

MD: ARSHAD </p>

ALISHARMEEN02@GMAIL.COM </p>

1. Write a program to accept percentage from the user and display the grade according to the following criteria: Marks Grade

90 ----A, 80 and <=90 ----B, =60 and <=80 ----C , below 60 -----D

In [13]: def DIVISION(a):
if a>90:
print("you have percentage ",a, "and your division is A")
elif a<=90 and a>80:
print("you have percentage ",a , "and your division is B")
elif a<=80 and a>=60:
print("you have percentage ",a, "and your division is c")
elif a<=60:
print("you have percentage ",a , "and your division is D")

In [6]: A = int(input("enter your percentage here :- "))
DIVISION(A)

enter your percentage here :- 91
you have percentage , 91 and your division is A

In [7]: A = int(input("enter your percentage here :- "))
DIVISION(A)

enter your percentage here :- 90
you have percentage , 90 and your division is B

In [8]: A = int(input("enter your percentage here :- "))
DIVISION(A)

enter your percentage here :- 85
you have percentage , 85 and your division is B

In [9]: A = int(input("enter your percentage here :- "))
DIVISION(A)

enter your percentage here :- 80
you have percentage , 80 and your division is c

In [10]: A = int(input("enter your percentage here :- "))
DIVISION(A)

enter your percentage here :- 75
you have percentage , 75 and your division is c

In [11]: A = int(input("enter your percentage here :- "))
DIVISION(A)

enter your percentage here :- 60
you have percentage , 60 and your division is c

In [14]: A = int(input("enter your percentage here :- "))
DIVISION(A)

enter your percentage here :- 59
you have percentage , 59 and your division is D

1. Write a program to accept the cost price of a bike and display the road tax to be paid according to the following criteria:

In [15]: def Tax(a):
if a>100000:
print("your cost price of a bike is ",a, "he road tax to be paid 15%")
elif a<=100000 and a>50000:
print("your cost price of a bike is ",a, "he road tax to be paid 10%")
elif a<=50000:
print("your cost price of a bike is ",a, "he road tax to be paid 5%")

In [16]: a = int(input("enter your percentage here :- "))
Tax(a)

enter your percentage here :- 115555
your cost price of a bike is , 115555 he road tax to be paid 15%

In [17]: a = int(input("enter your percentage here :- "))
Tax(a)

enter your percentage here :- 100000
your cost price of a bike is , 100000 he road tax to be paid 10%

In [18]: a = int(input("enter your percentage here :- "))
Tax(a)

enter your percentage here :- 50000
your cost price of a bike is , 50000 he road tax to be paid 5%

1. Accept any city from the user and display monuments of that city.

In [25]: def City(a):
if a=="Delhi" or a=="delhi":
print("monuments of that city , Red Fort")
elif a=="Agra" or a=="agra":
print("monuments of that city , Taj Mahal")
elif a=="Jaipur" or a=="jaipur":
print("monuments of that city , Jal Mahal")
else:
print("sorry ! you have entered wrong input")

In [26]: a = (input("enter your percentage here :- "))
City(a)

enter your percentage here :- delhi
monuments of that city , Red Fort

In [27]: a = (input("enter your percentage here :- "))
City(a)

enter your percentage here :- Delhi
monuments of that city , Red Fort

In [28]: a = (input("enter your percentage here :- "))
City(a)

enter your percentage here :- Agra
monuments of that city , Taj Mahal

In [29]: a = (input("enter your percentage here :- "))
City(a)

enter your percentage here :- agra
monuments of that city , Taj Mahal

In [30]: a = (input("enter your percentage here :- "))
City(a)

enter your percentage here :- jaipur
monuments of that city , Jal Mahal

In [31]: a = (input("enter your percentage here :- "))
City(a)

enter your percentage here :- Jaipur
monuments of that city , Jal Mahal

In [32]: a = (input("enter your percentage here :- "))
City(a)

enter your percentage here :- Mumbai
sorry ! you have entered wrong input

1. Check how many times a given number can be divided by 3 before it is less than or equal to 10.

In [4]: def Check():
a = int(input("enter the number here :-"))
b = a//3
print(b-3,"many times a given number can be divided by 3 before it is less than or equal to 10")

In [6]: Check()

enter the number here :-15
2 many times a given number can be divided by 3 before it is less than or equal to 10

In [8]: Check()

enter the number here :-9
0 many times a given number can be divided by 3 before it is less than or equal to 10

In [10]: Check()

enter the number here :-20
3 many times a given number can be divided by 3 before it is less than or equal to 10

In [11]: Check()

enter the number here :-25
5 many times a given number can be divided by 3 before it is less than or equal to 10

In [7]: Check()

enter the number here :-18
3 many times a given number can be divided by 3 before it is less than or equal to 10

1. Why and When to Use while Loop in Python give a detailed description with example

A while loop in Python is used to repeatedly execute a block of code as long as a specified condition is true. It allows you to create iterative processes that continue until a certain condition is met. This type of loop is particularly useful when you don't know in advance how many times the loop needs to run, and you want to keep iterating until a specific condition is satisfied.

