Creating a module in Julia

Related functions and descriptions can be added to a “module” in Julia to be used later.

```
module MyModule


end
```

In between these lines, you write your function definitions so that they can later be called by “importing” this module.

When you write your own modules, the functions that you choose to export can be used **without** the module name as prefix (this will be explained in a later section below). Those that you don't export can still be used, but only if they are prefixed with the module name. For example:

```
module MyModule
export my_awesome_function

function my_awesome_function()
  println("Hello World")
end

function my_secret_function()
  println("Hello from the other side!")
end

end
```

Once you have put all your function definitions in this way, save this file by any name `MyModule.jl`. It is advisable to use the name of the module to name the file, just for the sake of code readability.

After creating this file, there is one more thing you need to do before Julia can use it.

Adding to PATH

Julia should be able to find your module file!
When you load Julia, it searches a list of path stored in the LOAD_PATH variable.

```
julia> LOAD_PATH
3-element Array{String,1}:
 "@"
 "@v#.#"
 "@stdlib"
```

This above is what LOAD_PATH has by default on my system.

So, you first need to add whatever path your `MyModule.jl` file is located in to the LOAD_PATH variable.

```
push!(LOAD_PATH, "/path/of/folder/containing/module/file")
```

You have to do this **every time** you open Julia.

Exporting functions

You can really use either of `import` or `using` to use a module in Julia. There are minor differences between the two. Remember that we exported `my_awesome_function`, but not `my_secret_function`.

So, if you load `MyModule` using `using`, this will happen:

```
julia> using MyLearn2ClassifyManyAlgs

julia> my_awesome_function()
Hello World

julia> my_secret_function()
ERROR: UndefVarError: my_secret_function not defined
```

```
Stacktrace:
 [1] top-level scope at none:0

julia> MyLearn2ClassifyManyAlgs.my_secret_function()
Hello from the other side!
```

Since we didn't export `my_secret_function`, it has to be prefixed with the module name, otherwise Julia won't find it.

On the other hand, if you load `MyModule` using `import`, this will happen:

```
julia> import MyLearn2ClassifyManyAlgs

julia> my_awesome_function()
ERROR: UndefVarError: my_awesome_function not defined
Stacktrace:
 [1] top-level scope at none:0

julia> MyLearn2ClassifyManyAlgs.my_awesome_function()
Hello World

julia> my_secret_function()
ERROR: UndefVarError: my_secret_function not defined
Stacktrace:
 [1] top-level scope at none:0

julia> MyLearn2ClassifyManyAlgs.my_secret_function()
Hello from the other side!
```

Here, every function needs to be prefixed with the module name!

Adventuring

Now, since you all learned a lot of functions in the class and the homeworks, try building your own module and calling it inside your own Julia instance!