



Profundiza más

Recurso de Profundización

Clase 4: Tipos de datos, operadores y variables (Parte 2)

Autor: Damián Nicolalde Rodríguez

Objetivo del Documento: Reforzar el aprendizaje sobre operadores en Python al abordar aspectos de prioridad (para escribir expresiones correctas y evitar confusiones) y aplicaciones prácticas (cálculo de notas con operadores aritméticos y lógicos). Se recomienda utilizarlos junto con el contenido principal de la clase para consolidar la comprensión y la capacidad de aplicar estos conceptos en proyectos reales.

Recurso de Profundización 1: Tabla de Prioridad de Operadores en Python

Este recurso presenta un listado jerárquico de la precedencia de los operadores en Python, mostrando el orden en el que se evalúan las expresiones cuando se combinan distintos operadores. Conocer esta prioridad evita resultados inesperados y ayuda a escribir código más claro, pues, aunque se pueden utilizar paréntesis para forzar un orden específico, es fundamental entender el orden natural de evaluación. En Python, por ejemplo, la exponenciación (`**`) se evalúa antes que la multiplicación y la división, que a su vez preceden a la suma y la resta. Además, se ubican los operadores lógicos en un nivel diferente, ejecutándose después de las operaciones aritméticas y de comparación. La tabla ofrece ejemplos ilustrativos de cómo un cambio en el orden de evaluación puede modificar el resultado de la expresión, fomentando la escritura de algoritmos correctos y confiables.

Tabla 1. Tabla de Prioridad de Operadores en Python (Resumen):

Nivel de Prioridad (Mayor a Menor)	Operador(es)	Ejemplo	Comentario
1.	**	2 ** 3 * 4 se (2 ** 3) * 4	Se evalúa primero, pero interpreta como cuidado con la asociatividad (derecha a izquierda).



Profundiza más

2. Signos unarios	+x, -x, ~x (operadores unarios)	-5 * 2	Aplica el signo o negación bit a bit (para ~) antes de la operación siguiente.
3. Multiplicación, División, Módulo	*, /, //, %	10 / 2 * 3 se evalúa como (10 / 2) * 3 = 5 * 3 = 15	Se evalúan de izquierda a derecha; la división // (entera) y el módulo (%) comparten el mismo nivel.
4. Suma y Resta	+, -	10 - 2 + 4 se interpreta como (10 - 2) + 4	También se evalúan de izquierda a derecha.
5. Operadores de Comparación	==, !=, >, <, >=, <=, is, in	(10 - 2) > 5	Determinan valores booleanos y se evalúan tras las operaciones aritméticas.
6. Operadores Lógicos	not, and, or	not (3 > 1 and 2 > 5)	not tiene mayor prioridad que and, y and mayor que or.

Damián Nicolalde (2025)

Recurso de Profundización 2: Estudio de Caso – Comparación y Cálculo de Notas en Python

Este estudio de caso muestra cómo integrar operadores aritméticos (para calcular el promedio) y operadores lógicos (para determinar si el estudiante aprobó) en un programa que solicita al usuario tres notas de tipo float. A partir de estas notas, el programa calcula el promedio y verifica si supera un umbral (por ejemplo, 7.0). De esta forma, los estudiantes ven en la práctica cómo se combinan los conceptos de declaración de variables, tipos de datos y operadores para resolver un problema educativo. Se incluye pseudocódigo, un diagrama de flujo y ejemplos de salidas, reforzando la importancia de la planificación y la validación en el desarrollo de software.



Profundiza más

Pseudocódigo (Versión Simplificada):

INICIO

LEER nota1, nota2, nota3 (tipo float)

CALCULAR promedio = (nota1 + nota2 + nota3) / 3

LEER umbral (por ejemplo, 7.0)

SI (promedio >= umbral) ENTONCES

 MOSTRAR "Aprobado"

SINO

 MOSTRAR "Reprobado"

FIN SI

FIN

Diagrama de Flujo (Versión Simplificada):

1. **Inicio**
2. Leer nota1, nota2, nota3.
3. Calcular promedio = (nota1 + nota2 + nota3) / 3.
4. Comparar promedio con umbral (por ejemplo, 7.0).
5. Si promedio >= umbral, mostrar "Aprobado"; de lo contrario, "Reprobado".
6. **Fin**

Ejemplo de Salidas Posibles:

- **Caso 1:** Notas: 8.0, 7.5, 9.0 → Promedio = 8.167 → "Aprobado".
- **Caso 2:** Notas: 6.5, 5.0, 4.0 → Promedio = 5.167 → "Reprobado".
- **Caso 3:** Notas: 7.0, 7.0, 7.0 → Promedio = 7.0 → "Aprobado" (al igualar el umbral).

Este ejemplo demuestra cómo los operadores aritméticos (+, /) se utilizan para obtener el promedio y los operadores de comparación (por ejemplo, >=) y lógicos (en caso de necesitar evaluar más condiciones) determinan la acción final. La combinación de estas herramientas hace posible un flujo de decisión claro y eficiente, imprescindible en el desarrollo de aplicaciones de evaluación o reportes académicos.