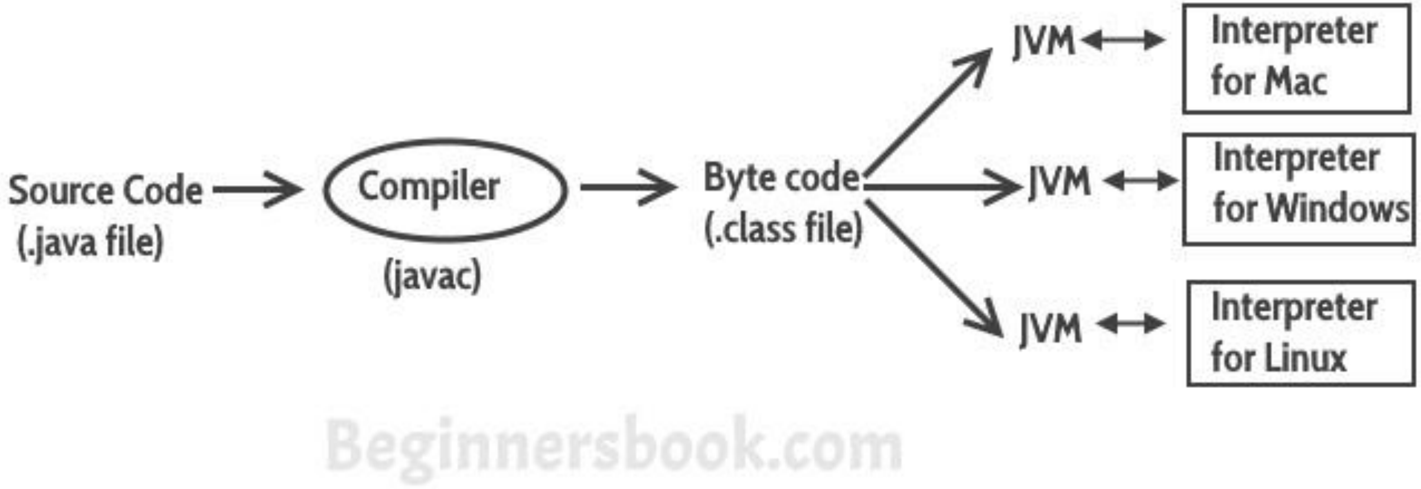
Introducción a JAVA

Características.

- Independiente de la plataforma.
- Orientado a objetos.
- Es Claro. Es simple. Sin sobrecarga de operadores (WTF), punteros, alojamiento explícito de memoria y herencia múltiple.
- Robusto. Detección temprana de errores.
- Seguro.
- Sistemas distribuidos.
- Multihilo.



Fuente: https://beginnersbook.com/2013/05/java-introduction/

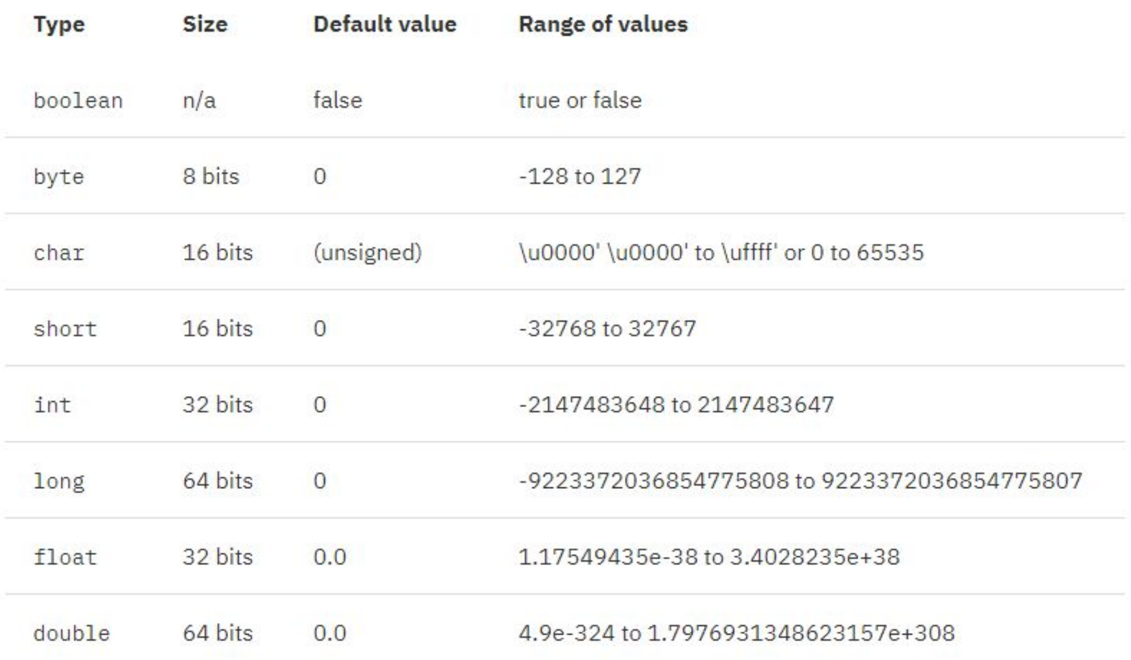
Palabras reservadas.

