Python Conditional & Looping Statements

1 - Conditional Statements

- Conditional statements in Python allow you to control the flow of your program based on certain conditions. The main conditional statements in Python are:
  - if statement
  - if-else statement
  - if-elif-else statement
  - Nested if statements
  - Ternary (conditional) operator

1.1 if statement

In [ ]:
```
x = 7
if x > 5:
    print("x is greater than 5")
```

In [ ]:
```
x=10
if x>5:
    print('This print is inside if condition')
print('This print is outside if condition')
```

1.2 if-else statement

In [ ]:
```
x = 7
if x > 5:
    print("x is greater than 5")
else:
    print("x is not greater than 5")
```

1.3 if-elif-else statement

In [ ]:
```
x = 3
if x > 5:
    print("x is greater than 5")
elif x == 5:
    print("x is equal to 5")
else:
    print("x is less than 5")
```

1.4 Nested if statements

In [ ]:
```
x = 10
y = 5
if x > 5:
    print('First condition is satisfied')
    if y > 3:
        print("Both conditions are satisfied")
    else:
        print(' Only second condition is not satisfied')
else:
    print('First condition is not satisfied')
```

1.5 Ternary (conditional) operator

In [ ]:
```
x = 10
"Greater than 5" if x > 5 else "Not greater than 5"
```

Chained Comparison:

In [ ]:
```
x =18
if 10 < x < 20:
    print("x is between 10 and 20.")
```

Conditional Statements-Practice Questions

Q1. Write a code in Python that prints whether the customer can buy the a mobile Phone or not based on his budget.

- Input the budget.
- Mobile price is 30000

Q2. Suppose the price of the 32 GB, 64 GB, 128 GB and 256GB Phones are 15000, 20000, 30000, 40000 respectively. Write a conditional statement to print the price based on the internal storage of the phone. If the entered storage is not available, print NA

Q3. Write a program for a candidate's selection prodedure for an examination with below conditions are given:

- If the candidate score is above 80 in First Round, he/she is eligible to appear in Second Round else not.
- If the candidate score is above 70 in Second Round, he/she is eligible to appear in Third Round else not.
- If the candidate score is above 60 in Third Round, he/she is SELECTED else REJECTED.

Q4. Print the Group of student based on given criteria:

- If marks are more than or equaL to 80 - Group1
- If marks are more than or equaL to 60 & less 80 - Group2
- If marks are more than or equaL to 40 & less 60 - Group3

Q5 Nested Login System:

- Ask for username and Password of a user
- If username is 'admin' ask for password
- If username is incorrect print "incorrect username"
- If password is incorrect print "incorrect password"
- If password is 'admin123' print "login successful"

Solutions:

In [ ]:

```
#Q1
price=90000
budget = int(input('Enter your budget: '))
if price <= budget:
    print('yes')
else:
    print('No')
```

In [ ]:
```
#Q2
Memory = int(input('Enter the memory: '))
if Memory == 32:
    print('The price of the phone is 15000')
elif Memory == 64:
    print('The price of the phone is 20000')
elif Memory ==128:
    print('The price of the phone is 30000')
else:
    print('Please enter a valid memory requirement')
```

In [ ]:
```
#Q3
score1=int(input('Score of 1st round '))
if score1>=80:
    print('Move to 2nd round')

    score2=int(input('Score of 2nd round '))
    if score2>=70:
        print('Move to 3rd round')

        score3=int(input('Score of 3rd round '))
        if score3>=60:
            print('SELECTED')

        else:
            print("REJECTED")
    else:
            print(" Can't move to 3rd round ")
else:
    print("Can't Move to 2nd round ")
```

In [ ]:
```
#Q4
marks= int(input('Enter the marks'))
if marks>=80:
    print('Group1')
elif marks<80 and marks>=60:
    print('Group=2')
elif marks<60 and marks>=40:
    print('Group=3')
```

In [ ]:
```
#Q5
username= input('Enter username')
if username=='admin':
    password= input('Enter Password')
    if password=='admin123':
        print('Login Successful')
    else:
        print('Wrong password')
else:
    print('wrong username')
```

2 - Looping Statements

- Looping statements in Python allow you to repeatedly execute a block of code.
- There are two main types of loops in Python:
  - for Loop
  - while Loop

2.1- 'for' Loop :

- The for loop is used for iterating over a sequence (such as a list, tuple, string, or range) or other iterable objects.

In [4]:
```
# Iterating over a list:
fruits = ["apple", "banana", 65]
for i in fruits:
    print(i)
```

```
apple
banana
65
```

In [ ]:
```
# Using range for numerical loops:
for i in range(10,20,3):
    print(i)
```

In [ ]:
```
# Iterating over a string:
word = "Python"
for i in word:
    print(i)
```

In [ ]:
```
#for loop with else
num=[1,2,3,4,5]
for i in num:
    print(i)
else:
    print("no items in list")
```

Practice Questions - for loop

Q1 Write a code in Python that prints the discounted price for each of the below-mentioned discounts;

Suppose the price of the phone is 40000. Calculate the prices of this phone after providing 5%, 10%, 15%, and 20% discounts respectively.

