

Assignment - 4

Q-1 What are Conditional statements. Show with Example.

→ Conditional statements to executes based on the given condition

→ 1. if statement

→ An if statement is written by using keyword "if".

→ Syntax:

if condition:
statements

→ Eg:

a = 20

b = 5

if a > b:

print("a is greater than b")

→ output: a is greater than b

→ 2. if..else statement:

→ if..else statement is used to execute the statement if the given condition is true or false.

→ Syntax:

if condition:
true statement

else:
false statement

→ Eg:

a = 20, b = 10

if a > b:

```
print("a is greater than b")  
else :  
    print("b is greater than a")
```

→ Output : a is greater than b

→ 3. if...elif statement

→ if...elif statement is use to check multiple conditions and execute accordingly.

→ Syntax :

```
if condition :  
    statement =  
elif condition :  
    statement "
```

→ Eg :

```
if 5 > 10  
    print("False")  
elif 10 > 5  
    print("True")
```

→ output : True

→ 4. if...elif...else statement

→ The else keyword executes statements which isn't executed by the preceding conditions.

→ Syntax :

```
if condition :  
    statement
```

```
elif condition :
```


statements

else:

statements

→ Eg

a = 10, b = 20

if a > b:

print("a is largest")

elif b > a:

print("b is largest")

else:

print("both are equal")

→ output : b is largest

→ 5. Nested if

→ Using if statement inside if statement
called nested if

→ Syntax

if condition:

statements

if condition:

statements

else:

statement

→ Ex

x = 30

if x > 10:

if x > 20:

print("x is greater than 20")

else:

```
print("x is greater than 20 but  
not 20")
```

→ output: x is greater than 20

Q-2 Explain Iterative statements.

→ 1 While loop

→ While loop executes a set of statements until a condition is true.

→ First of all, the given condition is checked, if it is true then statements are executed.

→ When the condition becomes false then the given statements are not executed but the statement outside the body of while statement is executed.

→ Syntax:

while condition:
statements

→ Eg:

i = 1

while i < 3:

print(i)

i += 1

→ output

1

2

→ Nested while loop

→ while loop in while loop called nested while loop

→ Syntax:

while condition:

while statements

while condition:
statements

→ Eg.	Output
i = 0, j = 4	
while i <= 3 :	0 4
while j <= 5:	1 25
print(i + " " + j)	32 6
j = j + 1	3 7
i += 1	

→ break : break statement is used to exit from loop

→ if condition will be true then break statement execute to exit early from loop

→ Eg	Output:
i = 1	1
while i < 6:	2
print(i)	3
if i == 3:	
break	
i += 1	

→ Continue : continue statement is used to skipped any statement and go for next iteration

Eg	Output
i = 1	2
while i < 6	3
print if i == 2:	4
print(i) continue	5
i += 1	

→ 2. For loop

→ for loop is used for iterating over a sequence.

→ for loop can execute a set of statement once for each item in a list, tuple etc.

→ Syntax:

for iterator in sequence:

statements

Output

→ Eg

```
for i in 'Python':  
    print(i)
```

P

y

t

h

o

n

→ Nested Loops

→ A loop inside a loop called nested loop

→ Eg

```
for x in '123':
```

```
    for y in '45':
```

```
        print(x+y)
```

Output

1 4

1 5

2 4

2 5

3 4

3 5

→ Pass statement:

→ for loops can't be empty, but if you for some reason have a for loop with no content, put in the pass statement to avoid getting an error.

Q3 Explain Range function and its use.

→ To loop through a set of code a specified number of times, we can use the range()

→ The range() returns a sequence of numbers, starting from 0 by default and increment by 1 and ends at specific number

Example 1	Output
for x in range(4):	0
print(x)	1
	2
	3

Example 2 with starting value	Output
for x in range(2,4):	2
print(x)	3
	4

Example 3 with starting, ending, increment value	Output
for x in range(0,9,2)	0
print(x)	2
	4
	6
	8

→ The range() function defaults to 0 as a starting value, however it is possible to specify the starting value by adding a parameter: range(2,6) which means values from 2 to 6 but not including 6

→ Else in for Loop

→ The else keyword in a for loop specifies a block of code to be executed when the loop is finished:

→ Eg	Output
for x in range(3):	0
print(x)	1
else:	2
print('finish')	finish

→ Eg:	Output
for x in range(6):	0
if x == 3:	1
break	2
else: print(x)	
print('Finish')	
else:	
print('Finish')	

Q.4. List Different List methods with example.

sr no	Method	Explanation	Example for list	Output
			fruits: fruit = ['apple', 'banana', 'cherry']	
1.	append	Adds an element at the end of the list	fruit.append("orange") print(fruits)	['apple', 'banana', 'cherry', 'orange']
2.	clear	Remove all the element from list	fruit.clear() print(fruit)	[]
3.	copy	Returns a copy of the list	x = fruit.copy() print(x)	['apple', 'banana', 'cherry', 'orange']
4.	count	Returns the number of elements with the specified value	x = fruit.count('cherry') print(x)	1
5.	index	Returns the index of the first element with the specified value	x = fruit.index('cherry') print(x)	2

6.	insert	adds an element at the specified position	fruit.insert(1, 'orange') print(fruit)	['apple', 'orange', 'banana', 'cherry']
7.	pop	Remove element from specified index	fruit.pop(1) print(fruit)	['apple', 'banana', 'cherry']
8.	remove	Removes element by value	fruit.remove('banana') print(fruit)	['apple', 'cherry']
9.	reverse	Reverse the list	fruit.reverse() print(fruit)	['cherry', 'apple']
10.	sort	Sorts the list	a = ['x', 'a', 'd'] a.sort() print(a)	['a', 'd', 'x']