



## Funciones sobre números

Desarrollar contenido

Evaluaciones

Herramientas

Descubrir contenido



## Funciones sobre números

Las funciones sobre números son todas aquellas que reciben como argumento uno o varios números y devuelven como resultado un valor numérico.



## Aritmética básica

(+) :: (Num a) => a -> a -> a

(x + y) es la suma de los números x e y.

```
λ> 2 + 3
5
```

(-) :: (Num a) => a -> a -> a

(x - y) es la diferencia de los números x e y.

```
λ> 2 - 3
-1
```

(\*) :: (Num a) => a -> a -> a

(x \* y) es el producto de los números x e y.

```
λ> 2 * 4
8
```

(/) :: (Fractional a) => a -> a -> a

(x / y) es el cociente de los números x e y.

```
λ> 2 / 4
0.5
```

div :: (Integral a) => a -> a -> a

div x y es el cociente de la división entera entre los números enteros x e y.

```
λ> div 6 4
1
```

rem :: (Integral a) => a -> a -> a

rem x y es el resto de la división entera entre los números enteros x e y.

```
λ> rem 6 4
2
λ> rem (-7) 4
-3
```

mod :: (Integral a) => a -> a -> a

mod x y es el valor del número entero x módulo el número entero y.

```
λ> mod 6 4
2
λ> mod (-7) 4
1
```

(^) :: (Num a, Integral b) => a -> b -> a

x ^ y es el valor del número x elevado al número entero positivo y.

```
λ> 6 ^ 2
36
```

(^^) :: (Fractional a, Integral b) => a -> b -> a

x ^^ y es el valor del número x elevado al número entero y.

```
λ> 6 ^^ (-2)
2.7777777777777776e-2
```

(\*\*) :: (Floating a) => a -> a -> a

x \*\* y es el valor del número x elevado al número y.

```
λ> 6 ** (0.5)
2.449489742783178
```



## Funciones de aproximación

`floor :: (RealFrac a, Integral b) => a -> b`  
(`floor x`) es el suelo del número `x`.

```
λ> floor 2.3456789
2
λ> floor (-2.3456789)
-3
```

`ceiling :: (RealFrac a, Integral b) => a -> b`  
(`ceiling x`) es el techo del número `x`.

```
λ> ceiling 2.3456789
3
λ> ceiling (-2.3456789)
-2
```

`round :: (RealFrac a, Integral b) => a -> b`  
(`round x`) es la aproximación del número `x` al entero más próximo.

```
λ> round 2.3456789
2
λ> round (-2.9876543)
-3
```

`truncate :: (RealFrac a, Integral b) => a -> b`  
(`truncate x`) es la parte entera del número `x`.

```
λ> truncate 2.3456789
2
λ> truncate (-2.9876543)
-2
```



## Funciones trigonométricas

`pi :: (Floating a) => a`  
`pi` es la constante  $\pi$ .

```
λ> pi
3.141592653589793
```

`sin :: (Floating a) => a -> a`  
(`sin x`) es el valor del seno del número `x` medido en radianes.

```
λ> sin (pi/2)
1.0
```

`cos :: (Floating a) => a -> a`  
(`cos x`) es el valor del coseno del número `x` medido en radianes.

```
λ> cos pi
-1
```

`tan :: (Floating a) => a -> a`  
(`tan x`) es el valor de la tangente del número `x` medido en radianes.

```
λ> tan (pi/4)
0.9999999999999999
```



## Operadores de relación

`(==) :: (Eq a) => a -> a -> Bool`  
(`x == y`) comprueba si los números `x` e `y` son iguales.

```
λ> 2 == 2
True
λ> 2 == 3
False
```

`(/=) :: (Eq a) => a -> a -> Bool`  
(`x /= y`) comprueba si los números `x` e `y` son distintos.

```
λ> 2 /= 2
False
λ> 2 /= 3
True
```

`(<) :: (Ord a) => a -> a -> Bool`  
(`x < y`) comprueba si el número `x` es estrictamente menor que el número `y`.

```
λ> 2 < 2
False
λ> 2 < 3
True
```

`(<=) :: (Ord a) => a -> a -> Bool`  
(`x <= y`) comprueba si el número `x` es menor o igual que el número `y`.

```
λ> 2 <= 1
False
```

```
λ> 2 <= 3
True
```

(>) :: (Ord a) => a -> a -> Bool  
(x > y) comprueba si el número x es estrictamente mayor que el número y.

```
λ> 2 > 2
False
λ> 3 > 2
True
```

(>=) :: (Ord a) => a -> a -> Bool  
(x >= y) comprueba si el número x es mayor o igual que el número y.

```
λ> 2 >= 2
True
λ> 2 >= 3
False
```



## Otras funciones

abs :: (Num a) => a -> a  
(abs x) es el valor absoluto del número x.

```
λ> abs 3
3
λ> abs (-3)
3
```

max :: (Ord a) => a -> a -> a  
(max x y) es el máximo de los números x e y.

```
λ> max 2 (-3)
2
```

min :: (Ord a) => a -> a -> a  
(min x y) es el mínimo de los números x e y.

```
λ> min 2 (-3)
-3
```

sqrt :: (Floating a) => a -> a  
(sqrt x) es la raíz cuadrada del número x.

```
λ> sqrt 2
1.4142135623730951
```

exp :: (Floating a) => a -> a  
(exp x) es el valor del número e elevado al número x.

```
λ> exp 1
2.718281828459045
λ> exp (-1/2)
0.6065306597126334
```

log :: (Floating a) => a -> a  
(log x) es el logaritmo natural del número x.

```
λ> log 2
0.6931471805599453
```

logBase :: (Floating a) => a -> a -> a  
(logBase b x) es el logaritmo en base b del número x.

```
λ> logBase 2 1024
10.0
```

odd :: (Integral a) => a -> Bool  
(odd x) comprueba si el número entero x es impar.

```
λ> odd 5
True
λ> odd (-4)
False
```

even :: (Integral a) => a -> Bool  
(even x) comprueba si el número entero x es par.

```
λ> even 5
False
λ> even (-4)
True
```

gcd :: (Integral a) => a -> a -> a  
(gcd x y) es el máximo común divisor de los números enteros x e y.

```
λ> gcd 30 42
6
λ> gcd 0 30
30
```

lcm :: (Integral a) => a -> a -> a  
(lcm x y) es el mínimo común múltiplo de los números enteros x e y.

```
λ> lcm 30 42
210
```

```
λ> lcm 0 30
0
```