

04_ejercicios

February 22, 2019

1 Ejercicios 04

1. Realice un programa para:

- Leer los datos del archivo **aluminio.dat** y poner los datos del elemento en un diccionario de la forma:

```
d = {'S': 'Al', 'Z':13, 'A':27, 'M': '26.98153863(12)', 'P': 1.0000, 'MS':26.9815386(8) '}
```

- Modifique el programa anterior para que las masas sean números (float) y descarte el valor de la incerteza (el número entre paréntesis)
- Agregue el código necesario para obtener una impresión de la forma:

Elemento: S

Número Atómico: 13

Número de Masa: 27

Masa: 26.98154

Note que la masa sólo debe contener 5 números decimales

2. Realice un programa para:

- Leer el archivo **elementos.dat** en el directorio **data** y guardar los datos en un diccionario, cuyas claves serán los símbolos del elemento:

```
elementos.keys() = ['C', 'H', 'O', 'N', 'Na', 'Cl', 'Ca', 'Au']
```

y los valores serán diccionarios tal como se definieron en el ejercicio anterior. Por ejemplo:

```
elementos['H'] = {'S': 'H', 'Z':1, 'A':1, 'M': 1.00782503207, 'P': 0.999885, 'MS':1.00782503207(12)}
```

- Imprimir todos los elementos, en un formato legible (y si le sale: agradable) para personas, ordenados en valores crecientes de masa.

3. **PARA ENTREGAR:** Adapte los programas realizados en el punto anterior para trabajar con funciones. Se requiere que escriba:

- Una función que lea un archivo (cuyo nombre es el argumento) y devuelva un diccionario donde cada clave es el símbolo del elemento.
- Una función que escriba en un string todos los elementos, ordenados alfabéticamente por clave, en una forma similar a:

```

s = """
Elemento: C
Z = 6
A = 12
Masa = 12.0000000
Abundancia = 0.9893
Masa Promedio = 12.0107

Elemento: Ca
Z = 20
A = 40
Masa = 39.96259098
Abundancia = 0.96941
Masa Promedio = 40.078

...
"""

```

Esta función tendrá un argumento requerido que es el diccionario con los elementos y un argumento opcional `reverse` con valor por defecto `False`. Este argumento indica si los elementos se ordenan alfabéticamente de la manera natural (a,b,c...,y,z) o inversa (z,y,x, ... b,a).

- Una función que reciba un nombre de archivo y un string y escriba el string en el archivo dado.
- Finalmente, escriba también el código llamando a las funciones anteriores para realizar el trabajo de lectura y escritura de los elementos en archivos.

Instrucciones de envío: Envíe el código en un archivo llamado `04_SuApellido.py` por correo electrónico con asunto (*subject*) `04_SuApellido`.

4. Escriba funciones para analizar la divisibilidad de enteros:

- La función `es_divisible1(x)` que retorna verdadero si `x` es divisible por alguno de 2,3,5,7 o falso en caso contrario.
- La función `es_divisible_por_lista` que cumple la misma función que `es_divisible1` pero recibe dos argumentos: el entero `x` y una variable del tipo lista que contiene los valores para los cuáles debemos examinar la divisibilidad. Las siguientes expresiones deben retornar el mismo valor:

```

es_divisible1(x)
es_divisible_por_lista(x, [2,3,5,7])
es_divisible_por_lista(x)

```

- La función `es_divisible_por` cuyo primer argumento (mandatorio) es `x`, y luego puede aceptar un número indeterminado de argumentos:

```

es_divisible_por(x) # retorna verdadero siempre
es_divisible_por(x, 2) # verdadero si x es par
es_divisible_por(x, 2, 3, 5, 7) # igual resultado que es_divisible1(x) e igual a es_divisible_por(x, 2, 3, 5, 7, 9, 11, 13) # o cualquier secuencia de argumentos debe funcionar

```