abstract assert boolean break byte case catch char class const
continue default do double else
enum extends final finally float for goto if implements import
instanceof int interface long native new
package private protected public return short static strictfp
super switch synchronized this throw
throws transient try void volatile while

Comentarios.

```
// This is a comment

/* This is a comment too */

/* This is a

multiline

comment *
```

Variables.



Type	Size	Default value	Range of values
boolean	n/a	false	true or false
byte	8 bits	0	-128 to 127
char	16 bits	(unsigned)	\u0000' \u0000' to \uffff' or 0 to 65535
short	16 bits	0	-32768 to 32767
int	32 bits	0	-2147483648 to 2147483647
long	64 bits	0	-9223372036854775808 to 9223372036854775807
float	32 bits	0.0	1.17549435e-38 to 3.4028235e+38
double	64 bits	0.0	4.9e-324 to 1.7976931348623157e+308

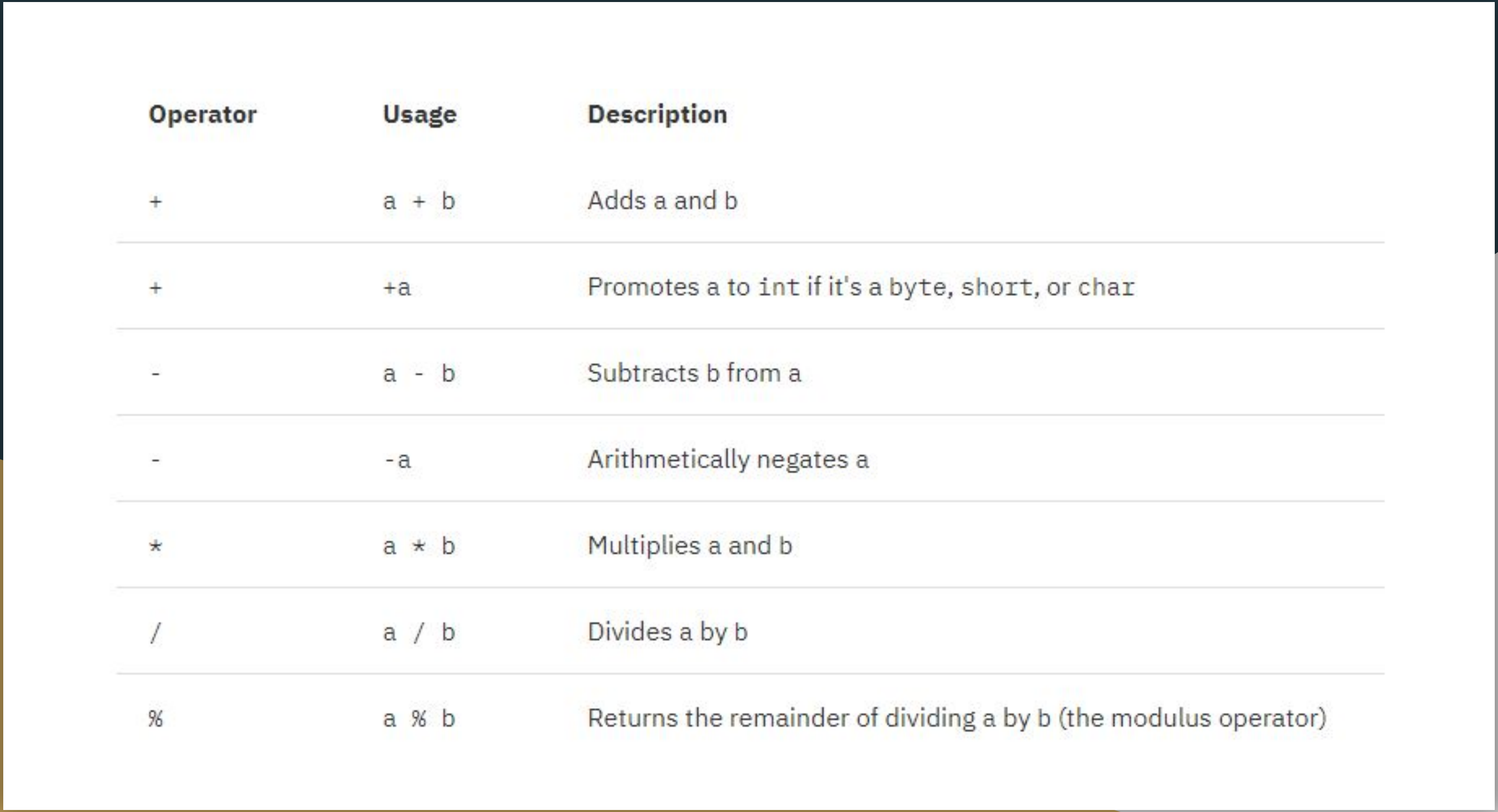
Strings.

```
String greeting = new String(“hello”) ;

String greeting2 = “Hello” ;

String concatenar = “Hello“ + “There” ;
```

Tipos de variable.

- Locales
- Estáticas (o de clase)
- De instancia.

