Week 4 - Exercises

Getting Started

1. Your project is ready to have PHP added to it. In order to do so, download the `install-composer.sh` script to your Downloads directory

2. Execute the following command in iTerm:

```Bash
sh ~/Downloads/install-composer.sh
```

3. All exercises from this workbook should be complete in a self-contained directory:

```
~/projects/deeper/exercises/week-04/workbook
```

So make sure you `cd` into it in iTerm before running any `composer install` commands!

Exercises

1. Create a photo submission form using Bootstrap. Fields should include:

   1. Name
   2. Date of Birth
   3. Email Address
   4. Picture

2. Create a PHP class which represents the form, with an additional "submitted" property which stores

the date/time of when the user submitted the form. All date properties should use the [Carbon](#) library. The "picture" property should hold the path to the file once it has been uploaded.

3. On submission, hydrate an instance of your class from the form data, and store it to a file (using the "submitted" property, formatted as a timestamp, as the filename - e.g. `1598183369.txt`)

4. Images should be saved in a subfolder called "uploads". i.e:

```
1 | ~/projects/deeper/exercises/week-04/workbook/uploads
```

5. Display a Bootstrap success / danger alert, depending on whether the submission was successful.

6. Below the form, display all uploaded images in a Bootstrap carousel.

Stretch Goal

1. Using `composer` , install [Google's official ReCAPTCHA library](#).
2. Do not process the form submission unless ReCAPTCHA has verified the user.

Useful Reading

- [Bootstrap Forms](#) - Documentation on Bootstrap forms
- [Bootstrap Carousel](#) - Documentation on Bootstrap's carousel component
- [Google's ReCAPTCHA library](#) - Documentation on Google's PHP ReCAPTCHA library, with code examples
- [Getting started with Carbon](#) - Carbon documentation

  - [fileputcontents](#) - PHP documentation on writing content to a file from a string

- [filegetcontents](#) - PHP documentation on reading the contents of a file to a string
- [serialize](#) - PHP documentation on serialising data to a string
- [unserialize](#) - PHP documentation on unserialising data from a string
- [glob](#) - PHP documentation on retrieving files matching a pattern
- [scandir](#) - PHP documentation on scanning a directory