

75.40 Algoritmos y Programación I

CURSO: 04 - Essaya

EJ1 - Área de polígonos

Alumno: Facundo Prieto - 106886

Ayudante a cargo: Javier Di Santo

Parte 1: Entorno de trabajo

Parte 1.1:

4. Tomar una captura de pantalla del resultado

```
facurpri in ~  
> python3  
Python 3.8.5 (default, Jul 27 2020, 08:42:51)  
[GCC 10.1.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> "Hola Algoritmos y Programación I"  
'Hola Algoritmos y Programación I'  
>>> 
```

Parte 1.2:

4. Tomar una captura de pantalla del resultado

```
facurpri in algoritmos-programacion-i-essaya/tps/ej1_area_de_polinomio on master [?]  
> python3 parte_1_2.py  
facurpri in algoritmos-programacion-i-essaya/tps/ej1_area_de_polinomio on master [?]  
> 
```

5. ¿Qué función debo usar para conseguir el mismo resultado que en la parte 1.1?

¿Por qué en la parte 1.1 vemos el resultado aun sin haber usado esta función?

- Deberíamos usar la función `print` pasando el texto que queremos que se imprima en pantalla
- Por que en la parte 1.1 estamos usando el shell de python que por defecto imprime en pantalla los resultados

Parte 2: norma.py

2. Tomar una captura de pantalla de la ejecución.

```
facurpri in algoritmos-programacion-i-essaya/tps/ej1_area_de_polinomio on master [?]  
> python3 norma.py  
facurpri in algoritmos-programacion-i-essaya/tps/ej1_area_de_polinomio on master [?]  
> 
```

5. Tomar captura de pantalla de la nueva ejecución y contestar las preguntas:

1. ¿Cuál es la salida del programa?
2. ¿Podemos saber en qué línea se generó el error? ¿Cómo?
3. ¿Qué hace la instrucción `assert`?
4. Solucionar el problema. Hallar el valor de `z` para que ya no de error.

```

facurpri in algoritmos-programacion-i-essaya/tps/ej1_area_de_polinomio on master [?]
> python3 norma.py
Traceback (most recent call last):
  File "norma.py", line 17, in <module>
    assert norma(-70, 14, z) == 111.0
AssertionError
facurpri in algoritmos-programacion-i-essaya/tps/ej1_area_de_polinomio on master [?]
> 

```

1. Es un mensaje de error
2. El error se generó en la línea 17, fue un AssertionError, osea, una de las pruebas fallo
3. La instrucción assert recibe una condición que se debe cumplir, cuando la cumple devuelve True cuando no False o error, se usa para hacer test automatizados en un programa/ funcion
4. $z = 85$ soluciona el problema

Parte 3: diferencia.py

3. Tomar captura de pantalla de la terminal mostrando su ejecución.

```

facurpri in algoritmos-programacion-i-essaya/tps/ej1_area_de_polinomio on master [?]
> python3 diferencia.py
Traceback (most recent call last):
  File "diferencia.py", line 11, in <module>
    assert diferencia(1, 2, 3, 1, 2, 3) == (0, 0, 0)
  File "diferencia.py", line 6, in diferencia
    return dif_x, dif_y, diff_z
NameError: name 'diff_z' is not defined
facurpri in algoritmos-programacion-i-essaya/tps/ej1_area_de_polinomio on master [?]
> 

```

4. ¿Se detectó algún error? ¿Cuál era? ¿Qué significa? ¿Qué línea estaba fallando?
 - Se detectaron 2 errores uno en la línea 11 y otro en la 6.
 - El de la línea 11 dice que uno de los assert falló.
 - El de la línea 6 dice que una función está devolviendo una variable no definida.

Parte 4: Depuración

4. ¿Qué error muestra? ¿En qué línea?
 - El error ocurre en la línea 11 y dice que una de las pruebas fallo
6. Renombrar la función y las variables de forma que sus nombres sean representativos. ¿Por qué es importante hacer esto?
 - Es importante tener funciones y variables con nombres representativos para hacer que sus funcionalidades sean explícitas de modo que el código termine siendo más limpio y legible, facilitando la manutención y colaboración.

Parte 5: Reutilizando funciones

4. ¿Cuál es la importancia de reutilizar funciones?
 - Es importante poder reutilizar funciones para evitar repetirse y copiar y pegar código.