1. Multiple choice

¿Cuál de estos tipos de datos primitivos no pertenece a Python?

- [ ] Boolean

- [x] Decimal

- [ ] Integer

2. Verdadero o falso

En una lista solo pueden haber datos del mismo tipo [F]

3. Multiple choice

Según el siguiente código seleccione la opción que corresponda a lo que será impreso por consola

datos = var1, var2, 5

print(type(datos))

- [x] <class 'tuple'>

- [ ] Este código no produce ninguna salida por consola

- [ ] 5

- [ ] <class 'list'>

4. Multiple choice

Para convertir una variable a texto (siempre que se pueda) podemos usar la clase str. Dado el siguente código:

variable = 12.34

Elija el fragmento de codigo correcto para convertir esa variable a string e imprimirla por consola.

- [ ] str(print(variable))

- [ ] print(variable.str)

- [x] print(str(variable))

- [ ] console.log(str variable)

Tip: al igual que en la aritmética, donde hay operaciones que se hacen primero (primero multiplicamos y luego sumamos) En Python también hay una precedencia de sus operadores.

Ver: https://interactivechaos.com/es/manual/tutorial-de-python/precedencia-de-operadores

5. Completar

Complete el operador faltante para que se imprima por consola "Hola"

A = 5

B = 3

if A ... B:

print("Chau")

else:

print("Hola")

6. Verdadero o falso

Si quiero repetir una operación por cada miembro de una lista es más fácil usar un bucle 'for'. [V]

7. La siguiente función calcula el promedio ponderado de una serie de precios y cantidades. Esta función que **debe devolver el promedio** tiene un error, indicar cuál es:

(Sabemos que este ejercicio puede ser intimidante, pero queríamos mostrarles una función que hace efectivamente algo. No se asusten, escriban lo que les parezca, nos interesa leerlos, es más bien para tener algo que discutir la clase siguiente. La respuesta es muy simple, que el árbol no les tape el bosque)

# Hagamos una función que calcule el promedio ponderado tomando una serie de precios

# y otra de cantidades. Debe devolver el promedio ponderado

def promedio\_ponderado(precios, cantidades):

if len(precios) != len(cantidades):

print("¡Error!: la lista de precios debe tener el mismo largo que la lista de cantidades")

return # Como hay un error, sale de la función con el return y no ejecuta el resto

valores = []

for idx, precio in enumerate(precios):

cantidad = cantidades [idx]

valor = precio \* cantidad

valores.append(valor)

valor\_total = sum(valores)

ponderadores = []

for valor in valores:

ponderadores.append(valor/valores)

precios\_ponderados = []

for idx, precio in enumerate(precios):

precios\_ponderados.append(precio\*ponderadores[idx])

promedio\_ponderado = sum(precios\_ponderados)