

# Archivos – escritura

Clase #15

IIC1103 – Introducción a la Programación

Marcos Sepúlveda ([marcos@ing.puc.cl](mailto:marcos@ing.puc.cl))

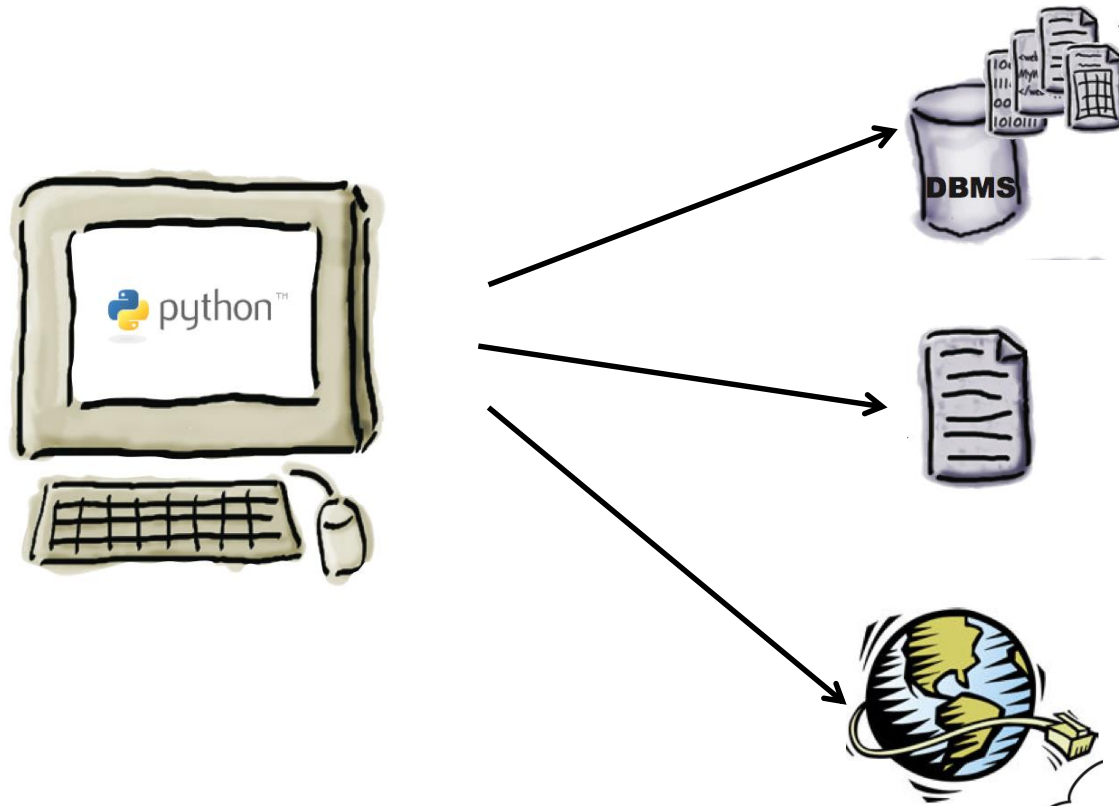
# Veremos hoy ...

---

- ▶ Escribir archivos
- ▶ Excepciones
- ▶ Ejercicios

# Escribir Archivos – persistencia

- ▶ ¿Qué sucede con nuestros datos luego de terminar de ejecutar un programa?
- ▶ Para usar los datos **después**, debemos guardarlos en una unidad de almacenamiento secundario.



# Leer archivos – abrir, procesar y cerrar

```
mi_archivo = open("mis_datos.txt")
```

1. Abrir

```
# Hacer algo con los datos en el archivo
```

2. Procesar

```
mi_archivo.close()
```

3. Cerrar

# Escribir archivos – abrir, procesar y cerrar

```
mi_archivo = open("mis_datos.txt", 'w')
```

1. Abrir o crear para escribir

```
# Hacer algo con los datos en el archivo
```

2. Procesar

```
mi_archivo.close()
```

3. Cerrar

# Modos básicos de acceso a un archivo

Modo	Descripción
'r'	Lee un archivo. Si el archivo no existe, entonces Python mostrará un mensaje de error. (Si se omite 'r' cuando se abre un archivo, entonces Python asume que es de lectura).
'w'	Escribe en un archivo. Si el archivo existe, entonces el archivo se sobrescribe perdiendo la información previa. Si el archivo no existe, entonces se crea.
'a'	Concatena la nueva información al final de la ya existente en el archivo. Si el archivo no existe, entonces se crea con la nueva información.

