

Herramientas Computacionales

2016661

Introducción a python

Ricardo Amézquita

Departamento de Física
Universidad Nacional de Colombia
Sede Bogotá


- 1 El interprete de python
- 2 Python básico
- 3 Condicional
- 4 Ciclo mientras
- 5 Introducción a las funciones
- 6 Funciones recursivas
- 7 Retornando múltiples datos desde una función
- 8 Parámetros pasados a una función por defecto
- 9 Funciones lambda
- 10 Listas
- 11 Tuplas

El interprete de python

Usando el interprete como una calculadora

Operadores aritméticos

Operador	Descripción	Ejemplo	Resultado
+	Suma	3+4	7
-	Resta	5-3	2
*	Multiplicación	4*7	28
/	División	15/4 15./4.	3 ¹ 3.75
%	Residuo	5 %2	1
**	Exponente	2**3	8
//	División Entera	15/4 15./4.	3 3.

¹3.75 si se usa `from __future__ import division` 

Usando el interprete como una calculadora

Operadores de comparación

Operador	Descripción	Ejemplo	Resultado
==	Igual a	3==5	False
!=	Diferente a	3!=5	True
<>	Diferente a	3<>5	True
>	Mayor que	3>5	False
<	Menor que	3<5	True
>=	Mayor igual que	3>=3	True
<=	Menor igual que	5<=5	True

Usando el interprete como una calculadora

Operadores lógicos

Operador	Descripción	Ejemplo	Resultado
and	Y	$(3 \neq 5) \text{ and } (7 < 10)$	True
or	O	$(3 == 5) \text{ or } (7 > 10)$	False
not	Negado	$\text{not}((3 \neq 5) \text{ and } (7 < 10))$	True

Usando el interprete como una calculadora

Operadores de bits

Operador	Descripción	Ejemplo	Resultado
&	Y	98&51 (0110 0010)&(0011 0011)	34 (0010 0010)
	O	98 51 (0110 0010) (0011 0011)	115 (0111 0011)
^	O exclusivo	98^51 (0110 0010)^(0011 0011)	81 (01010001)
<<	Desp. a la izq.	3<<2 (0000 0011)<<(0000 0010)	12 (0000 1100)
>>	Desp. a la der.	156>>2 (1001 1100)	39 (00100111)

Usando el interprete como una calculadora

Operadores de asignación

Operador	Ejemplo	Equivalente a
=	a=10	
+=	a+=3	a=a+3
-=	a-=4	a=a-4
=	a=8	a=a*8
/=	a/=4	a=a/4
%=	a%=3	a=a%3
=	a=8	a=a**8
//=	a//=3	a=a//3

Usando el interprete como una calculadora

Otros operadores

Operador	Resultado
<code>abs(x)</code>	Valor absoluto de x
<code>int(x)</code>	x convertido a entero
<code>long(x)</code>	x convertido a entero largo
<code>float(x)</code>	x convertido a flotante
<code>complex(re, im)</code>	numero complejo $re + i im$
<code>divmod(x,y)</code>	La pareja $(x//y, x \% y)$

Variables y tipos de datos numéricos

- boolean
- int
- long
- float
- complex

modulo math

`math.acos``math.acosh``math.asin``math.asinh``math.atan``math.atan2``math.atanh``math.ceil``math.copysign``math.cos``math.cosh``math.degrees``math.e``math.erf``math.erfc``math.exp``math.expm1``math.fabs``math.factorial``math.floor``math.fmod``math.frexp``math.fsum``math.gamma``math.hypot``math.isinf``math.isnan``math.ldexp``math.lgamma``math.log``math.log10``math.log1p``math.modf``math.pi``math.pow``math.radians``math.sin``math.sinh``math.sqrt``math.tan``math.tanh``math.trunc`

