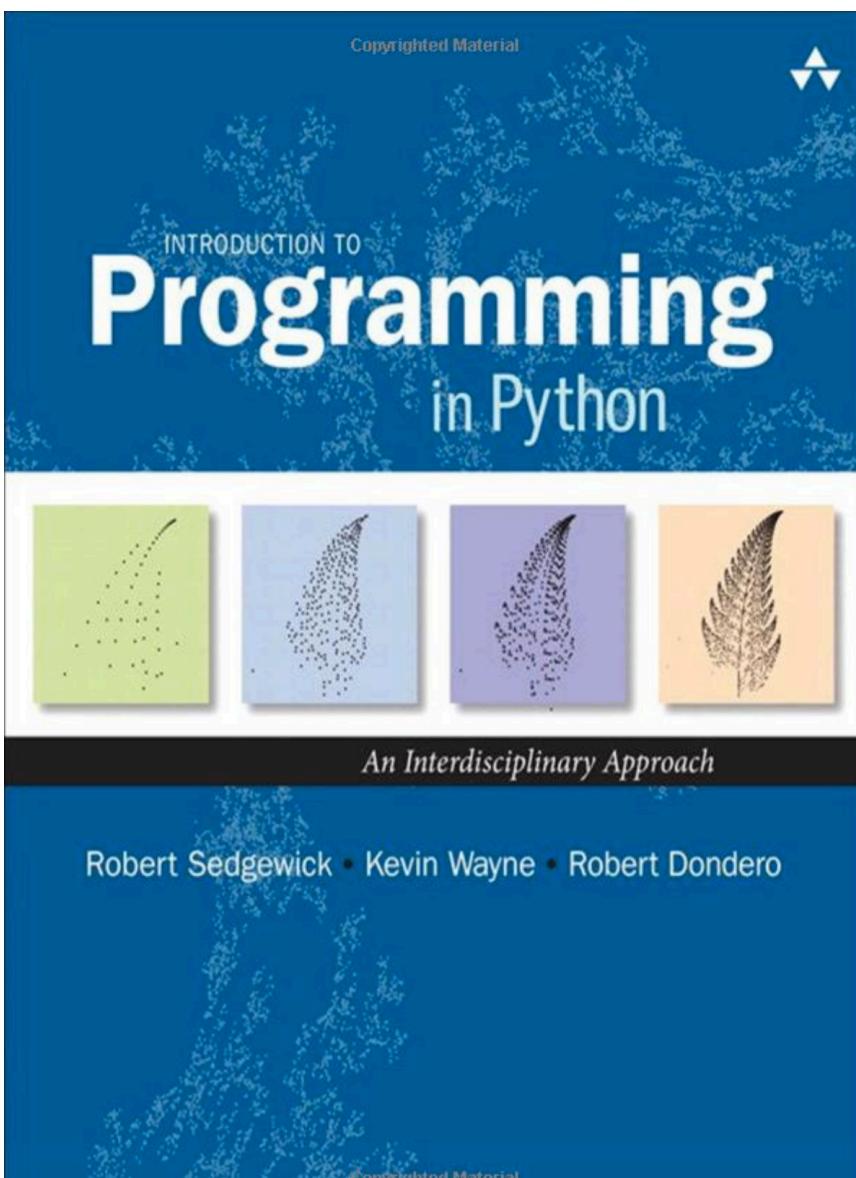


# Taller de Programación

## Clase 12: Diccionarios y Manejo de Archivos

Daniela Opitz  
dopitz@udd.cl



Basada en presentaciones oficiales de libro *Introduction to Programming in Python* (Sedgewick, Wayne, Dondero).

Disponible en <https://introcs.cs.princeton.edu/python>

# Outline

- Ejemplos con diccionarios
- Funciones para trabajar con strings
- Manejo de archivos
- Cómo resolver problemas?

