

Tecnologías de la Información II

Clase 06: Ciclos II parte & Funciones

Daniela Opitz, Diego Caro
dcaro@udd.cl



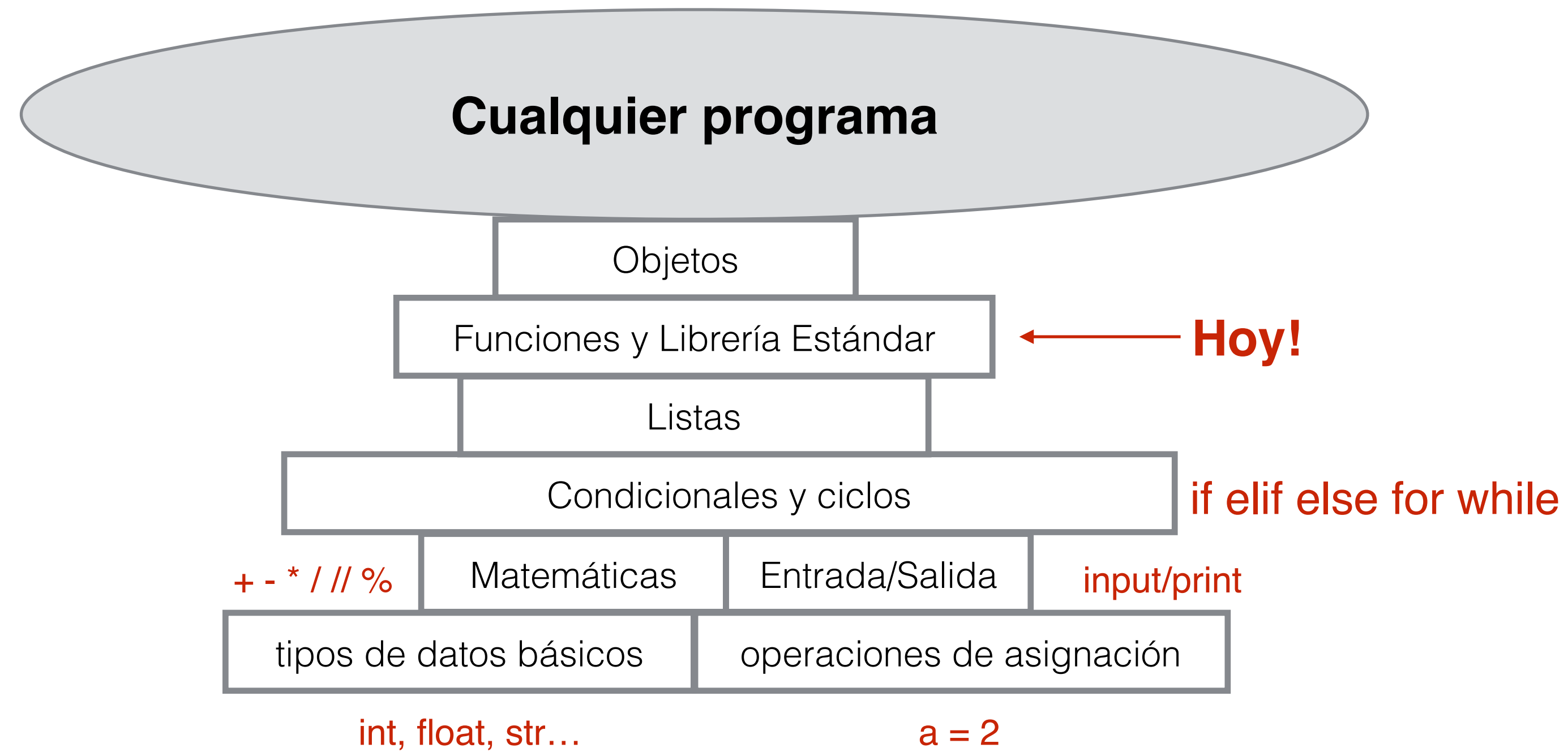
Basada en presentaciones oficiales de libro Introduction to Programming in Python (Sedgewick, Wayne, Dondero).

