**BINARY TO DECIMAL**

decimal\_value = 0

n = int(binary\_value[3])

decimal\_value = decimal\_value + n \* math.pow(2, 0)

#Code to do the conversion

n = int(binary\_value[2])

decimal\_value = decimal\_value + n \* math.pow(2, 1)

n = int(binary\_value[1])

decimal\_value = decimal\_value + n \* math.pow(2, 2)

n = int(binary\_value[0])

decimal\_value = decimal\_value + n \* math.pow(2, 3)

**IF, ELSE FUNCTION**

if (a < 6): #Check if user's passport is valid for entry

print('You have to renew your passport.') #Modify to display that user's passport is valid for entry

return True #Modify accordingly

else:

print('You do not have to renew your passport.') #Modify to display that user needs to renew his passport

return False #Modify accordingly

**IF FUNCTION**

if (age >= 75):

new\_amount = ((amount\*95)/100)

print('New amount to pay is: ', '$' , new\_amount)

**IF, ELIF, ELSE FUNCTION**

if BMI < 18.5:

print("Risk of developing osteoporosis")

elif BMI < 23:

print("Healthy")

elif BMI < 27.5:

print ("Low risk of developing heart disease, stroke, etc.")

else: print("High risk of developing heart disease, stroke, etc.")

**CHANGE VARIABLE**

x=5

y=3

z=x

w=y

y=z

x=w

**LIST**

electrical\_cost = float(input('Enter the electrical cost($) for 3 months separated by a ";":'))

electrical\_list = electrical\_cost.split(';')

a, b, c, = float(electrical\_list[0], float(electrical\_list[1]), float(electrical\_list[2])

**SPLIT FUNCTION (float or int must be below and not In the input)**

mthly\_cost = (input('Enter the electrical cost($) for 3 months separated by a ";":'))

mthly\_cost\_list = mthly\_cost.split(';')

total = float(mthly\_cost\_list[0]) + float(mthly\_cost\_list[1]) + float(mthly\_cost\_list[2]) #Compute the total electrical costs over 3 months

average = (total / 3)

**REPLACE FUNCTION**

base = 'ACGT' String format (don’t need use base and complement)

complement = 'TGCA'

new\_string = dna1.replace("T", "X").replace("G", "Y").replace("A", "T").replace("C", "G").replace("X", "A").replace("Y", "C")

**STRING FORMATTING**

\n 🡪 new line

\t 🡪 new tab (blank space “fill in the blanks ”)

‘ {} {} ‘ .format(1, 2) 🡪 printed version: “ 1 2 “

**DECIMAL PLACES AND OTHERS**

{:.2f} = 2sf / 2dp // = floor division ( to find quotient) % = modulus ( to find remainder )

[::-1] = 123 (reverse function (eg. 123 becomes 321))

**SPACING IN PYTHON**

print ('{:10s} {:10s} {:10s}\n'.format('a', 'b', 'a to the power of b'))

can change to {:25s} {:\_<10s} = underline {:^10s} = centralize

s = string d = integer f = float

**IMPORT MATH FUNCTION**

>>> import math

>>> math.sqrt ( any number ) = square root of that number

>>> math.pi = pi (3.142…)

>>> math.pow (2, 2) = power of number in this case 22

**WHILE LOOP**

|  |  |
| --- | --- |
| Adds the even numbers from 1 to 20 | While loop using lists |
| Number = 2  total = 0  while number <= 20:  total = total + number  number = number + 2  print (total) | temp\_list = [12, 13, 14, 45, 32]  i = 0  total = 0  while i < len(temp\_list):  total += temp\_list[i]  i += 1  average = total / len(temp\_list)  print (average) |

**FOR LOOP**

|  |  |
| --- | --- |
| For loop using lists | For loop using lists and range |
| total = 0  for i in temp\_list:  total += i  average = total / len(temp\_list)  print (average) | total = 0  for i in range(0, len(temp\_list)):  total += temp\_list[i]  average = total / len(temp\_list)  print (average) |
| Add break to end loop | total = 0  counter = 0  for i in range(0, len(temp\_list)):  total += temp\_list[i]  if(temp\_list[i] > 25):  print ("WARNING")  break  total = temp\_list[i]  counter += 1  average = total / counter  print (average) |
|  | Counter is added to take the average of the numbers which are only > 25 and not all of the numbers in the list. |

**NESTED LOOP**

|  |  |
| --- | --- |
| Nested loop with range |  |
| for i in range(3):  for j in range(6):  print('\*',end='')  print() |  |
| end= marks the end of the loop and also keeps the cursor in place so that the printed version is proper |  |

Import statistic

Enumerate

[x for x in list if ‘sth’ in x]

List(map(func, list))

Lambda x: func , list