Loops and Recursion

Let's take a look at the basic loop. This is very similar to the way Python does things.

```
# iex

for x <- [1, 2, 4] do
   IO.puts x
end
```

Notice the output. Not only does it do the prints, it also returns all the returned values in a string.

```
for x <- [1, 2, 4] do
   x + 1     # Automatic collection (comprehensions in Python)
end
```

The usual loop from one to 10 is through a similar syntax:

```
for x <- 1..10 do
  x + 1
end
```

The Elixir Way

Let's try to sum the elements of a list in the Elixir way.

```
# File: 07loop.exs
defmodule Math do
  def sum_list([], accumulator) do
    accumulator
  end

  # Recall: [head | tail] = [1, 2, 4]

  def sum_list([head | tail], accumulator) do
    sum_list(tail, head + accumulator)
  end
end


IO.puts Math.sum_list([1, 2, 4], 0)    # second arg necessary
IO.puts Math.sum_list([], 0)
```

This is extremely powerful and makes your code much more readable once you understand it. Also, it's much easier to test since all functions are independent of each other.