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

Outline

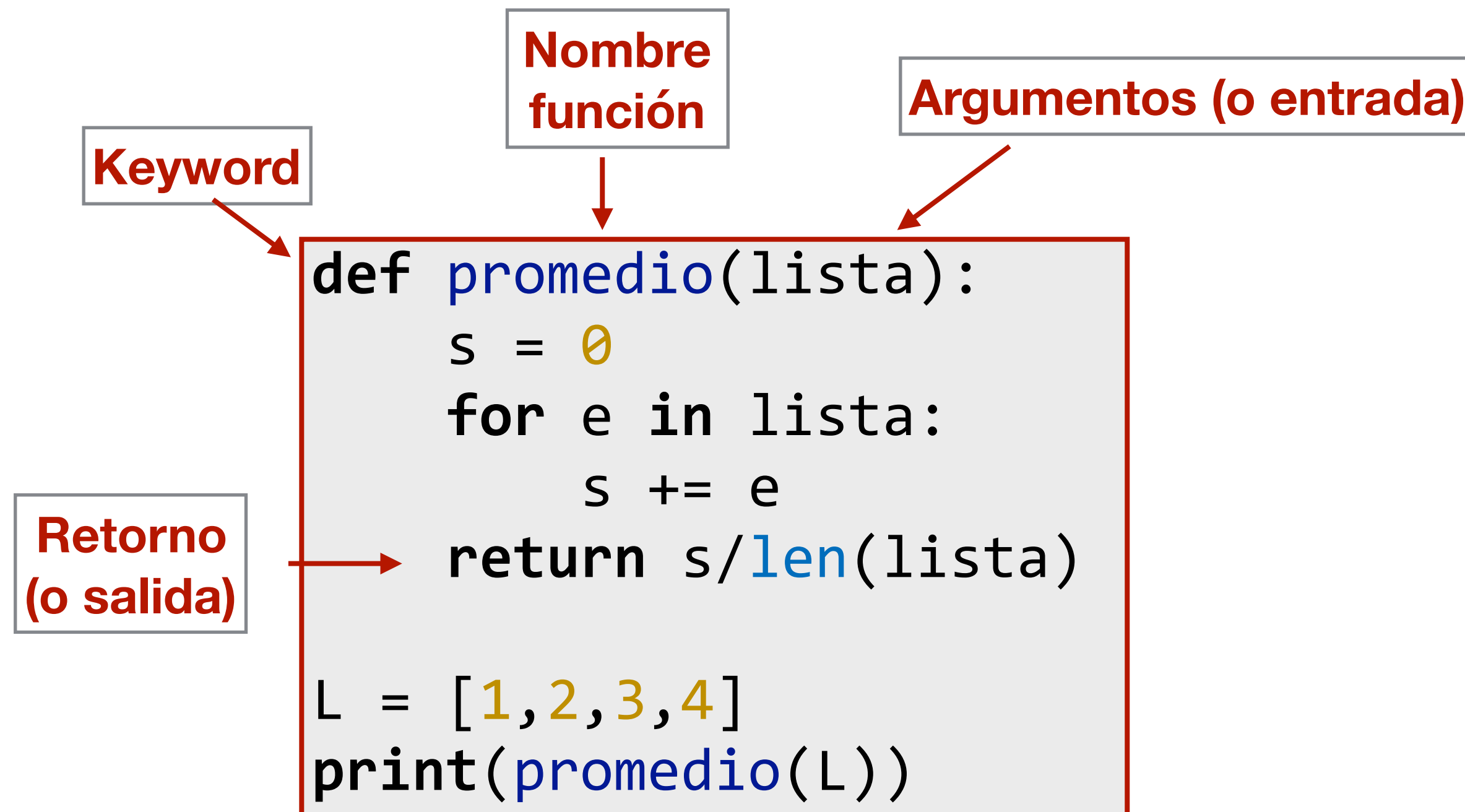
- Funciones
- Scope: variables parámetros, variables locales, variables globales
- Funciones en la librería estándar: sorted, reversed, sum, min, max, etc...
- Manejo de archivos

¿Dónde estamos?



Funciones

- **Funciones:** bloque de código que se ejecuta cuando se solicita.
- **Objetivo:** dividir el programa en partes más pequeñas que se pueden reutilizar.



También puede recibir
varios parámetros

```
def suma(a, b):  
    c = a+b  
    return c  
  
s = suma(2, 5)
```

Funciones: **return**

- **return** se utiliza para devolver el resultado de ejecutar el algoritmo de la función.
- Efectos secundario: la función se detiene (incluso si return está en un ciclo!).
- **Nota:** si una función no tiene return, por defecto devuelve None.

```
1 def es_primo(n):  
2     if n < 2: return False  
3     i = 2  
4     while i*i <= n:  
5         if n % i == 0: return False  
6         i += 1  
7     return True  
8  
9 print('9 es primo?', es_primo(9))
```

Ver demo en <https://goo.gl/VXUBGC>

Scope

- Scope (o ámbito) define la visibilidad de variables.
- El scope por defecto es el global. Todas las funciones tienen acceso al scope global.
- Cada función tiene su propio scope local, y solo se puede acceder a esas variables desde la función.
- Los parámetros de las funciones son parte del scope local.

```
1 x = 'hola'
2 y = 'miau'
3
4 def mymin(x, y):
5     if x < y: return x
6     else: return y
7
8 m = mymin(9,2)
9
10 print (m)
```

Scope global

x = 'hola'
y = 'miau'

Scope local: mymin

x = 9
y = 2

Parámetros: alias o paso por referencia

- Los parámetros de una función son alias (referencias). Consecuencia: si variables son mutables, es posible modificar su contenido... aunque estén en otro scope!.
- Sin embargo, si las variables son de tipo inmutable no hay problema.
 - Tipos de datos inmutables: int, float, str, bool.

```
1 def inc(L):  
2     L.append(33)  
3  
4 a = [1, 2]  
5 print(a)  
6 inc(a)  
7 print(a)
```

```
1 def inc(j):  
2     j += 1  
3  
4 i = 99  
5 print(i)  
6 inc(i)  
7 print(i)
```

```
1 def inc(j):  
2     j += 1  
3     return j  
4  
5 i = 99  
6 print(i)  
7 i = inc(i)  
8 print(i)
```

Usando Funciones

¿Calcule la variación en el tiempo de viaje?

