

start is the starting index where the slicing of string starts.

stop is the ending index where the slicing of string stops.

step determines the increment b/w each index for slicing, it is an optional argument.

We can also use colon (:) operator b/w start, stop and step.

Example : $s = \text{"Hello World"}$

$$s[0] = 'H'$$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|
| H | e | l | l | o | | W | o | r | l | d |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

$$s[0:4] = \text{'Hello'}$$

$$s[:] = \text{'Hello World'}$$

$$s[2:6] = \text{'llO'}$$

$$s[0:] = \text{'Hello World'}$$

$$s[3:11] = \text{'Hello World'}$$

$$s[7:10] = \text{'oWorld'}$$

Negative slicing also done in the string.
It starts from the rightmost character having index (-1) and second element from the right side is having index (-2) and so on.

Example: $s = \text{'Python'}$

$$s[-6:-3] = \text{'Pyt'}$$

$$s[-1] = \text{'n'}$$

$$s[-4:] = \text{'thon'}$$

$$s[-5] = \text{'y'}$$

$$s[::-1] = \text{nohtyP}$$

| | | | | | |
|----|----|----|----|----|----|
| P | y | t | h | o | n |
| -6 | -5 | -4 | -3 | -2 | -1 |

WAP to find the first 4 characters
and last 3 characters of the string:
"India won the match" to the console.

$s = \text{"India won the match"}$

print(s[0:4])

print(s[-3:])

Output: Indi
tch

| | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|
| -19 | -18 | -17 | -16 | -15 | -14 | -13 | -12 | -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 |
| I | n | d | i | a | w | o | n | t | h | e | m | a | t | C | h | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |