

Programación de Computadores: Listas, Ejecución Iterativa y Librerías en Python

Juan F. Pérez

Departamento MACC
Matemáticas Aplicadas y Ciencias de la Computación
Universidad del Rosario

juanferna.perez@urosario.edu.co

Segundo Semestre de 2017

Contenidos

- 1 Listas
- 2 Ejecución Iterativa en Python
- 3 Ejecución Iterativa: Número de iteraciones
- 4 Iteración condicional

Listas

Listas en Python

Un tipo de variable para almacenar varios valores, no solo uno
Podemos acceder a la lista recorriendo todos los elementos uno por uno
Podemos acceder a elementos específicos de la lista

Listas en Python

En un script de Python:

```
# -*- coding: utf-8 -*-  
lista_de_numeros = [1, 2, 3, 4, 5]  
print lista_de_numeros  
lista_de_strings = ['hola', 'esta', 'es', 'una', 'lista']  
print lista_de_strings  
print lista_de_numeros[1]  
print lista_de_numeros[3]  
print lista_de_strings[0]  
print lista_de_strings[2]  
print lista_de_strings[4]
```

Listas en Python: range()

```
# -*- coding: utf-8 -*-  
otra_lista = range(5)  
print otra_lista  
otra_lista_mas = range(2,5)  
print otra_lista_mas
```

Listas en Python: append

```
# -*- coding: utf-8 -*-  
mi_lista = range(5)  
print mi_lista  
mi_lista.append(10)  
print mi_lista  
mi_lista.append(50)  
print mi_lista
```

Ejecución Iterativa en Python

Recorriendo listas (para una sola lista)

```
# -*- coding: utf-8 -*-  
frutas_disp = ['fresa', 'uva', 'naranja', 'banano']  
for i in frutas_disp:  
    print u"La fruta %s está disponible" % i
```

Recorriendo listas (con contador)

```
# -*- coding: utf-8 -*-  
frutas_disp = ['fresa', 'uva', 'naranja', 'banano']  
frutas_cant = [40, 56, 25, 102]  
for i in range(4):  
    print "i=%d" % i  
    print u"La fruta %s está disponible" \  
        % frutas_disp[i]  
    print u"Hay %d frutas disponibles de \  
este tipo" % frutas_cant[i]
```

Utilizando listas

```
# -*- coding: utf-8 -*-  
frutas_inventario = [40, 56, 25, 102]  
frutas_pedido = [100, 200, 350, 500]  
frutas_nuevo_inventario = []  
for i in range(4):  
    nuevo_inventario_temp = \  
        frutas_inventario[i] + \  
frutas_pedido[i]  
    frutas_nuevo_inventario.append(\  
nuevo_inventario_temp)  
print u"El nuevo inventario es", \  
frutas_nuevo_inventario
```

Ejecución Iterativa: Número de iteraciones

Ejecución iterativa - Número de iteraciones

```
# -*- coding: utf-8 -*-  
for i in range(10):  
    print " %f:  %f+ %f = %f" \  
        % (i, 2**i, 3**i, 2**i + 3**i)
```

Ejecución iterativa - Número de iteraciones

```
# -*- coding: utf-8 -*-
def fact(numero):
    if isinstance( numero, int ) and numero > 0:
        result = 1
        for i in range(1,numero):
            result = result*i
        print "i: %d \tfact(i): %d" %(i,result)

mi_num = 5
fact_mi_num = fact(mi_num)

mi_num = 15
fact_mi_nsum = fact(mi_num)
```

Ejecución iterativa - Número de iteraciones

La sucesión de Fibonacci empieza con los números 0 y 1, y los demás términos son iguales a la suma de los dos anteriores. Escriba un programa que calcule e imprima los primeros n términos de la sucesión de Fibonacci. Pruebe su programa con $n = 2$, $n = 5$ y $n = 20$.

Iteración condicional

Iteración condicional

```
# -*- coding: utf-8 -*-  
remanente = 10  
print "Remanente inicial: %d" % remanente  
while remanente > 0:  
    remanente = remanente - 1  
    print "Remanente actual: %d" % remanente
```

Iteración condicional

```
# -*- coding: utf-8 -*-  
limite = 10  
i = 1  
while i < limite:  
    j = 1  
    while j < 10:  
        suma_cuadrados = i**2 + j**2  
        print " %d^2 + %d^2: %d" \  
            %(i, j, suma_cuadrados)  
        j = j + 1  
        suma_cuadrados = i**2 + j**2  
    i = i + 1
```

Iteración condicional

```
# -*- coding: utf-8 -*-
```

```
def fact(numero):
    if isinstance( numero, int ) and numero > 0:
        result = 1
        for i in range(1,numero):
            result = result*i
        return result

limite = 1E6
print u"Todos los números naturales n tales que \
factorial de n es menor que %d " %limite
i = 1
res = fact(i)
while res < limite:
    print u"Factorial de %d: %d" %(i,res)
    i = i + 1
    res = fact(i)
```