

Curso Elixir

Sección 4

Keyword lists

En Elixir, una lista de palabras clave es una lista especial de tuplas de dos elementos, cuyos primeros elemento son átomos; éstas tienen el mismo funcionamiento que las listas:

```
iex> [foo: "bar", hello: "world", number: 1, number: 5.6]
iex> [{:foo, "bar"}, {:hello, "world"}]
```

Las tres características de las listas de palabras clave que resaltan su importancia son:

- Las claves son átomos.
- Las claves están ordenadas.
- Las claves pueden no ser únicas.

Es por esto que las listas de palabras clave son comúnmente usadas para pasar opciones a funciones.

Para acceder a los valores de Keyword lists usamos

```
iex> list = [foo: "bar", hello: "world", number: 1, number: 5.6]
iex> list[:foo]
"bar"
```

Maps

Los mapas son el tipo de datos utilizado por excelencia para almacenar pares de clave/valor. A diferencia de las listas de palabras clave, los mapas permiten claves de cualquier tipo y no mantienen un orden.

```
iex> map = %{:foo => "bar", "hello" => :world}
```

o

```
iex> key = "hello"
iex> %{key => "world"}
%{"hello" => "world"}
```

Si un elemento duplicado es agregado al mapa, este reemplazará el valor anterior:

```
iex> %{:foo => "bar", :foo => "hello world"}
%{:foo: "hello world"}
```

Como podemos ver en la salida anterior, hay una sintaxis especial para los mapas que solo contienen átomos como claves:

```
iex> %{foo: "bar", hello: "world"}  
%{foo: "bar", hello: "world"}  
iex> %{foo: "bar", hello: "world"} == %{foo => "bar", :hello => "world"}  
true
```

Para acceder a los valores de los mapas podemos usar:

```
iex> map = %{foo => "bar", :bar => "hello world", "hello"=> "Hello"}  
iex> map["hello"]  
iex> map.foo
```

solo podemos usar la sintaxis map.foo en caso de que la llave sea un atom.

Otra característica interesante de los mapas es que poseen su propia sintaxis para realizar operaciones de actualización:

```
iex> map = %{foo => "bar", :bar => "hello world"}  
iex> %{map | foo: "baz"}
```