

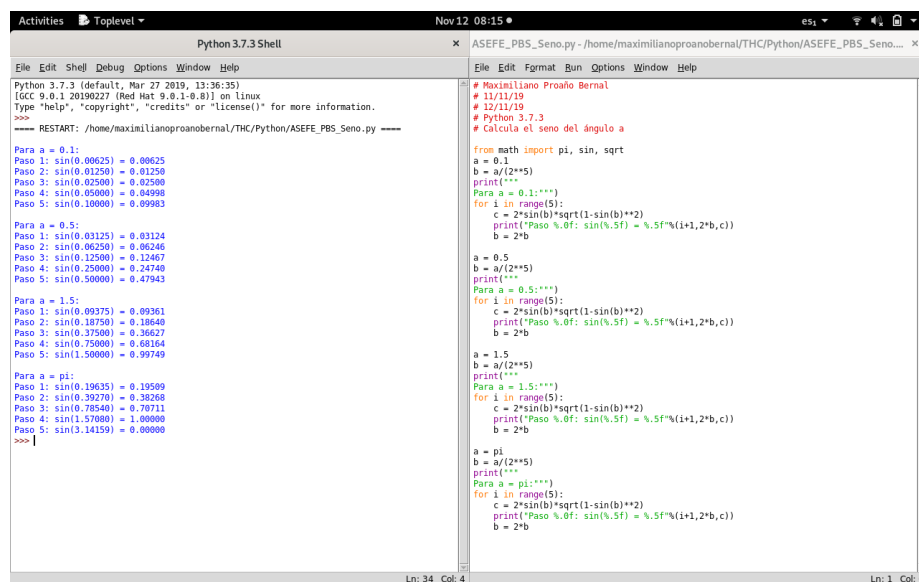
Reporte del archivo Seno.tex

Maximiliano Proaño Bernal

November 11, 2019

Cuando concluí la práctica pude notar que, analizando cómo se llevó a cabo la actividad, es evidente que tenía muchas similitudes con la práctica de Raíz Cuadrada. A pesar de que era un diferente concepto, el procedimiento y los pasos a seguir fueron muy similares.

Al empezar escribiendo el código del problema no tuve mayor complicación. Sin embargo, en las últimas líneas noté que algo no funcionaba como debía de hacerlo. Tras un pequeño análisis de mi código, observe que tenía una confusión con las variables que usé. Afortunadamente pude corregir este error inmediatamente



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (default, Mar 27 2019, 13:36:35)
[GCC 9.0.1 20190227 (Red Hat 9.0.1-0.8)] on linux
Type "help()", "copyright()", "credits()" or "license()" for more information.
>>>
==== RESTART: /home/maximilianooproanobernal/THC/Python/ASEFE_PBS_Seno.py ====

Para a = 0.1:
Paso 1: sin(0.00625) = 0.00625
Paso 2: sin(0.01250) = 0.01250
Paso 3: sin(0.02500) = 0.02500
Paso 4: sin(0.05000) = 0.04998
Paso 5: sin(0.10000) = 0.09983

Para a = 0.5:
Paso 1: sin(0.03125) = 0.03124
Paso 2: sin(0.06250) = 0.06246
Paso 3: sin(0.12500) = 0.12467
Paso 4: sin(0.25000) = 0.24740
Paso 5: sin(0.50000) = 0.47943

Para a = 1.5:
Paso 1: sin(0.09375) = 0.09361
Paso 2: sin(0.18750) = 0.18640
Paso 3: sin(0.37500) = 0.36627
Paso 4: sin(0.75000) = 0.68164
Paso 5: sin(1.50000) = 0.99749

Para a = pi:
Paso 1: sin(0.19635) = 0.19589
Paso 2: sin(0.39270) = 0.38268
Paso 3: sin(0.78540) = 0.70711
Paso 4: sin(1.57080) = 1.00000
Paso 5: sin(3.14159) = 0.00000
>>>

# Maximiliano Proaño Bernal
# 11/11/19
# 12/11/19
# Python 3.7.3
# Calcula el seno del ángulo a

from math import pi, sin, sqrt
a = 0.1
b = a/(2**5)
print("...")
Para a = 0.1:***
for i in range(5):
    c = 2*sin(b)*sqrt(1-sin(b)**2)
    print("Paso %.8f: sin(%.5f) = %.5f"%(i+1,2*b,c))
    b = 2*b

