KV6006 practical session - 5 - Extension Material

This is stuff I cut from the session because it seemed excessive, but it's still interesting and might be useful at some point. You can, however, safely ignore it if you wish.

People in space: Prettier output via a basic GUI

In worksheet 2 we retrieved data about all the people currently in space.

Not many users like viewing data in a terminal, so let's build them a GUI window. There are dozens of ways of doing this; we're going to use one of the simplest, a toolkit called GUIzero. Documentation for GUIzero is here: https://lawsie.github.io/guizero/.

Edit the program in Thonny so it looks like this (you can omit the comments if you wish):

```
# examples/peopleinspace/peopleinspace-gui-1.py
import requests
from guizero import App, Text

app = App(title="People iiiin Spaaaaaace!", height=150)

# Get data and parse it as JSON
r = requests.get('http://api.open-notify.org/astros.json')
data = r.json()

# Write some text to the GUI window
message0 = Text(app, " ", height=2) # spacer to push other text down the window a lit
message1 = Text(app, "Number of people in space: ", size=24)
message2 = Text(app, data['number'], size=48, color='red')

# Now show the window, containing the messages
app.display()
```

Run that, and (after a short time) you should see a nice neat GUI window.

If you want to expand that to include names, you'll need to add something like:

```
people_string = ""
for person in data['people']:
    people_string += person['name']
    people_string += "\r" # Add a new line
message3 = Text(app, people_string, size=14, color='blue')
```

…then you'll encounter a bug where the window isn't tall enough. But I'm sure you can fix that.

Electricity generation: drawing a pie chart with Matplotlib

You retrieved real-time electricity generation data, and output the fuel mix. Let's do the same thing… in pie chart form.

You an type this in, or see `elecgenapi-graph.py` in the examples directory.

```
# examples/elecgenapi/elecgenapi-chart.py
import matplotlib.pyplt as plt

# [...]
mix = r.json()

# Give ourselves some empty lists
fueltype = []
percentage =[]

for fuel in mix['data']['generationmix']:
    fueltype.append(fuel['fuel'])
    percentage.append(fuel['perc'])

# Set up a chart
fig1, ax1 = plt.subplots()
# Plot a pie chart of the percentage data, using fueltype as labels
ax1.pie(percentage, labels=fueltype, autopct='%1.1f%%', shadow=False, startangle=90)
ax1.axis('equal')

plt.show()
```

Run that, and after a few seconds (possibly *quite a few seconds*) you should have a pie chart. An ugly one, probably, but you can immerse yourself in the matplotlib documentation at a later date.