

Actividad 03 – Python

Hernández Lomelí Diego Armando

Seminario de algoritmia

Lineamientos de actividad.

- El reporte está en formato Google Docs o PDF.
- El reporte sigue las pautas del [Formato de Actividades](#).
- El reporte tiene desarrollada todas las pautas del [Formato de Actividades](#).
- Se muestra resultado correcto de las funciones para el cálculo de áreas de figuras del punto a., captura de pantalla de los resultados y código.
- Se muestra resultado correcto del cálculo del signo zodiacal del punto b., tres capturas de pantalla de los resultados y código.
- Se muestra resultado correcto del cálculo del [Número e](#) del punto c., tres capturas de pantalla de los resultados y código.

Desarrollo

Se comenzó por desarrollar el código para resolver los puntos siguientes. El código se encuentra en su sección posterior a esta.

1. Programa 3 funciones para calcular el área de las figuras: cuadrado, triángulo y círculo. Usa `input()` para pedir al usuario las variables para calcular el área de cada figura. Una función por figura. Prueba las funciones y toma capturas de pantalla de la salida de las pruebas realizadas a las funciones.

```
PS C:\Users\Armando\Documents\Github\Actividad 3> & C:/Users/Armando/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Armando/Documents/Github/Actividad 3/AreaFiguras.py"
Ingrese la medida de un lado del cuadrado:
6
El area es: 36

Ingrese la base del triangulo
3
Ingrese la altura del triangulo
4
El area del triangulo es: 6.0

Ingrese el radio del triangulo
5
El area del circulo es: 78.53981633974483
```

El flujo de este punto se basa en las 3 funciones definidas al inicio del script y su utilización es justo después de la declaración de los metodos

2. Haz un script para pedirle al usuario el día y mes de nacimiento, y de acuerdo a su día y mes de nacimiento mostrar en pantalla su signo zodiacal. El día debe ser leído como entero (1-31) al igual el mes (1-12). No validar la entrada, suponemos que el

usuario siempre ingresará día y mes correctos. Prueba tu script tres veces con diferentes fechas y toma capturas de pantalla de estas pruebas.

El desarrollo de este script trata de imitar las cláusulas de guarda, con ellas podemos detener el flujo de la ejecución del método para llegar a un punto en específico, también aprovechamos para no utilizar un switch, además, para saber el signo tomamos el mes, y después definimos un rango en los días, pues si el día ingresado es menor a un tope, pertenece a un signo específico, en caso contrario, el signo es el siguiente del signo anteriormente dicho.

```
PS C:\Users\Armando\Documents\GitHub\Actividad 3> & C:/Users/Armando/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Armando/Documents/GitHub/Actividad 3/2 zodiaco.py"
¿Que signo zodiacal eres?

Ingrese el mes de nacimiento:
10
Ingrese el día de nacimiento:
20
El signo zodiacal es: Libra

PS C:\Users\Armando\Documents\GitHub\Actividad 3> & C:/Users/Armando/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Armando/Documents/GitHub/Actividad 3/2 zodiaco.py"
¿Que signo zodiacal eres?

Ingrese el mes de nacimiento:
5
Ingrese el día de nacimiento:
25
El signo zodiacal es: Géminis

PS C:\Users\Armando\Documents\GitHub\Actividad 3> & C:/Users/Armando/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Armando/Documents/GitHub/Actividad 3/2 zodiaco.py"
¿Que signo zodiacal eres?

Ingrese el mes de nacimiento:
12
Ingrese el día de nacimiento:
3
El signo zodiacal es: Sagitario
```

3. Haz un script que haga el cálculo del
$$\text{Número } e: e = \sum_{n=0}^{\infty} \frac{1}{n!}$$
 donde ∞ es el límite del ciclo, por lo que entre mayor sea el límite del ciclo, mayor precisión del número e .

Prueba tu script con tres diferentes límites en donde demuestres con capturas de pantalla que entre más grande el límite es más preciso el valor del [Número e](#).

