Elixir Solution for Checking if an Integer is a Palindrome

To implement a function in Elixir that checks if an integer is a palindrome,
the following steps are used:

1. Negative numbers are not palindromes, so immediately return false for them.
2. Convert the integer to a list of its digits.
3. Check if this list of digits is the same forwards and backwards.

Elixir is a functional programming language, thus utilizing functions like Integer.digits/1 for conversion and Enum.reverse/1 for list manipulation provides a clean and effective solution.

Here is the Elixir code for the solution:

```
defmodule Palindrome do
  def is_palindrome(number) when number < 0, do: false
  def is_palindrome(number), do: number |> integer_to_list() |> is_list_palindrome()

  defp integer_to_list(number), do: Integer.digits(number)

  defp is_list_palindrome(list) do
    list == Enum.reverse(list)
  end
end
```

This module, Palindrome, provides the is_palindrome/1 function, which determines if the given integer is a palindrome or not.