

Clases Wrapper en Java

¿Qué son las clases Wrapper?

Para todos los tipos de datos primitivos, existen unas clases llamadas **Wrapper**, también conocidas como **envoltorio**, ya que proveen una serie de mecanismos que nos permiten envolver a un tipo de dato primitivo permitiéndonos con ello el tratarlos como si fueran objetos.

¿Por qué no trabajamos directamente con Wrappers?

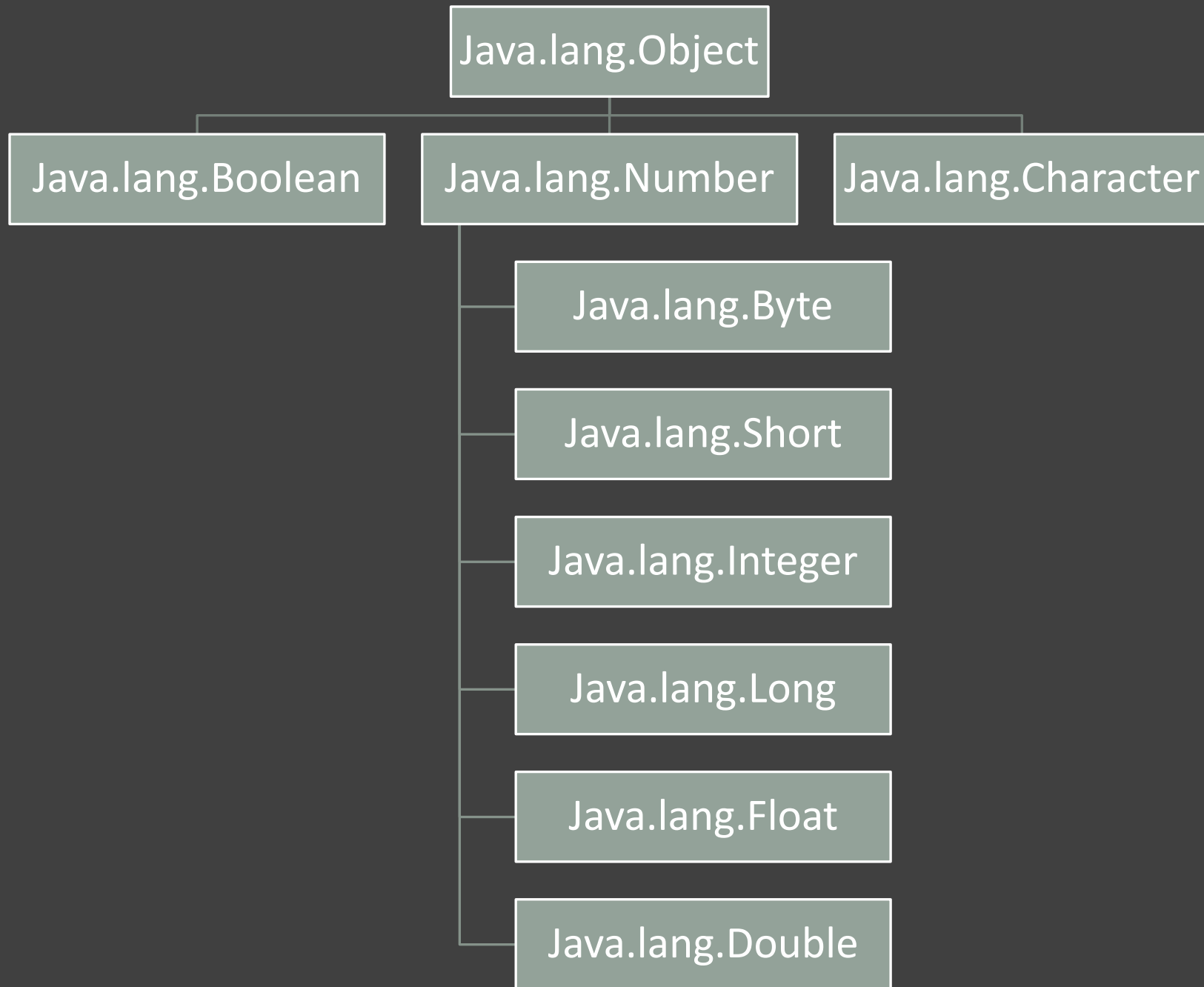
Los primitivos han estado presentes desde los orígenes de Java en 1996. A día de hoy **los tipos primitivos siguen utilizándose principalmente por el notable rendimiento que ofrecen**.

Aunque es un tema que ha generado una gran polémica principalmente por los detractores de los tipos primitivos como, por ejemplo, uno de los más famosos, Simon Ritter quien quería eliminarlos en el lanzamiento de la versión 10 de JDK desde hace tiempo.

Tipos de wrappers en Java

Existen 8 tipos primitivos predefinidos en Java, cada uno de los tipos de tipos de datos primitivos tienen asociados su correspondiente clase **Wrapper** para poder realizar las conversiones de tipos primitivos a objetos (tipos no primitivos).

