In [1]:
```
salary=eval(input('enter the salary:'))
tax_per=eval(input('tax percentage:'))
tax_pay=salary*tax_per/100
print(tax_pay)
```

```
10000.0
```

In [ ]:
```
firm 500 *4=2000

def TAX_PAY():
    salary=eval(input('enter the salary:'))
    tax_per=eval(input('tax percentage:'))
    tax_pay=salary*tax_per/100
    print(tax_pay)

TAX_PAY ======= 500

1time
```

Loops

- Loops are used to itearte the code multiple times
- functions are used to reuse the code block
- that code block can be repeated multiple times
- whenever you are doing same task multiple times then think about loop
- We have two loops
  - for loop
  - while loop
- Any loop we need three things
  - Intialization (start point)
  - increment or decrement
  - Condition to stop

For loop

In [ ]:
```
for <variable> in range(<number>):
    <write your code here>
```

In [2]:
```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

In [5]:
```
# print(0)
# print(1)
# print(2)
for trump in range(3):
    print(trump)
```

```
0
1
2
```

In [9]:
```
# print(0,end=' ')
# print(1,end=' ')
# print(2)
for i in range(3):
    print(i,end=' ')
```

```
0 1 2
```

case-1:range(stop)

- start=0 : python index always starts with zero
- increment by 1
- last=stop-1

In [10]:
```
for i in range(10):
    print(i,end=' ')
```

```
0 1 2 3 4 5 6 7 8 9
```

In [11]:
```
print('good night')
print('good night')
print('good night')
```

```
good night
good night
good night
```

In [12]:
```
for i in range(3):
    print('good night')
```

```
good night
good night
good night
```

Case-2:range(start,stop)

- Suppose I want to start my loop with a particular number
- then add start also inside range
- start=start

- increment by 1
- last= stop-1

range(10,20)

- start=10,increment by1 , last=20-1=19

In [13]:
```
for i in range(10,20):
    print(i,end=' ')
```

```
10 11 12 13 14 15 16 17 18 19
```

In [14]:
```
# wap ask the user print the square first 5 number
# square of 1 is 1
# square of 2 is 4
# so on

for i in range(1,6):
    print(f"the square of {i} is:{i*i}")
```

```
the square of 1 is:1
the square of 2 is:4
the square of 3 is:9
the square of 4 is:16
the square of 5 is:25
```

In [15]:
```
# wap ask the user enter a number the prompt should happen 5 times
# and print the square of the number
for i in range(5):
    num=eval(input('enter the number:'))
    print(f"the square of {num} is:{num*num}")
```

```
the square of 20 is:400
the square of 30 is:900
the square of 40 is:1600
the square of 50 is:2500
the square of 60 is:3600
```

In [16]:
```
def square():
    num=eval(input('enter the number:'))
    print(f"the square of {num} is:{num*num}")

for i in range(2):
    square()
```

```
the square of 30 is:900
the square of 50 is:2500
```

In [18]:
```
for i in range(133,136):
    num=eval(input('enter the number:'))
    print(f"the square of {num} is:{num*num}")
```

```
the square of 4 is:16
the square of 5 is:25
the square of 6 is:36
```

Case-3: range(start,stop,step)

- start=start only

- step:
  - look about step sign is it postive or negative sign
  - positive means increment
    - last:stop-1
  - negative means decrement
    - last:stop+1

In [19]:
```
# example1: range(3,19,2)
# start=3
# step=+2 postive direction
# last = 19-1=18
# 3,5,7,9,11,13,15,17
for i in range(3,19,2):
    print(i,end=' ')
```

```
3 5 7 9 11 13 15 17
```

In [20]:
```
# example2: range(3,19,-2)
# start=3
# step=-2 negative direction
# last = 19+1=20
for i in range(3,19,-2):
    print(i)
```

In [21]:
```
for i in range(3,-19,-2):
    print(i,end=' ')

