

El conjunto de los enteros en Haskell

Diferencia entre `Int` e `Integer` en Haskell

- `Int` es el tipo (conjunto de pertenencia) de los enteros que la computadora puede representar, esto depende de la arquitectura interna de cada máquina.
- `Integer` representa al conjunto de los enteros matemáticos.

El significado del operador `::` en Haskell es "tiene tipo" Entonces en Haskell, la expresión `8 :: Int` indica que 8 "tiene tipo" `Int`, es decir, es un entero.

Escriba `8 :: Integer` en Hugs y explique el resultado.

El beneficio de usar `Integer` es teórico, propiedades que se verifican en `Integer`, podrán no verificarse en `Int`.

Realizar las siguientes pruebas en Hugs:

1. Calcular 2^2
2. Calcular 2^4
3. Calcular 2^{32}
4. Calcular 2^{64}
5. Calcular 2^{200}
6. Calcular $2^4 :: \text{Int}$
7. Calcular $2^{32} :: \text{Int}$
8. Calcular $2^{31} :: \text{Int}$
9. Calcular $2^{31} - 1 :: \text{Int}$

Estudiar los resultados obtenidos y analizar posibles explicaciones

Abrir en el bloc de notas el archivo `primero.hs`

Copiar al archivo `primero.hs` las siguientes definiciones de funciones:

```
f :: Int -> Int
f n = 2^n
```

```
g :: Int -> Integer
g n = 2^n
```

Estudiar su significado.

1. Calcular $f\ 2$
2. Calcular $g\ 4$
3. Calcular $f\ 32$
4. Calcular $f\ 64$
5. Calcular $f\ 200$
6. Calcular $g\ 200$
7. Calcular $g\ 35$
8. Otras aplicaciones de f o g que deseen

Explicar los resultados.

Hugs evalúa expresiones booleanas:

Escriba y ejecute las siguientes expresiones en Hugs.

- 1) $7 < 0$
- 2) $8 > 2$
- 3) $\text{div}\ 8\ 4 == 2$
- 4) $\text{div}\ 4\ 3 == 0$
- 5) $\text{mod}\ 8\ 4 == 0$
- 6) $\text{not}\ (\text{mod}\ 8\ 4 == 7)$
- 7) $4 /= 4$
- 8) $\text{div}\ 8\ 9 /= 20$
- 9) $\text{mod}\ 50\ 10 <= 0$
- 10) $\text{div}\ 50\ 10 >= 6$