



CAPITULO 6 (CONT):

RECURSIVIDAD

$$y = f(x) = \frac{2x+1}{3}$$



- Vimos como utilizar en nuestro programa VARIABLES

LOCALES

GLOBALES

- Además de esto, aprendimos que en el PASO de parámetros, podemos actuar desde la función sobre el argumento o NO.
- También conocimos, como funciona Python internamente para organizarse cuando llamamos a una función desde el cuerpo de otra función y así sucesivamente. Con la PILA!.



TEMARIO

- 6 Funciones.
 - 6.8 Recursión.
- Repaso de:
 - Paso de parámetros
 - Procedimientos que modifican argumentos
 - Problemas.

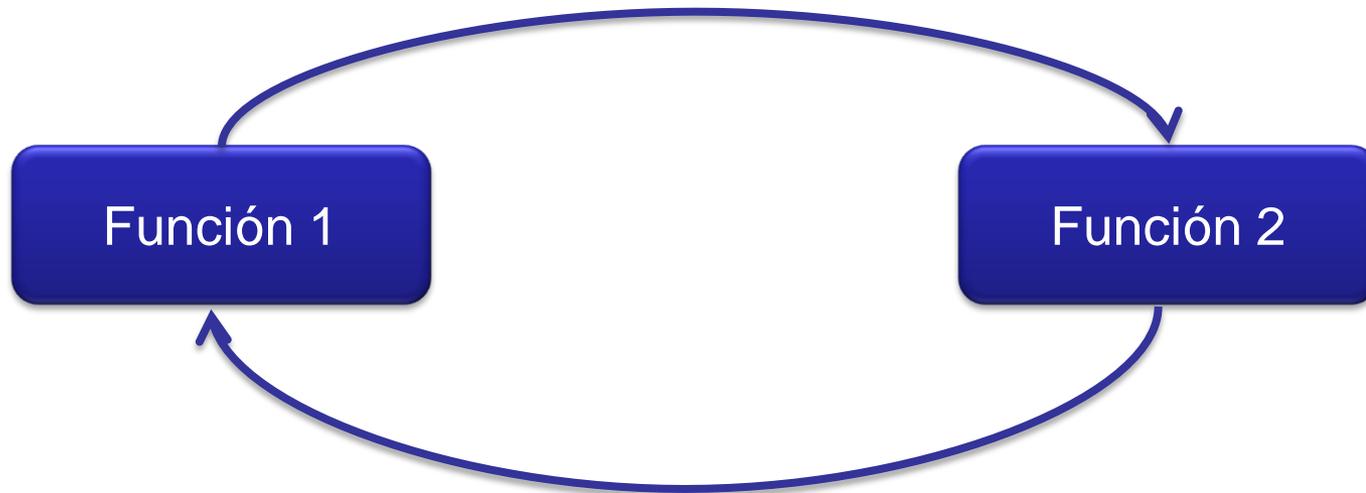


TEMARIO

- 6 Funciones.
 - 6.8 Recursión.
 - 6.8.1 Calculo recursivo del factorial
 - 6.8.6 Recursión Indirecta
- Repaso de:
 - Paso de parámetros
 - Procedimientos que modifican argumentos
 - Problemas.



- Hemos visto que DESDE una función, podemos llamar a OTRA función.



- Que pasa, si esta segunda FUNCION, llama a la PRIMERA que la invoco a ella?

- Y peor aun, que pasaría si una función se llama a si misma?



- Ambos casos son posibles!.
- Una función que se llama a si mismo es **RECURSIVIDAD DIRECTA**.
- El anterior, que incluye una función numero 2, es llamado **RECURSIVIDAD INDIRECTA**.



- 6 Funciones.

- 6.8 Recursión.

- **6.8.1 Calculo recursivo del factorial**

- 6.8.6 Recursión Indirecta

- Repaso de:
 - Paso de parámetros
 - Procedimientos que modifican argumentos
 - Problemas.



- Veamos esto con un ejemplo: el **calculo del factorial!**
- El calculo recursivo del factorial de un numero natural.
- Veamos la **definición matemática** del mismo:

$$n! = \begin{cases} 1, & \text{si } n = 0 \text{ o } n = 1; \\ n \cdot (n - 1)!, & \text{si } n > 1. \end{cases}$$



- Analicemos la definición: la función se define en torno de si misma:
 - Indica cuanto vale el factorial de 1.
 - Indica que el factorial de un numero, es igual al factorial del numero anterior multiplicado por el numero.

