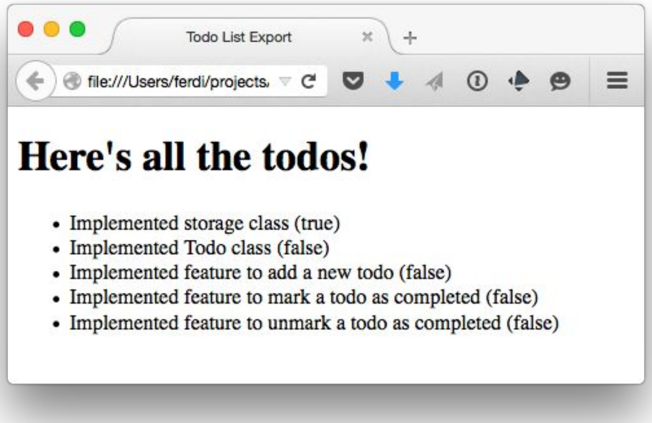
Exercises - Session 15

In case you get stuck anywhere, don't be afraid to ask the coaches! They are here to help and will gladly explain everything to you!
Take notes during the exercises. Even if you never look at them again, they will help you memorise things!

Todo List HTML Exporter

1. Create an HTML document that includes some todos by hand. When you open the file in the browser, it could look like this, for example:



Here's all the todos!

- Implemented storage class (true)
- Implemented Todo class (false)
- Implemented feature to add a new todo (false)
- Implemented feature to mark a todo as completed (false)
- Implemented feature to unmark a todo as completed (false)

2. Find the place in the code where the `CsvExporter` is used. Think about the interface (common methods) all of the exporter classes need to have to be used interchangeably.

3. Create a new `HtmlExporter`, similar to the `CsvExporter`, which generates the HTML file you created.

4. Integrate the new HTML export you created into the existing app so it is callable similar to the CSV export. The end result should look like this:

```
% ruby todo_list.rb
Welcome to the todo_list app!
Type help for listing all commands.
> export
[TODOLIST] export...
Please enter an export format (csv, html): html
Please enter a file name: export.html
Written HTML export to file export.html.
> exit
```

Optional Part

1. Think of last week's exercise. The RSS feed is a form of XML. How does it compare to CSV? What are the advantages and disadvantages of each of the data formats?

2. Can you define a todo list export in XML? What could it look like?