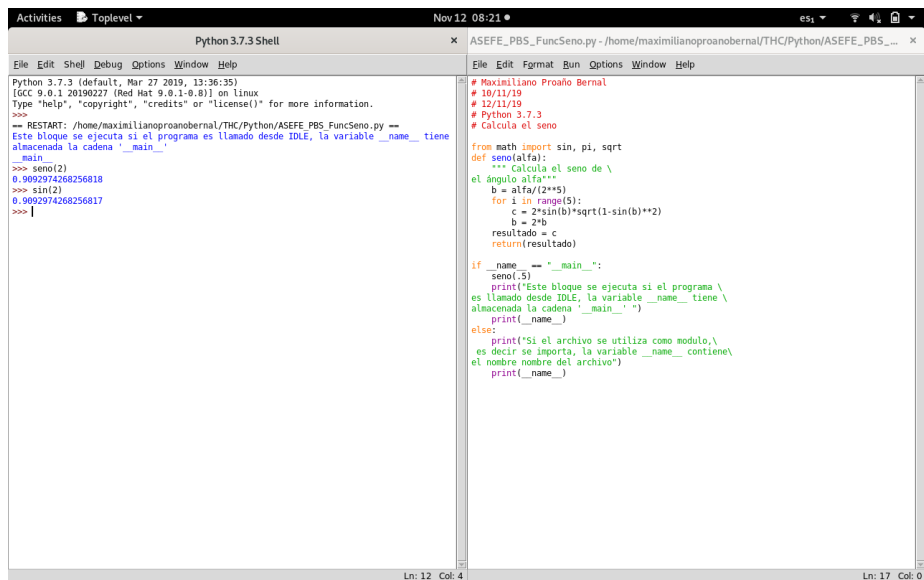
a = 0.5
b = a/(2**5)
print("...")
Para a = 0.5:***
for i in range(5):
    c = 2*sin(b)*sqrt(1-sin(b)**2)
    print("Paso %.8f: sin(%.5f) = %.5f"%(i+1,2*b,c))
    b = 2*b

a = 1.5
b = a/(2**5)
print("...")
Para a = 1.5:***
for i in range(5):
    c = 2*sin(b)*sqrt(1-sin(b)**2)
    print("Paso %.8f: sin(%.5f) = %.5f"%(i+1,2*b,c))
    b = 2*b

a = pi
b = a/(2**5)
print("...")
Para a = pi:***
for i in range(5):
    c = 2*sin(b)*sqrt(1-sin(b)**2)
    print("Paso %.8f: sin(%.5f) = %.5f"%(i+1,2*b,c))
    b = 2*b
```

Figure 1:

Una vez resuelto el problema con el archivo "ASEFE PBS Seno.py" no fue difícil implementar la misma idea al archivo "ASEFE PBS FuncSeno.py".



The image shows a screenshot of a computer screen with two windows. The left window is titled 'Python 3.7.3 Shell' and shows the Python interpreter's prompt. The right window is titled 'ASEFE_PBS_FuncSeno.py' and shows a Python script. The script defines a function 'seno(alfa)' that calculates the sine of an angle using a Taylor series approximation. The script also includes a main block that prints the result of the function for a specific angle.

```
Python 3.7.3 Shell:
Python 3.7.3 (default, Mar 27 2019, 13:36:35)
[GCC 9.0.1 20190227 (Red Hat 9.0.1-0.8)] on linux
Type "help()", "copyright()", "credits()" or "license()" for more information.
>>>
-- RESTART: /home/maximilianooproanobernal/THC/Python/ASEFE_PBS_FuncSeno.py --
Este bloque se ejecuta si el programa es llamado desde IDLE, la variable __name__ tiene
almacenada la cadena '__main__'
__main__
>>> seno(2)
0.9092974268256818
>>> sin(2)
0.9092974268256817
>>> ]
```

```
ASEFE_PBS_FuncSeno.py:
# Maximiliano Proaño Bernal
# 10/11/19
# 12/11/19
# Python 3.7.3
# Calcula el seno

from math import sin, pi, sqrt
def seno(alfa):
    """Calcula el seno de \
    el ángulo alfa"""
    b = alfa/(2*pi)
    for i in range(5):
        c = 2*sin(b)*sqrt(1-sin(b)**2)
        b = 2*b
        resultado = c
    return(resultado)

if __name__ == "__main__":
    seno(5)
    print("Este bloque se ejecuta si el programa \
    es llamado desde IDLE, la variable __name__ tiene \
    almacenada la cadena '__main__'")
    print(__name__)
else:
    print("Si el archivo se utiliza como modulo, \
    es decir se importa, la variable __name__ contiene \
    el nombre nombre del archivo")
    print(__name__)
```

Figure 2:

En la figura podemos observar que el error de la función "seno" (la que debíamos hacer nosotros) con la función "sin" (la que ofrece Python) es increíblemente pequeño, lo que implica que la función "seno" realiza de manera casi perfecta su propósito.

Para concluir, se hacen unas preguntas al final del archivo Seno.tex: 1. ¿El valor de la potencia 5 es adecuado para todos los ángulos? 2. ¿Entre mayor sea n la precisión incrementará?

La respuesta a la pregunta 1 es: Sí. No importa cuál sea el ángulo en cuestión, la fórmula utilizada para calcular la cifra no tiene falla.

La respuesta a la pregunta 2 es: Sí. Este es un proceso que, al tener valores más grandes de n , el ángulo inicial será muy parecido a su seno, lo que incrementa la precisión del procedimiento.

Finalmente, quiero agregar que, a pesar de ser una práctica parecida a la anterior, sigue siendo un excelente ejercicio para estudiantes principiantes en programación.