Python básico

Primer programa en python

```
#!/usr/bin/python
# -*- coding: utf8 -*-
print 'Este es un primer programa en python'
a=10
b=20.
# Este es un comentario
print 'Este es un número entero ',a
print 'Este es un número flotante ',b
a=complex(10,3)
print 'Este es un número complejo ',a
```

Imprimiendo resultados a hacia la pantalla

```
#!/usr/bin/python
# -*- coding: utf8 -*-
q=305
j=8
k=20
print 'q, j, k son',q,j,k
print 'q es {}, j es {}, y k es {}'.format(q,j,k)
print 'j={1}, q={0}, y k={2}'.format(q,j,k)
```

Imprimiendo resultados a hacia la pantalla

```
#!/usr/bin/python
# -*- coding: utf8 -*-
a=9
print "{:1d}".format(a)
print "{:01d}".format(a)
print "{:2d}".format(a)
print "{:02d}".format(a)
print "{:3d}".format(a)
print "{:03d}".format(a)
print "{:4d}".format(a)
print "{:04d}".format(a)
print "{:5d}".format(a)
print "{:05d}".format(a)
```

Imprimiendo resultados a hacia la pantalla

```
#!/usr/bin/python
# -*- coding: utf8 -*-
import math
print "{:5.3f}".format(math.pi)
print "{:6.3f}".format(math.pi)
print "{:7.3f}".format(math.pi)
print "{:8.3f}".format(math.pi)
print "{:9.3f}".format(math.pi)
```


Recibiendo información desde el teclado

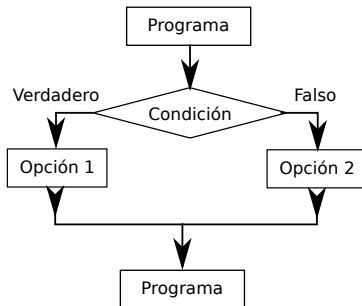
```
#!/usr/bin/python
# -*- coding: utf8 -*-
a,b,c=input("teclea 3 números ")
print a,b,c
w=raw_input("Teclea una palabra con mas de 5 letras")
print w
print w[2:5]
print w[3]
print w[4:1:-1]
print w[::-1]
print w[:2]
```

Ejercicios

- 1 Haga un programa que pregunte un numero e imprima su seno y coseno con 5 cifras significativas.
- 2 Haga un programa que pregunte la parte real e imaginaria de un numero complejo, e imprima el cuadrado de su modulo.
- 3 Haga un programa que pregunte 5 números e imprima una tabla con estos valores y sus cuadrados.
- 4 Usando `raw_input`, haga un programa que pregunte un numero e imprima su raíz cuadrada
- 5 Haga un programa que pregunte una palabra e imprima ésta al revés
- 6 Usando una consola de `ipython`, cree una variable que contenga una cadena de caracteres y busque en su documentación como pasar esta cadena de caracteres a letras mayúsculas
- 7 Haga un programa que pregunte una palabra y que la imprima en mayúsculas
- 8 Haga un programa que pregunte un numero binario de 4 dígitos, e imprima su valor en decimal

Condicional

¿Que es un condicional?



Condicionales en Python

Ejemplo 1

```
print 'antes del if'
a=3
if a==3:
    print 'en el if'
print 'despues del if'
```

Ejemplo 2

```
print 'antes del if'
a=3
if a==3:
    print 'en el if'
else:
    print 'en el else'
print 'despues del if-else'
```

Condicionales en Python

Ejemplo 3

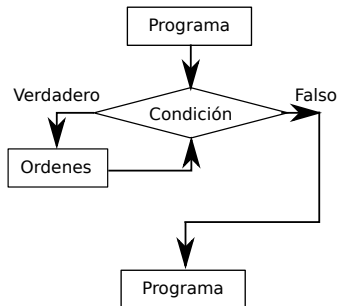
```
print 'antes del if'
a=4
if a==3:
    print 'en el if'
elif a==4:
    print 'en el elif'
else:
    print 'en el else'
print 'despues del if-elif-else'
```