# start=3  dire =nega  last= stop+1= -19+1=-18
```

```
3 1 -1 -3 -5 -7 -9 -11 -13 -15 -17
```

In [22]:
```
for i in range(3,-19,2):
    print(i,end=' ')
```

In [ ]:
```
range(3,15,3) # P
range(3,15,-3) # np
range(3,-15,3) # np
range(-3,15,3) # p
range(3,-15,-3) # p
range(-3,-15,3) # np
range(-3,15,-3) # np
range(15,3,3) # np
range(15,3,-3) # p
range(15,-3,3) # np
range(-15,3,3) # p
range(-15,3,-3) # np
range(-15,-3,3) # p
range(-15,-3,-3) # np
```

In [1]:
```
for i in range(20,-20,-10):
    print(i)

# start=20
# dire = -ve step=-10
```

```
# last= -20+1=-19
# 20 10 0 -10
```

```
20
10
0
-10
```

In [6]:
```
# Q4) Print the 7th table
# 7x1=7
# 7x2=14
# 7x3=21
# 7xi=num*i
num=eval(input('enter the table number:'))
for i in range(1,11):
    print(f"{num}X{i}={num*i}")
```

```
14X1=14
14X2=28
14X3=42
14X4=56
14X5=70
14X6=84
14X7=98
14X8=112
14X9=126
14X10=140
```

In [8]:
```
#Q5) Find the number of divisors of 75
# if you divide 75 with any number the reminder should be zero
# 75%1==0  True  then print 1
# 75%2==0  False
# 75%3==0  True  then print 3
# 75%4==0  F
# 75%5==0  T
#
# 75%15==0  T
#
# 75%15==0 T
# 75%75==0

# if 75%i==0


num=eval(input('which number divisiors you want:'))
for i in range(1,num+1):
    if num%i==0:
        print(f"{i} is a divisor of {num}")
```

```
1 is a divisor of 75
3 is a divisor of 75
5 is a divisor of 75
15 is a divisor of 75
25 is a divisor of 75
75 is a divisor of 75
```

In [9]:
```
# Q6) sum of first 10 natural numbers
# natural : starts from 1
# 1+2+3+4+5+6+7+8+9+10=55
# n(n+1)/2
```

```
n=10
n*(n+1)/2
```

Out[9]: 55.0

summation wrapper

- starting we intialise the sum value: summ=0
- inside loop we will add a simple line : summ=summ+i
- summ=0
- for loop
    - summ=summ+i

In [ ]:
```
0+1=1
1+2=3
3+3=6
6+4=10
10+5=15
15+6=21
21+7=28
28+8=36
36+9=45
45+10=55
op=0
op+i=op ==== >  op=op+i
```

In [12]:
```
summ=0
########################
for i in range(1,11):
    summ=summ+i
################################
print(summ)
```

```
55
```

In [13]:
```
for val in range(4):
    print(val)
```

```
0
1
2
3
```

In [14]:
```
val
```

Out[14]: 3

In [15]:
```
# Q7) Find the sum of divisors of 75
# 75 : 1,3,5,15,25,75
#      1+3+5+15+25+75

summ=0

num=eval(input('enter the num:'))
for i in range(1,num+1):
```

```
    if num%i==0:
        print(i)
        summ=summ+i
summ
```

```
1
3
5
15
25
75
```

Out[15]: 124

counter wrapper

- Counter means counting the success ones
- count=0
- for loop
    - count=count+1

In [16]:
```
# Q7) Find the number of divisors of 75
count=0
num=eval(input('enter the num:'))
for i in range(1,num+1):#
    if num%i==0:
        print(i)
        count=count+1
print('the number of divisors are:',count)
```

