

IIC1103

Clase 5: Funciones recursivas.

Nicolás Alvarado - Francisca Cattán

Universidad Católica de Chile

8 de enero, 2024

En la clase pasada...

- ▶ Aprendimos a hacer importar módulos usando la instrucción `import`. En particular, se usaron los módulos `math`, `random`.
- ▶ Aprendimos a escribir funciones propias con la siguiente estructura:

```
1 def nombre_funcion(x1, x2, ...):  
2     #funcionamiento  
3     return output
```

Funciones recursivas

Qué es para ustedes la recursión?

Funciones recursivas

Qué es para ustedes la recursión?

La **recursión** es la estrategia de implementación de funciones en la cual se usa la misma definición de la función, como parte del funcionamiento de la misma.

Funciones recursivas

Qué es para ustedes la recursión?

La **recursión** es la estrategia de implementación de funciones en la cual se usa la misma definición de la función, como parte del funcionamiento de la misma.

```
1 def fun(n):  
2     if n==0 or n==1:  
3         return 1  
4     else:  
5         return n*fun(n-1)
```

Funciones recursivas

Importante!!!!

- ▶ Las llamadas a función se repiten.
- ▶ La recursión **debe** terminar en algún momento.

WHEN THE RETURN CONDITION IS WRONG



Funciones recursivas

Probemos el ejemplo anterior en colab sin las líneas 2 y 3.

Funciones recursivas

Un par de observaciones:

- ▶ Las funciones recursivas deben tener casos base.
- ▶ Una función recursiva funcionará bien en cuanto las llamadas recursivas lleguen a casos base.



Funciones recursivas

En general tenemos la siguiente estructura:

```
1 def recursion():  
2     ...  
3     recursion() #llamada recursiva  
4     ...
```

Funciones recursivas

Veamos algunos ejemplos:

```
1 def fun(n):  
2     if n!=0:  
3         return n + fun(n-1)  
4     else:  
5         return 0
```

Funciones recursivas

Veamos algunos ejemplos:

```
1 def fun(n):  
2     if n!=0:  
3         return n + fun(n-1)  
4     else:  
5         return 0
```

```
1 def fun(n):  
2     if n == 0:  
3         return 0  
4     elif n == 1:  
5         return 1  
6     else:  
7         return fun(n-1)+ fun(n-2)  
8 for i in range(10):  
9     print(fun(i))
```

Funciones recursivas

Otro ejemplo:

```
1 def fun(n, x):
2     if x == 0:
3         imprimir = "*_" * (n-1) + "*"
4         print(imprimir)
5         fun(n, x + 1)
6     elif x == n:
7         return
8     elif n > x:
9         imprimir = "_"*(n-(n - x))+"*_" * ((n - x)-1) + "*"
10        print(imprimir)
11        fun(n, x + 1)
12 a = int(input())
13 fun(a, 0)
```

Funciones recursivas

Resolvamos algunos ejercicios:

- ▶ Escriba una función recursiva que reste de 3 unidades a un entero positivo y llegue a 0.
- ▶ Escriba una función recursiva que calcule la potencia positiva de un número.
- ▶ Escriba una función recursiva que encuentre el máximo común divisor entre dos números.
- ▶ Escriba una función recursiva que calcule la suma armónica.

Funciones recursivas

Sigamos con ejemplos:

```
1 def fun1(n, k):
2     if k == 0 or k == n:
3         return 1
4     else:
5         return fun1(n-1, k-1) + fun1(n-1, k)
6 def fun2(n):
7     a = []
8     for i in range(n):
9         b = []
10        for j in range(i + 1):
11            b.append(fun1(i, j))
12        a.append(b)
13    return a
```

Funciones recursivas

Sigamos con ejemplos:

```
1 def fun1(x, y=None):
2     if y is None:
3         y = x - 1
4     if x < 2:
5         return False
6     elif y == 1:
7         return True
8     elif x % y == 0:
9         return False
10    else:
11        return fun1(x, y - 1)
12 def fun2(inicio, fin):
13     if inicio <= fin:
14         if fun1(inicio):
15             print(inicio)
16         fun2(inicio + 1, fin)
```

Funciones recursivas

Resolvamos algunos ejercicios:

- ▶ Escriba una función que reciba un intervalo de números y los imprima de mayor a menor. Su función debe ser recursiva.
- ▶ Escriba una función recursiva que encuentre el minimo común multiplo entre dos números.
- ▶ Escriba una función recursiva que reciba como input dos strings. La funcion debe retornar la concatenación del primer string con el segundo, repetidas veces hasta superar 7 veces el largo del primer string.