



Java™ Enum



Octavio Robleto



octavio.robleto@gmail.com



<https://octaviorobleto.com>



Enum

- **Es una lista de constantes con nombre que definen un nuevo tipo de datos.** Un objeto de un tipo de enumeración solo puede contener los valores definidos por la lista. Por lo tanto, una enumeración le brinda una manera de definir con precisión un nuevo tipo de datos que tiene **un número fijo de valores válidos**

```
// creamos un Enum local
enum ColoresBandera {
    AZUL, BLANCO;
}
```

- **Como se utiliza?**

```
ColoresBandera color = ColoresBandera.AZUL;
System.out.println(color);
```



Enum como Clases

- **Un tipo enumerado puede ser declarado dentro o fuera de una clase**, pero no dentro de un método. Por tanto no podemos declarar un enum dentro de un método main (programa principal); si lo hacemos nos saltará el error de compilación "enum types must not be local" (los tipos enumerados no deben ser locales a un método)

Constructor privado

```
EnumDiaSemana diaSemana  
= new EnumDiaSemana();
```

```
public enum DiaSemana {  
    //Constantes con constructor  
    LUNES("Lun.",1),  
    MARTES("Mar.",2),  
    MIERCOLES("Mie.",3),  
    JUEVES("Jue.",2),  
    VIERNES("Vie.",2),  
    SABADO("Sab.",2),  
    DOMINGO("Dom.",2);  
  
    private final String abreviatura;  
    private final int numero;  
    /**  
     * @author <a href="mailto:octavio.robledo@gmail.com">Ing. Octavio Robledo</a>  
     * @date 18/09/2020  
     * @param abreviatura  
     * @param numero  
     * Este constructor debe ser privado ya que no se debe ni puede instanciar el objeto EnumDiaSemana  
     */  
    private DiaSemana(String abreviatura, int numero) {  
        this.abreviatura = abreviatura;  
        this.numero = numero;  
    }  
  
    public String getAbreviatura() {  
        return abreviatura;  
    }  
  
    public int getNumero() {  
        return numero;  
    }  
}
```



Una de sus Ventajas

- Se pueden usar en switch

```
DiaSemana diaSemana = DiaSemana.DOMINGO;

switch (diaSemana) {
case LUNES:
case MARTES:
case MIERCOLES:
case JUEVES:
case VIERNES:
    System.out.println("Dia Hábil");
    break;
case SABADO:
case DOMINGO:
    System.out.println("Dia No Hábil");
    break;
}
```



Métodos



- **values():** El método se puede usar para devolver todos los valores presentes dentro de enum.
- **ordinal():** el orden es importante en las enumeraciones. Al usar el método se puede encontrar cada índice de la constante enum, al igual que el índice de un arreglo.
- **valueOf():** El método devuelve la constante enum del valor de cadena especificado si existe, de lo contrario mostrara un error.

