



# Java™ Envoltorio



Octavio Robleto



octavio.robleto@gmail.com

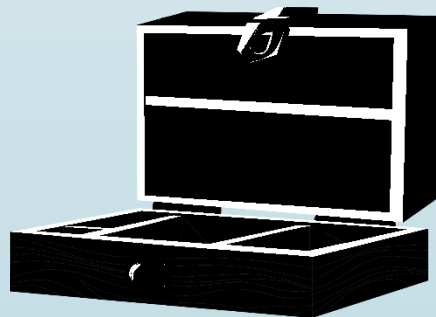


<https://octaviorobleto.com>



# Wrapper o clases “Envoltorio” en JAVA

- Muchas veces nos encontramos con problemas de conversión de tipos de datos, principalmente cuando le solicitamos datos al usuario el API de java nos provee clases útiles para manejar estas situaciones.
- Proveen un mecanismo para “envolver” las variables de tipo primitivas en un objeto para que estos puedan ser incluidos en actividades que estén reservadas para objetos, lo que permite usar variables como objetos.
- Además nos dan un variedad de funciones de utilidad a los primitivos. La mayoría de estas funciones están relacionadas con varias conversiones.



# Tipos

Primitivo	Envoltorio	Constructor
<b>byte</b>	Byte	byte o String
<b>short</b>	Short	short o String
<b>int</b>	Integer	int o String
<b>long</b>	Long	long o String
<b>boolean</b>	Boolean	boolean o String
<b>float</b>	Float	float o String
<b>double</b>	Double	double o String
<b>char</b>	Character	char



# Constructores

- Todas las clases wrapper excepto Character proveen de 2 constructores: uno que toma un primitivo del tipo que está siendo construido, y otro que toma un String como representación del tipo que está siendo construido.

```
Integer i1 = new Integer(42);  
Integer i2 = new Integer("42");  
Integer i3 = 42;
```



# Los Métodos `valueOf()`

- Los 2 (normalmente son 2) métodos **`valueOf()`** proveídos en la mayoría de las clases wrapper nos dan otro enfoque para poder crear objetos wrapper. Ambos métodos cogen una representación String del tipo apropiado de primitivo como primer argumento, y el segundo método (cuando se provee) coge un argumento adicional, `int radix`, el cual indica en que base será representado el primer argumento:

```
// Convierte el 10 (Binario) a 2 (Decimal) y le asigna el valor de 2 al objeto  
// Integer i4  
Integer i4 = Integer.valueOf("10", 2);  
System.out.println(i4);  
  
// le asigna el valor de 10 al objeto Integer i5  
Integer i5 = Integer.valueOf("10");  
System.out.println(i5);
```



# Los Métodos Value()

- Cuando necesitamos convertir un valor de un contenedor numérico a un primitivo, usamos uno de los métodos **xxxValue()**. Todos estos métodos en esta familia son métodos que no necesitan argumentos.

```
// usamos el metodo para obtener el valor primitivo del objeto i5  
int i6 = i5.intValue();  
System.out.println(i6);
```



# Los Métodos Parse()

- con este método, convertimos una cadena de caracteres, un String, en un dato numérico. Lo podemos convertir en un número entero (int), sin decimales, o en un número con decimales del tipo double o float, u otro.
- Xxx.parseXxx()

```
// convertimos un String a un entero  
int i7 = Integer.parseInt("85");  
System.out.println(i7);
```

