04 Tasks and Agents

Agent

In these exercises you, are provided with an implementation of `Stack` module, as described during the presentation. Use the provided functions to familiarize yourself with the process of setting up a stack, pushing and popping new values.

Exercise

Based on the `Stack` module implementation, create module called `Counter` with the following functions:

- increment/0 - increments current counter value by 1
- decrement/0 - decrements current counter value by 1

Useful tips

When using the function `Agent.start_link/2` to start a new Agent process, you can pass a second argument, `name: __MODULE__`, to register the `Counter` module in the Registry. This allows you to use Agent functions like: `Agent.get(__MODULE__, fun)` without needing to remember the PID of the Agent process.

Question

- If we use `name: __MODULE__` instead of relying on the PID of the Agent, can we start multiple Agents? Why or why not?
- If we rely on the PID of the Agent process, can we start multiple processes at the same time?

```
defmodule Stack do
  def new(), do: Agent.start_link(fn () -> [] end)

  def stop(stack_pid), do: Agent.stop(stack_pid)

  def push(stack_pid, value) do
    Agent.update(stack_pid, fn (stack) -> [value | stack] end)
  end

  def pop(stack_pid) do
    Agent.get_and_update(stack_pid, fn ([head | tail]) ->
      {head, tail}
    end)
  end
end
```

Tasks

The `Task` module contains functions for spawning separate process to perform computations and awaiting their results. A key advantage of using tassks is that they can be supervised.

Exercise

Using provided implementation of the `Zoo` module, within the `list_animals/0` function, spawn new a new task and await its result. The task should iterate over list `@mammals` and apply `String.capitalize/1` to each element. While waiting for the task to complete, capitalize the elements of the `@birds` lists. Finally, return the concatenated lists.

```
defmodule Zoo do
  @mammals ["monkey", "bear", "giraffe", "penguin"]
  @birds ["eagle", "vulture", "condor"]

  def list_animals do
    ### Your code here
  end
end

Zoo.list_animals()
```