Día	Tiempo de viaje en minutos
1	67
2	45
3	84
s	19,553

Día	Tiempo de viaje en minutos
1	70
2	70
3	70
s	0

Desviación Estándar

$$s = \sqrt{\frac{1}{n - 1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Promedio

Usando Funciones

```
1 from math import sqrt
2
3 L = [67, 45, 84]
4
5 s = 0
6 for e in L:
7     s += e
8 prom = s / len(L)
9
10 d = 0
11 for e in L:
12     d += (e - prom)*(e - prom)
13 sd = sqrt(d / (len(L)-1))
14 print(sd)
```

```
1 from math import sqrt
2
3 def prom(seq):
4     return sum(seq)/len(seq)
5
6 def desviacion(seq):
7     p = prom(seq)
8     d = 0
9     for e in seq:
10         d += (e - p)*(e - p)
11     return sqrt(d / (len(seq)-1))
12
13 L = [67, 45, 84]
14 print(desviacion(L))
```

L = [4, 8, 15, 16, 23, 42]

	Código Típico	Función en Python
Encontrar el máximo valor en una lista	<pre>maxi = L[0] for elem in L: if elem > maxi: maxi = elem print(maxi)</pre>	<pre>maxi = max(L)</pre>
Encontrar el mínimo valor en una lista	<pre>mini = L[0] for elem in L: if elem < mini: mini = elem print(mini)</pre>	<pre>mini = min(L)</pre>
Obtener el promedio	<pre># promedio suma = 0.0 for elem in L: suma = suma + elem prom = suma/N</pre>	<pre>prom = sum(L)/len(L)</pre>
Copiar elementos a otra lista	<pre>L2 = [] for elem in L: L2.append(elem)</pre>	<pre>L2 = L.copy()</pre>
Invertir elementos del arreglo	<pre>for i in range(N): temp = L[i] L[i] = L[N-i-1] L[N-i-1] = temp</pre>	<pre>L = list(reversed(L))</pre>
Extraer nombre y extensión de un archivo	<pre>nombre = 'hola.py' partes = [] for i in range(len(nombre)): if nombre[i] == '.': n = nombre[0:i] e = nombre[i+1:] partes.append(n) partes.append(e) break print('nombre archivo:', partes[0]) print('extension archivo:', partes[1])</pre>	<pre>s = 'hola.py' partes = s.split('.') nombre = partes[0] extension = partes[1]</pre>

DRY: Don't repeat yourself!

Si tienes más de una forma de expresar un algoritmo, escoge la más sencilla.



DRY

* https://en.wikipedia.org/wiki/Don%27t_repeat_yourself

Manejo de Archivos (Ejercicio)

Usted ha sido contratada por la Organización de las Naciones Unidas como consultor(a) experto Data Science. Su primer trabajo es estudiar las economías de los países de acuerdo al nivel de desigualdad de ingresos. La desigualdad en los ingresos se puede cuantificar a través del Coeficiente de Gini. Este coeficiente varía entre 0 y 1, donde 0 indica igualdad absoluta y 1 desigualdad absoluta.

Su primera tarea como Data Scientist es reportar los 5 países con más desigualdad en los ingresos. Con este fin, usted debe hacer un programa que lea el archivo [gini_by_country.csv](#) y que imprima en pantalla estos 5 países.

Nota: el archivo [gini_by_country.csv](#) contiene varias líneas, una por país. Cada línea contiene dos campos separados por coma, donde el primer campo indica el país y el segundo campo indica su Coeficiente de Gini. Por ejemplo, la línea `Chile,0.454` indica que Chile tiene un Coeficiente Gini de **0.454** (fuente: <http://stats.oecd.org/Index.aspx?DataSetCode=IDD>)

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()`

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`

Output

```
$ 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
```


Resumen

Conceptos

- **Función:** bloque de código que solo se activa cuando se ejecuta.
- **Scope:** ámbito en el que está disponible una variable

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

https://docs.python.org/3/reference/lexical_analysis.html

Funciones

- **min, max, round, sum:** obtiene mínimo, máximo, redondea al entero más cercano, suma.
- **sorted:** devuelve una lista ordenada
- **reversed:** devuelve una secuencia invertida

		Built-in Functions		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

<https://docs.python.org/3/library/functions.html>