Ejercicios

- Escriba un programa que pregunte un numero e indique si este es divisible por 2,3,5
- Escriba un programa que pregunte 3 números cualquiera y los imprima en orden ascendente y en orden descendente

Ciclo mientras

Ciclo mientras en python



Ciclo mientras en python

```
a=0
print 'antes del ciclo'
while a<10:
    print a
    a=a+1
print'despues del ciclo'
```

Ejercicios

- Escriba un programa que pregunte un numero entero e imprima los números pares menores a el.
- Escriba un programa que pregunte un numero n y calcule $\sum_{i=1}^n i$.
- Escriba un programa que pregunte un numero n y calcule $n!$.
- Escriba un programa que pregunte un numero n e imprima los primeros n términos de la serie de Fibonacci.
- Escriba un programa que pregunte un numero e indique si este es primo.

Introducción a las funciones

Función sencilla

```
#!/usr/bin/python
# -*- coding: utf8 -*-
a=10 # Definicion de variable global
b=0
def f(x,y):
    a=x+y #Definicion de variable local
    global b
    b=10
    return a
print f(3,4)
print a
print b
```

Ejercicios

- Escriba una función que calcule el factorial de un numero

Funciones recursivas

Funciones recursivas

```
#!/usr/bin/python
# -*- coding: utf8 -*-
def factorial(x):
    if x==0 or x==1:
        return 1
    return x*factorial(x-1)
print factorial(4)
```

 $4 \times \text{factorial}(3)$
 $3 \times \text{factorial}(2)$
 $2 \times \text{factorial}(1)$

1

Ejercicios

- Escriba un programa que usando una función recursiva que calcule $\sum_{i=1}^n i$, teniendo en cuenta que $\sum_{i=1}^n i = n + \sum_{i=1}^{n-1} i$, y que $\sum_{i=1}^1 i = 1$
- Escriba un programa que usando una función recursiva calcule x^n , teniendo en cuenta que $x^n = x \times x^{n-1}$ y que $x^0 = 1$
- ¿Que pasa en el programa anterior, si n no es un numero entero?

Retornando múltiples datos desde una función

Retornando múltiples datos de una función

```
#!/usr/bin/python
# -*- coding: utf8 -*-
def menor_mayor(x,y):
    if x>y:
        mayor,menor=x,y
    else:
        mayor,menor=y,x

    return menor, mayor

a,b=menor_mayor(6,3)
print a,b
```

Parámetros pasados a una función por defecto

Parámetros de funciones

```
#!/usr/bin/python
# -*- coding: utf8 -*-
def saludo(veces,nombre='SinNombre'):
    for i in range(veces):
        print nombre

    return None
saludo(4)
saludo(3,'yo')
saludo(2,nombre='tu')
```

Funciones lambda

Funciones Lambda

```
#!/usr/bin/python
# -*- coding: utf8 -*-
f= lambda x,y:x+y
g= lambda a:a**2
print f(4,5),g(8)
```

Listas

Creando listas-1

```
a=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
print a  
print a[0]  
print a[-1]  
print a[:5]  
print a[3:4]  
print a[::-1]
```

Creando listas-2

```
b=[1,'dos',3,'cuatro']  
print b  
print b[1]
```

Pregunta:

¿Cual es el resultado de la siguiente instrucción?

```
print b[3][3]
```

Usando listas de listas como una lista multidimensional

```
c=[[4,5],[2,3],[8,9]]  
print c  
print c[0]  
print c[1][1]
```

Operaciones aritméticas entre listas

```
d=10*[1]  
print d
```

```
a=[1,2,3]+[4,5,6]+[7]  
print a
```

Modificando listas

```
a=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
print a
```

Modificando un elemento de la lista

```
a[4]='cinco'  
print a
```

Modificando varios elementos de la lista

```
a[2:5]=['tres','cuatro','cinco']  
print a  
a[::2]=['*', '*', '*', '*', '*']  
print a
```

