# Actividad 2: Elementos de la Programación Python 1

Valenzuela Carrillo María Inés

29 de Enero del 2016

## 1. Introducción

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Python es un lenguaje de programación multiparadigma. Lo cual significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programableM.

En esta actividad se muestra la ejecución de varios programas en Python con ejercicios básicos.

## 2. Problema 1

Para este problema se deja caer una pelota desde el techo de una torre de altura h. A partir de lo anterior se desea saber la altura de la pelota respecto a la torre a un determinado tiempo después de haber sido dejada caer.

El código para conocer la altura de la pelota es:

```
h = float(input("Proporciona la altura de la torre: "))
t = float(input("Ingresa el tiempo: "))
s = 0.5*9.81*t**2
print("La altura de la pelota es", h-s, "metros")
```

Se pide modificar el código anterior, para hacer un programa que solicite la altura de la torre en metros, y que imprima el tiempo que tarda la pelota en llegar al suelo.

Haciendo ciertas modificaciones el resultado para el código fue:

```
import math
h = float(input("Proporciona la altura de la torre: "))
g = 9.8
t = sqrt((2*h)/g)
print("El tiempo que tarda en llegar al suelo es", t , "segundos")
```

Resultado al correrlo con una altura de 10 metros es:

```
Proporciona la altura de la torre: 10
('El tiempo que tarda en llegar al suelo es', 1.4285714285714286, 'segundos')
```

# 3. Problema 2

Este problema trata sobre un satélite orbitando la tierra a una altura h con un periodo T. Se pide hacer un programa que pida al usuario el valor de T y le regrese la altura a la que debería estar el satélite para tener dicho periodo. Para poder realizar el programa se utilizó la expresión:

$$(R+h)^3 = \frac{GMT^2}{4\pi^2}$$

donde $G = 6.67x10^{-11}m^3kg^{-1}s^{-2}$ es la constante de Gravitación Universal de Newton, $M = 5.97x10^{24}kg$ es la masa de la Tierra y $R = 6371km$ es su radio.

El código el programa realizado es:

```
from math import pi
G=6.67e-11
M=5.97e24
R=6371e3
T = float(input("ingrese el periodo del satélite:"))
h=((G*M*T**2/(4*pi**2))**(1./3))-R
print("La altitud del satelite es de", h , "metros")
```

Resultado para un periodo de un día:

```
ingrese el periodo del satélite:86400
('La altitud del satelite es de', 35855910.176174976, 'metros')
```

Resultado para un periodo de 90 minutos:

```
ingrese el periodo del satélite:5400
('La altitud del satelite es de', 279321.6253728606, 'metros')
```

Resultado para un periodo de 45 minutos:

```
ingrese el periodo del satélite:2700
('La altitud del satelite es de', -2181559.8978108233, 'metros')
```

El último resultado da negativo, lo cual indica que el satéite no puede tener dicho periodo.

# 4. Problema 3

Un punto en dos dimensiones puede ser descrito de diversas maneras entre las cueles se encuentran las coordenadas polares (r,$\theta$) y las cartesianas (x,y), entre estas diferentes maneras de describir un punto existe cierta relación dada por: $x = r\cos\theta$, $y = r\sin\theta$.

El siguiente programa calcula las coordenadas cartesianas a partir de las coordenadas polares.

```
from math import sin,cos,pi
r = float(input("Introduce r: "))
d = float(input("Ingresa theta en grados: "))
theta = d*pi/180
x = r*cos(theta)
y = r*sin(theta)
print("x =",x," y =",y)
```

Se pide realizar un programa similar para calcular las coordenadas esféricas (r,$\theta$, $\phi$) a partir de las coordenadas cartesianas (x,y,z).

El código del programa es:

```
from math import pi,sqrt,atan,acos
x = float(input("Introduce x: "))
y = float(input("Introduce y: "))
z = float(input("Introduce z: "))
r = sqrt((x**2)+(y**2)+(z**2))
theta= atan(y/x)
phi=acos(z/r)
p=phi*180/pi
d=theta*180/pi
print("r =",r," theta =", d, "phi=",p )
```

El resultado al correr el programa es:

```
Introduce x: 3

Introduce y: 3

Introduce z: 3
('r =', 5.196152422706632, ' theta =', 45.0, 'phi=', 54.735610317245346)
```

# 5. Problema 4

Se sabe que los números pares son divisibles entre 2, es decir que su residuo es cero (2n), mientras que los impares tienen residuo 1 (2n+1). Nos apoyamos en la operación de módulo % que indica el residuo en una división.

El siguiente código utiliza *if* y *else*, se ingresa un número y se saca el residuo de el dividido entre dos. Si el residuo es 0 se imprime .evenz si este es 1 .odd".

```
n = int(input("Enter an integer: "))
if n\%2==0:
    print("even")
else:
    print("odd")
```

El código que acontinuación se muestra admite 2 enteros, pide ingresar uno par y otro impar, en este caso se utiliza el *while* para poner una condición, mientras los números dados no sean como se piden el programa seguirá imprimiendo que se ingrese uno para y uno impar, al momento de romperse la condición, es decir, se ingresen como se indica el programa termina.

```
print("Enter two integers, one even, one odd.")
m = int(input("Enter the first integer: "))
n = int(input("Enter the second integer: "))
while (m+n)%2==0:
    print("One must be even and the other odd.")
    m = int(input("Enter the first integer: "))
    n = int(input("Enter the second integer: "))
print("The numbers you chose are",m,"and",n)
```

# 6. Problema 5

Los Números de Fibonacci es una sucesión de números enteros aparecen en toda la naturaleza.

El siguiente programa calcula la secuencia de Fibonacci, mediante la condición *while*.

```
f1,f2 = 1,1
while f2<1000:
      print(f2)
      f1,f2 = f2,f1+f2
```

Se pide crear un programa basado en el código anterior que imprima la secuencia de los números de Catalan, que son dados por la fórmula de recurrencia:

$$C_0 = 1, \quad C_{(n+1)} = \frac{2(2n+1)}{(n+2)} C_n$$

El código para los números de catalan es:

```
n,C1,C2 = 0,1,1
while(C2 < 1000000):
    print(C2)
    C2= C1*(4*n+2)/(n+2)
    n=n+1
    C1=C2
```

El correrlo arroja el siguiente resultado:

```
In [9]: run "catalan.py"
1
1
2
5
14
42
132
429
1430
4862
16796
58786
208012
742900
```

# Referencias

[1] Wikipedia, the free encyclopedia, *Python*.