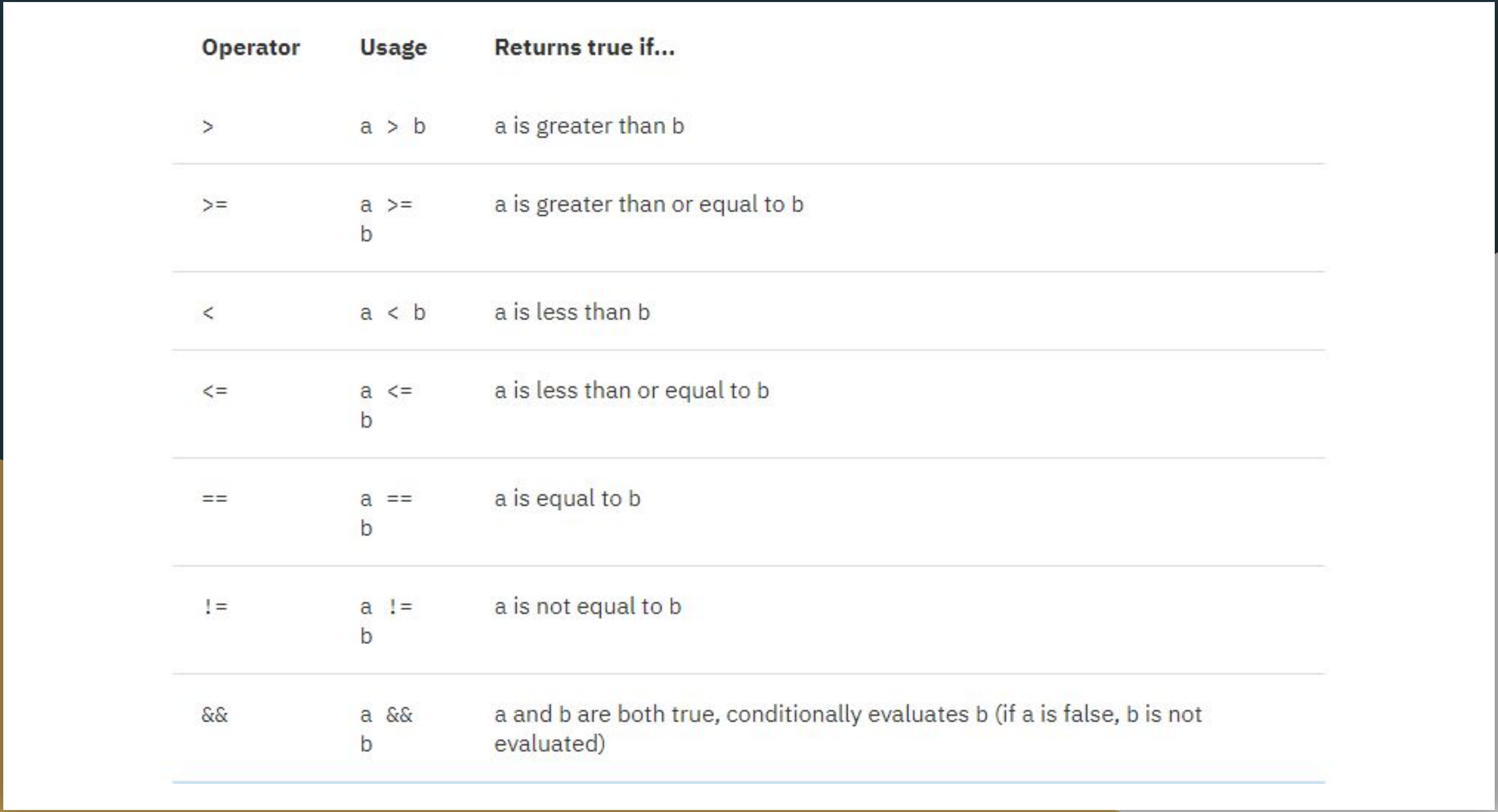
Operadores.



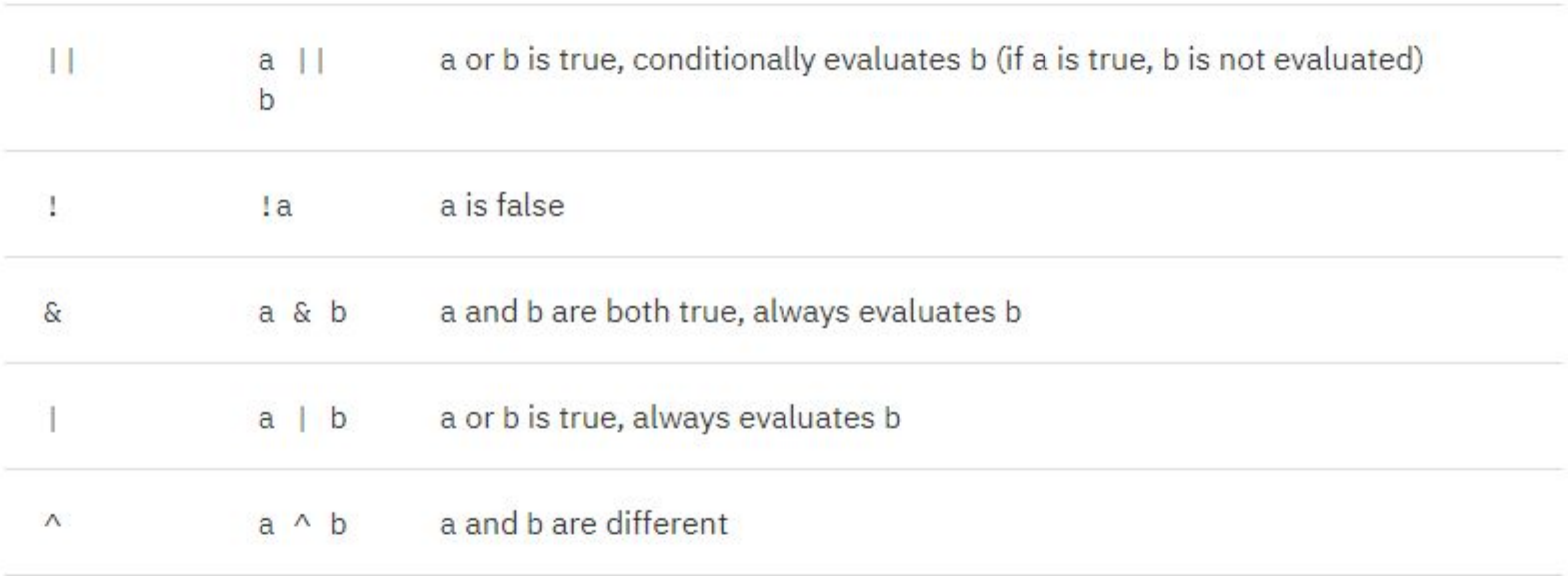
Operator	Usage	Description
`+`	`a + b`	Adds a and b
`+`	`+a`	Promotes a to `int` if it's a `byte`, `short`, or `char`
`-`	`a - b`	Subtracts b from a
`-`	`-a`	Arithmetically negates a
`*`	`a * b`	Multiplies a and b
`/`	`a / b`	Divides a by b
`%`	`a % b`	Returns the remainder of dividing a by b (the modulus operator)