Modificando listas-1

Adicionando elementos al final de la lista

```
a.append(14)
a.append(15)
print a
```

Extendiendo la lista

```
a.extend([16,17])
print a
```

Eliminando elementos de la lista

```
a[3:5]=[]
print a
```

Copiando listas

Ejemplo 1

```
a=[1,2,3,4]  
b=a  
a[0]=0  
print a  
print b
```

Ejemplo 2

```
a=[1,2,3,4]  
b=a[:]  
a[0]=0  
print a  
print b
```

¿Cual es la diferencia entre estos 2 ejemplos?

Ejercicio

Escriba un programa con una función que reciba una lista y retorne la lista ordenada. La función `len(l)` retorna la cantidad de elementos de la lista `l`.


```
#!/usr/bin/python
#-*- coding: utf8 -*-

a=[1, 4, 6, 8, 9, 11, -3, -4, 6, 10]

for i in range(len(a)):
    for j in range(i+1,len(a)):
        if a[i]>a[j]: a[i],a[j]=a[j],a[i]

print a
```

¿Como hacemos para que este programa ordene la lista en forma descendente?

Solución python

```
#!/usr/bin/python
#-*- coding: utf8 -*-

a=[1, 4, 6, 8, 9, 11, -3, -4, 6, 10]
a.sort()
print a
```

¿Como hacemos para que este programa ordene la lista en forma descendente?

Otros métodos de las listas

```
l=[1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,9]
```

Buscando información en la lista

```
print l.count(4) # Cuantos cuatros hay?  
print l.index(4) # Donde esta el primer 4?
```

Otra forma de agregar información a una lista

```
# Inserte 'cuatro' en el indice 3  
l.insert(3,'cuatro')
```

Otros métodos de las listas

Sacando elementos de una lista

```
# Retorne y remueva el ultimo item de la lista  
print l.pop()
```

```
# Retorne y remueva el n-simo item de la lista  
print l.pop(3)
```

```
#Remueva el primer 4 que se encuentre en la lista  
l.remove(4)
```

Otros métodos de las listas

Formas fáciles de cambiar el orden en una lista

#Invierta la lista

```
l.reverse()
```

#Ordene la lista

```
l.sort()
```

Más funciones y métodos que operan en listas

```
l=[0,3,5,1]
```

```
l1=[0,3,4,1]
```

```
cmp(l,l1) #Retorna 0 si las listas son iguales
```

```
len(l)
```

```
max(l)
```

```
min(l)
```

Otros métodos y funciones que pueden aplicarse a listas

`x in s--> Booleano`

```
s=[1,2,3,4,'c',6,7,'o',9]
```

```
5 in s
```

```
'c' in s
```

```
2 in s
```

`x not in s--> Booleano`

```
s=[1,2,3,4,'c',6,7,'o',9]
```

```
5 not in s
```

```
'c' not in s
```

```
2 not in s
```

`len(s) --> entero`

```
s=[1,2,3,4,'c',6,7,'o',9]  
len(s)
```

`min(s) --> elemento`

```
s=[1,2,3,4,5,6,7,8,9]  
min(s)
```

`max(s) --> elemento`

```
s=[1,2,3,4,5,6,7,8,9]  
max(s)
```


`s.index(x)` —> Entero

```
s=[3,2,56,3,7,8,9]
```

```
s.index(56)
```

`s.count(x)` —> Entero

```
s=[1,2,3,4,3,5,3,6,7,8,7,5,4]
```

```
s.count(3)
```

`s.sort([reverse=False])`

```
s=[1,2,4,5,6,8,4,32,5,7,8]  
s.sort() ; print s  
s.sort(reverse=True) ; print s
```

`s.reverse()`

```
s=[1,2,4,5,6,8]  
s.reverse() ; print s
```

`sorted(s,[reverse=False])` —> Lista

```
s=[1,2,4,5,6,8,4,32,5,7,8]  
y=sorted(s) ; print y  
z=sorted(s,reverse=True) ; print z
```

