


```
In [4]: s='abcdefghijklmnopqrstuvwxyz'
print(s[0])
print(s[2])
print(s[-1])
print(s[-2])
print(s[-10])
```

a
c
z
y
q

2.14

```
In [12]: s='goodbye'
print('g' in s)
print('g' in s[-1])
print('g' in s[0] and 'a' in s[1])
print('x' in s[-2])
print('d' in s[-4])
print(s[0]==s[-1])
sting='tion'
print([s[-1],s[-2],s[-3],s[-4]]==sting)
```

True
False
False
False
True
False
False

2.15

```
In [17]: s1='anachronistically'
s2='counterintuitive'
print(len(s1)>len(s2))
wrds='floccinaucinihiipilification'
print('e' in wrds)
s3='counterrevolution'
s4='counter'
s5='resolution'
print(len(s3)==len(s4)+len(s5))
```

True
False
True

2.16

```
In [18]: a=6
b=7
c=(a+b)/2
print(c)
inventory=['papers','staples','pencils']
print(inventory)
first='john'
middle='fitzgerald'
last='kennedy'
fullname=first + ' ' + middle + ' ' + last
print(fullname)
```

```
6.5
['papers', 'staples', 'pencils']
john fitzgerald kennedy
```

2.17

```
In [26]: print((17-9)<10)
inventory=['papers','staples','pencils']
first='john'
middle='fitzgerald'
last='kennedy'
fullname=first + ' ' + middle + ' ' + last
print(len(inventory)>(len(fullname)*5))
c=6.5
print(c<=24)
a=6
b=7
print(a<6.75<b)
print(len(first)<len(middle)<len(last))
print(inventory==[] or len(inventory)<10)
```

```
True
False
True
True
False
True
```

2.18

```
In [28]: flowers=['rose','bougainvillea','yucca','marigold','daylilly','lilly of vallr
y']
print('patato' in flowers)
thorny=[flowers[0],flowers[1],flowers[2]]
print(thorny)
poisonous=[flowers[-1]]
print(poisonous)
dangerous=[thorny + poisonous]
print(dangerous)
```

False

['rose', 'bougainvillea', 'yucca']

['lilly of vallry']

[['rose', 'bougainvillea', 'yucca', 'lilly of vallry']]

```
In [41]: answers=['y','y','n','y','n','y','y','n','n','n','y','y','y','n','y','n','y',
'n','n','n','n','n']
num_yes=answers.count('y')
print(num_yes)
num_no=answers.count('n')
print(num_no)
percentage_yes=(num_yes/len(answers))*100
print(percentage_yes,"%")
srt=answers.sort()
print(srt)
```

10

12

45.45454545454545 %

None

2.19 ^

2.20

```
In [44]: s='fat'
rev=s[::-1]
print(rev)
```

taf

2.21

```
In [45]: s='lionel'  
         t='messi'  
         u=s[0] + t[0]  
         print(u)
```

lm

2.22

```
In [56]: lst=[3,4,6,8,3,23,32,1,6,19,21]  
         a=max(lst)  
         b=min(lst)  
         rng=a-b  
         print("range of list is:",rng)
```

range of list is: 31

2.23

```
In [76]: months1=['jan','feb','mar','may']
monthst=('jan','feb','mar','may')
newmonth1=months1.insert(-1,'apr')
print(newmonth1)
appl=months1.append('jun')
print(appl)
appt=monthst.append('jun')
print(appt)
reml=months1.remove('feb')
print(reml)
remt=monthst.append('feb')
print(remt)
revl=months1.reverse()
print(revl)
revt=monthst.reverse()
print(revt)
srtl=months1.sort()
print(srtl)
srtt=monthst.sort()
print(srtt)
```

None

None

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-76-b1af28971122> in <module>
      5 appl=months1.append('jun')
      6 print(appl)
----> 7 appt=monthst.append('jun')
      8 print(appt)
      9 reml=months1.remove('feb')
```

AttributeError: 'tuple' object has no attribute 'append'

2.24

```
In [75]: grades=['A','B','D','C','C','C','A','A','B','F','F','A','B','B']
A=grades.count('A')
B=grades.count('B')
C=grades.count('C')
D=grades.count('D')
F=grades.count('F')
print("number of grades A B C D and F ARE:",[A,B,C,D,F])
```

number of grades A B C D and F ARE: [4, 4, 3, 1, 2]

2.25

```
In [77]: grades=('A','B','D','C','C','C','A','A','B','F','F','A','B','B')
A=grades.count('A')
B=grades.count('B')
C=grades.count('C')
D=grades.count('D')
F=grades.count('F')
print("number of grades A B C D and F ARE:",[A,B,C,D,F])
```

number of grades A B C D and F ARE: [4, 4, 3, 1, 2]

2.26

```
In [85]: limbaord=[10,10]
x=0
y=0
print([x,y]<=limbaord)
x1=10
y2=10
print([x1,y2]<=limbaord)
x2=6
y3=-6
print([x2,y3]<=limbaord)
x3=-7
y4=8
print([x3,y4]<=limbaord)
```

True
True
True
True

2.27

```
In [3]: import math
l1=16
angle1=75
rad1=(math.pi*75)/180
height1=l1*math.sin(rad1)
print(height1,"feet")
l1=20
angle1=0
rad1=(math.pi*0)/180
height1=l1*math.sin(rad1)
print(height1,"feet")
l1=24
angle1=45
rad1=(math.pi*45)/180
height1=l1*math.sin(rad1)
print(height1,"feet")
l1=24
angle1=80
rad1=(math.pi*80)/180
height1=l1*math.sin(rad1)
print(height1,"feet")
```

```
15.454813220625093 feet
0.0 feet
16.970562748477143 feet
23.63538607229299 feet
```

2.28

```
In [9]: lst=[2,3,-5,8,5,7,5,3,5]
index=lst.index(5)
print(index)
element=lst[-5]
print(element)
lst.sort(reverse=True)
print(lst)
lst=[2,3,-5,8,5,7,5,3,5]
lst.remove(2)
lst.append(2)
print(lst)
```

```
4
5
[8, 7, 5, 5, 5, 3, 3, 2, -5]
[3, -5, 8, 5, 7, 5, 3, 5, 2]
```

2.29


```
In [2]: print(0==(1==2))#since 0 is the numeric term of false so false = false is true
print((2+3==4)==(+5==7))# since false = false is true
print((1<-1)==(3>4))#since false = false is true
```

True

True

True

2.30

```
In [6]: s='saim'
lst=list(s)
print(lst)
#the list converter function is simple it takes the variable to which the string is assigned to as an argument and convert that string into standard form of list in which each substring of the list becomes an individual character of the list
```

['s', ' ', 'a', ' ', 'i', ' ', 'm']

2.31

```
In [12]: lst=[2,3,4]
lst.extend([5,6])#extend method inserts the argument elements in the end of the list contrary to append method it resumes the list with new elements instead of adding only one element
print(lst)
lst2=lst.copy()#copy method makes the copy of the given list and assigns it to a new variable
print(lst2)
lst.clear()#clear method removes the elements in the list and makes it an empty list
print(lst)
```

[2, 3, 4, 5, 6]

[2, 3, 4, 5, 6]

[]