# Actividad

El siguiente diccionario contiene información relacionada a la pobreza regional de nuestro país. Las llaves (keys) corresponden a los nombres las regiones de Chile, y los valores (values) corresponden a la información del porcentaje de pobreza de ingreso en la respectiva región (Fuente: Casen 2015). Imprima el nombre la región con mayor población en situación de pobreza de ingresos y su respectivo porcentaje.

```
pobreza_reg={'Tarapaca' : 7.1, 'Antofagasta' : 5.4, 'Atacama' : 6.9, 'Coquimbo' :  
13.8, 'Valparaíso' : 12.0, 'Libertador Bernardo OHiggins' : 13.7, 'Maule' : 18.7,  
'Biobío' : 17.6, 'La Araucanía' : 23.6, 'Los Lagos' : 16.1, 'Aysen' : 6.5,  
'Magallanes y La Antártica Chilena' : 4.4, 'Región Metropolitana de Santiago' : 7.1,  
'Los Ríos' : 16.8, 'Arica y Parinacota' : 9.7}
```

# Métodos para Trabajar con Strings

- **str.strip()**: Borra todo lo que está antes y después de la cadena
- **str.lstrip()**: Borra lo que está al inicio de la cadena
- **str.rstrip()**: Borra lo que está al final de la cadena
- **str.replace(str1, str2, n)**: Reemplaza la cadena str1 por la cadenas str2 una máximo de n veces.
- **str.split(separador, maxsplit)**: Retorna una lista de strings después de romper la cadena por un separador específico.

# Métodos para Trabajar con Strings

```
1cadena = " - - - programaresmuydivertido - - -"
2
3#Usando strip() para borrar todos los '-'
4print ("String después de remover '-' antes y después: ")
5print (cadena.strip('-'))
6
7# Usando lstrip para borrar los '-' anteriores
8print ("String después de remover todos '-' anteriores: ")
9print (cadena.lstrip('-'))
10
11# Usando rstrip para borrar los '-' posteriores
12print ("String después de remover todos los '-' posteriores: ")
13print (cadena.rstrip('-'))
```

# Métodos para Trabajar con Strings

```
1# Usando split() para separar
2x = 'azul,rojo,verde'
3print('String para separar por comas: ' + x + '\n')
4
5print('String para separar por comas: ' + str(x.split(',')) + '\n')
6print(x.split(',', 1))
7
8#Usando replace() para reemplazar
9print("String despues de reemplazar 'divertido' por 'aburrido'")
10print(cadena.replace('divertido', 'aburrido', 1))
```

# Manipulando Archivos



# Manejo de Archivos

- En Python, los archivos se pueden leer como una secuencia de líneas.
- Archivo se abre con función `open('rutaarchivo.txt', modo)`:
  - `modo = "r" # lectura`
  - `modo = "w" # escritura`
- Debes cerrar archivo con método `.close()`

Output

Archivo de  
entrada

gini\_by\_country.csv

```
Slovenia,0.251
Denmark,0.256
Slovak Republic,0.247
Czech Republic,0.257
Iceland,0.246
Norway,0.257
Finland,0.257
Belgium,0.266
Austria,0.274
Germany,0.289
```

Código

```
f = open("gini_by_country.csv","r")
for l in f:
    print(l)
f.close()
```

La variable `l` contiene cada línea  
del archivo `gini_by_country.csv`

```
$ python3 archivos.py
Slovenia,0.251
```

```
Denmark,0.256
```

```
Slovak Republic,0.247
```

```
Czech Republic,0.257
```

```
Iceland,0.246
```

```
Norway,0.257
```

```
Finland,0.257
```

```
Belgium,0.266
```

```
Austria,0.274
```

# Leyendo Archivos

```
file = open('archivo.txt', 'r') #Abre el archivo
for linea in file:
    print (linea)
    <instrucciones>
file.close() # Cierra el archivo
```