- A. Without using any loop
- B. Using for loop

In [6]:
```
price= 40000
print('price after 5% discount', price-5/100*price)
print('price after 10% discount', price-10/100*price)
print('price after 15% discount', price-15/100*price)
print('price after 20% discount', price-20/100*price)
```

```
price after 5% discount 38000.0
price after 10% discount 36000.0
price after 15% discount 34000.0
price after 20% discount 32000.0
```

In [7]:
```
price= 40000
for i in [5,10,15,20]:
    print('price after',i, '% discount', price-i/100*price)
```

```
price after 5 % discount 38000.0
price after 10 % discount 36000.0
price after 15 % discount 34000.0
price after 20 % discount 32000.0
```

In [10]:
```
price= 40000
for i in range(5,21,5):
    print('price after',i,'% discount',
          price-i/100*price)
```

```
price after 5 % discount 38000.0
price after 10 % discount 36000.0
price after 15 % discount 34000.0
price after 20 % discount 32000.0
```

Q2. Print the multiplication table of 5 using :

- for loop

In [ ]:
```
num = 5
for i in range(1, 11):
    print(num, "x", i, "=", num * i)
```

Q3. Print even numbers from 2 to 10 using :

- A. for loop
- B. Both for and if without using step in range function

In [ ]:
```
for i in [2,4,6,8,10]:
    print(i)
```

In [ ]:
```
for i in range(2,11,2):
    print(i)
```

In [ ]:
```
for i in range(2,11):
    if i%2==0:
        print(i)
```

2.2- 'while' Loop :

- The while loop is used for repeatedly executing a block of code as long as a given condition is true.

In [ ]:
```
# Simple while loop:
count = 0
while count < 5:
    print(count)
    count += 1
```

In [ ]:
```
i=0
while i in range(5):
    print(i)
    i +=1
```

In [ ]:
```
#while loops can also have an else clause.
#The else block is executed when the loop condition becomes False.
count = 0
while count < 5:
    print(count)
    count += 1
else:
    print("Loop condition is False.")
```

Practice Question- While loop

Q1 Write a code in Python that prints the discounted price for each of the below-mentioned discounts;

- using while loop Suppose the price of the phone is 40000. Calculate the prices of this phone after providing 5%, 10%, 15%, and 20% discounts respectively.

In [ ]:
```
#Q1 Using while Loop
Price = 40000
i = 5
while i <= 20:
    discounted_price = Price - Price * (i / 100)
    print(discounted_price)
    i =i + 5
```

Q2. Print the multiplication table of 5 using :

- while loop

In [ ]:
```
count=1
while count<=10:
    print('5 X',count,'=',5*count)
    count +=1
```

Q3. Print even numbers from 2 to 10 using :

- while loop

Any and All functions:

- The any() and all() functions work with iterable conditions.
- any() returns True if at least one element in the iterable is True
- all() returns True if all elements are True.

In [ ]:
```
values = [False, 0, '', True,None, 5, 'Hello']
print(any(values))
print(all(values))
```

In [ ]:
```
values = []
print(any(values))
print(all(values))
```

Q. Given a list of integers, write a Python expression using any() to check if at least one element is greater than 10.

In [ ]:
```
numbers = [5, 8, 12, 3, 7]
result = any(x > 10 for x in numbers)
print(result)
```

Q. Write a code to check and print whether a list has all even numbers or not

In [ ]:
```
numbers = [2, 4, 6, 8, 10,12]
if all (x % 2 == 0 for x in numbers):
    print("All numbers are even.")
else:
    print("all numbers are not even")
```

Q. Write a code to check and print whether a list has atleast one even number

In [ ]:
```
numbers = [1,3,5,7,4]
if any (x % 2 == 0 for x in numbers):
    print("One number are even.")
else:
    print("all numbers are not even")
```

Q. Create a list of boolean values and use all() to check if all values are True.

In [ ]:
```
boolean_values = [True, True, True, True]
result = all(boolean_values)
print(result)
```

zip() function

- Use this to iterate over multiple iterables simultaneously.

In [ ]:
```
#Pairing elements from two lists:
names = ['Alice', 'Bob', 'Charlie']
ages = [25, 30, 22]
for name, age in zip(names, ages):
    print(name,age)
```

In [ ]:
```
# Combining three lists:
cities = ["New York", "Paris", "Tokyo"]
populations = [8175133, 2243833, 13929286]
countries = ["USA", "France", "Japan"]

combined_data = list(zip(cities, populations, countries))
print(combined_data)
```

In [ ]:
```
# Unzipping data:
data = [('apple', 3), ('banana', 5), ('orange', 2)]
fruits, quantities = zip(*data)
print(fruits)
print(quantities)
```

In [ ]:
```
# Iterating over multiple lists simultaneously:
names = ["Alice", "Bob", "Charlie"]
ages = [25, 30, 22]
for name, age in zip(names, ages):
    print(name, 'is',age, 'years old.')
```

enumerate() function

- The enumerate() function allows you to iterate over both the index and the value of an iterable simultaneously.

In [ ]:
```
fruits = ['apple', 'banana', 'cherry']
for index, value in enumerate(fruits):
    print(index,value)
```

In [ ]:
```
# Starting Index at a Specific Value:
colors = ["red", "green", "blue"]
for index, color in enumerate(colors, start=3):
    print(index,color)
```

In [ ]:
```
# Using Enumerate with Strings:
message = "Hello"
for index, char in enumerate(message):
    print(index,char)
```

****