`range([inicio,] final [, paso])` —> lista

```
a= range(10) ; print a
b= range(5,100,2) ; print b
c= range(100,2,-3)
```

`enumerate(s [, inicio])` —> lista de tuplas

```
s=['a', 'b', 'c', 'd', 'e']
print enumerate(s)

for n,l in enumerate(s):
    print n, l
```

`filter(función, secuencia) —> secuencia`

```
f= lambda x: x<10  
s=[1,2,3,4,5,6,7,8,9,0,12,13,14,15,16,17]  
print filter(f,s)
```

`map(función, secuencia) —> secuencia`

```
f= lambda x: x**2  
s=range(10)  
print map(f,s)
```

`reduce(función, secuencia) —> Valor`

```
f= lambda x,y: x*y  
s=range(1,4)  
print reduce(f,s)
```

```
g= lambda x,y: x+y  
s=range(100)  
print reduce(g,s)
```

`sum(secuencia) —> valor`

```
s= range(100)  
print sum(s)
```

Los ciclos for y las listas

```
l=[0,3,5,1,'seis','siete']  
for i in l:  
    print i
```

Función range

```
range([start,] stop[, step])
```

Retorna una lista de enteros que inician en **start**, finalizando antes de incluir **stop**, con paso **step**.

Por defecto *start* = 0 y *step* = 1

Tuplas

Creando tuplas

```
a=(1, 2, 3)
b= 34,56,"ww"
print a
print a[2]
```

Diferencia entre listas y tuplas

- Las tuplas son listas que no se pueden modificar después de creadas
- Los métodos normales de las listas no funcionan en ellas

Similitudes entre listas y tuplas

- El indexado en las tuplas funciona igual que en las listas
- Las funciones que trabajan en listas, también trabajan en tuplas

Tarea

Buscar que son los diccionarios (dictionaries) y los conjuntos (sets) en python.

Escriba un programa que calcule de manera aproximada el valor de:

$$y = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!}$$

Problema 1

Calculando el factorial:

Solución 1:

```
def factorial(n):  
    if (n==0) or (n==1): return 1  
    f=1  
    for k in range(1,n+1)  
        f=f*k  
    return f
```

Solución 2:

```
def factorial(n):  
    if (n==0) or (n==1): return 1  
    return n*factorial(n-1)
```

Problema 1

Calculando el factorial:

Solución 3:

```
def factorial(n):  
    a=range(1,n+1)  
    f=lambda x,y:x*y  
    return reduce(f,a)
```

Solución 4:

```
from math import factorial
```

Problema 1

Calculando la sumatoria

Solución 1:

```
def sumatoria(x):  
    s=0  
    for n in range(20):  
        s=s+x**(2*n+1)/factorial(2*n+1)  
    return s
```

Solución 2:

```
def sumatoria(x):  
    n=range(20)  
    f=lambda m: x**(2*m+1)/factorial(2*m+1)  
    x1=map(f,n)  
    return sum(x1)
```

Ejercicios

- 1 Encuentre el 10001avo número primo
- 2 Calcule la suma de los números primos menores a 2000000.

Sigue.....

Ejercicios

- ❶ La siguiente secuencia iterativa está definida para el conjunto de los enteros positivos:

$$n \rightarrow n/2 \text{ si } n \text{ es par}$$

$$n \rightarrow 3n + 1 \text{ si } n \text{ es impar}$$

Usando la regla anterior e iniciando con 13 se genera la siguiente secuencia:

$$13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

Esta cadena (que inicia en 13 y termina en 1) contiene 10 elementos. ¿Cual numero menor a un millón produce la secuencia mas larga?

Nota:

- ❶ Una vez la cadena se inicia los términos pueden ser mayores a un millón.
- ❷ Se supone que sin importar en que numero inician, todas las secuencias terminan en 1.