

Actividad #2 (Elementos de la programación en Python 1)

Instructor: Carlos Lizárraga Celaya

Student: Antonio Cota Rodríguez

Introducción, ¿Qué es Python ?

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Programas

Caída de una pelota

El primer programa que se realizó en esta práctica fue de el *caida.py* el código es el siguiente :

```
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")
```

Lo único que se le modificó a este programa fue que se solicitará explícitamente al usuario la altura en metros, a continuación se muestra un ejemplo que se desarrolló en **IPython Notebook**.



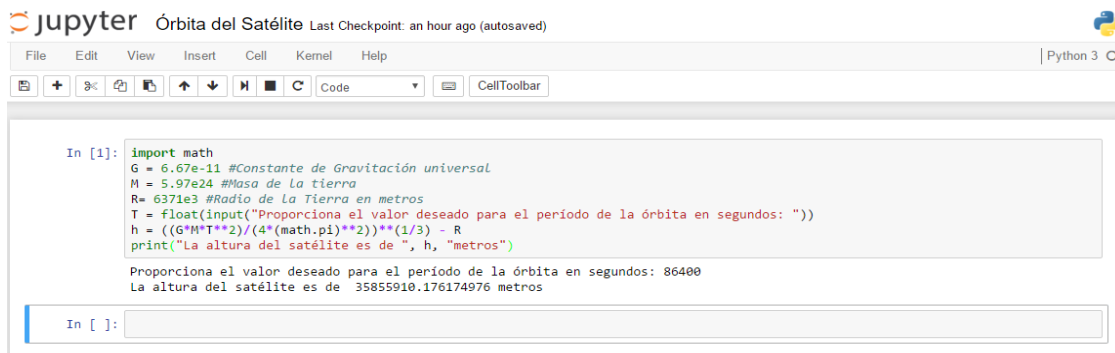
FIGURE 1 – Caída.py

Altura de un satélite

El siguiente código es para calcular la altura de un satélite en órbita alrededor de la tierra en función del período (en segundos) dado por el usuario :

```
import math
G = 6.67e-11 #Constante de Gravitación universal
M = 5.97e24 #Masa de la Tierra en kilogramos
R = 6371e3 #Radio de la Tierra en metros
T = float(input("Proporciona el valor deseado para el período de la órbita en segundos: "))
h = ((G*M*T**2)/(4*(math.pi)**2))**(1/3) - R
print("La altura del satélite es de ", h, "metros")
```

Los siguiente ejemplos que se hicieron fue para un período de un día y otro de 90 minutos.



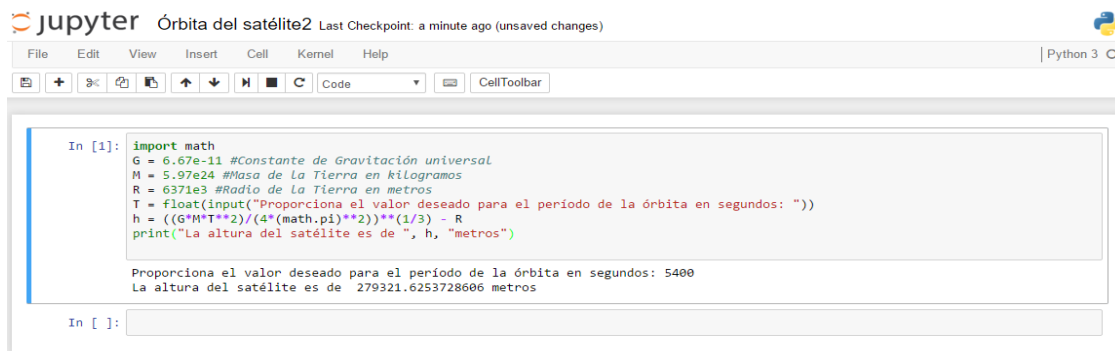
```

In [1]: import math
        G = 6.67e-11 #Constante de Gravitación universal
        M = 5.97e24 #Masa de La Tierra en kilogramos
        R = 6371e3 #Radio de La Tierra en metros
        T = float(input("Proporciona el valor deseado para el periodo de la órbita en segundos: "))
        h = ((G*M*T**2)/(4*(math.pi**2))**(1/3) - R
        print("La altura del satélite es de ", h, "metros")

Proporciona el valor deseado para el periodo de la órbita en segundos: 86400
La altura del satélite es de  35855910.176174976 metros

```

FIGURE 2 – Período de 1 día



```

In [1]: import math
        G = 6.67e-11 #Constante de Gravitación universal
        M = 5.97e24 #Masa de La Tierra en kilogramos
        R = 6371e3 #Radio de La Tierra en metros
        T = float(input("Proporciona el valor deseado para el periodo de la órbita en segundos: "))
        h = ((G*M*T**2)/(4*(math.pi**2))**(1/3) - R
        print("La altura del satélite es de ", h, "metros")

Proporciona el valor deseado para el periodo de la órbita en segundos: 5400
La altura del satélite es de  279321.6253728606 metros

```

FIGURE 3 – Período de 90 minutos

Transformación de coordenadas

El siguiente código muestra una forma de realizar una transformación de coordenadas (de cartesianas a esféricas) $(x, y, z) \rightarrow (r, \theta, \phi)$:

```

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