# Archivos de texto – nuevo .txt



El archivo existe pero  
está vacío, o no existe

# Escribir archivos – alternativa 1

---

```
data = open("nuevo.txt", 'w')
print("Hola!", file=data)
print("Escribir archivos es fácil", file=data)
data.close()
```



## Escribir archivos – alternativa 2

---

```
data = open("nuevo.txt", 'w')  
# recordar: '\n' es un caracter especial para salto de línea  
data.write("Hola!\n")  
data.write("Escribir archivos es fácil\n")  
data.close()
```

## Escribir archivos – alternativa 3

---

```
data = open("nuevo.txt", 'w')  
# recordar: '\n' es un caracter especial para salto de línea  
texto = "Hola!\nEscribir archivos es fácil\n"  
data.write(texto)  
data.close()
```

## Escribir archivos – alternativa 4

---

```
# se puede combinar print y write
data = open("nuevo.txt", 'w')
print("Hola!", file=data)
texto = "Escribir archivos es fácil\n"
data.write(texto)
data.close()
```

# Archivos de texto – nuevo.txt

```
1  Hola!  
2  Escribir archivos es fácil
```

Ahora el archivo  
contiene las dos líneas

# Excepciones

- El manejo de **try-except-finally** permite capturar un eventual error en un bloque, evitando así que el programa termine de forma inesperada con un mensaje de error (aquellos con letras rojas).



# Escritura sin manejo de errores

```
### sin manejo de errores
```

```
data = open("nuevo.txt", 'w')  
print("Hola!", file=data)  
print(a) # genera error  
print("Escribir archivos es fácil", file=data)  
data.close()
```

```
>>>  
Traceback (most recent call last):  
  File "C:\Users\Marcos\Desktop\Python\Clases\clase 21\excepciones.py", line 5, in <module>  
    print(a)  
NameError: name 'a' is not defined
```

nuevo.txt



The diagram consists of a rectangular box representing a file named 'nuevo.txt'. Below this box, there is a blue-outlined callout box with a tail pointing towards the file box. Inside the callout box, the text 'Archivo queda vacío' (File is empty) is written in black.

Archivo queda  
vacío

# Escritura con manejo de errores

```
### con manejo de errores
```

```
try:
    data = open("nuevo.txt", 'w')
    print("Hola!", file=data)
    print(a) # genera error
    print("Escribir archivos es fácil", file=data)
except:
    print ("*** Error escribiendo archivo")
finally:
    data.close()
    print ("Uff! ya lo cerramos")
```

```
>>>
*** Error escribiendo archivo
Uff! ya lo cerramos
```

nuevo.txt

Hola!

Alcanza a escribir  
primera línea

# Archivos de texto – nuevo .txt

```
1  Hola!  
2  Escribir archivos es fácil
```

Siempre escribimos de nuevo el archivo.  
¿Cómo agregamos más líneas?



# Escribir archivos – *append*

- ▶ Se puede abrir un archivo para agregar información adicional al final de éste usando 'a'.

```
try:
    data = open("nuevo.txt", 'a')
    print("Es fácil agregar texto", file=data)
except:
    print ("*** Error escribiendo archivo")
finally:
    data.close()
    print ("Uff! ya lo cerramos")
```

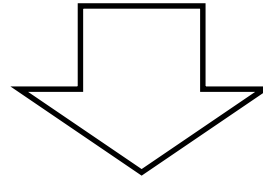
# Archivos de texto – nuevo.txt

```
1  Hola!  
2  Escribir archivos es fácil  
3  Es fácil agregar texto
```

Se agregó una línea  
al final

# Movimiento posición en archivo con seek

```
1 Hola!  
2 Escribir archivos es fácil  
3 Es fácil agregar texto
```



`data.seek(0)`

`data.seek(5)`

0	1	2	3	4	5	6	7	8	...
H	o	l	a	!	\n	E	s	c	...

