My first report.

Tomás Belmar da Costa

Spring Term 2022

Introduction

For my first Elixir program I decided to make a module called Arithmetic. This module includes the following simple arithmetic functions:
- add(a, b)
- subtract(a, b)
- multiply(a, b)
- divide(a, b)
- pow(a, b)

Setting up the File

I began by creating a file `Arithmetic.ex` by typing

```
defmodule Arithmetic do

end
```

Creating the functions

The add, subtract, multiply, and divide functions were all relatively straightforward. All I had to do was define a function with their names that takes two parameters as inputs, and then use the +, -, *, and / operators built into elixir, as such.

```
  def add(a, b) do
    a + b
  end

  def subtract(a, b) do
    a - b
  end
```

```
def multiply(a, b) do
  a * b
end

def divide(a, b) do
  a / b
end
```

I confirmed that these functions worked by importing the module with the command `c "Arithmetic.ex"` and calling `Arithmetic.add(x, y)` to check that the expected results were being output.

Building the `Arithmetic.pow()` function was a little more difficult however since there is no built in power function in Elixir. I solved this recursively like so:

```
def pow(_, 0) do 1 end

def pow(a, b) do
  a * pow(a, b-1)
end
```

The base case accepts any argument that's being raised to the power of 0. That will always return 1. In any other case, the function will recursively multiply the input number by itself $\widehat{b}$ times. This power function only works for positive powers however.

I once again tested this with several numbers and confirmed that it does in fact work.