Curso Elixir

Sección 3

Operadores básicos

Anteriormente vimos los operadores `+, -, *, /, ++, --, las funciones div/2 y rem/2`

A continuación veremos algunos otros operadores básicos:

Strings

La concatenación de cadenas se realiza con <>

```
iex> "foo" <> "bar"
"foobar"
```

Operadores Boleanos

or, and y not

```
iex> true and true
true
iex> false or is_atom(:example)
true

iex(1)>  1 and true
** (BadBooleanError) expected a boolean on left-side of "and", got: 1
```

Además de estos operadores booleanos, Elixir también proporciona ||, && y ! que aceptan argumentos de cualquier tipo.

```
# or
iex> 1 || true
1
iex> false || 11
11

# and
iex> nil && 13
nil
iex> true && 17
17

# !
iex> !true
false
iex> !1
false
iex> !nil
true
```

como regla usa and, or o not cuando los parámetros sean boleanos. Y si alguno de los parámetros es de cualquier tipo usa &&, || o !

Elixir también provee los operadores de de comparación ==, !=, ===, !==, <=,>=, < y >

```
iex> 1 == 1
true
iex> 1 != 2
true
iex> 1 < 2
true
```

La diferencia de los comparadores == y === es que la última es más estricta al comparar enteros y flotantes

```
iex> 1 == 1.0
true
iex> 1 === 1.0
false
```

En elixir es posible comparar diferentes tipos

```
iex> 1 < :atom
true
```

La razón por la que podemos comparar diferentes tipos de datos es el pragmatismo. Los algoritmos de clasificación no necesitan preocuparse por los diferentes tipos de datos para ordenar. Este es el orden de clasificación general:

```
number < atom < reference < function < port < pid < tuple < map < list < bitstring
```