`data.seek(3)`

# Leer archivos – seek y tell

```
try:
    data = open("nuevo.txt")
    print(data.readline().strip())

    posSegundaLinea = data.tell()

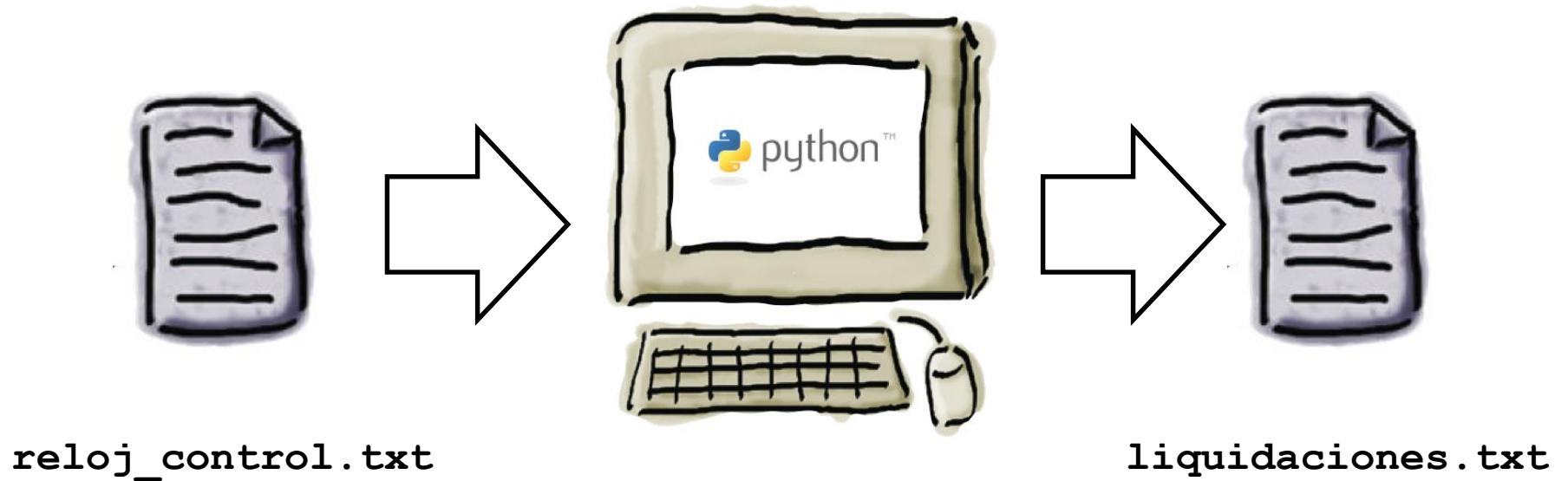
    print("Leo 3 veces la segunda línea:")
    for j in range(0,3):
        data.seek(posSegundaLinea)
        print(data.readline().strip())
except:
    print ("*** Error leyendo archivo")
finally:
    data.close()
    print ("Uff! ya lo cerramos")
```

```
>>>
Hola!
Leo 3 veces la segunda línea:
Escribir archivos es fácil
Escribir archivos es fácil
Escribir archivos es fácil
Uff! ya lo cerramos
```

# Ejercicio – leer y escribir archivos

- ▶ En el archivo `reloj_control.txt` se encuentra la información registrada por un marcador de ingreso y salida del personal de una compañía:
  - `RUT ; Horas Trabajadas ; Valor Hora Trabajada ; Descuento (%)`
- ▶ Se requiere generar el archivo `liquidaciones.txt` con las liquidaciones de los empleados, que contenga los siguientes campos separados por `' ; '`:
  - `RUT; Salario Bruto; Salario Líquido`
  - `Salario Bruto = Horas Trabajadas * Valor Hora Trabajada`
  - `Salario Líquido = Salario Bruto * (1-Descuento/100.0)`

# Ejercicio – leer y escribir archivos



# Ejercicio – reloj\_control.txt

1	11.232.213-0 ; 23 ; 1450 ; 3
2	11.232.214-0 ; 24 ; 1450 ; 3
3	11.232.215-0 ; 22 ; 1450 ; 3
4	11.232.216-0 ; 10 ; 1450 ; 3
5	11.235.217-0 ; 32 ; 1250 ; 2
6	11.235.218-0 ; 31 ; 1250 ; 2
7	11.235.219-0 ; 33 ; 1250 ; 2
8	11.235.220-0 ; 34 ; 1250 ; 2
9	11.245.221-0 ; 23 ; 1450 ; 3
10	11.242.222-0 ; 24 ; 1450 ; 3
11	11.242.223-0 ; 22 ; 1450 ; 3
12	11.242.224-0 ; 10 ; 1450 ; 3
13	11.245.225-0 ; 32 ; 1250 ; 2
14	11.245.226-0 ; 31 ; 1250 ; 2

# Ejercicio – leer y escribir archivos

---

```
try:
    lector = open("reloj_control.txt")
    escritor = open("liquidaciones.txt", 'w')
    for linea in lector:
        rut, hh, valor_h, dcto = linea.split(";")
        rut = rut.strip()
        bruto = int(hh)*int(valor_h)
        liquido = bruto*(1-int(dcto)/100.00)
        print(rut,bruto,liquido,sep=';',file=escritor)
except:
    print ("Error escribiendo un archivo")
finally:
    lector.close()
    escritor.close()
    print ("Uff! cerramos los archivos")
```



# Ejercicio – liquidaciones.txt

1	11.232.213-0;33350;32349.5
2	11.232.214-0;34800;33756.0
3	11.232.215-0;31900;30943.0
4	11.232.216-0;14500;14065.0
5	11.235.217-0;40000;39200.0
6	11.235.218-0;38750;37975.0
7	11.235.219-0;41250;40425.0
8	11.235.220-0;42500;41650.0
9	11.245.221-0;33350;32349.5
10	11.242.222-0;34800;33756.0
11	11.242.223-0;31900;30943.0
12	11.242.224-0;14500;14065.0
13	11.245.225-0;40000;39200.0
14	11.245.226-0;38750;37975.0