**Meerkat Software Training**

**Web Programming in Python**

**Exercise: Content and Styling**

**Primary Target:** A good place to look for tutorials and docs related to web programming is the site: [www.w3schools.com](http://www.w3schools.com/). The aim of this exercise is to become as comfortable with basic HTML and CSS files. This is all about specifying content and styling.

1. Work through the first stages of the HTML5 tutorial found at: [www.w3schools.com/html](http://www.w3schools.com/html)

HTML Introduction

HTML Basic

HTML Elements

HTML Attributes

HTML Headings

HTML Paragraphs

HTML CSS

HTML Links

2 .Work through the first stages of the CSS tutorial found at: [www.w3schools.com/css](http://www.w3schools.com/css)

CSS Introduction

CSS Syntax

CSS How To

CSS Colors

CSS Backgrounds

CSS Borders

CSS Margins

CSS Padding

CSS Height/Width

CSS Box Model

**Advanced target:** The CSS in Meerkat Frontend is assembled from SASS files. Read about SASS here: [sass-lang.com/guide](http://sass-lang.com/guide). SASS is a preprocessor that generates standard .css files from .scss files. Among many useful features, it allows us to abstract values (e.g. colours, sizes etc...) into variables. The SASS is compiled into CSS as part of our build process (we will discuss this in the next talk). You can find the .scss files in:

[meerkat\_frontend/meerkat\_frontend/src/sass].

1. Read about SASS using the link above.
2. Browse the SASS files for Meerkat Frontend and see if you can make sense of them.
3. Make some changes and see if they are visible in a web browser.

(You will need to run the commands [gulp clean] and then [gulp] in the Frontend docker container to compile the sass)

**Meerkat Software Training**

**Web Programming in Python**

**Exercise: A Meerkat Frontend Case Study**

In this exercise we will step through the process of adding a simple new web page to Meerkat Frontend. The web page will show some location specific details for a location ID number specified in the url. i.e. the location ID for Petra is 12. You will add a web page at the url [/location/12] that returns some information about Petra.

**Basic Target:** Work through the following steps...

1. Create a simple Jinja2 template.

Create a file [meerkat\_frontend/meerkat\_frontend/templates/location.html]. Initialise the file as a simple HTML page...TODO

1. Create a python flask page function.

TODO

1. View the work so far

One of the docker containers that we run in our development environment is a web server. It allows you to make http requests to your own computer at the [localhost] domain name, or the ip address [127.0.0.1]. To view the changes we've made so far, we'll need to restart the Frontend docker container by running the command [docker compose restart frontend]. Then open a web browser and type [127.0.0.1/location/12] into the address bar. If all goes well you should see a web page showing details about petra, if an error occurs, use the command [docker compose logs frontend].

1. Add some styling.

TODO

1. View the work so far

Remember that you will have to restart the front end docker container. However, this time you will also have to compile the SASS. This is done as part of the gulp build process, so you will need to run the commands [gulp clean] and then [gulp] in the Frontend docker container.

**Advanced Target:** Change the page design to match the other pages in Meerkat Frontend.

The file [meerkat\_frontend/meerkat\_frontend/templates/error.html] is a simple example of how to create a page that matches the design of the rest of the site. Merge the contents of this file into your newly created JINJA2 template so that your location page has a visual design that matches the visual design of the rest of the site.