

# Curso de Python



# Ejercicio

- Se requiere un programa al cual se le ingresen los coeficientes a,b,c y genere la solución de las raíces.

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

# Funciones



- ▶ Es una estructura de control.
- ▶ Bloque de código reutilizable que se encarga de realizar una determinada tarea.

# Estructura de una Función

```
def mostrar():  
    print "Hola mundo"  
  
mostrar()
```

# Argumentos en las funciones

```
def arbitrarios(parametro_fijo, *args, **kwargs):  
    print parametro_fijo  
    for i in args:  
        print i  
  
    *** parametros arbitraratios, tipo diccionarios  
    for clave in kwargs:  
        print "El valor de", clave, "es", kwargs[clave]  
  
arbitrarios("Hola", "args 1", "args 2", "args 3"  
, clave1="valor uno", clave2="valor dos")
```

- ▶ \*args: La función espera recibir mas parámetros, los almacena en una tupla.
- ▶ \*\*kwargs: La función espera recibir pares de parámetros clave = valor

# Recursividad

Es la forma en la cual se especifica un proceso basado en su propia definición. Siendo un poco más precisos, y para evitar el aparente círculo sin fin en esta definición:



Mostrar texto una cantidad  
repetida de veces

**HOLA  
MUNDO!**

**HOLA  
MUNDO!**

**HOLA  
MUNDO!**

# Hallar el factorial de un número

$$1! = 1$$

$$2! = 2(1) = 2$$

$$3! = 3(2)(1) = 6$$

$$4! = 4(3)(2)(1) = 24$$

$$5! = 5(4)(3)(2)(1) = 120$$



# Retos:

- Serie de Fibonacci: Resolver la serie para una cantidad determinada de valores. La cantidad valores deberán ser ingresados por el usuario. Se debe mostrar la serie.

0	1	1	2	3	5	8	13	21
34	55	89	144	233	377			
610	987	1597	2584	....				

# Módulos

- ▶ Son archivos de código con extensión .py, y ubicados en un directorio donde Python sea capaz de encontrarlos .
- ▶ Los módulos contienen normalmente métodos o algoritmos de programación útiles o necesarios para mejorar nuestro código actual.

# Llamada de un modulo

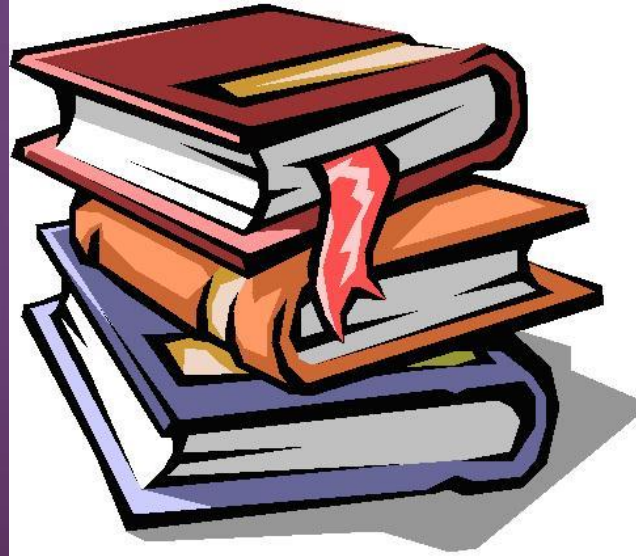
```
import modulo  
  
modulo.(nombre de la funcion)(parametros)
```

```
import modulo as md  
  
md.(nombre de la funcion)(parametros)
```

```
from modulo import (nombre de la funcion)  
  
(nombre de la funcion)(parametros)
```

# Diccionarios

- Un diccionario es como una contenedor de llaves, las cuales tienen un valor asociado. Las llaves no están ordenadas y no hay llaves repetidas. La única manera de acceder a un valor es a través de su llave.



# Veamos

- Estructura de un diccionario

```
usuario={"Luis":1234,"Carlos": 4321}
```

Llave

Valor