

Sumatorias y productorias

Sumatorias y productorias

¿Qué aprenderás?

Introducción

Sumatorias

Productorias

1
2
2
3
4



¡Comencemos!

¿Qué aprenderás?

- Leer expresiones de sumatoria.
- Escribir código para resolver sumatorias.

Introducción

Es muy frecuente para un programador tener que implementar fórmulas matemáticas. Por ejemplo, al realizar algún cálculo específico del negocio, proyecciones de venta, o incluso algo más directo como trabajar en una simulación de algún área científica. Es por eso que tenemos que sentirnos cómodos como programadores viendo e implementando en código diversos tipos de fórmulas matemáticas.

¡Vamos con todo!



Sumatorias

El símbolo conocido que se utiliza para representar la sumatoria \sum proporciona una forma de representar sumas simples o complejas con grandes cantidades de sumandos.

Por ejemplo, dado una sucesión representada por i , lo siguiente, representa la suma de cada término de dicha sucesión.

$$\sum_{i=5}^{100} i = 5 + 6 + 7 + \dots + 100$$

El número inferior nos da el primer número de la iteración que toma el valor i , el superior corresponde al último. Para este caso i toma los valores $5 \leq i \leq 100$, es decir en esta expresión el primer número de la suma es 5 y el último es 100.

En código sería:

```
suma = 0
for i in (5..100)
  suma += i
end
print suma
```

5040

No siempre sumaremos de uno en uno. Veamos la siguiente expresión.

$$\sum_{i=3}^9 2i = 2 * 3 + 2 * 4 + \dots + 2 * 9$$

En este caso, al anteponer el 2 en la expresión, se representa la multiplicación entre dicho número y la variable. Implementar esto en código también es bastante directo.

```
suma = 0
for i in (3..9)
  suma += 2*i
end
print suma
```

84

La expresión puede llegar a ser un poco más compleja, pero siempre es la misma idea.

$$\sum_{i=1}^{10} i^2 + 2i$$

```
suma = 0
for i in (1..10)
  suma += (i**2)+2*i
end
print suma
495
```

495

Productorias

Existe una expresión similar a las sumatorias, llamadas productorias. El funcionamiento es el mismo que para las sumatorias, pero en este caso los números se multiplican en lugar de sumarse.

$$\prod_{i=1}^{10} i = 1 * 2 * 3 * \dots$$

```
producto = 1 # Es importante no inicializar el producto en 0, porque
cualquier multiplicación por cero dará como resultado cero.
for i in (1..10)
  producto *= i
end
print producto
```

3628800