[Curso de Dart](https://platzi.com/clases/dart/)

**ArtículoOperadores aritméticos, de asignacion y lógicos**

[**ArkangelB**](https://platzi.com/@ArkangelB)

24 de Febrero de 2019

**Operadores**

Los operadores utilizados en Dart son los siguientes:

Unario PostExpr

expr**++** expr**--** () **[]** . ?.

Unario PreExpr

-**expr** !**expr** ~**expr** ++**expr** --**expr**

Multiplicativo

\* / % ~/

Adición

+ -

Shift

<< >>

Bitwise

& ^ |

Relacional y tipo prueba

>= > <= < **as** **is** **is**!

Igualdad

**== !=**

Lógicos

&& ||

Si Nulo

??

Condicional

expr1 ? expr2 : expr3

Cascada

..

Asignación

\*= /= ~/= %= += -= <<= >>= &= ^= |= ??=

Con los operadores puedes crear expresiones, algunos ejemplos son:

a++  
a + b  
a = b  
a == b  
c ? a : b  
a is T

En la tabla de operadores, cada operador tiene mayor precedencia que los operadores de las filas siguientes. Por ejemplo, el operador multiplicativo % tiene mayor precedencia que (y por lo tanto ejecuta antes) el operador de igualdad ==, que tiene mayor precedencia que el operador lógico AND (&&). Esa precedencia significa que las siguientes dos líneas de código se ejecutan de la misma manera:

// Los paréntesis mejoras la lectura  
if ((n % i == 0) && (d % i == 0)) …

// Mas difícil de leer, pero equivalente.  
if (n % i == 0 && d % i == 0) …

**Operadores Aritméticos**

Dart soporta los operadores aritméticos habituales:

* Suma ( +)
* Resta ( - )
* -expr Unario menos también conocido como negación (invierte el signo de la expresión)
* Multiplicación
* / División
* ~/ Divide, regresando el valor entero
* % regresa el restante del entero en una división (modulo)

Dart también soporta operadores de incrementos y decrementos de prefijos y postfijos.

var El cual se leería; var = var + 1 (expression value is var + 1)  
var El cual se leería; var = var + 1 (expression value is var)  
–var El cual se leería; var = var – 1 (expression value is var – 1)  
var-- El cual se leería; var = var – 1 (expression value is var)

**Operadores de Igualdad y relacionales**

Enumeraremos el significado de la igualdad y de los operadores relacionales.

* == Igual
* != No igual (diferente)
* > Mayor que
* < Menor que
* >= Mayor que o igual a
* <= Menor que o igual a

**Operadores de prueba de tipo**

Los operadores as, is, and is! son útiles para verificar los tipos en tiempo de ejecución.

as Typecast (También utilizado para especificar prefijos de biblioteca)  
is Verdadero si el objeto es el tipo especificado  
is! Falso si el objeto tiene el tipo especificado

**Operadores de Asignación**

Como ya has visto, puedes asignar valores usando el operador =. Para asignar sólo si la variable asignada es nula, utilice el operador ??=.

// Asigna valor a variable a  
a = value;

// Asigna valor a b, si b es nulo; de lo contrario b se mantiene igual  
b ??= value;

Operadores que existen  
= –= /= %= >>= ^=  
+= \*= ~/= <<= &= |=

//Así es como trabajan  
Donde op es un operador a op= b a = a op b  
Ejemplo a += b a = a + b

**Operadores Lógicos**

Puedes invertir o combinar expresiones booleanas utilizando los operadores lógicos

!expr invierte la expresión que le sigue (cambia falso a verdadero y viceversa)  
|| OR lógico  
&& AND lógico

**Operadores Bitwise y Shift**

Puedes manipular los bits individuales de los números en Dart. Por lo general, se utilizan estos operadores de bits y de desplazamiento con enteros.

* & AND
* | OR
* ^ XOR
* ~expr Complemento unario bitwise (0s a 1s; 1s a 0s)
* << Desplazamiento hacia la izq
* >> Desplazamiento hacia la derecha

El uso de estos lo estaremos viendo mientras avancemos en el curso algunos de estos no serán visto pero te invito a probarlos y conocer la forma en que estos son utilizados, son fiel seguidor de la documentación te invito a visitarla por si tiene alguna duda o bien déjala en nuestra sección de comentarios y te ayudaremos a resolverla.