

Retos de Programación en Python: Escuela, Tienda y Maestros

13 de junio de 2025

1. Introducción

Este documento presenta seis retos de programación en Python, diseñados para ser resueltos por seis equipos. Cada reto aplica conceptos de funciones y módulos de Python a problemas prácticos relacionados con el ámbito escolar, una tienda y los maestros. Los retos incluyen una descripción detallada, ejemplos de entrada y la salida esperada. Cada equipo debe desarrollar un programa en Python que cumpla con los requisitos especificados, utilizando funciones pre-construidas, funciones personalizadas y los módulos `math` y `statistics` según corresponda.

2. Instrucciones

- Cada equipo debe resolver el reto asignado, escribiendo un programa en Python.
- Asegúrese de importar los módulos `math` y `statistics` según las indicaciones de cada reto.
- La salida del programa debe coincidir con el formato especificado en los ejemplos de salida.
- Ejecute y pruebe el programa con las entradas proporcionadas y, opcionalmente, con otros casos de prueba.
- Colabore en equipo para diseñar una solución eficiente y clara.

3. Retos para los Equipos

3.1. Reto 1: Promedio de Calificaciones Escolares (Equipo 1)

Descripción: Desarrolle una función `promedio_calificaciones` que calcule el promedio de una lista de calificaciones de estudiantes y determine si el grupo aprueba (promedio $\geq 6,0$). La función debe:

- Validar que la lista no esté vacía usando `len()`.
- Calcular el promedio con `statistics.mean()`.

- Devolver una cadena que indique el promedio (redondeado a 2 decimales) y si el grupo aprueba o no.

Importe `statistics` con el alias `stats`. Pruebe con la lista de calificaciones `[7.5, 8.0, 5.5, 6.5]`.

Ejemplo de Entrada:

```
calificaciones = [7.5, 8.0, 5.5, 6.5]
promedio_calificaciones(calificaciones)
```

Salida Esperada:

```
Promedio: 6.88, Grupo: Aprobado
```

3.2. Reto 2: Cálculo de Descuentos en Tienda (Equipo 2)

Descripción: Cree una función `calcular_descuento` que determine el precio final de un producto tras aplicar un descuento basado en su precio original. La función recibe el precio y un porcentaje de descuento, y usa la fórmula:

$$\text{Precio final} = \text{Precio original} \cdot \left(1 - \frac{\text{Porcentaje}}{100}\right)$$

Valide que el precio sea positivo y el porcentaje esté entre 0 y 100 usando una función interna. Importe `math` con alias `m` y use `m.floor()` para redondear el precio final hacia abajo a un entero. Devuelva el precio final.

Ejemplo de Entrada:

```
precio = 1500
porcentaje = 20
calcular_descuento(precio, porcentaje)
```

Salida Esperada:

```
1200
```

3.3. Reto 3: Evaluación de Asistencia de Maestros (Equipo 3)

Descripción: Defina una función `evaluar_asistencia` que analice una lista de asistencias diarias de un maestro (representadas como 1 para presente y 0 para ausente). La función debe:

- Calcular el porcentaje de asistencia usando `sum()` y `len()`.
- Devolver una cadena que indique el porcentaje (redondeado a 2 decimales) y si el maestro cumple ($\geq 90\%$) o no.

Importe `math` y use `math.ceil()` para redondear el porcentaje hacia arriba en el mensaje. Pruebe con `[1, 1, 0, 1, 1]`.

Ejemplo de Entrada:

```
asistencias = [1, 1, 0, 1, 1]
evaluar_asistencia(asistencias)
```

Salida Esperada:

```
Porcentaje de asistencia: 80.00%, Cumple: No
```

3.4. Reto 4: Distribución de Salones Escolares (Equipo 4)

Descripción: Escriba una función `distribuir_salones` que calcule cuántos salones circulares son necesarios para un número de estudiantes, dado un radio fijo por salón y una densidad máxima de estudiantes por metro cuadrado. Use la fórmula del área circular ($\text{Área} = \pi \cdot r^2$) y divida el total de estudiantes entre el área por salón multiplicada por la densidad. Importe `math` como `m` y use `m.pi` y `m.ceil()` para redondear hacia arriba el número de salones. Devuelva el número de salones.

Ejemplo de Entrada:

```
estudiantes = 100
radio = 5
densidad = 0.5
distribuir_salones(estudiantes, radio, densidad)
```

Salida Esperada:

```
3
```

3.5. Reto 5: Análisis de Ventas en Tienda (Equipo 5)

Descripción: Cree una función `analizar_ventas` que procese una lista de montos de ventas diarias en una tienda. La función debe:

- Validar que la lista no esté vacía con `len()`.
- Calcular la media y la desviación estándar con `statistics.mean()` y `statistics.stdev()`.
- Identificar días con ventas excepcionales (superiores a media + desviación estándar).

Importe `statistics` como `stats` y devuelva un diccionario con la media, desviación estándar (redondeadas a 2 decimales), y el número de días excepcionales.

Ejemplo de Entrada:

```
ventas = [200, 250, 300, 500, 220]
analizar_ventas(ventas)
```

Salida Esperada:

```
{ 'media': 294.00, 'desviacion_estandar': 117.35,
  'excepcionales': 1 }
```

3.6. Reto 6: Planificación de Cursos para Maestros (Equipo 6)

Descripción: Desarrolle una función `planificar_cursos` que ayude a un maestro a planificar cursos según el número de estudiantes y la capacidad máxima por curso. La función recibe el total de estudiantes y una lista de capacidades máximas propuestas, y debe:

- Calcular la capacidad promedio con `statistics.mean()`.
- Estimar el número de cursos necesarios dividiendo el total de estudiantes por la capacidad promedio, redondeado hacia arriba con `math.ceil()`.
- Validar que las capacidades sean positivas.

Importe `math` como `m` y `statistics` como `stats`. Devuelva un diccionario con la capacidad promedio (redondeada a 2 decimales) y el número de cursos.

Ejemplo de Entrada:

```
estudiantes = 50
capacidades = [15, 20, 25]
planificar_cursos(estudiantes, capacidades)
```

Salida Esperada:

```
{'capacidad_promedio': 20.00, 'cursos': 3}
```

4. Conclusión

Estos retos promueven el aprendizaje práctico de Python aplicado a escenarios escolares, comerciales y docentes. Cada equipo debe colaborar para desarrollar una solución eficiente, utilizando funciones y módulos de Python de manera efectiva. ¡Éxito en la resolución de los retos!