```

No se presentó problema alguno, la siguiente imagen es un ejemplo con valores dados por el usuario :



```

Jupyter Untitled1 Last Checkpoint: 6 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Help Python 3
In [1]: 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)

Introduce r: 5
Ingresa theta en grados: 40
x = 3.83022221559489 y = 3.2139380484326963

In [3]: from math import sin,cos,pi
r = float(input("Introduce r: "))
d = float(input("Ingresa theta en grados: "))
p = float(input("Ingresa phi en grados: "))
theta = d*pi/180
phi = p*pi/180
x = r*sin(theta)*cos(phi)
y = r*sin(theta)*sin(phi)
z = r*cos(theta)
print("x =",x," y =",y," z=",z)

Introduce r: 10
Ingresa theta en grados: 35
Ingresa phi en grados: 20
x = 4.698463103929542 y = 3.2898992833716556 z= 8.191520442889917

```

FIGURE 4 – Ejemplo de una transformación de coordenadas

Par o impar

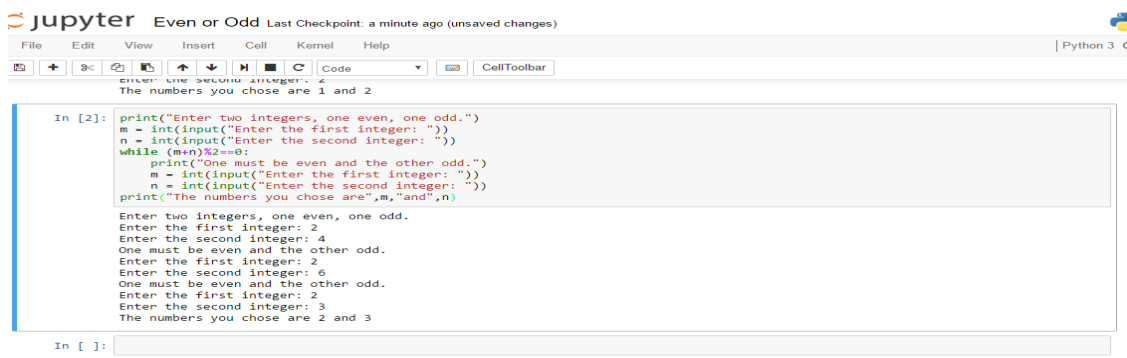
En esta parte se creó un programa en el cual se debía introducir dos valores cualesquiera enteros, el código es el siguiente :

```

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)

```

Cuando el usuario introducía dos valores enteros el programa imprimía una orden donde se le pedía que se debían introducir un entero *par* y otro *impar* sólo así el programa imprimía los valores que se introdujeron, esto debido a que en el código escrito se especifica que el residuo debe ser diferente de cero, se muestra un ejemplo a continuación :



```

Jupyter Even or Odd Last Checkpoint: a minute ago (unsaved changes)
File Edit View Insert Cell Kernel Help Python 3
In [2]: 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)

Enter two integers, one even, one odd.
Enter the first integer: 2
Enter the second integer: 4
One must be even and the other odd.
Enter the first integer: 2
Enter the second integer: 6
One must be even and the other odd.
Enter the first integer: 2
Enter the second integer: 3
The numbers you chose are 2 and 3

```

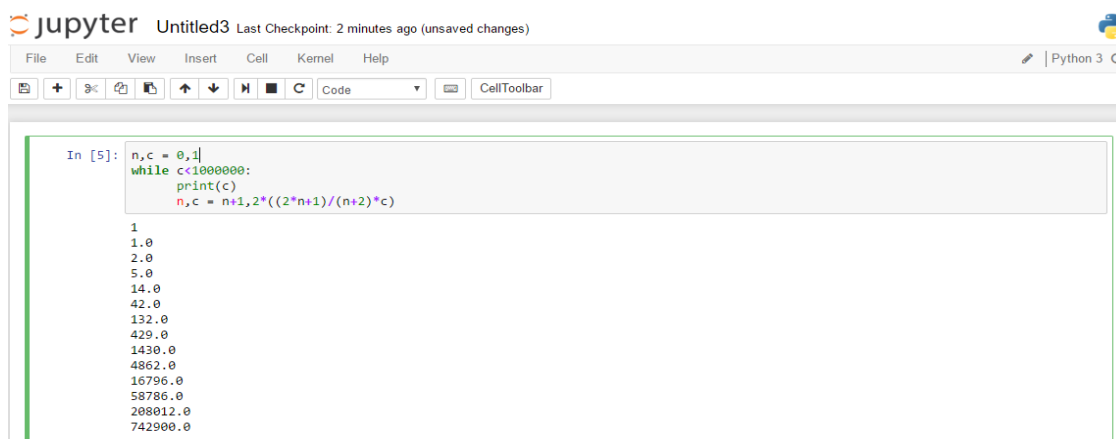
FIGURE 5 – Par o impar

Números de Catalán

De la misma forma que se hizo el programa de la serie de Fibonacci, esta vez se creó un programa que imprime los números de Catalán menores de un millón, el código es el siguiente

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

El resultado fue el siguiente



The screenshot shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar. The code cell contains the following Python code:

```
In [5]: n,c = 0,1
while c<1000000:
    print(c)
    n,c = n+1,2*((2*n+1)/(n+2)*c)
```

The output of the code is a list of Catalan numbers, each followed by a decimal point, indicating they are floats:

```
1
1.0
2.0
5.0
14.0
42.0
132.0
429.0
1430.0
4862.0
16796.0
58786.0
208012.0
742900.0
```

FIGURE 6 – Números de Catalán