

Módulo 1

Lección 3

Operadores y expresiones



Lección 3 – Operadores y expresiones

Un operador es un símbolo del lenguaje de programación, el cual es capaz de realizar operaciones con los valores.

Los operadores más comunes son los que están asociados con las operaciones aritméticas más conocidas:

```
+   # Suma
-   # Resta
*   # Multiplicación
/   # División
//  # División parte entera
%   # Módulo o Residuo
**  # Exponenciación
```

El orden en el que se presentan hace referencia a algo llamado precedencia, se utilizará la convención matemática PENDAS, más adelante se detalla.

Note que los operadores tienen un espacio a cada lado, esto es solo por motivos de facilitar la legibilidad del código. Cuando los datos y operadores se unen, forman juntos expresiones.



```
[ ] #Operador Exponenciación
    print(2 ** 3)      # Entre enteros, el resultado es entero
    print(2 ** 3.)     # Si hay flotantes entonces el resultado es flotante
```

Ejercitación

- Revisar que pasa con todos los operadores cuando se usan enteros y flotantes en la expresión.
- Qué pasa con los números negativos?
- Todos son operadores binarios, pero cuál también es operador unario?
- Ahora, qué pasa si no se entrega un número?

```
[ ] 1//2
```

0

```
[ ] (2 + 3 * 5) #Cuál es el resultaod y por qué?
```

```
[ ] print(2 ** 2 ** 3)
```

```
[ ] print((5 * ((25 % 13) + 100) / (2 * 13)) // 2)
```

```
[ ] -4 ** -1
```

-0.25

Algunos operadores actúan antes que otros, a esto se le llama – jerarquía de prioridades:

- El operador ****** (exponencial) tiene la prioridad más alta;
- Posteriormente los operadores unarios **+** y **-** (nota: los operadores unarios a la derecha del operador exponencial enlazan con mayor fuerza, por ejemplo $4 ** -1$ es igual a 0.25)



- Después *, /, //, y `%`
- Finalmente, la prioridad más baja: los operadores binarios + y -.

```
[ ] (2 ** 3 ** 2 ** 2)
```

```
2417851639229258349412352
```

```
[ ] 3 ** 2 ** 2
```

```
81
```

Ejercitación

A continuación una historia:

```
[Fuente: 2.4.7 LAB Variables, Fundamentos de Python 1, skills for all]
```

Érase una vez en la Tierra de las Manzanas, Juan tenía tres manzanas, María tenía cinco manzanas, y Adán tenía seis manzanas. Todos eran muy felices y vivieron por muchísimo tiempo. Fin de la Historia.

Tu tarea es:

- Crear las variables: john, mary, y adam;
- Asignar valores a las variables. El valor debe de ser igual al número de manzanas que cada quien tenía;
- Una vez almacenados los números en las variables, imprimir las variables en una línea, y separar cada una de ellas con una coma;



- Después se debe crear una nueva variable llamada `total_apples` y se debe igualar a la suma de las tres variables anteriores;
- Imprime el valor almacenado en `total_apples` en la consola;
- Actualiza el código para que ahora la cantidad de manzanas sea ingresada por el usuario utilizando la función `input()`

Ejercitación

Debe completar el código para ejecutar la siguiente expresión:

$$\frac{1}{x + \frac{1}{x + \frac{1}{x + \frac{1}{x}}}}$$

El resultado debe de ser asignado a `y`. Se cauteloso, observa los operadores y priorízalos. Utiliza cuantos paréntesis sean necesarios.

Aquí algunos casos de prueba

Fuente: Elaboración propia



Entrada	Salida
1	y = 0.600000000000000001
10	y = 0.09901951266867294
100	y = 0.009999000199950014
-5	y = -0.19258202567760344

Ejercitación

La tarea es preparar un código simple para evaluar o encontrar el tiempo final de un periodo de tiempo dado, expresándolo en horas y minutos. La hora de inicio se da como un par de horas (0..23) y minutos (0..59). El resultado debe ser mostrado en la consola.

Por ejemplo, si el evento comienza a las 12:17 y dura 59 minutos, terminará a las 13:16.

No te preocupes si tu código no es perfecto – está bien si acepta una hora invalida – lo más importante es que el código produzca una salida correcta acorde a la entrada dada.

Prueba el código cuidadosamente. Pista: utilizar el operador % puede ser clave para el éxito.



Entrada	Salida
Hora: 12, Minuto: 17, Duración: 59	13:16
Hora: 23, Minuto: 58, Duración: 642	10:40
Hora: 0, Minuto: 1, Duración: 2939	1:0

```
[ ] hour = int(input("Hora de inicio (horas): "))
    mins = int(input("Minuto de inicio (minutos): "))
    dura = int(input("Duración del evento (minutos): "))
    mins = mins + dura # encuentra el número total de minutos
    hour = hour + mins // 60 # encuentra el número de horas ocultas en los minutos y actualiza
    mins = mins % 60 # corrige los minutos para que estén en un rango de (0..59)
    hour = hour % 24 # corrige las horas para que estén en un rango de (0..23)
    print(hour, ":", mins, sep='')

```

```
Hora de inicio (horas): 12
Minuto de inicio (minutos): 17
Duración del evento (minutos): 59
13:16

```