```
1
3
5
15
25
75
the number of divisors are: 6
```

In [17]:
```
# Q8)
# ask the get 5 random numbers
# means you need a get a random number inside for loop
# the loop should run 5 times
# now perform the even and odd opeation
# 1) even count # 2) sum of even numbers  3) odd count 4) sum of add numbers
```

In [ ]:
```
 # even_Count=0
# odd_count=0
# even_sum=0
# odd_sum=0
# for loop
#   randm
#  if even
#    summ
#    counter
# else
#    summ
#    count
```

In [20]:
```
import random
even_count,odd_count=0,0
even_sum,odd_sum=0,0
for i in range(5):
    num=random.randint(1,100)
    if num%2==0:
        print(f"{num} is an even")
        even_count=even_count+1
        even_sum=even_sum+num
    else:
        print(f"{num} is an odd")
        odd_count=odd_count+1
        odd_sum=odd_sum+num
print('number of evens are:',even_count)
print('number of odds are:',odd_count)
print('sum of evens are:',even_sum)
print('sum of odds are:',odd_sum)
```

```
37 is an odd
50 is an even
18 is an even
44 is an even
95 is an odd
number of evens are: 3
number of odds are: 2
sum of evens are: 112
sum of odds are: 132
```

In [21]:
```
# Q9) Game program
# user enter one num
# another number will generate randomly
# if both numbers are equal then you won
# otherwise you lost
# i want to give 3 chances

# for loop
#   num1=random
#   num2= user keybaord
#   if num1==num2
#     print won
#  else
#    print fail
```

In [22]:
```
for i in range(3):
    n1=random.randint(1,10)
    n2=eval(input('enter a num:'))
```

```
    if n1==n2:
        print('you won')
    else:
        print('better luck next time')
```

```
you won
better luck next time
better luck next time
```

In [25]:
```
for i in range(3):
    n1=random.randint(1,10)
    print(n1)
    n2=eval(input('enter a num:'))
    if n1==n2:
        print('you won')
        break
    else:
        print('better luck next time')
```

```
2
better luck next time
3
you won
```

In [ ]:
```
# case-1: we apply the break === Completed
# case-2: whenever the user enter the wronh answer
#         better luck next time
#         how many chances you are left
# case-3: if you lost all the chances
#         print try after 24hours time
#         dont give time.sleep 24hours katam your zindagi
# whatsapp group online and offline
```

in

- in for loop we have range and in operators
- range is math related word, whenever we use range it expects a number inside range
- in operator expects a string

In [1]:
```
for i in range(10):
    print(i,end=' ')
```

```
0 1 2 3 4 5 6 7 8 9
```

In [2]:
```
for i in range('python'):
    print(i,end=' ')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[2], line 1
----> 1 for i in range('python'):
      2     print(i,end=' ')

TypeError: 'str' object cannot be interpreted as an integer
```

In [3]:
```
for i in 'python':
    print(i,end=' ')
```

```
p y t h o n
```

In [5]:
```
'p' in 'python'

'y' in 'python'

't' in 'python'

'h' in 'python'

'o' in 'python'

'n' in 'python'

i in 'python'
```

Out[5]:
```
False
```

In [ ]:
```
'A'=='a'  # F  sense
'A'>'a'   # T  make double sense
'A'<'a'   # F
```

ASCII

- AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

In [6]:
```
#'A' ==== 65
#'a' ==== 97

'A'>'a'   # 65>97
```

Out[6]:
```
False
```

- How to convert char to number
- How to convert number to char

char-ord

In [7]:
```
ord('A')
```

Out[7]:
```
65
```

In [8]:
```
chr(65)
```

Out[8]:
```
'A'
```

- so **ord** converting char to number
- **chr** converting number to char

In [10]:
```
# wap ask the user get the ASCII number A to Z
# print: A:65
#        B:66
ord('A')
ord('B')

ord(i)

# i will pass the letter you tell me number
# you are doing i will pass the number and give letter to you
```

Out[10]: 66

In [12]:
```
for i in range(65,97):
    print(f'{i}:{chr(i)}')
```

