

Repaso y Tips Prueba:

Unidad 1:

Estructura de Scripting: es la estructura del código.

- Bloque de importación.
- Bloque de Definición: Def. constantes, Def. Funciones, Def. Variables
- Bloque de Desarrollo: todos los while, if, elses.
- Bloque de salida: prints o returns.

Listas y Strings:

lista: ["hola", "mundo", "como", "están"] index:[0, 1, 2, 3]		
Funcion:	Como se usa:	Salida esperada.
len(): función que entrega la cantidad de elementos	len(lista)	Ej: 4
append(): dado un elemento, lo agrega al final de la lista.	lista.append(cosa)	["hola", ..., "están", cosa]
remove(): dado un elemento, remueve tal de la lista (el 1° que encuentre)	Lista.remove("hola")	["mundo", "como", "están"]
pop(): dado el índice de un elemento, lo remueve	Lista.pop(1)	lista=["hola", "como", "están"] Lista.pop(1)=["mundo"]
*Sort(): ordena los valores de manera ascendente	Lista.sort()	["como", "están", "hola", "mundo",]
*Insert(): dado el índice y el elemento, agrega este a la lista en el lugar dado	Lista.insert(1, "a")	["hola", "a", "mundo", "como", "están"]

String: "el clásico hola mundo" Índices como en la lista, pero en cada carácter.		
+ : concatena 2 strings	string + "haha"	"el clásico hola mundohaha"
*: repite un string	"a"*3	"aaa"
Len(): entrega el largo del string	len(string)	21
split(): separa un string en listas, según el carácter dado	String.split() String.split(" ")	["el", "clásico", "hola", "mundo"]
Strip(): elimina, tanto del inicio, como del final, el string dado	String.strip("el")	" clásico hola mundo"
\n = salto de linea	\0 = fin de archivo	

- Tips y ayudas:
 - Intenten trabajar con los tipos de datos que dan en la U: int, float, string, list,boolean.
 - No trabajen con un While True / while (val == TRUE):
*(while valor == 0)

Unidad 2:

Recursion: forma donde la función en vez de dar un resultado, por la salida, se llama a si misma 1 o mas veces.

Funciones importadas: (from Y = dar una librería en específico) Import X (as Z = darle un "apodo")

Funciones propias:

```
def nombreFuncion(parametros):
```

- Usen cammelCase o esta_ forma.

- Mantengan el mismo orden de scripting.

- Revisen las salidas, si solo puede haber 1 valor en el return, o si es recursión, solo recursión.

Archivos:

-Abrir:

"r" = read	indica que abriremos el archivo en modo de lectura.
"w" = write	indica que abriremos el archivo en modo de escritura (sobrescritura)
"a" = append	indica que abriremos el archivo en modo de escritura (agregar)

```
NombreVariable = open('Ruta_Nombre_del_Archivo.extension', "modo")
```

#Recordar que siempre leeran Strings

*CADA FUNCION DEBE TENER SU ARCHIVO.CLOSE() ANTES DEL RETURN O RECUERDEN PONER ARCHIVO.CLOSE()

archivo.close(): Cierra el archivo, y guarda cambios de ser necesario.

#Ejemplos de Lectura

```
def lecturaArchivo(rutaYNombre):
```

```
    archivo = open(rutaYNombre, "r")
```

```
    lista = []
```

```
    for linea in archivo:
```

```
        lista.append(linea)
```

```
    archivo.close()
```

```
    return lista
```

```
#nos entrega algo asi = ["Bastian Onetto\n", "Andres Ramos\n",..., "Ximena Gomez\0"]
```

```
def lecturaArchivoMod(rutaYNombre):
```

```
    archivo = open(rutaYNombre, "r")
```

```
    lista = []
```

```
    for linea in archivo:
```

```
        linea2 = linea.strip("\n")
```

```
        linea2.strip("\0")
```

```
        lista.append(linea2)
```

```
    archivo.close()
```

```
    return lista
```

```
#nos entrega algo asi = ["Bastian Onetto", "Andres Ramos",..., "Ximena Gomez"]
```

-Escribir:

```
#Definicion Escritura
```

```
#teniendo una lista ya preparada
```

```
def escribirArchivoWrite(lista):
```

```
    #write(string)
```

```
    #si nuestra lista contiene numericos
```

```
    archivo = open("resultado_escritura.txt", "w")
```

```
    for elemento in lista:
```

```
        archivo.write(str(elemento)+"\n")
```

```
    archivo.close()
```

```
def escribirArchivoAppend(lista):
```

```
    #write(string)
```

```
    #si nuestra lista contiene numericos
```

```
    archivo = open("resultado_escritura.txt", "a")
```

```
    for elemento in lista:
```

```
        archivo.write(str(elemento)+"\n")
```

```
archivo.close()
```

****Tener en mente la diferencia entre Write y Append**

Rutas:

es la indicacion de la carpeta en especifico donde se encuentra el archivo

ejemplo:

C:\Users\centr\Desktop\carpeta

El problema es que para que se lea en Python, deben dar vuelta los \

C:/Users/centr/Desktop/carpeta

Punto interesante: python toma como ruta previa, la ruta donde se encuentre el .py ejecutado

```
===== RESTART: C:/Users/centr/Desktop/CodigoAyudantia5.py =====
```

por lo que solo falta rellenar lo que falte.

-Bastian "Sora" Onetto