Here's a detailed description of why and when to use a while loop in Python, along with an example:

Why Use a while Loop:

Dynamic Iteration: Unlike a for loop, which is better suited for known and fixed iteration counts, a while loop is suitable when the number of iterations is not predetermined and depends on a specific condition. It provides more flexibility for handling changing situations.

Conditional Execution: while loops allow you to repeatedly execute a block of code as long as a certain condition remains true. This is useful for situations where you want to keep performing a task until a certain criteria is met.

Continuous Monitoring: while loops are commonly used for monitoring real-time data or user input. You can keep checking for new data or responses and react accordingly.

When to Use a while Loop:

Input Validation: You can use a while loop to repeatedly prompt the user for input until they provide valid data. This ensures that your program doesn't proceed with incorrect or unexpected values.

Implementing Algorithms: Some algorithms require dynamic iteration until a specific convergence or termination condition is met. A while loop can be used to repeatedly update values or perform calculations until the desired result is achieved.

Game Loops: In game development, a while loop is often used to create the main game loop. The loop keeps running as long as the game is still active, updating the game state and handling user input.

Resource Management: When dealing with limited resources or quotas, you can use a while loop to perform actions until the resource is depleted or a specific condition is met.

Example: Calculating Factorial Using a While Loop:

Let's consider an example where we want to calculate the factorial of a given positive integer using a while loop.

In [5]: def factorial(n):
if n < 0:
return "Factorial is not defined for negative numbers"
elif n == 0:
return 1
else:
result = 1
while n > 0:
result *= n
n -= 1
return result

num = int(input("Enter a positive integer: "))
print(f"The factorial of {num} is {factorial(num)}")

Enter a positive integer: 7
The factorial of 7 is 5040

In this example, the while loop keeps multiplying the result by the value of n and then decrementing n until n becomes 0. This calculates the factorial of the input number.

Remember to use caution when using while loops to avoid infinite loops. Ensure that the loop's condition eventually becomes False to prevent your program from getting stuck.

1. Use nested while loop to print 3 different pattern.

Certainly! Here are three different patterns that you can print using nested while loops in Python: Pattern 1: Right Triangle Pattern This pattern prints a right triangle of stars.

In [6]: def print_right_triangle(n):
i = 1
while i <= n:
j = 1
while j <= i:
print(" *", end=" ")
j += 1
print()
i += 1

print("Pattern 1: Right Triangle")
print_right_triangle(5)

Pattern 1: Right Triangle
* *
* * *
* * * *
* * * * *

In [7]: #Pattern 2: Hollow Rectangle Pattern
#This pattern prints a hollow rectangle of stars.
def print_hollow_rectangle(rows, cols):
i = 1
while i <= rows:
j = 1
while j <= cols:
if i == 1 or i == rows or j == 1 or j == cols:
print(" *", end=" ")
else:
print(" ", end=" ")
j += 1
print()
i += 1

print("Pattern 2: Hollow Rectangle")
print_hollow_rectangle(5, 7)

Pattern 2: Hollow Rectangle
* * * * * * *
* * * * *
* * * * *
* * * * *
* * * * * *

In [8]: #Pattern 3: Pyramid Pattern
#This pattern prints a pyramid of stars.
def print_pyramid(n):
i = 1
while i <= n:
j = 1
while j <= n - i + 1:
print(" *", end=" ")
j += 1
j = 1
while j <= 2 * i - 1:
print(" *", end=" ")
j += 1
print()
i += 1

print("Pattern 3: Pyramid")
print_pyramid(4)

Pattern 3: Pyramid
*
* * *
* * * *
* * * * *

1. Reverse a while loop to display numbers from 10 to 1.

In [12]: #To display numbers from 10 to 1 using a while loop, you can start the loop with an initial
#value of 10 and decrement the value in each iteration. Here's how you can achieve this:
def reverse_countdown(start):
while start >= 1:
print(start, end=" ")
start -= 1

print("Countdown from 10 to 1:")
reverse_countdown(10)

Countdown from 10 to 1:
10,9,8,7,6,5,4,3,2,1,

1. Reverse a while loop to display numbers from 1 to 10

In [13]: def countup(start, end):
while start <= end:
print(start, end=",")
start += 1

print("Countup from 1 to 10:")
countup(1, 10)

Countup from 1 to 10:
1,2,3,4,5,6,7,8,9,10,

Thank You ,That's All </p>