



ISIS-1221

INTRODUCCIÓN A LA PROGRAMACIÓN

Objetivos

1. Practicar el manejo de cadenas de caracteres
2. Aprender a crear y manipular diccionarios
3. Fomentar la habilidad de descomponer un problema en subproblemas y de implementar funciones que los resuelven, lo que se conoce comúnmente como la técnica de "Dividir y Conquistar"
4. Practicar el uso de instrucciones condicionales para la solución de problemas por casos

Preparación

Cree una carpeta de trabajo y descargue allí el archivo ModuloLibreria.py que se encuentra adjunto a este enunciado en Brightspace. Implemente en este mismo archivo las funciones que se piden más adelante.

Diccionarios para manejar elementos con características comunes

Los diccionarios nos permiten manejar fácilmente elementos de la realidad que tienen las mismas características (información). Por ejemplo: supongamos que de todos los libros de la librería se guarda la misma información: nombre, código, autor, año de publicación, cantidad, precio y costo de producción. Y supongamos también que necesitamos manejar la información de los siguientes 4 libros:

	nombre	código	autor	año de publicación	cantidad	precio	costo
Libro 1	Harry Potter y la piedra filosofal	HPJK1997	J.K. Rowling	1997	200	25000	9000
Libro 2	Los Juegos del Hambre	JHSC2008	Suzanne Collins	2008	100	27000	12000
Libro 3	El Hobbit	EHJR1937	J.R.R. Tolkien	1937	50	35000	15000
Libro 4	Hamlet	HWS1589	William Shakespeare	1589	20	26000	13000

Esto lo podemos hacer usando 4 diccionarios para guardar estas características comunes (nombre, código, autor, año de publicación, cantidad, precio y costo de producción) como claves y sus respectivos valores con la información específica de cada libro. En Python, podemos crear estos 4 diccionarios con las siguientes instrucciones:

```
Libro1 = {"nombre": "Harry Potter y la piedra Filosofal", "código": "HPJK1997", "autor":  
          "J.K. Rowling", "añoPublicacion": 1997, "cantidad": 200, "precio": 25000,  
          "costoProduccion": 9000}  
Libro2 = {"nombre": "Los Juegos del Hambre", "código": "JHSC2008", "autor": "Suzanne  
          Collins", "añoPublicacion": 2008, "cantidad": 100, "precio": 27000,  
          "costoProduccion": 12000}
```

```

Libro3 = {"nombre": "El Hobbit", "código": "EHJR1937", "autor": "J.R.R. Tolkien",
"añopublicacion": 1937, "cantidad": 50, "precio": 35000,
"costoProduccion": 15000}
Libro4 = {"nombre": "Hamlet", "código": "HWS1589", "autor": "William Shakespearde",
"añopublicacion": 1589, "cantidad": 20, "precio": 26000,
"costoProduccion": 13000}

```

Si ejecutamos esto en la terminal de Spyder, junto con la impresión de los datos, obtendremos:

```

In [7]: Libro1 = {"nombre": "Harry Potter y la piedra Filosofal", "código": "HPJK1997", "autor": "J.K. Rowling",
"añopublicacion": 1997, "cantidad": 200, "precio": 25000, "costoProduccion": 9000}

In [8]: Libro2 = {"nombre": "Los Juegos del Hambre", "código": "JHSC2008", "autor": "Suzanne Collins", "añopublicacion": 2008,
"cantidad": 100, "precio": 27000, "costoProduccion": 12000}

In [9]: Libro3 = {"nombre": "El Hobbit", "código": "EHJR1937", "autor": "J.R.R. Tolkien", "añopublicacion": 1937, "cantidad":
50, "precio": 35000, "costoProduccion": 15000}

In [10]: Libro4 = {"nombre": "Hamlet", "código": "HWS1589", "autor": "William Shakespearde", "añopublicacion": 1589, "cantidad":
20, "precio": 26000, "costoProduccion": 13000}

In [11]: print ("Estos son los libros \nLibro1", libro1, "\nLibro2", libro2, "\nLibro3", libro3, "\nLibro4", libro4 )
Estos son los libros
Libro1 {'nombre': 'Harry Potter y la piedra filosofal', 'código': 'HPJK1997', 'autor': 'J.K. Rowling', 'añopublicacion': 1997,
'cantidad': 200, 'precio': 25000, 'costoProduccion': 9000}
Libro2 {'nombre': 'Los Juegos del Hambre', 'código': 'JHSC2008', 'autor': 'Suzanne Collins', 'añopublicacion': 2008,
'cantidad': 100, 'precio': 27000, 'costoProduccion': 12000}
Libro3 {'nombre': 'El Hobbit', 'código': 'EHJR1937', 'autor': 'J.R.R. tolkien', 'añopublicacion': 1937, 'cantidad': 50,
'precio': 35000, 'costoProduccion': 15000}
Libro4 {'nombre': 'Hamlet', 'código': 'HWS1589', 'autor': 'William Shakespeare', 'añopublicacion': 1589, 'cantidad': 20,
'precio': 26000, 'costoProduccion': 13000}

