**C01-PYTHON LAB RECORD**

1. **PGM1**

**Familiarizin text editor,ide,code analysys tools**

IDE stands for Integrated Development Environment. It’s a coding tool which allows you to write, test, and debug your code in an easier way, as they typically offer code completion or code insight by highlighting, resource management, debugging tools,… And even though the IDE is a strictly defined concept, it’s starting to be redefined as other tools such as notebooks start gaining more and more features that traditionally belong to IDEs.

Comparison between IDLE and Thonny

Thonny is built for education and you can download the latest version from the Thonny website. The download options are at the top right. Thonny looks quite different to IDLE - it has different panels for the editor, the shell and the variables watcher plus (show view) lots of other options as well. It has a powerful debugger built in and other tools which let you manage packages and plugins.

The Idle editor comes built-in with Python and is the one that many tutorials use by default. It's a fine, basic, editor that also has a Python shell built in for interactive programming.When you start Idle up, you get the shell window. This allows you to execute python commands and see the results immediately without having to create a program. This can be useful for trying things out.

1. **PGM 2**

**Display future leap years from current year to a final year entered by user.**

**INPUT :**

s=int(input("enter start year"))

e=int(input("enter end year:"))

if(s<e):

print("leap years are:",end="")

for i in range(s,e):

if i%4==0 and i%100!=0:

print(i,end=" ")

**output**

enter start year2000

enter end year:2500

leap years are:2004 2008 2012 2016 2020 2024 2028 2032 2036 2040 2044 2048 2052 2056 2060 2064 2068 2072 2076 2080 2084 2088 2092 2096 2104 2108 2112 2116 2120 2124 2128 2132 2136 2140 2144 2148 2152 2156 2160 2164 2168 2172 2176 2180 2184 2188 2192 2196 2204 2208 2212 2216 2220 2224 2228 2232 2236 2240 2244 2248 2252 2256 2260 2264 2268 2272 2276 2280 2284 2288 2292 2296 2304 2308 2312 2316 2320 2324 2328 2332 2336 2340 2344 2348 2352 2356 2360 2364 2368 2372 2376 2380 2384 2388 2392 2396 2404 2408 2412 2416 2420 2424 2428 2432 2436 2440 2444 2448 2452 2456 2460 2464 2468 2472 2476 2480 2484 2488 2492 2496 2888 2892 2896 2904 2908 2912 2916 2920 2924 2928 2932 2936 2940 2944 2948 2952 2956 2960 2964 2968 2972 2976 2980 2984 2988 2992 2996

