Welcome to the Menpo Playground!

This directory contains:

1. A full isolated Python installation containing the Menpo Project with all its dependencies ready to go
2. A set of notebooks that you can run immediately to see how to use Menpo
3. Two command line tools, `menpofit`, and `menpodetect`, that can be used to locate bounding boxes and landmarks in challenging in-the-wild facial images

A guide to to use each of the above follows. All of the contents of this directory are **command line tools**, and you will need to open a terminal in order to run them.

Facial landmark localization

This Playground includes pre-trained models for performing face detection (thanks to `dlib`) and facial landmark localization. These are exposed as simple command line tools, so you don't even need to know Python to use them.

To detect 68 landmarks on faces for a set of images in `~/my_images/`, run:

```
> ./menpofit ~/my_images
```

The annotations will be stored in simple `.pts` files next to each image. If more than one face is found, a number will be appended to the file stem to differentiate them. You can open these files in any text editor to extract the data.

Detecting bounding boxes alone is very similar (with the same output style), just run:

```
> ./menpodetect ~/my_images
```

Isolated Python installation

Inside the `./src` directory is a full installation of Python along with all the complex software needed to run the Menpo Project.

You can access this Python installation in two ways. Firstly, you can run `python` command in the root of this directory as follows:

```
> ./python
```

This will invoke the Python interpreter shipped with this Playground in an interactive REPL mode, and you'll see something like this:

```
Python 3.5.1 |Continuum Analytics, Inc.| (default, Dec  7 2015, 11:24:55)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

If you are new to Python, try executing the following commands, hitting enter after each one:

```
>>> print('hello from python!')
>>> x = 2
>>> y = 4
>>> z = x + y
>>> print(z)
>>> import menpo
>>> print(menpo.__version__)
```

You can of course write more involved Python scripts which perform complex computer vision tasks. Make a file called `image_info.py` in this directory and use a text editor to set the contents to the following:

```
import menpo.io as mio

img = mio.import_image('~/Downloads/my_img.jpg')
print(img)
```

Now do an Image search for an interesting picture and download it to `~/Downloads/my_img.jpg`. To run our script, we can just pass it to the `python` command as we saw earlier:

```
> python ./image_info.py
```

You should see some information about your downloaded image printed.

Example IPython Notebooks

Scripts are great for larger structured programs or for tasks that you want to leave running for a long time, but for exploratory research work we can do a lot better.

IPython Notebooks are an evolved version of the basic `python` interface we saw earlier, but instead of running in a terminal they run in the browser.

This folder includes a set of Notebooks that explain how to use much of the Menpo Project. The best way to understand Notebooks is to just start playing with them. To do so, run:

```
> ./launch_ipython_notebooks
```

Your browser should open up to a file browser GUI. If it doesn't, try going to http://localhost:8080 manually.

Go through the folders and launch the notebooks. You can browse the already run content, but the documents are also fully interactive - try executing the cells yourself by pressing `shift` + `enter`. Now edit the content of a cell and re-run it with `shift` + `enter` again. Notice how you can quickly iterate between typing code and running it - this is the beauty of the Notebook interface.