Día 3

Temas

1. Listas
   1. Crear listas
   2. for (aplicado a listas)
   3. Añadir valores a la lista
   4. Length

Actividades de la mañana

1. Palíndromos
   1. Formar equipos.
   2. Cada equipo debe formar un palíndromo original (oración).
2. 100 mexicanos dijeron/ Ahorcados
   1. Formar 4 equipos.
   2. Se pone los espacios por cada letra de una frase oculta en el pintarrón.
   3. Pasa una representante de cada equipo al frente.
   4. Cada 10- 15 segundos (dependiendo de la edad del grupo) se va abriendo una letra.
   5. Entre menos letras tenga la oración a descifrar menos puntos ganará el equipo que la adivine.
3. Programando pulseras
4. Todas las niñas tienen en su escritorio un puñado de piedras de pulseras, resorte, y una bolsita de papelitos con instrucciones.
5. Las niñas sacan un papelito en donde viene la acción a realizar, ej:

* añadir(“rojo”)

Esto indica que las niñas tienen que añadir una piedra a su pulsera de color rojo.

* for i in range(5):

añadir(“blanco”)

Esto indica que las niñas van a añadir 5 piedras de color blanco.

1. Pesa y ordena
2. Dividir el grupo en 4 equipos
3. Cada equipo tiene una balanza y 9 recipientes iguales excepto por el peso.
4. Cada equipo tendrá que pensar en una serie de pasos para ordenar los recipientes de menor a mayor peso (en el número menor de pasos posibles).

Actividades de programación

La lista del mercado

1. Crea un archivo y nómbralo lista1.py
2. Crea una lista que contenga nombres de frutas (pera, manzana, plátano).

Cada elemento deberá estar separado por una coma y la lista está encerrada entre corchetes. Ejemplo:

lista=[“pera”, “manzana”, “plátano”]

1. Ahora, haz que el programa imprima cada uno de los elementos.

Recuerda que la computadora enumera los elementos, comenzando desde el 0, por eso pera sería el elemento 0, manzana el 1, plátano el 2.

Para imprimir cierto elemento escribe print(lista[2])

Este ejemplo imprimiría “plátano”.

1. Ahora intenta añadir más frutas a la lista. Ejemplo:

lista.append(“fresa”)

Solución: 1\_Listas

Mis amigas

1. Crea un archivo y nómbralo amigas.py
2. Crea una lista con los nombres de 5 amigas
3. Imprime sus nombres, pero esta vez usando “for”

¿cómo lo puedes hacer?

for i in range(len(personas)):

#Recorres la lista con cada posicion i

print(personas[i]) #imprimes la posicion i de la lista

o también puede ser

for i in personas:

print(i) #donde i representa un elemento de la lista

Solución: 2\_Listas

Analizador de oraciones

1. Crea un archivo y nómbralo AnalisisDeOraciones.py
2. Crea una variable llamada “cadena” que acepte como input una oración
3. Imprime el número total de letras, espacios, consonantes, vocales, y el número específico de vocales (a,e,i,o,u) que contiene
4. Puedes tratar a la variable “cadena” como una lista y recorrerla como tal, haciendo comparaciones en cada elemento de la lista

Solución: 5\_AnalisisDeOraciones

Encontrar el mayor

1. Crea un archivo y nómbralo NumeroMayor.py
2. Crea una lista llamada “numeros” e inserta 9 números enteros al azar,
3. Recorre la lista y encuentra el número mayor.

Solución: NumeroMayor.py

¿Estás ahí?

1. Crea un archivo y nómbralo encontrarElemento.py
2. Crea una lista de enteros con números al azar.
3. Recibe como input el número a buscar deseado.
4. Recorre la lista hasta encontrar o no el número.

Solución: encontrarElemento.py

Reverse a una lista

1. Crea un archivo y nómbralo reversa.py
2. Recibe el input de una oración y guárdalo en una variable llamada “oracion”.
3. Separa esa oración en una lista de caracteres:

caracteres = list(oracion)

1. Recorre la mitad de la lista y ve haciendo swap entre los elementos correspondientes:

#For para recorrer la mitad de la lista

for i in range(tamaño // 2):

#Aquí se realiza el swap entre los elementos

temp = caracteres[i]

caracteres[i] = caracteres[tamaño - 1 - i]

caracteres[tamaño - 1 - i] = temp

1. Como paso extra puedes concatenar los elementos e imprimir el string.

oracion = "".join(caracteres)

print(oracion)

Solución: reversa.py

Elementos al cuadrado

1. Crea un programa donde ingreses datos a una lista, después recorre esta lista y eleva cada elemento al cuadrado

PISTAS:

1. Para determinar cuándo el usuario dejará de ingresar datos a la lista puedes poner un mensaje como este:

"Ingresa un número entero para agregar a la lista, para terminar ingrese 0: ", o alguna otra forma que tu gustes

1. El operador para elevar un num. al cuadrado es: "num\*\*exp"

Solución: ElementosAlCuadrado.py