• EJ:

$$0! = 1$$

$$1! = 1$$

$$2! = 1! \bullet 2 = 1 \bullet 2 = 2$$

$$3! = 2! \bullet 3 = 1 \bullet 2 \bullet 3 = 6$$

$$4! = 3! \bullet 4 = 1 \bullet 2 \bullet 3 \bullet 4 = 24$$

.....



- Intentemos **CODIFICAR** esto:
- Debemos **definir una función**, que reciba un argumento, un numero entero.
- Si el numero que recibió es 0 o 1, el resultado es 1.
- Si el numero es otro, debe calcular el resultado como el numero recibido multiplicado por el llamado a la función con el numero anterior...
- Devolver al programa principal el **RESULTADO**



- Intentemos CODIFICAR esto:

```
1 def factorial(n):  
2     if n == 0 or n == 1:  
3         resultado = 1  
4     elif n > 1:  
5         resultado = n * factorial(n-1)  
6     return resultado
```



Recursión

Ing. Ventre, Luis O.

- Como sería el llamado a función si le envié $n=5$?...hágalo con PAPEL y LAPIZ!

1)  $5! = 5 \cdot 4!$

2)  $5! \rightarrow 4!$

3)  $4! = 4 \cdot 3!$

4)  $4! \rightarrow 3!$

5)  $3! = 3 \cdot 2!$

6)  $3! \rightarrow 2!$

7)  $2! = 2 \cdot 1!$

8)  $2! \rightarrow 1!$

9)  $1!$



- Como sería el llamado a función si le envió $n=5$?

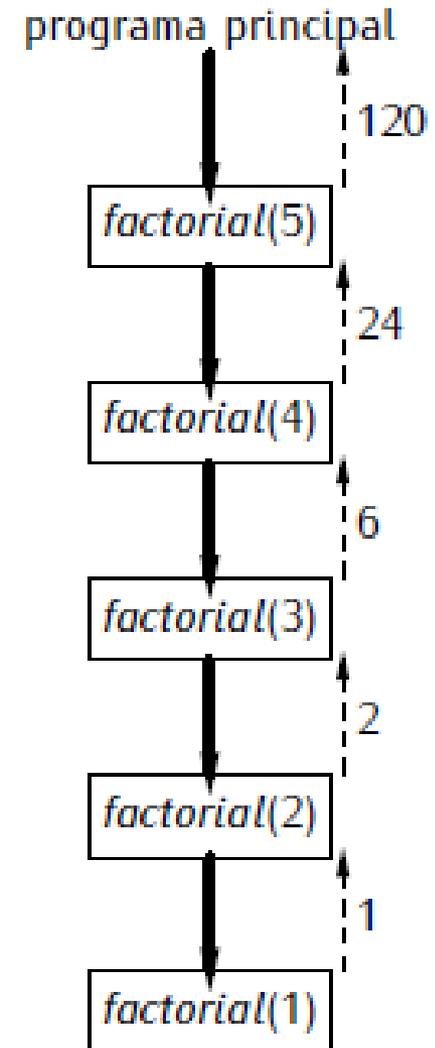
10)					
	$5!$	$4!$	$3!$	$2!$	$1! = 1$
11)					
	$5!$	$4!$	$3!$	$2! \leftarrow 1! = 1$	
12)					
	$5!$	$4!$	$3!$	$2! = 2 \cdot 1$	
13)					
	$5!$	$4!$	$3! \leftarrow 2! = 2$		
14)					
	$5!$	$4!$	$3! = 3 \cdot 2$		

15)			
	$5!$	$4! \leftarrow 3! = 6$	
16)			
	$5!$	$4! = 4 \cdot 6$	
17)			
	$5! \leftarrow 4! = 24$		
18)			
	$5! = 5 \cdot 24$		



- El árbol de llamadas a función para $n = 5$ será:

```
1 def factorial(n):  
2     if n == 0 or n == 1:  
3         resultado = 1  
4     elif n > 1:  
5         resultado = n * factorial(n-1)  
6     return resultado
```





- Regresión INFINITA....puede ver algún error?

```
1 def factorial(n):  
2     if n == 1:  
3         return 1  
4     else:  
5         return n * factorial(n-1)
```

Calcule de acuerdo a este programa el resultado de factorial con $n = 0$



TEMARIO

- 6 Funciones.

- 6.8 Recursión.