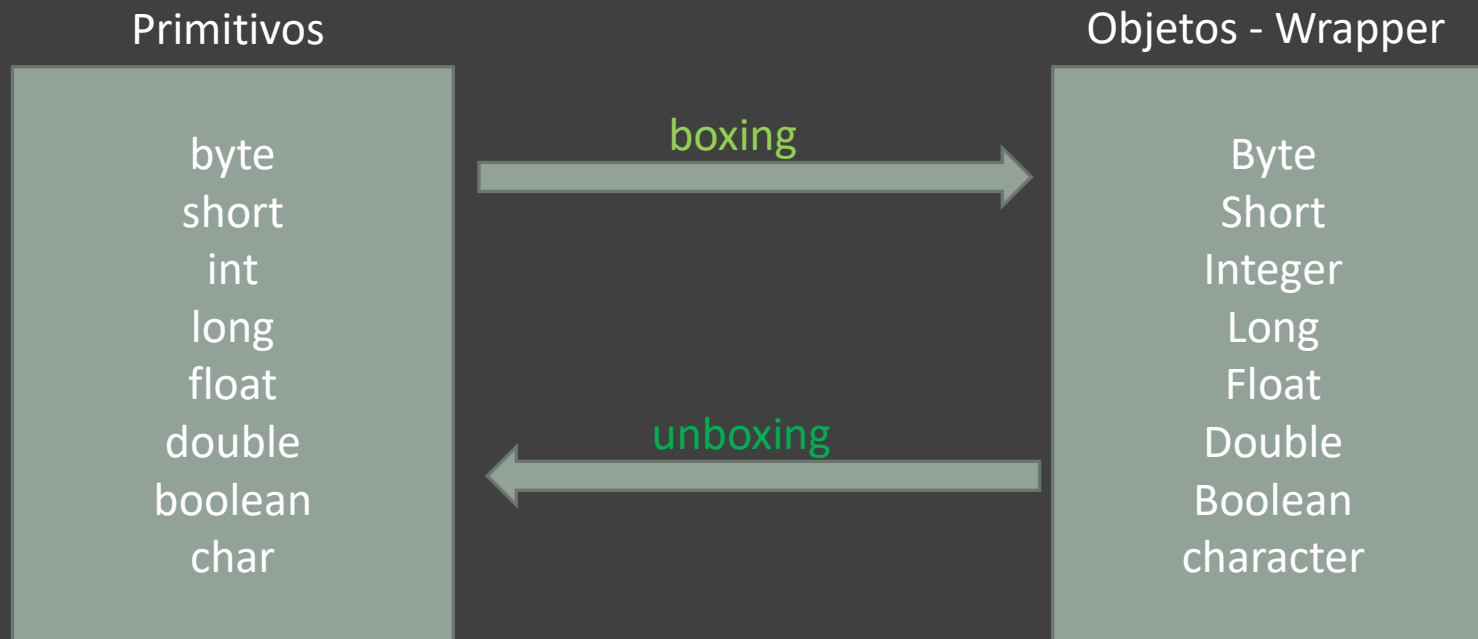
Tipos primitivos (no son objetos y por tanto no poseen métodos)	Wrappers (son objeto y por tanto poseen métodos)
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean



Jerarquía
de las clases
Wrapper

Wrapeando vs Deswrapeando

Dependiendo del punto sobre el que partimos: tipo de dato primitivo o no primitivo, y por tanto, considerado como un objeto; podemos realizar dos tipos de conversiones:



Autoboxing (Wrapping automático)

Es el proceso de conversión **automática** que realiza el compilador de Java para que un tipo primitivo pase a ser un objeto utilizando para ello su clase de envoltura ("Wrapper").

Aunque existen otras formas de realizar el **autoboxing**, el proceso más sencillo sería:

```
Character chWrapped = 'a'; //Autoboxing
```

```
Integer intWrapped = 100; //Autoboxing
```

Lo que realmente hace java internamente cuando realizamos una **autoboxing** es una especie de conversión (casteo) hacía un objeto :

```
char ch = 'a';
```

```
Character chWrapped = (Character) ch;
```

Auto Unboxing (Deswrapping automático)

Es el proceso de conversión **automática** que realiza el compilador de Java para que un objeto Wrapper pase a ser un tipo primitivo perdiendo con ello sus métodos.

El **auto unboxing** tiene lugar cada vez que un objeto se debe convertir a un tipo primitivo.

Por lo tanto, el **auto unboxing** puede ocurrir:

1 - Cuando asignamos un tipo primitivo a un objeto wrapper.

```
Integer miEnteroObjeto = 2;
```

```
int miEnteroPrimitivo = miEnteroObjeto; //Auto Unboxing
```

2 - Cuando pasamos un objeto wrapper como argumento a una función que espera un primitivo de ese tipo.

Boxing (Wrapping manual)

El proceso de conversión **no automático** que realizamos con el fin de pasar un tipo primitivo a un objeto mediante su clase de envoltura (“Wrapper”):

```
int numPrimitive = 6;
```

```
Integer numWrapper = Integer.valueOf(numPrimitive); //Boxing
```


Unboxing (Deswrapping manual)

El proceso de conversión **no automático** que realizamos con el fin de pasar un tipo no primitivo (Wrapper) a un tipo primitivo.

```
int numPrimitive = 6;
```

```
Integer numWrapper = Integer.valueOf(numPrimitive); //Boxing
```

```
int numUnWrapped = numWrapper.intValue(); //Unboxing
```