`++`	`a++`	Increments a by 1; computes the value of a before incrementing
`++`	`++a`	Increments a by 1; computes the value of a after incrementing
`--`	`a--`	Decrements a by 1; computes the value of a before decrementing
`--`	`--a`	Decrements a by 1; computes the value of a after decrementing
`+=`	`a += b`	Shorthand for `a = a + b`
`-=`	`a -= b`	Shorthand for `a = a - b`
`*=`	`a *= b`	Shorthand for `a = a * b`
`%=`	`a %= b`	Shorthand for `a = a % b`

Operadores de condición y relación.



Operator	Usage	Returns true if...
`>`	`a > b`	a is greater than b
`>=`	`a >= b`	a is greater than or equal to b
`<`	`a < b`	a is less than b
`<=`	`a <= b`	a is less than or equal to b
`==`	`a == b`	a is equal to b
`!=`	`a != b`	a is not equal to b
`&&`	`a && b`	a and b are both true, conditionally evaluates b (if a is false, b is not evaluated)



`||`	`a || b`	a or b is true, conditionally evaluates b (if a is true, b is not evaluated)
`!`	`!a`	a is false
`&`	`a & b`	a and b are both true, always evaluates b
`|`	`a | b`	a or b is true, always evaluates b
`^`	`a ^ b`	a and b are different

Ciclos repetitivos.

- Ciclo for
    - `for (int aa = 0; aa < 3; aa++)  { // hacer algo }`
- Ciclo while
    - `while (condition) { //hacer algo }`
- Ciclo do-while
    - `do { //hacer algo } while (condition) ;`

Switch

```
switch (variable or an integer expression)
{
     case constant:
     //Java code
     ;
     case constant:
     //Java code
     ;
     default:
     //Java code
     ;
}
```

Arreglos.

```
int[] integers = new int[5]; // crea un arreglo de 5 elementos enteros

int[] integers = new int[] { 1, 2, 3, 4, 5 }; // crea e inicializa un arreglo de
5 elementos enteros

int[] integers = { 1, 2, 3, 4, 5 }; // crea e inicializa un arreglo de 5
elementos enteros

int element = arrayName [elementIndex];

int arraySize = arrayName.length;
```

Salida por pantalla.

```
System.out.println("This is my
first program in java");
```

Ingreso por teclado.

```java
Scanner scan = new Scanner(System.in);
int num = scan.nextInt();
scan.close();
```

Buenas prácticas.

- Clases pequeñas.
- Nombrar métodos con cuidado.
- Métodos pequeños
- Usar comentarios
- Usar un estilo consistente

Fuente: https://www.ibm.com/developerworks/java/tutorials/j-introtojava1/