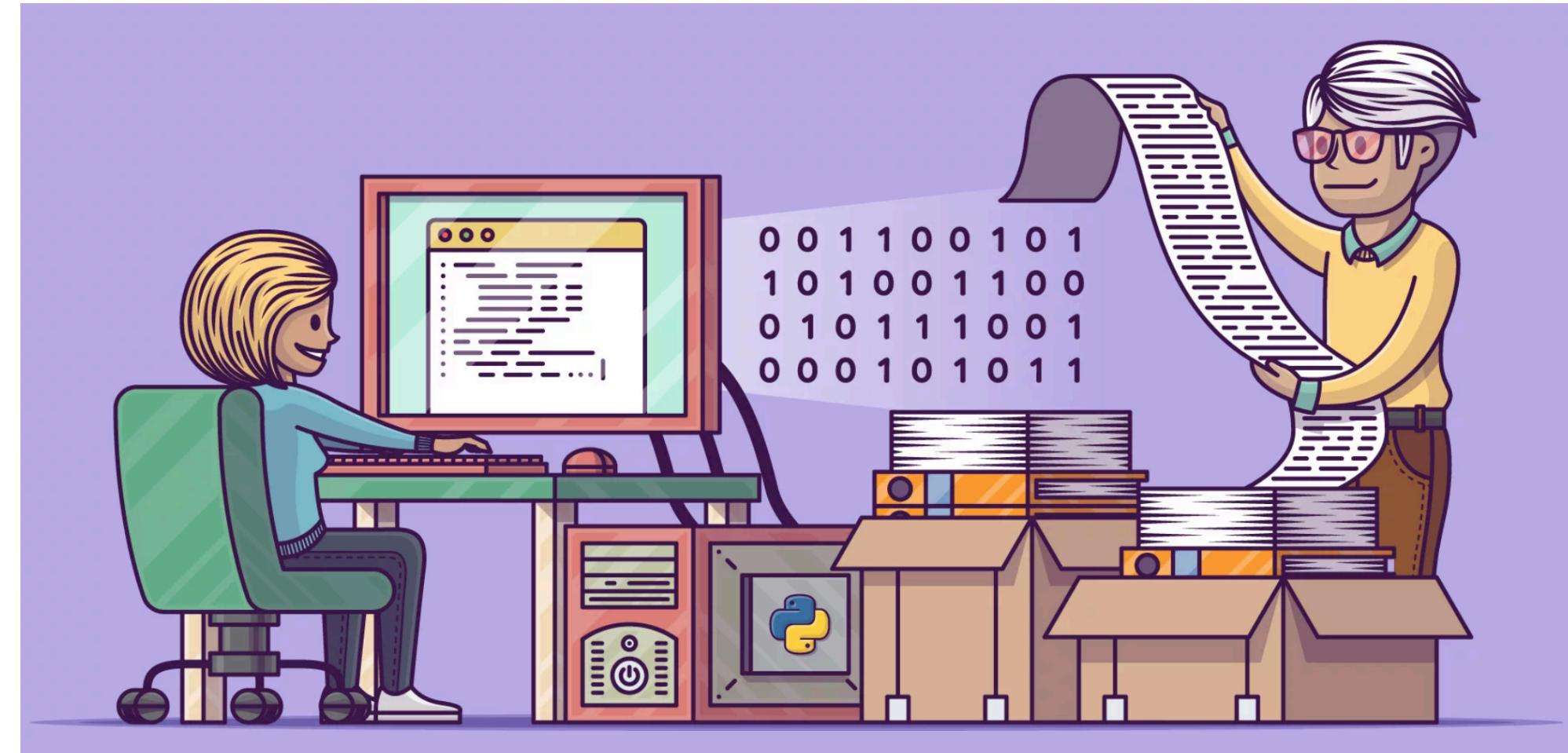
Necesito escribir  
`file.close()` para  
cerrarlo.

```
with open('archivo.txt', 'r') as file:
    for linea in file:
        print (linea)
        <instrucciones>
```

No necesito escribir  
`file.close()` para  
cerrarlo. El archivo se  
cierra automáticamente al  
terminar!

# Escribiendo Archivos

```
with open('archivo_nuevo.txt', 'w') as f:  
    data = 'Texto para escribir'  
    f.write(data)
```



# ¿Cómo Plantear y Resolver Problemas?

Libro del matemático húngaro **George Pólya**, describe métodos para resolver problemas y elaborar pequeñas demostraciones.

Sugiere que un problema matemático puede ser resuelto mediante los siguientes pasos:

1. Entender el problema.
2. Crear un plan.
3. Llevar a cabo el plan.
4. Revisar e interpretar el resultado

- Si no puedes resolver ese problema, entonces existe un problema más sencillo que éste que sí podrás resolver: encuéntralo.
- Si no puedes resolver el problema propuesto, intenta resolver primero un problema relacionado. ¿Podrías imaginar un problema relacionado más accesible?



# Actividad 2

El archivo 2020-05-01-CasosConfirmados.csv contiene el numero de casos confirmados de covid-19 por comuna.

Programe un código que imprima en pantalla una lista con las cinco comunas de la región Metropolitana con más contagiados ordenados de mayor a menor.

```
1 f = open('2020-05-01-CasosConfirmados.csv', 'r')
2 for l in f:
3     print(l)
4 f.close()
```