

Ejercicios Temas Semana 5

Ejercicios Diccionarios, JSON, Archivos y Excepciones

Jonatan Gómez Perdomo, Ph. D.

jgomezpe@unal.edu.co

Arles Rodríguez, Ph.D.

aerodriguezp@unal.edu.co

Camilo Cubides, Ph.D. (c)

eccubidesg@unal.edu.co

Carlos Andrés Sierra, M.Sc.

casierrav@unal.edu.co

Research Group on Artificial Life – Grupo de investigación en vida artificial – (Alife)

Computer and System Department

Engineering School

Universidad Nacional de Colombia

Agenda

1 Diccionarios

2 Archivos

3 JSON

4 Excepciones



Problemas de diccionarios I

Problemas

- ➊ Desarrollar un algoritmo que imprima de manera ascendente los valores (todos del mismo tipo) de un diccionario.
- ➋ Desarrollar un algoritmo que verifique si todas las `clave:valor` de un diccionario se encuentran en otro diccionario.
- ➌ Desarrollar una función que reciba dos diccionarios como parámetros y los mezcle, es decir, que se construya un nuevo diccionario con las llaves de los dos diccionarios; si hay una clave repetida en ambos diccionarios, se debe asignar el valor que tenga la clave en el primer diccionario.
- ➍ Desarrollar un programa que dada una listas de personas, cada persona representada como el siguiente ejemplo:
`{"nombres": "Pedro Julio", "apellidos": "Tristán Merchán", "edad": 101}`, imprima los nombres y apellidos de las personas que están en un rango de edades.

Agenda

1 Diccionarios

2 Archivos

3 JSON

4 Excepciones



Ejercicios I

Problemas

- 1 Dado el archivo de texto files/SalesJan2009.csv, procese el archivo para obtener las compras realizadas en un país dado. Ejemplo:

Input	Output
United Kingdom	100



Ejercicios II

Problemas

- ② Dado el archivo de texto files/SalesJan2009.csv, procese el archivo para obtener las compras realizadas con un medio de pago dado. Ejemplo:

Input	Output
Visa	521



Agenda

- 1 Diccionarios
- 2 Archivos
- 3 JSON**
- 4 Excepciones



Ejercicios I

Problemas

Para el JSON con la estructura mostrada

```
{
  "jadiazcoronado":{
    "nombres": "Juan Antonio",
    "apellidos": "Díaz Coronado",
    "edad":19,
    "colombiano":true,
    "deportes":["Fútbol","Ajedrez","Gimnasia"]
  },
  ...
  "dmlunasol":{
    "nombres": "Dorotea Maritza",
    "apellidos": "Luna Sol",
    "edad":25,
    "colombiano":false,
    "deportes":["Baloncesto","Ajedrez","Gimnasia"]
  }
}
```


Ejercicios II

Problemas

Cree un programa que lea de un archivo con dicho JSON y

- 1 Imprima los nombres completos (nombre y apellidos) de las personas que practican el deporte ingresado por el usuario.
- 2 Imprima los nombres completos (nombre y apellidos) de las personas que estén en un rango de edades dado por el usuario.
- 3 Cree un JSON de deportes como sigue:

```
{  
  "Ajedrez": ["jadiazcoronado", ..., "dmlunasol"],  
  "Futbol": ["jadiazcoronado", ...],  
  "Gimnasia": ["jadiazcoronado", ..., "dmlunasol"],  
  ...  
  "Baloncesto": [..., "dmlunasol"]  
}
```



Ejercicios III

Problemas

- 4 Desarrolle un programa que lea dos archivos JSON, y encuentre los componentes clave:valor que son iguales en ambos. Genere un nuevo archivo JSON con las coincidencias exactas entre los dos archivos.
- 5 Desarrolle un programa que lea un archivo JSON, en el cual se encuentran las notas de los estudiantes del curso. Cada llave corresponde al código de cada estudiante, y su valor es una lista con las notas obtenidas en las actividades del curso. Se debe generar un nuevo archivo JSON que para uno de los estudiantes solo guarde el promedio de las notas obtenidas.



Ejercicios IV

Problemas

- 6 Desarrollar un programa que lea un archivo JSON que contiene una serie de cadenas de caracteres en minúscula, cada una con su propia llave. Estas llaves tienen una codificación, a forma de encriptación, en donde las vocales están descritas como otros símbolos: \$ en vez de a, # en vez de e, * en vez de i, ¬ en vez de o, y + en vez de u. Una vez leído el archivo, realice una desencriptación de todas las cadenas, es decir, convierta los símbolos a sus vocales correspondientes (si la cadena de entrada es "h¬l\$", la cadena resultante sería "hola"), y guarde el resultado en un nuevo archivo JSON.



Agenda

- 1 Diccionarios
- 2 Archivos
- 3 JSON
- 4 Excepciones



Ejercicios I

Problemas

- 1 Capture la excepción que evita que el usuario acceda a posiciones que no se encuentran definidas en la lista dada y muestre el mensaje Intenta acceder una posición que no está en el arreglo:

```
lista = [1, 2, 3, 4]
lista[5]
```

Si se ejecuta sin el manejo de la excepción se produce la siguiente salida:

```
IndexError      Traceback (most recent call last)
<ipython-input-42-64245f71fd49> in <module>
      1 lista = [1, 2, 3, 4]
----> 2 lista[5]

IndexError: list index out of range
```

Ejercicios II

Problemas

- ② Capture la excepción para evitar que un programador sume una cadena de texto a un número y muestre el mensaje
Los tipos de datos no cuadran para hacer la operación:

```
def operar(a, b):  
    return a+b  
  
def main():  
    a = int(input())  
    b = 'hola'  
    operar(a, b)  
  
main()
```



Ejercicios III

Problemas

Si se ejecuta el anterior programa, sin el manejo de la excepción, se produce la siguiente salida (con cualquier número dado por el usuario):

```
-----  
TypeError          Traceback (most recent call last)  
<ipython-input-43-63f2edb5e1e0> in <module>  
      7      operar(a, b)  
      8  
----> 9 main()  
  
<ipython-input-43-63f2edb5e1e0> in main()  
      5      a = int(input())  
      6      b = 'hola'  
----> 7      operar(a, b)  
      8  
      9 main()  
...  

```

Ejercicios IV

Problemas

Si se ejecuta el anterior programa, sin el manejo de la excepción, se produce la siguiente salida (con cualquier número dado por el usuario):

```
...
<ipython-input-43-63f2edb5e1e0> in operar(a, b)
      1 def operar(a, b):
----> 2     return a+b
      3
      4 def main():
      5     a = int(input())

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```



Ejercicios V

Problemas

- 3 Capture la excepción cuando se trata de obtener una llave que no se encuentra en un diccionario y muestre el mensaje
Intenta acceder una llave que no está en el diccionario:

```
def main():  
    dict = {'James': 'Java', 'Dennis' : 'C', 'Das': 'Python'}  
    print(dict['Ada'])  
  
main()
```



Ejercicios VI

Problemas

Si se ejecuta el anterior programa, sin el manejo de la excepción, se produce la siguiente salida:

```
-----  
KeyError      Traceback (most recent call last)  
<ipython-input-45-174134c9fede> in <module>  
      3      print(dict['Ada'])  
      4  
----> 5 main()  
  
<ipython-input-45-174134c9fede> in main()  
      1 def main():  
      2     dict = {'James': 'Java', 'Dennis' : 'C', 'Das':'Python'}  
----> 3     print(dict['Ada'])  
      4  
      5 main()  
  
KeyError: 'Ada'
```