```

Algo un poco más sofisticado sería escribir un programa dentro del cual exista una función que nos permita crear un libro (diccionario) dada su información. Como se muestra en el siguiente ejemplo:

```

ModuloLibreria.py
1 # -*- coding: utf-8 -*-
2 def crear_libro(nom: str, cod: str, autor: int, adp: int, cant: int, pdv: float, cpu: float)->dict:
3     dic_libro = { "nombre": nom,
4                   "codigo": cod,
5                   "autor": autor,
6                   "añopublicacion": adp,
7                   "cantidad": cant,
8                   "precio": pdv,
9                   "costoProduccion": cpu}
10    return dic_libro
11
12 #PROGRAMA PRINCIPAL
13 libro1 = crear_libro("Harry Potter y la piedra filosofal", "HPJK1997", "J.K. Rowling", 1997, 200 , 25000, 9000)
14 libro2 = crear_libro("Los Juegos del Hambre", "JHSC2008", "Suzanne Collins", 2008, 100 , 27000, 12000)
15 libro3 = crear_libro("El Hobbit", "EHJR1937", "J.R.R. tolkien", 1937, 50 , 35000, 15000)
16 libro4 = crear_libro("Hamlet", "HWS1589", "William Shakespeare", 1589, 20 , 26000, 13000)
17

```

El resultado de la ejecución de este programa es el mismo que obtuvimos creando los libros directamente en la terminal de Spyder. Partiendo de esta base, extienda el programa anterior (que fue el que usted mismo descargó de Brightspace) para implementar las funciones que se describen a continuación. **No olvide probar todas sus funciones desde la terminal de Spyder.**

Función 1: mayor_ganancia

Escriba una función llamada `mayor_ganancia` que reciba por parámetro los 4 libros (diccionarios) y retorne el libro (diccionario) que tiene mayor ganancia por unidad. Tenga en cuenta que la ganancia es el precio de venta – el costo de producción.

Función 2: hacer_pedido

Escriba una función llamada `hacer_pedido` que reciba por parámetro los 4 libros, el nombre del libro y retorne verdadero si se pudo hacer el pedido o false en caso de lo contrario. Tenga que solo se puede hacer un pedido si la cantidad de libros es menor o igual a 50 y siempre se piden 100 libros extra.

Función 3: publicacion_antes_anio

Escriba una función llamada `publicacion_antes_anio` que reciba como parámetro los 4 libros y un año. Debe retornar todos los libros publicados antes de ese año en un diccionario. El formato es llave = nombre del libro y el valor = año de publicación.

Función 4: ganancias_venta_libro

Escriba una función llamada `ganancias_venta_libro` que reciba como parámetro los cuatro libros, el nombre del libro y la cantidad de unidades a vender. Debe retornar en un diccionario el nombre del libro y las ganancias si se hiciera la venta. Tenga en cuenta que la ganancia por unidad es igual a precio de venta – costo de producción.

Función 5: venta_por_mayor

Escriba una función llamada `venta_por_mayor` que reciba los 4 libros, un nombre y la cantidad de unidades a vender. Debe retornar el costo total de la venta y el descuento recibido en un diccionario. Tenga en cuenta que recibe el 10% si compra una cantidad mayor al 25% de la cantidad que se tiene pero menor al 50%, recibe un 20% de descuento si compra más del 50% de la cantidad pero menos del 75% y recibe el 30% de descuento si compra más del 75% de la cantidad de los libros en la tienda. No debe importar si el nombre está en mayúscula o minúscula. Es decir que `juan` y `JUAN` son el mismo nombre.

Pruebas

Para estar seguro de que sus funciones quedaron bien implementadas, modifique los datos de creación de los libros y ejecute una vez más sus funciones desde la consola de Spyder.

Entrega

Entregue el archivo modificado a través de Brightspace en la actividad designada como “L3 - Dictionarios”.