>>>

1. **PGM3**

**LIST COMPREHENSION**

**A. Generate positive list of numbers from a given list of integers**

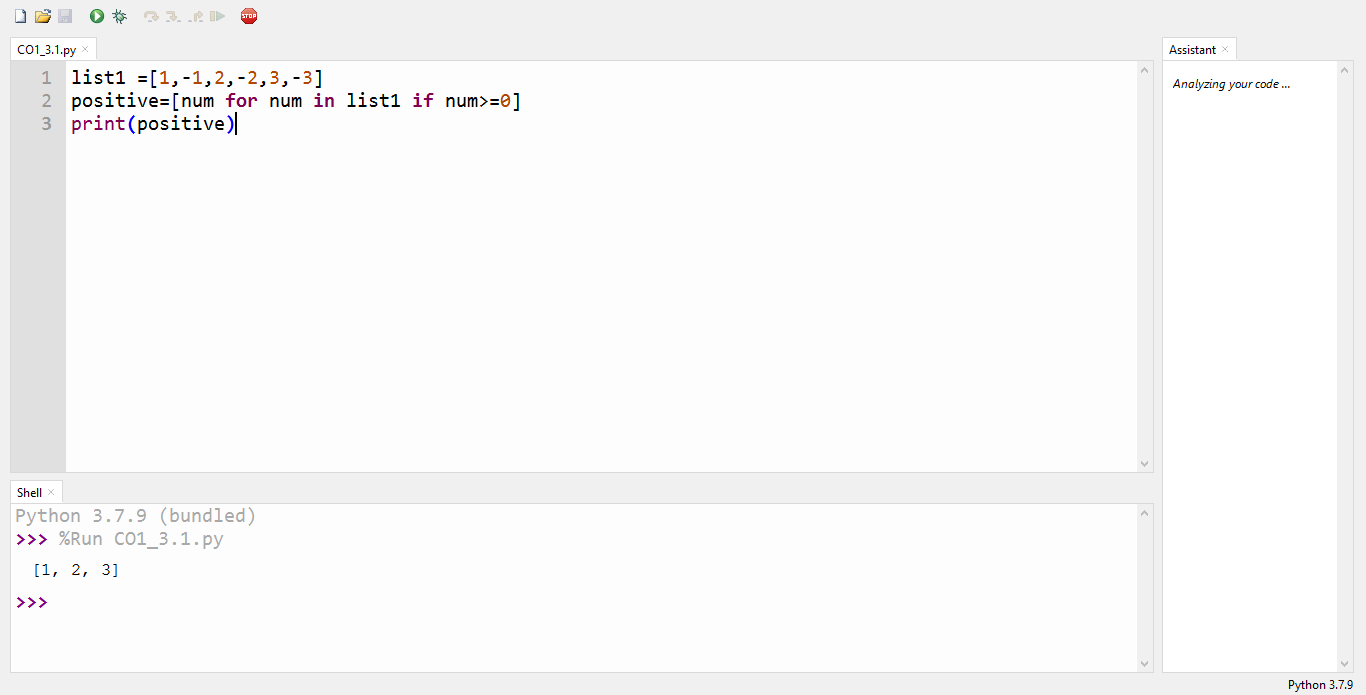
**INPUT :**

list1 =[1,-1,2,-2,3,-3]

positive=[num for num in list1 if num>=0]

print(positive)

**OUT PUT**



**B. Square of N number**

**INPUT :**

n=int(input("enter limit:"))

squarelist=[i\*\*2 for i in range(1,n+1)]

print("square of N numbers:",squarelist)

**output**

enter limit:5

square of N numbers: [1, 4, 9, 16, 25]

**C. : Form a list of vowels selected from a given word**

**INPUT :**

word=str(input("enter the word:"))

print("the original string is:"+word)

print("the vowels are :",end="")

for i in word:

if i in 'aeiouAEIOU':

print([i],end="")

**out out**

enter the word:word

the original string is:word

the vowels are :['o']

>>>

**D. List ordinal value of each element of a word**

w=input("enter a word:")

print("ordinal values corresponding to each element is")

for i in w:

print(i,end=":")

print(ord(i),end=" ")

**output**

enter a word:parvathi

ordinal values corresponding to each element is

p:112 a:97 r:114 v:118 a:97 t:116 h:104 i:105

1. **Pgm4**

**Count the occurrences of each word in a line of text.**

**INPUT :**

str1=input("enter a string")

wordlist=str1.split()

count=[]

for w in wordlist:

count.append(wordlist.count(w))

print("count of occurence:"+str(list(zip(wordlist,count))))

**output**

enter a string parvathi

count of occurence:[('parvathi', 1)]

1. **Pgm5**

**Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead**

**INPUT :**

n=[]

s=int(input("enter a limit"))

print("enter {s} values")

for i in range (0,s):

n.append(int(input()))

print("\nthe list is after asssigning:\n")

for i in range(0,len(n)):

if n[i]>=100:

print("over")

else:

print(n[i])

**output**

enter a limit5

enter {s} values

12

123

14

13

100

the list is after asssigning:

12

over

14

13

Over

1. **Pgm6**

**Store a list of first names. Count the occurrences of ‘a’ within the list**

**INPUT :**

list=["a","b","a"]

occ=list.count("a")

print("count of occurences of a:",occ)

a\_list=("a","b","a")

occ=a\_list.count("a")

print("count of occurnces of a :",occ)

**output**

count of occurnces of a : 2

1. **Pgm7**

**Enter 2 lists of integers. Check**

**(a) Whether list are of same length**

**(b) whether list sums to same value**

**(c) whether any value occur in both**

**INPUT :**

lst=[1,3,5,7,9,11,34]

lst1=[5,13,45,7,20,65,1]

s=int(0)

c=int(0)

if(len(lst)==len(lst1)):

print("lists are of same length")

else:

print("lists have diff length")

for i in range(0,len(lst) and len(lst1)):

s=s+lst[i]

c=c+lst1[i]

if(s==c):

print("equal sum")

else:

print("not same sum")

print("elements that matched are:")

l=[]

for i in range(0,len(lst)):

for j in range(0,len(lst1)):

if lst[i]==lst1[j]:

l.append(lst[i] and lst1[j])

else:

continue

print(l)

**output**

lists are of same length

not same sum

elements that matched are:

[1, 5, 7]

>>>

1. **Pgm8**

**Get a string from an input string where all occurrences of first character replaced with ‘$’, except first character.**

**INPUT :**

str1=input("enter a string:")

char=str1[0]

str1=str1.replace(char,'$')

str1=char+str1[1:]

print(str1)

