|  |  |
| --- | --- |
|  | Trabajo grupal o individual |

## Identificación del trabajo

|  |  |
| --- | --- |
| 1. **Módulo:** | 3 |
| 1. **Asignatura:** | Scripting |
| 1. **RA:** | Aplica estructuras de datos complejos, utilizando listas, tuplas y diccionarios, en la manipulación de datos |
| 1. **Docente Online:** | Gonzalo Cárdenas |
| 1. **Fecha de entrega**: | 24-07-2023 |
|  |  |
|  |  |

## Identificación del/los estudiante/s

|  |  |
| --- | --- |
| **Nombre y apellido** | **Carrera** |
| William Huera | Técnico en Data Science |
|  |  |
|  |  |
|  |  |
|  |  |

## Introducción

En la empresa de desarrollo de software donde trabajas, tienes un rol importante: recibir y preparar a alumnos en práctica para mejorar sus habilidades de scripting. Estos alumnos ya poseen conocimientos básicos de scripting, y tu tarea consiste en llevarlos al siguiente nivel, enseñándoles el uso de estructuras complejas de datos.

Para cumplir con esta meta, te han solicitado crear un programa de diagnóstico del estado del conocimiento de los alumnos. El objetivo es que a través de este programa, los alumnos puedan demostrar sus habilidades y comprensión de los conceptos clave relacionados con las estructuras complejas de datos.

**Problemática**

## Desarrollo

**Item1**:

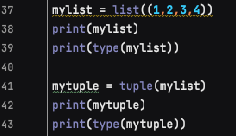
Analiza el uso de tuplas y diccionarios en términos de las funciones y variables definidas, en lo referente a la llamada y el resultado esperado, a partir de un script, preparando un informe que dé cuenta del análisis realizado para las estructuras. Para ello

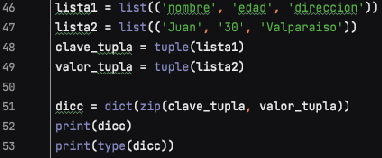
1. Identifica el tipo de estructura de dato usado en la función suma, indicando la manera en que se usa en un mínimo de 3 líneas y un máximo de 5.
2. Identifica el tipo de estructura de dato usado en la función saluda, indicando la manera en que se usa en un mínimo de 3 líneas y un máximo de 5.
3. Identifica el tipo de estructura de dato usado en la función invierte, indicando la manera en que se usa en un mínimo de 3 líneas y un máximo de 5.
4. Identifica el tipo de estructura de dato usado en la función completa, indicando la manera en que se usa en un mínimo de 3 líneas y un máximo de 5.

|  |
| --- |
| Código:  **def** suma(a, b, c):  res **=** a **+** b **+** c  print('la suma es:', res)    tp **=** 55, 55.22, 33.33  suma(**\***tp)    la función recibe una tupla (estructura de datos inmutable) como paramétro, recorre cada valor con el operador de desempaquetado, que descompone la tupla en tres argumentos individuales y los suma, lo almacena en la variable “res” y finalmente imprime la respuesta. |
| **def** saluda(nombre, edad, sexo, nacionalidad):  print('Hola', nombre)  print('Tienes', edad)  print('eres del sexo', sexo)  print('eres de nacionalidad', nacionalidad)    dicc **=** {'nombre':'Miguel C.', 'edad':22, 'sexo':'Masculino','nacionalidad':'Mexicana'}  saluda(**\*\***dicc)    la función saluda, recibe un diccionario como parámetro, de igual forma con el operador desempaquetador, envia cada uno de las variables y finalmente las imprime. |
| **def** invierte(lst):  print(lst)  lst\_rev **=** lst[::**-**1]  print(lst\_rev)    lst **=** ["IPP", "Python", "Scripting", "Curso", "Material"]  invierte(lst)    la función invierte, recibe un objeto tipo lista como parámetro, y con el operador [::-1], se invierte el orden de los elementos de la lista, finalmente imprime la lista |
| **def** completa(dct1, dct2):  dct2.update(dct1)  print(dct2)    dict1 **=** {'bookA':1,'bookB':2,'bookC':3}  dict2 **=** {'bookC':2,'bookD':4,'bookE':5}  completa(dict1,dict2)    La función completa, recibe dos objetos de tipo diccionario como parámetros, con el operador “update” actualiza los valores del diccionario dict2 con los valores de dict1, para el caso de bookC:3 y agrega los elementos bookD:4, bookE:5 y finalmente imprime el diccionario |
| Git: <https://github.com/whuera/scripting/blob/master/modulo2/m2-t1.py> |

**Item2**:

Determina las variables de entrada y salida, las transformaciones o conversiones solicitadas en cada punto para los scripts descritos a continuación, de manera de explicar a los practicantes lo que realiza cada uno de los scripts descritos:





1. Determina, para el primer script, la identificación y descripción de los valores de las variables iniciales y de las salidas.
2. Determina, para el primer script, las transformaciones o conversiones entre los distintos tipos de datos.
3. Determina, para el segundo script, la identificación y descripción de los valores de las variables iniciales y de las salidas.
4. Determina, para el segundo script, las transformaciones o conversiones entre los distintos tipos de datos

|  |
| --- |
| Código |
| Git: <https://github.com/whuera/scripting/blob/master/modulo2/m2-t1-function.py> |

**Item3**:

Aplica los conceptos aprendidos, desarrollando un ejercicio mediante la creación de un script que identifique la estructura de datos idónea para el ingreso de notas de 10 alumnos, generando 2 fórmulas: una que calcule el promedio de notas y otra que calcule la desviación estándar.

Se evaluará lo siguiente:

1. Determinar la estructura de datos adecuada, con la justificación del caso, en un mínimo de 3 líneas y un máximo de 8 líneas como comentario dentro del código fuente.
2. Generar la función solicitada de cálculo del promedio.
3. Generar la función solicitada de cálculo de desviación estándar.
4. Crear el script con la sección principal de ingreso de datos, y despliegue por pantalla de los resultados

|  |
| --- |
| Código: |
| Git: <https://github.com/whuera/scripting/blob/master/modulo2/m2-t1-function-array.py> |

## Conclusión

## Bibliografía

<https://www.w3schools.com/python/>

Documentación guía, material de apoyo del módulo 3