Waes:

hours=input("Enter hours::::")

wages=int(input("Enter hour wage::::"))

hours=hours.split()

work\_hours=[int(x) for x in hours ]

total\_hours=sum(work\_hours)

if total\_hours <= 40:

    total\_wages=total\_hours \* wages

else:

    over=total\_hours-40

    total\_wages= (total\_hours \* wages) + (over \* wages \* 1.5)

print(total\_wages)

Output:

Enter hours::::8 8 8 8 8

Enter hour wage::::12

480

$ python weekly\_wages.py

Enter hours::::9 9 9 9 9

Enter hour wage::::12

630.0

2,

Remove dups

L1=[3,5,7,9,3,6,5,2,3,7,8,9,10,12]

L2=[]

for i in L1:

    if i not in L2:

        L2.append(i)

print(L2)

$ python Removedups.py

[3, 5, 7, 9, 6, 2, 8, 10, 12]

3.

Even or odd

L1=[1,2,3,4,5,6,8,9]

odd=[x for x in L1 if x%2==0]

even=[x for x in L1 if x%2!=0]

print(odd)

print(even)

4.

lst = [5, 4, 3, 3, 4, 5]

rev = lst[::-1]

if lst == rev:

    print('Yes Palindrome')

else:

    print('Not Plaindrome')

Output:

$ python palindrome.py

Yes Palindrome

5.

Rotate a list

1 2 3 4 5 6

Rotation 1: 2 3 4 5 6 1

Rotation 2: 3 4 5 6 1 2

L1=[1,2,3,4,5,6]

n=int(input("No of rotations:::"))

L2=L1[n:]+L1[:n]

print(L2)

$ python rotate.py

No of rotations:::2

[3, 4, 5, 6, 1, 2]

6. Shuffle a list:

import random as rd

print(rd.random())

print(rd.randint(1,100))

#93

print(rd.randint(1,100))

#94

print(rd.randrange(1,10,2))

#1

print(rd.randrange(1,10,2))

#3

for i in range(5):

    print(rd.randint(1,100),end=" ")

#42 30 68 42 16 -> 1st run

# 37 67 3 22 26  --> 2nd Run

#

print("\n")

rd.seed(10)

for i in range(5):

    print(rd.randint(1,100),end=" ")

# 74 5 55 62 74   - Run1

# 74 5 55 62 74  -- Run2

# 74 5 55 62 74  -- Run3

# 74 5 55 62 74  -- Run4

L1=[1,2,3,4,5,6,7]

rd.shuffle(L1)

print(L1)

# [2, 4, 6, 3, 7, 5, 1]

# [3, 1, 2, 7, 6, 5, 4]

# [4, 5, 1, 6, 3, 7, 2]

# [6, 1, 2, 7, 5, 3, 4]

List permutation:

PERMUTATIONS:

Lst=[‘A”,”B”,”C”]

A B C

A C B

B A C

B C A

C A B

C B A

n!%(n-r)!

A B C D

R=2

A B

A C

A D

B A   
B C

B D

C A

C B

C D

D A

D B

D C

COMBINATIONS:

n!/(n-r)!r!

L1=[“A”,”B”,”C”,”D”]

A B

A C

A D

B C

B D

D A

Total : 6

PRODUCT:

Product(iterable,repeat=1)

(cartisian product)

[1,2,3],[“a”,”b”]

1,a 1,b 1,c 2,a 2,b 2,c 3,a 3,b

import itertools as it

lst=["A","B","C","D"]

#permutations(iterable,r=None)

#combinations(iterable,r=None)

#product(iterable,repeat=1)

perms=it.permutations(lst,r=2)

print(type(perms))

#<class 'itertools.permutations'>

perms=list(perms)

for i in perms:

    print(i,end=" ")

# ('A', 'B') ('A', 'C') ('A', 'D') ('B', 'A') ('B', 'C') ('B', 'D') ('C', 'A') ('C', 'B') ('C', 'D') ('D', 'A') ('D', 'B') ('D', 'C')

b.

combi=it.combinations(lst,r=2)

combi=list(combi)

for i in combi:

    print(combi)

('A', 'B')

('A', 'C')

('A', 'D')

('B', 'C')

('B', 'D')

('C', 'D')

c.

product=it.product(lst,repeat=2)

product=list(product)

for i in product:

    print(i,end=" ")

('A', 'A') ('A', 'B') ('A', 'C') ('A', 'D') ('B', 'A') ('B', 'B') ('B', 'C') ('B', 'D') ('C', 'A') ('C', 'B') ('C', 'C') ('C', 'D') ('D', 'A') ('D', 'B') ('D', 'C') ('D', 'D')

MEAN:

Mean-Median-Mode

Lst=[6,8,4,2]

Mean=(6+8+4+2) /4

Median:

2 3 4 4 5 6 7 8 8 9 10 11 12

Median=(7+8)/2

Mode: No of more repeatable element

8

Find Longest List:

A person's hand holding a white board with text

AI-generated content may be incorrect.

lists=[[1,2,3],[1,1,1,1,1],[2,2,3,3]]

print(max(lists,key=len))

# [1, 1, 1, 1, 1]