**output**

enter a string:malayalam

malayala$

1. **Pgm9**

**Create a string from given string where first and last characters exchanged.**

**INPUT :**

str=input("enter a string")

new\_str=str[-1:]+str[1:-1]+str[:1]

print("new string:",new\_str)

**output**

enter a string parvathi

new string: iarvathp

>>>

1. **Pgm 10**

**: Accept the radius from user and find area of circle.**

**INPUT :**

pi=3.14

r=float(input("input the radius of circle:"))

result=3.14\*r\*\*2

print("the area of circle with radius is:",result)

**output**

input the radius of circle:3

the area of circle with radius is: 28.26

>>>

1. **Pgm11**

**: Find biggest of 3 numbers entered**

**INPUT :**

x=int(input("enter 1st number"))

y=int(input("enter 2nd number:"))

z=int(input("enter 3rd number"))

if(x>y) and (x>z):

largest=x

elif(y>x) and (y>z):

largest=y

else:

largest=z

print("largest no is",largest)

**output**

enter 1st number12

enter 2nd number:1

enter 3rd number122

largest no is 122

>>>

1. **Pgm12**

**: Accept a file name from user and print extension of that**

**INPUT :**

file=input("enter filename:")

f=file.split(".")

print("extension of file is:"+f[-1])

**output**

enter filename:hai.python

extension of file is:python

>>>

1. **Pgm13**

**: Create a list of colors from comma-separated color names entered by user.Display first and last colors.**

**INPUT :**

a=[]

for i in range(3):

b=input("enter the color")

a.append(b)

print(a)

print(a[0])

print(a[2])

**output**

enter the color red

enter the color blue

enter the color black

['red', 'blue', 'black']

red

black

1. **pgm14**

**Accept an integer n and compute n+nn+nnn**

**INPUT :**

n=int(input("enter a number:"))

x=int("%s"%n)

y=int("%s%s"%(n,n))

z=int("%s%s%s"%(n,n,n))

print("n+nn+nnn:",x+y+z)

**output**

enter a number:2

n+nn+nnn: 246

1. **pgm15**

**: Print out all colors from color-list1 not contained in color-list2.**

**INPUT :**

color\_list\_1=set(["white","pink","red","blue"])

color\_list\_2=set(["red","green","pink"])

print(color\_list\_1.difference(color\_list\_2))

**output**

{'blue', 'white'}

1. **Pgm16**

**Create a single string separated with space from two strings by swapping the character at position 1.**

**INPUT :**

a="python"

b="java"

p1=a[0]

p2=b[0]

c=b[0]+a[1:len(a)]+" "+a[0]+b[1:len(b)]

print(a[1:len(a)])

print(c)

**output**

ython

jython pava

>>>

1. **Pgm17**

**Sort dictionary in ascending and descending order**

**INPUT :**

import operator

d={1:2,3:4,4:3,2:1,0:0}

print('original dictionary:',d)

sorted\_d=sorted(d.items(),key=operator.itemgetter(1))

print('dictionary in acending order by value',sorted\_d)

sorted\_d=dict(sorted(d.items(),key=operator.itemgetter(1),reverse=True))

print('dictionary in descending order by value:',sorted\_d)

**output**

original dictionary: {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}

dictionary in acending order by value [(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)]

dictionary in descending order by value: {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}

>>>

1. **Pgm18**

**Merge two dictionaries**

**INPUT :**

d1={'a':100,'b':200}

d2={'x':300,'y':200}

print("dictionary1=:",d1)

print("dictionary2=:",d2)

d=d1.copy()

d.update(d2)

print("merged dictionary:",d)

**output**

dictionary1=: {'a': 100, 'b': 200}

dictionary2=: {'x': 300, 'y': 200}

merged dictionary: {'a': 100, 'b': 200, 'x': 300, 'y': 200}

>>>

1. **Pgm 19**

**: Find gcd of 2 numbers.**

**INPUT :**

x=int(input("enter 1st no"))

y=int(input("enter 2nd no"))

i=1

while(i<=x and i<=y):

if(x%i==0 and y%i==0):

gcd=i

i=i+1

print("gcd:",gcd)

**output**

enter 1st no3

enter 2nd no56

gcd: 1

1. **pgm20**

**From a list of integers, create a list removing even numbers.**

**INPUT :**

num=[7,8,120,25,44,20,27]

print("original list:",num)

num=[x for x in num if x%2!=0]

print("list after remooving even nos",num)

**output**

original list: [7, 8, 120, 25, 44, 20, 27]

list after remooving even nos [7, 25, 27]