

Curso Elixir

Sección 8

Ciclos a través de la recursividad

Debido a la inmutabilidad de las variables en Elixir, la forma de escribir ciclos cambia en comparación con lenguajes imperativos como php.

```
for ($i = 1; $i <= 10; $i++) {  
  echo $i;  
}
```

En el código anterior, mutamos la variable `i` y como ya hemos visto en Elixir no es posible realizar esto. En lugar de esto podemos utilizar la recursividad.

Veamos a continuación un ejemplo de como usar recursividad y crear una función que sume los valores de una lista.

```
defmodule Math do  
  def sum_list(list) do  
    sum_list(list, 0)  
  end  
  
  def sum_list([head | tail], accumulator) do  
    sum_list(tail, head + accumulator)  
  end  
  
  def sum_list([], accumulator), do: accumulator  
end  
  
iex> Math.sum_list([1,2,3,4,5])  
15
```

Invocamos `sum_list` con la lista `[1, 2, 3, 4, 5]` como argumentos e invocamos `sum_list` con la lista y `0` como argumentos nuevamente. Intentaremos cada cláusula hasta encontrar una que coincida de acuerdo con las reglas de coincidencia de patrones. En este caso, la lista `[1, 2, 3, 4, 5]` coincide con `[head | tail]` que une `head` a `1` y `tail` a `[2, 3, 4, 5]`. Y el acumulador se establece en `0`.

Luego, agregamos el encabezado de la lista al `head + acumulador` y llamamos a `sum_list` nuevamente, recursivamente, pasando el final de la lista como su primer argumento. La cola volverá a coincidir `[head | tail]` hasta que la lista esté vacía, como se ve a continuación:

```
sum_list [1, 2, 3, 4, 5]  
sum_list [1, 2, 3, 4, 5], 0  
sum_list [2, 3, 4, 5], 1  
sum_list [3, 4, 5], 3  
sum_list [4, 5], 6  
sum_list [5], 10  
sum_list [], 15
```

Cuando la lista esté vacía coincidirá con `sum_list([], accumulator)` y regresará el acumulador.

Reduce y algoritmos map

El módulo Enum, que veremos en el próximo capítulo, ya ofrece muchas funciones para trabajar con listas. Por ejemplo, el ejemplo anterior podrían escribirse como

```
iex> Enum.reduce([1, 2, 3, 4, 5], fn(x, acc) -> x + acc end)
```

Actividad 5

Edita el módulo Math y agrega una función para calcular el factorial de un número