- 6.8.1 Calculo recursivo del factorial

- 6.8.6 Recursión Indirecta

- Repaso de:

- Paso de parámetros
 - Procedimientos que modifican argumentos
 - Problemas.



- Recordemos: Las funciones vistas son **RECURSIONES DIRECTAS**, ya que las funciones se LLAMAN a si mismas.
- Existe otra modalidad en la cual, una función llama a otra, y esta ultima invoca nuevamente la primera. Esto es **RECURSION INDIRECTA**.
- Veremos un ejemplo sencillo, e ilustrativo de la **RECURSION INDIRECTA**



- Ejemplo:
- Este programa nos permitirá decidir si un número es PAR, observando si el anterior es IMPAR.

- un número n es par si $n - 1$ es impar,
- un número n es impar si $n - 1$ es par.

- Pero necesitamos un caso base!

- 0 es par.



- Debemos implementar dos funciones en python:
- La función PAR: La cual recibe un argumento entero.
 - Si el valor es IGUAL a 0, devuelve TRUE.
 - Sino devuelve el resultado de llamar a IMPAR($n-1$)

```
1 def par(n):  
2     if n == 0:  
3         return True  
4     else:  
5         return impar(n-1)
```

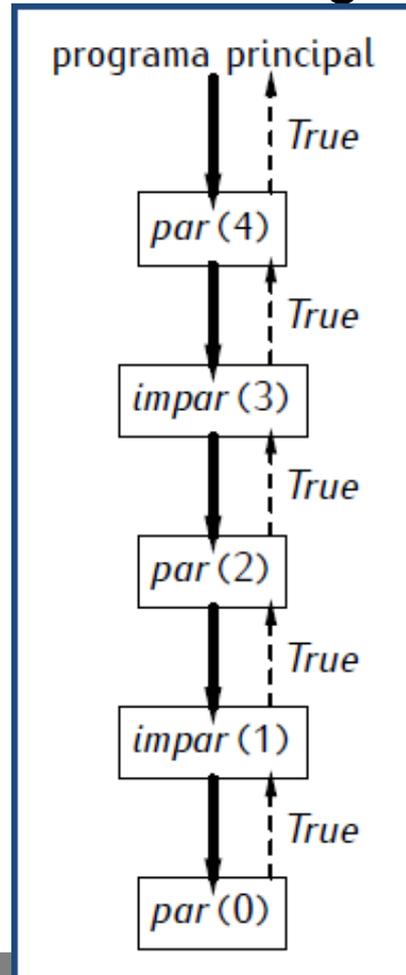


- Debemos implementar dos funciones en python:
- La función IMPAR: La cual recibe un argumento entero.
 - Si el valor es IGUAL a 0, devuelve FALSE.
 - Sino devuelve el resultado de llamar a PAR(n-1)

```
7 def impar(n):  
8     if n == 0:  
9         return False  
10    else:  
11        return par(n-1)
```



- Con esto comprendido, analicemos el árbol de llamadas por ejemplo, para la función PAR argumento 4.





TEMARIO

- 6 Funciones.
 - 6.8 Recursión.
 - 6.8.1 Calculo recursivo del factorial
 - 6.8.6 Recursión Indirecta
- Repaso de:
 - Paso de parámetros
 - Procedimientos que modifican argumentos
 - Problemas.



- Realice un bucle FOR IN, que imprima los números del 0 al 5 y utilice la función RANGE.

```
>>> for i in range(6):  
...     print i  
...
```

- Ahora realice un bucle FOR IN, que dada la siguiente LISTA imprima SUS elementos: lista = [1, 5, 10, 15, 20]

```
>>> lista = [1,5,10,15,20]  
>>>  
>>> for i in lista:  
...     print i
```



- Ahora realice un bucle FOR IN que dada la LISTA anterior, imprima sus elementos PERO RECORRIENDO sus elementos INDEXADOS por el bucle. «lista[i]» y utilice la función RANGE y LEN

```
>>> lista = [1,5,10,15,20]
>>>
>>> for i in range(len(lista)):
...     print lista[i]
```



NO OLVIDAR

Ing. Ventre, Luis O.



- Si un parámetro modifica su valor con una asignación, obtendrá una nueva zona de memoria y perderá relación el argumento que le dio origen.
- Operaciones como append, del o **asignaciones a elementos indexados**, modifican la propia lista, por lo que se afecta el parámetro y el argumento!
- Las cadenas son inmutables, no pueden cambiar su valor mediante operaciones como DEL, APPEND, o asignación directa a un elemento indexado.



