Práctico #1

Test Automation With Python Noviembre 2016.

Fecha de Entrega: 26/nov/2016
Forma de entrega: Via Email a jbc.develop@gmail.com
Se Evalua:

- Cantidad de ejercicios realizados
- PEP 8 (con flake8 3.0)
- PEP 20: Zen de Python
- Uso de características de Python

1. Strings

1.1 Facturas

Dado una número de facturas, implementar una función que devuleva el string ‘Cantidad de facturas: <nro> ‘ donde nro es el número que se pasa como argumento. Si las facturas son mas de 12, se tiene que devolver ‘Cantidad de facturas: muchas’.

1.2 Ambos

Dado un string s, implementar la función ambos que devuelve un string construido con los dos primeros y dos últimos caracteres. Por ejemplo, aplicar ambos a ‘primavera’ devuelve ‘prra’. Si s posee menos de dos caracteres, el resultado es el string vacío.

1.3. Fix

Dado un string s, implementar una función fix que reemplaza todas las ocurrencias del primer caracter por ‘*’ a excepción de la primera ocurrencia. Por ejemplo evaluar fix a la palabra ‘burbuja’ devuelve ‘bur*uja’.
Ayuda, estudiar la función replace.

1.4. Mezclar

Dados dos strings a y b, implementar la función mezclar que devuelve el string a y b separados por un espacio, excepto las primeros caracteres de cada string que son intercambiados. Por ejemplo, mezclar(‘mix’, ‘pod’) devuelve ‘pix mod’.

2. Listas

2.1. Macheos

Implementar la función macheos que dada una lista de strings devuelve un número representando la cantidad de strings que tienen más de dos caracteres y cuyos últimos dos strings son iguales.
Nota: python no posee operador ++ pero += funciona.

2.2. front_x

Dada una lista de strings, implementar la funcion front_x que devuelve una lista ordenada exceptuando las palabras que comiencen con x, las cuales deben ir al principio. Por ejemplo, ['mix', 'xyz', 'apple', 'xanadu', 'aardvark'] devuelve ['xanadu', 'xyz', 'aardvark', 'apple', 'mix']

2.3. sort_last

Dada una lista de tuplas no vacias, implementar la funcion sort_last que devuelve una lista con las tuplas ordenadas de forma incremental según el último elemento de la tupla. Ejemplo: aplicar sort_last a [(1, 7), (1, 3), (3, 4, 5), (2, 2)] devuelve [(2, 2), (1, 3), (3, 4, 5), (1, 7)]

2.4. Tablas de multiplicar

Implementar la función tablas que dado un argumento nro, devuelve la tabla de multiplicar de nro del 1 al 10.

3. Diccionarios

3.1. Mapeo

Implementar la función mapeo que toma un string y devuelve un diccionario con cada caracter como clave y la posición del caracter como valor. Ejemplo, evaluar mapeo a ‘cosa’ devuelve {‘c’: 0, ‘o’:1, ‘s’:2, ‘a’: 3}

3.2. Busqueda reversa

Implementar la función busqueda_reversa que dado un diccionario y un objeto cualquiera, permita buscar por valores de diccionarios en vez de claves. Ejemplo:
>>> d = {‘c’: 0, ‘o’:1, ‘s’:2, ‘a’: 3}
>>> busqueda_reversa(d, 3)
’a’

4. Tipos combinados

4.1. Invitados

Imaginen que poseemos un diccionario de la forma **nombre** -> **estado** (clave -> valor), el estado representa si la persona cuyo nombre es **nombre** irá o no a tu cumpleaños, por ejemplo:

>>> invitados = {"María": "Asistirá", "Luis": "Asistirá", "Ángel": "No asistirá",
... "Pedro": "Asistirá", "Carla": "No asistirá"}

Implementar la función invitados que devuelve solo aquellas personas que asistirán al cumpleaños.

4.2. Justificar

Dado un string implementar la función justificar que fija la longitud de cada línea en 80 caracteres y justifica cada línea.

5. Orientación a Objetos

5.1 Puerta

Un cerrojo con combinación tiene las siguientes propiedades básicas: la combinación (una secuencia de tres números); el cerrojo se puede abrir proporcionando la combinación; y la combinación se puede cambiar, pero solamente por alguien que conoce la combinación actual. Diseñe una clase con métodos abrir, cerrar y cambiar_combinacion, y atributos para almacenar la combinación y el estado de la puerta, cerrada o abierta. La combinación debería asignarse en el constructor.

5.2. Jerarquía de Clases

Establezca una jerarquía de clases que represente a los estudiantes de una universidad sabiendo que todos los estudiantes se caracterizan por un nombre y un número. Hay varios tipos de estudiantes: los estudiantes ocasionales, sean de cursos de verano o de cursos específicos (se matriculan de un curso determinado), los que cursan solo una tecnicatura, licenciatura. Además, la universidad imparte cursos de especialización gratuitos para sus empleados.

5.3. Triágulo

Escriba una clase, triángulo, que represente un triángulo. La clase debe incluir los siguientes métodos que devuelven un valor lógico indicando el tipo del triángulo:
es_rectangulo (para triángulos rectángulos)
es_escaleno (todos los lados distintos)
es_isosceles (dos lados iguales y el otro distinto)
es_equilatero (los tres lados iguales)

5.4. Una Persona

Construya una estructura de clases que represente una serie de personas caracterizadas por el nombre (compuesto de nombre de pila y dos apellidos) y el número del DNI. Debe ser posible imprimir los datos completos de una persona y devolver el nombre o el DNI independientemente.

5.5. Genelogía

Modifique el ejemplo anterior para poder construir un árbol genealógico donde se establezca dinámicamente un vínculo que indique qué persona es el padre y cual la madre de una persona dada.