```
66:B
67:C
68:D
69:E
70:F
71:G
72:H
73:I
74:J
75:K
76:L
77:M
78:N
79:O
80:P
81:Q
82:R
83:S
84:T
85:U
86:V
87:W
88:X
89:Y
90:Z
91:[
92:\
93:]
94:^
95:_
96:`
```

In [13]:
```
for i in 'ABCDEFGHIJKLMNOPQRSTUVWXYZ':
    print(f'{i}:{ord(i)}')
```

```
A:65
B:66
C:67
D:68
E:69
F:70
G:71
H:72
I:73
J:74
K:75
L:76
M:77
N:78
O:79
P:80
Q:81
R:82
S:83
T:84
U:85
V:86
W:87
X:88
Y:89
Z:90
```

package: string

In [ ]:
```
import <packagename>
dir(<packagename>)
in that A TO Z will be given by one method tell me that
```

In [14]:
```
import string
dir(string)
```

Out[14]:
```
['Formatter',
 'Template',
 '_ChainMap',
 '__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 '_re',
 '_sentinel_dict',
 '_string',
 'ascii_letters',
 'ascii_lowercase',
 'ascii_uppercase',
 'capwords',
 'digits',
 'hexdigits',
 'octdigits',
 'printable',
 'punctuation',
 'whitespace']
```

In [15]:
```
string.ascii_letters
```

Out[15]:
```
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

In [16]:
```
string.ascii_lowercase
```

Out[16]:
```
'abcdefghijklmnopqrstuvwxyz'
```

In [17]:
```
string.ascii_uppercase
```

Out[17]:
```
'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

In [18]:
```
string.punctuation
```

Out[18]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [19]:
```
string.digits
```

Out[19]:
```
'0123456789'
```

In [21]:
```
for i in string.ascii_uppercase:
    print(f'{i}:{ord(i)}',end=' ')
```

```
A:65 B:66 C:67 D:68 E:69 F:70 G:71 H:72 I:73 J:74 K:75 L:76 M:77 N:78 O:79 P:80
Q:81 R:82 S:83 T:84 U:85 V:86 W:87 X:88 Y:89 Z:90
```

In [22]:
```
for i in string.ascii_lowercase:
    print(f'{i}:{ord(i)}',end=' ')
```

```
a:97 b:98 c:99 d:100 e:101 f:102 g:103 h:104 i:105 j:106 k:107 l:108 m:109 n:110
o:111 p:112 q:113 r:114 s:115 t:116 u:117 v:118 w:119 x:120 y:121 z:122
```

In [23]:
```
for i in string.punctuation:
    print(f'{i}:{ord(i)}',end=' ')
```

```
!:33 ":34 #:35 $:36 %:37 &:38 ':39 (:40 ):41 *:42 +:43 ,:44 -:45 .:46 /:47 ::58
;:59 <:60 =:61 >:62 ?:63 @:64 [:91 \:92 ]:93 ^:94 _:95 `:96 {:123 |:124 }:125 ~:1
26
```

In [24]:
```
range(33,127)
```

Out[24]:
```
range(33, 127)
```

In [25]:
```
for i in range(33,127):
    print(f'{i}:{chr(i)}',end=' ')
```

```
33:! 34:" 35:# 36:$ 37:% 38:& 39:' 40:( 41:) 42:* 43:+ 44:, 45:- 46:. 47:/ 48:0 4
9:1 50:2 51:3 52:4 53:5 54:6 55:7 56:8 57:9 58:: 59:; 60:< 61:= 62:> 63:? 64:@ 6
5:A 66:B 67:C 68:D 69:E 70:F 71:G 72:H 73:I 74:J 75:K 76:L 77:M 78:N 79:O 80:P 8
1:Q 82:R 83:S 84:T 85:U 86:V 87:W 88:X 89:Y 90:Z 91:[ 92:\ 93:] 94:^ 95:_ 96:` 9
7:a 98:b 99:c 100:d 101:e 102:f 103:g 104:h 105:i 106:j 107:k 108:l 109:m 110:n 1
11:o 112:p 113:q 114:r 115:s 116:t 117:u 118:v 119:w 120:x 121:y 122:z 123:{ 124:
| 125:} 126:~
```

In [26]:
```
for i in range(1,34):
    print(f'{i}:{chr(i)}',end=' ')
```