EJERCICIO FINAL 1

Ing. Ventre, Luis O.

- Realice un programa que a través de una función llamada **ingreso_dim()**, solicite al usuario el INGRESO de la dimensión de una matriz N y M las cuales deben ser GLOBALES.
- Atraves de un procedimiento llamado **cargar_datos(matriz)**, solicitar el ingreso por teclado de los datos de la matriz. El procedimiento debe modificar los elementos del argumento recibido. No usar return.



- Una vez ingresada la matriz, a través de un procedimiento llamado **imprime_matriz(matriz)** imprimir la matriz por pantalla de forma ordenada y visible como matriz.
- Realice una función llamada **suma_columnas(matriz)**, que reciba como argumento la matriz y devuelva al programa principal una lista con los totales de sumar cada columna de la matriz.
- Desde el programa principal deberá imprimir la lista resultante con la suma de las columnas de la matriz.



- Ejemplo de ejecución:

Por favor ingrese la dimensiones de la matriz: n x m: 2 3

Ahora por favor ingrese los 6 elementos de la matriz: 1 2 3 4 5 6

La matriz ingresada es:

```
1 2 3
4 5 6
```

La suma de las columnas es:

```
5 7 9
```

EJERCICIO FINAL 2

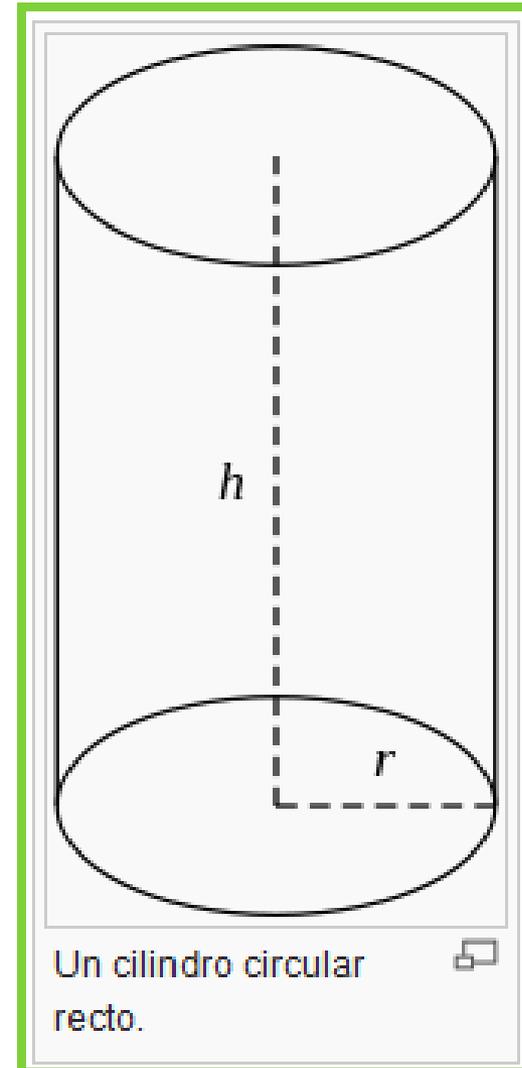
Ing. Ventre, Luis O.

- El objetivo es desarrollar
- Programa que calcule el volumen de un cilindro.

- Recordemos la formula es:

$$V = \pi r^2 \cdot h$$

- Área del circulo por altura!





- Realice un programa que a través de una función llamada **ingreso_datos()**, solicite al usuario el INGRESO del radio del cilindro y de la altura, y lo almacene en variables GLOBALES.
- A través de una función **calcula_volumen(radio,altura)** hacer el calculo del volumen del cilindro, equivalente a AREA por altura. Para esto debe hacer uso de otra funcion que devuelva el AREA.



EJERCICIO FINAL 2

Ing. Ventre, Luis O.

- Crear una función **calcula_area(radio)** para hacer el calculo del área del circulo, equivalente a $PI * \text{radio al cuadrado}$. Esta función debe devolver el valor del área.
- Desde el programa principal imprimir el valor del área y del volumen.



Lo visto!

Ing. Ventre, Luis O.

Repasando!...

- **Una función puede llamar a otra?**
- **Una función puede llamarse a SI MISMA?**
 - **RECURSIVIDAD DIRECTA**
- **Una función, puede llamar a otra y esta a la primera?**
 - **RECURSIVIDAD INDIRECTA**