In [21]:

```
#string selection from start point to the given end point
title = 'Amazing'
print(title[0:5])
```

```
Amazi
```

In [17]:

```
#entire string with start point but no end reference
title = 'Amazing'
print(title[2:])
```

```
azing
```

In [18]:

```
#entire string with no start reference but with end point
title = 'Amazing'
print(title[:5])
```

```
Amazi
```

In [27]:

```
#string with start reference and dynamic end point
title = 'Amazing'
print(title[2:-2])
```

```
azi
```

In [20]:

```
#veriable copy and print
title = 'Amazing'
Category = title[:]
print(Category)
```

```
Amazing
```

In [29]:

```
#veriable methods upper/lower
title = 'Amazing'
print(title.lower())
```

```
amazing
```

In [42]:

```
# fuction producses text in proper case"
title = 'Amazing Day'
print('title'.title())
```

```
Title
```

In [30]:

```
#veriable methods replace
title = 'Amazing'
print(title.replace('Amazing', 'Amazing Day'))
```

```
Amazing Day
```

In [33]:

```
#veriable methods find returns index of the length
title = 'Amazing Day'
print(title.find('ing'))
```

```
4
```

In [41]:

```
# producses boolean value "true or false"
title = 'Amazing Day'
print('Day' in title)
```

```
True
```

In [43]:

```
#Operators division returs float
print(10 / 3)
```

```
3.3333333333333335
```

In [44]:

```
#Operators division returs int
print(10 // 3)
```

```
3
```

In [45]:

```
#Operators division modulis returs remainder
print(10 % 3)
```

```
1
```

In [48]:

```
#Operators multiplication

print(10 * 3)
```

```
30
```

In [54]:

```
#Operators power

print(2 ** 3)
```

```
8
```

In [51]:

```
#Operators augmented/incremented assignment
x=10
x += 3
print(x)
```

```
13
```

In [55]:

```
#operators calculations
#1-parenthesis
#2-exponentiation
#3-multiplication or division
#4-addition or substraction

x=(10+2) * 2 ** 3

print(x)
```

```
96
```

In [56]:

```
#Math fuctions round off
x = 5.4
print(round(x))
```

```
5
```

In [57]:

```
#Math fuctions ABS-absolute returns positive number
x = -5.4
print(abs(x))
```

```
5.4
```

In [58]:

```
#Math module for using built in calculations used as math.
import math
math.
```

In [66]:

```
#for Loop for known criteria with start, end and step value

for i in range(10,0,-1):

    print(i)
```

```
10
9
8
7
6
5
4
3
2
1
```

In [67]:

```
#While Loop unknown criteria
number=int(input("enter number"))
rating=1
while rating<=10:
    product=number*rating
    print(number, 'x', rating, '=', product)
    rating=rating+1
```

```
enter number2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

In [68]:

```
number=int(input("enter number"))
rating=1
while rating<=number:
    product=number*rating
    print('Test')
    rating=rating+1
```

```
enter number5
Test
Test
Test
Test
Test
```

In [12]:

```
#If Statements
price = 1000000
good_credit = False
avg_credit = False
if good_credit:
    down_payment = 0.1 * price
elif avg_credit:
    down_payment = 0.2 * price
else:
    down_payment = 0.3 * price
print(f"Down Payment: ${down_payment}")
```

```
Down Payment: $300000.0
```

In [45]:

```
#logical Operators
high_income = True
good_credit = False
if high_income and good_credit:
    eligibility = "Eligible for loan"
elif high_income:
    eligibility = "Not eligible due to bad credit"
elif good_credit:
    eligibility = "Not eligibile due to low income"
else:
    eligibility = "Not eligible due low income and bad credit"
    print(f"Eligibility:{eligibility}")
```

In [44]:

```
#logical Operators
high_income = True
good_credit = False
if high_income and good_credit:
    eligibility = "Eligible for loan"
elif high_income and not good_credit:
    eligibility = "Not eligible due to bad credit"
elif good_credit and not high_income:
    eligibility = "Not eligibile due to low income"
else:
    eligibility = "Not eligible due low income and bad credit"
    print(f"Eligibility:{eligibility}")
```

In [46]:

```
#Comparison Operator
temperature = 30
if temperature > 30:
    print("Hot Day")
else:
    print("Good Day")
```

```
Good Day
```

In [50]:

```
weight = int(input('Enter your weight :'))
unit = input('unit in (L)bs or (K)g :')
if unit.upper() == "L":
    converted = weight * 0.45
    print(f"Your weight is {converted} kilos")
else:
    converted = weight / 0.45
    print(f"Your weight is {converted} pounds")
```

```
Enter your weight :98
unit in (L)bs or (K)g :l
Your weight is 44.1 kilos
```

In [56]:
```
#Loop While numerical sequence
i = 1
while i <=5:
    print(i)
    i +=1
print("end")
```

```
1
2
3
4
5
```

In [63]:
```
i = 1
while i <=5:
    print('*' * i)
    i +=1
print("end")
```

```
*
**
***
****
*****
end
```

In [72]:
```
number = 8
attempt_count = 0
attempt_limit = 3
while attempt_count < attempt_limit:
    enter_number = int(input("Enter number :"))
    attempt_count +=1
    if enter_number == number:
        print("You are right")
        break
    else:
        print("Retry")
```

```
Enter number :1
Retry
Enter number :4
Retry
Enter number :8
You are right
```

In [78]:

```
command = ""
started = False
while True:
    command = input("> ").lower()
    if command == "start":
        if started:
            print("Its already started")
        else:
            started = True
            print("Its Started...!")
    elif command == "stop":
        if not started:
            print("Its already stopped")
        else:
            started = False
            print("Its Stopped...!")
    elif command == "help":
        print("""
start - Use start option
stop - Use stop option
quit - Use quit option
        """)
    elif command == "quit":
        break
    else:
        print("Sorry use the keyword from available option")
```

```
> help

start - Use start option
stop - Use stop option
quit - Use quit option

> start
Its Started...!
> start
Its already started
> stop
Its Stopped...!
> stop
Its already stopped
> quit
```

In [89]:
```
#Loop - For
prices = [10, 20, 30]
total = 0
for cart in prices:
    total += price
    print(f"Cart Total: {total}")
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-89-095a9aae4755> in <module>
      3 total = 0
      4 for cart in prices:
----> 5     total += price
      6     print(f"Cart Total: {total}")

TypeError: unsupported operand type(s) for +=: 'int' and 'tuple'
```

In [90]:
```
for i in range(5,0,-1):

    print(i)
```

```
5
4
3
2
1
```

In [93]:
```
#Nested Loop
for x in range(4):
    for y in range(3):
        print(f"({x}, {y})")
```

```
(0, 0)
(0, 1)
(0, 2)
(1, 0)
(1, 1)
(1, 2)
(2, 0)
(2, 1)
(2, 2)
(3, 0)
(3, 1)
(3, 2)
```

In [102]:
```
#For loop using * to draw F
numbers = [5,2,5,2,2]
for x_count in numbers:
    print ('x' * x_count)
```

```
xxxxx
xx
xxxxx
xx
xx
```

In [108]:

```
#For loop using Nested loop to draw F
numbers = [5, 2, 5, 2, 2]
for x_count in numbers:
    output = ' '
    for count in range(x_count):
        output += 'x'
    print(output)
```

```
 xxxxx
 xx
 xxxxx
 xx
 xx
```

In [109]:

```
#Finding Maximum number from a list
numbers = [2,5,15,20,4,6,9]
max = numbers[0]
for max_number in numbers:
    if max_number > max:
        max = max_number
print(max)
```

```
20
```

In [115]:

```
#Replace/rename value of item in a list
names = ['a','b','c','d','e']
names[2] = 'G'
print(names)
```

```
['a', 'b', 'G', 'd', 'e']
```

In [116]:

```
#List Functions - Operations in a list
numbers = [4,3,7,4,6,1,2]
numbers.append(20)
print(numbers)
```

```
[4, 3, 7, 4, 6, 1, 2, 20]
```

In [117]:

```
#Insert
numbers = [4,3,7,4,6,1,2]
numbers.insert(4,21)
print(numbers)
```

```
[4, 3, 7, 4, 21, 6, 1, 2]
```

In [118]:

```
#Remove item
numbers = [4,3,7,4,6,1,2]
numbers.remove(6)
print(numbers)
```

```
[4, 3, 7, 4, 1, 2]
```

In [119]:

```
#Clear list
numbers = [4,3,7,4,6,1,2]
numbers.clear()
print(numbers)
```

```
[]
```

In [121]:

```
#Pop - remove last item
numbers = [4,3,7,4,6,1,2]
numbers.pop()
print(numbers)
```

```
[4, 3, 7, 4, 6, 1]
```

In [123]:

```
#Index - returns index of first occurance of the number
numbers = [4,3,7,4,6,1,2]
print(numbers.index(7))
```

```
2
```

In [ ]:

```
#Pop - remove last item
numbers = [4,3,7,4,6,1,2]
numbers.pop()
print(numbers)
```

In [124]:

```
#in - check the existance of the charecter
numbers = [4,3,7,4,6,1,2]
print(40 in numbers)
```

```
False
```

In [126]:

```
#Count - check the count of the numbers
numbers = [4,3,7,4,6,1,2]
print(numbers.count(4))
```

```
2
```

In [127]:

```
#Sort list ascending
numbers = [4,3,7,4,6,1,2]
numbers.sort()
print(numbers)
```

```
[1, 2, 3, 4, 4, 6, 7]
```

In [130]:

```
#Sort list descending
numbers = [4,3,7,4,6,1,2]
numbers.sort()
numbers.reverse()
print(numbers)
```

```
[7, 6, 4, 4, 3, 2, 1]
```

In [133]:

```
#Copy list
numbers = [4,3,7,4,6,1,2]
numbers2=numbers.copy()
numbers.append(10)
print(numbers2)
```

```
[4, 3, 7, 4, 6, 1, 2]
```

In [134]:

```
#Remove duplicates
numbers = [4,3,3,7,2,4,4,6,1,2]
unique=[]
for number in numbers:
    if number not in unique:
        unique.append(number)
print(unique)
```

```
[4, 3, 7, 2, 6, 1]
```