```
1:□ 2:□ 3:□ 4:□ 5:□ 6:□ 7:□ 8 9:          10:
 14:□ 15:□ 16:□ 17:□ 18:□ 19:□ 20:□ 21:□ 22:□ 23:□ 24:□ 25:□ 26:□ 27:□ 28:□ 29:□
30:□ 31:□ 32:  33:!
```

In [27]:
```
for i in range(127,250):
    print(f'{i}:{chr(i)}',end=' ')
```

```
127:  128:□ 129:□ 130:□ 131:□ 132:□ 133:□ 134:□ 135:□ 136:□ 137:□ 138:□ 139:□ 14
0:□ 141:□ 142:□ 143:□ 144:□ 145:□ 146:□ 147:□ 148:□ 149:□ 150:□ 151:□ 152:□ 153:□
154:□ 155:□ 156:□ 157:□ 158:□ 159:□ 160:  161:¡ 162:¢ 163:£ 164:¤ 165:¥ 166:¦ 16
7:§ 168:¨ 169:© 170:ª 171:« 172:¬ 173: 174:® 175:¯ 176:° 177:± 178:² 179:³ 180:´
181:µ 182:¶ 183:· 184:¸ 185:¹ 186:º 187:» 188:¼ 189:½ 190:¾ 191:¿ 192:À 193:Á 19
4:Â 195:Ã 196:Ä 197:Å 198:Æ 199:Ç 200:È 201:É 202:Ê 203:Ë 204:Ì 205:Í 206:Î 207:Ï
208:Ð 209:Ñ 210:Ò 211:Ó 212:Ô 213:Õ 214:Ö 215:× 216:Ø 217:Ù 218:Ú 219:Û 220:Ü 22
1:Ý 222:Þ 223:ß 224:à 225:á 226:â 227:ã 228:ä 229:å 230:æ 231:ç 232:è 233:é 234:ê
235:ë 236:ì 237:í 238:î 239:ï 240:ð 241:ñ 242:ò 243:ó 244:ô 245:õ 246:ö 247:÷ 24
8:ø 249:ù
```

In [28]:
```
for i in range(1632,1700):
    print(f'{i}:{chr(i)}',end=' ')
```