Desarrolle este punto en base al proveído en la plataforma, con el solo tuve que ajustar los parámetros por recibir y hacer que el usuario ingrese la entrada de los parametros

```

PS C:\Users\Armando\Documents\GitHub\Actividad 3> & C:/Users/Armando/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Armando/Documents/GitHub/Actividad 3/3 Numero e.py"
Calculando e:

Ingrese un limite entero:
5
Resultado del calculo: 2.708333333333333

PS C:\Users\Armando\Documents\GitHub\Actividad 3> & C:/Users/Armando/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Armando/Documents/GitHub/Actividad 3/3 Numero e.py"
Calculando e:

Ingrese un limite entero:
10
Resultado del calculo: 2.7182815255731922

PS C:\Users\Armando\Documents\GitHub\Actividad 3> & C:/Users/Armando/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Armando/Documents/GitHub/Actividad 3/3 Numero e.py"
Calculando e:

Ingrese un limite entero:
100
Resultado del calculo: 2.7182818284590455

```

Conclusiones

Con esta actividad pude aprender a definir funciones individuales con Python y empezara familiarizarme con la escritura del lenguaje, que difiere en gran parte a código de lenguajes de nivel inferior como **c++** o **java**

Referencias

Referencias

BOITES, M. D. (2022 de septiembre de 15). Obtenido de Youtube: https://youtu.be/uqCRDpaSH_I

Código

Subida a git

```
Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 3 (master)
$ git init
Reinitialized existing Git repository in C:/Users/Armando/Documents/GitHub/Actividad 3/.git/

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 3 (master)
$ git commit -m "Subida"
On branch master
nothing to commit, working tree clean

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 3 (main)
$ git remote add origin https://github.com/Diego-Armando-H/Actividad-3.git

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 3 (main)
$ git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 4 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 80.86 KiB | 8.98 MiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Diego-Armando-H/Actividad-3.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Punto 1(Area figuras).

```
import math

def calcularCuadrado(a):
    return a*a

def calcularAreaTriangulo(base, altura):
    return (base * altura)/2

def calcularAreaCirculo(radio):
    return math.pi * (radio*radio)

# Calculo del cuadrado
ladoCuadrado = int(input("Ingrese la medida de un lado del cuadrado:\n"))

print("El area es: ", calcularCuadrado(ladoCuadrado))
# Calculo del triangulo
base = int(input("\nIngrese la base del triangulo\n"))
altura = int(input("Ingrese la altura del triangulo\n"))

print("El area del triangulo es: ", calcularAreaTriangulo(base, altura))
# Calculo del circulo
radio = int(input("\nIngrese el radio del triangulo\n"))
```

```
print("El area del circulo es: ", calcularAreaCirculo(radio))
```

Punto 2 (Signos zodiacales)

```
def signoZodiacal(mes, dia):  
    if(mes == 1):  
        if(dia < 20):  
            return "Capricornio"  
        return "Acuario"  
    if(mes == 2):  
        if(dia < 19):  
            return "Acuario"  
        return "Piscis"  
    if(mes == 3):  
        if(dia < 21):  
            return "Piscis"  
        return "Aries"  
    if(mes == 4):  
        if(dia < 20):  
            return "Aries"  
        return "Tauro"  
    if(mes == 5):  
        if(dia < 21):  
            return "Tauro"  
        return "Géminis"  
    if(mes == 6):  
        if(dia < 21):  
            return "Géminis"  
        return "Cancer"  
    if(mes == 7):  
        if(dia < 23):  
            return "Cancer"  
        return "Leo"  
    if(mes == 8):  
        if(dia < 23):  
            return "Leo"  
        return "Virgo"  
    if(mes == 9):  
        if(dia < 23):  
            return "Virgo"  
        return "Libra"  
    if(mes == 10):  
        if(dia < 22):
```

```

        return "Libra"
    return "Escorpio"
if(mes == 11):
    if(dia < 23):
        return "Escorpio"
    return "Sagitario"
if(mes == 12):
    if(dia < 23):
        return "Sagitario"
    return "Diciembre"

print("¿Que signo zodiacal eres?\n4")
mes = int(input("Ingrese el mes de nacimiento:\n"))
dia = int(input("Ingrese el día de nacimiento:\n"))

print("El signo zodiacal es: ", signoZodiacal(mes, dia))

```

Punto 3 (Calculo del número e).

```

def factorial(n):
    fac = 1
    for i in range(1, n+1):
        fac = fac * i
    return fac

print("Calculando e:\n")
limite = int(input("Ingrese un limite entero:\n"))
# Inicializamos los valores para el calculo
n = 0
e = 0
while n < limite:
    e += 1/factorial(n) # se llama a la función factorial creada por ti
    n = n + 1

print("Resultado del calculo: ", e)

```