Introduction

As reference, we use this doc, so if your team have any doubt, don't hesite in jumping in the docs.

Exercises

The basic structure of file to testing a modules is:

```
defmodule AssertionTest do
  use ExUnit.Case

  test "the truth" do
    assert true
  end
end
```

We recommend to you create this file and execute it. To do this, use

```
$ elixir assertion_test.exs
```

1. Look at this function:

```
def sum_list list, start_value
  Enum.reduce(list, start_value, fn(x) -> start_value += list end)
end
```

Create a test suite testing if it works for the following cases:

```
list = [1,2,3,4], start_value = 0
list = [1], start_value = 1
list = [0], start_value = 10
```

2. I have a challenge for you. This test will pass or not? And why?

```
test "???" do
  refute 1 + 1 == 3
end
```

3. This exercises is special. What we, as developers are looking for when we create code to test our own code? Answer this before keep going? So, looking at this point of view, I'll give a piece of code and you'll create a few testes on top of this and evaluate it. What do you think about it?

So, let's describe the problem. Suppose you're a Biologist and need a program to count nucleotids in a DNA sequence. Pretty simple, right?

```
def compare(a, b) do
  if a == b, do: 1, else: 0
end

def count(dna, exp) do
  case dna do
    [ head | tail ] -> count(tail, exp)
    '' -> 0
  end
end
```

This code work when in this test code:

```
test "empty dna string has no adenine" do
  assert NucleotideCount.count('', ?A) == 0
end
```

But we all know this isn't only test case it should pass. Instead of that, let's a few test cases. I already described it for you:

```
test "repetitive cytosine gets counted" do
  ...
end

test "counts only thymine" do
  ...
end
```

And know, if we look in our code, it ins't passing. Fix it.

When I test are implemented and passing, you should improve your code. What I mean? There's a function already implemented by standard lib of Elixir that will make our code much cleaner. Go ahead and improve the code!

docs