```
1632:٠ 1633:١ 1634:٢ 1635:٣ 1636:٤ 1637:٥ 1638:٦ 1639:٧ 1640:٨ 1641:٩ 1642:٪ 164
3:٫ 1644:٬ 1645:ٕ1655 :ٗ1654 :ٓ1653 :ّ1652 :ِ1651 :ُ1650 :ٔ1649 :ٖ1648 :ق1647 :ب1646 *
1 :خ1666 :ح1665 :پ1664 :ث1663 :پ1662 :ٹ1661 :ټ1660 :ب1659 :ٹ1658 :ٹ1657 :ى1656
:ڈ1678 :ډ1677 :ډ1676 :ډ1675 :ډ1674 :ډ1673 :ڈ1672 :چ1671 :چ1670 :خ1669 :چ1668 :667
:1690 :ڑ1689 :ڑ1688 :ړ1687 :ړ1686 :ړ1685 :ړ1684 :ړ1683 :ړ1682 :ڑ1681 :ڈ1680 :ډ1679
ف:1699 ﭪ:1698 ف:1697 غ:1696 ظ:1695 ض:1694 ﭞ:1693 ﭩ:1692 ﭫ:1691 ﭨ
```

In [29]:
```
for i in range(2308,2400):
    print(f'{i}:{chr(i)}',end=' ')
```

2308:ঌ 2309:অ 2310:আ 2311:ই 2312:ঈ 2313:উ 2314:ঊ 2315:ঋ 2316:ঌ 2317:঍ 2318:ঐ 2319:এ 2320:ঐ 2321:঑ 2322:ও 2323:ও 2324:ঔ 2325:ক 2326:খ 2327:গ 2328:ঘ 2329:ঙ 2330:চ 2331:ছ 2332:জ 2333:ঝ 2334:ঞ 2335:ট 2336:ঠ 2337:ড 2338:ঢ 2339:ণ 2340:ত 2341:থ 2342:দ 2343:ধ 2344:ন 2345:ন 2346:প 2347:ফ 2348:ব 2349:ভ 2350:ম 2351:য 2352:র 2353:ऱ 2354:ল 2355:ळ 2356:ऴ 2357:ব 2358:শ 2359:ষ 2360:স 2361:হ 2362:ঃ 2363:ঁ 2364:় 2365:ऽ 2366:া 2367:ি 2368:ী 2369:ু 2370:ূ 2371:ৃ 2372:ৄ 2373:ॅ 2374:ে 2375:ে 2376:ৈ 2377:ॉ 2378:ো 2379:ো 2380:ৌ 2381:্ 2382:ি 2383:ৗ 2384:ॐ 2385:ং 2386:ু 2387:ে 2388:ং 2389:ॅ 2390:ু 2391:ু 2392:क़ 2393:ख़ 2394:ग़ 2395:ज़ 2396:ड़ 2397:ढ़ 2398:फ़ 2399:য়

In [30]:
```
for i in range(2693,2750):
    print(f'{i}:{chr(i)}',end=' ')
```

2693:অ 2694:আ 2695:ই 2696:ঈ 2697:উ 2698:ঊ 2699:ঋ 2700:ঌ 2701:঎ 2702:□ 2703:এ 2704:ঐ 2705:঑ 2706:□ 2707:ও 2708:ঔ 2709:ক 2710:খ 2711:গ 2712:ঘ 2713:ঙ 2714:য 2715:ছ 2716:ঝ 2717:ঞ 2718:ঞ 2719:ট 2720:ঠ 2721:ড 2722:ঢ 2723:ণ 2724:ত 2725:থ 2726:দ 2727:ধ 2728:ন 2729:□ 2730:প 2731:ফ 2732:ব 2733:ভ 2734:ম 2735:য 2736:র 2737:□ 2738:ল 2739:ল 2740:□ 2741:ব 2742:শ 2743:ষ 2744:স 2745:হ 2746:□ 2747:□ 2748:় 2749:ঽ

In [31]:
```
for i in range(3077,3150):
    print(f'{i}:{chr(i)}',end=' ')
```

3077:అ 3078:ఆ 3079:ఇ 3080:ఈ 3081:ఉ 3082:ఊ 3083:ఋ 3084:ఌ 3085:□ 3086:ఎ 3087:ఏ 3088:ఐ 3089:□ 3090:ఒ 3091:ఓ 3092:ఔ 3093:క 3094:ఖ 3095:గ 3096:ఘ 3097:ఙ 3098:చ 3099:ఛ 3100:జ 3101:ఝ 3102:ఞ 3103:ట 3104:ఠ 3105:డ 3106:ఢ 3107:ణ 3108:త 3109:థ 3110:ద 3111:ధ 3112:న 3113:□ 3114:ప 3115:ఫ 3116:బ 3117:భ 3118:మ 3119:య 3120:ర 3121:ఱ 3122:ల 3123:ళ 3124:ఴ 3125:వ 3126:శ 3127:ష 3128:స 3129:హ 3130:□ 3131:□ 3132:ఽ 3133:ఽ 3134:ా 3135:ీ 3136:ీ 3137:ు 3138:ూ 3139:ృ 3140:ౄ 3141:□ 3142:ె 3143:ే 3144:ై 3145:□ 3146:ొ 3147:ో 3148:ౌ 3149:్

In [ ]: