Adding Tables to our PDFs

Up to now, we've generated an extra simple PDF file, that just includes a title.

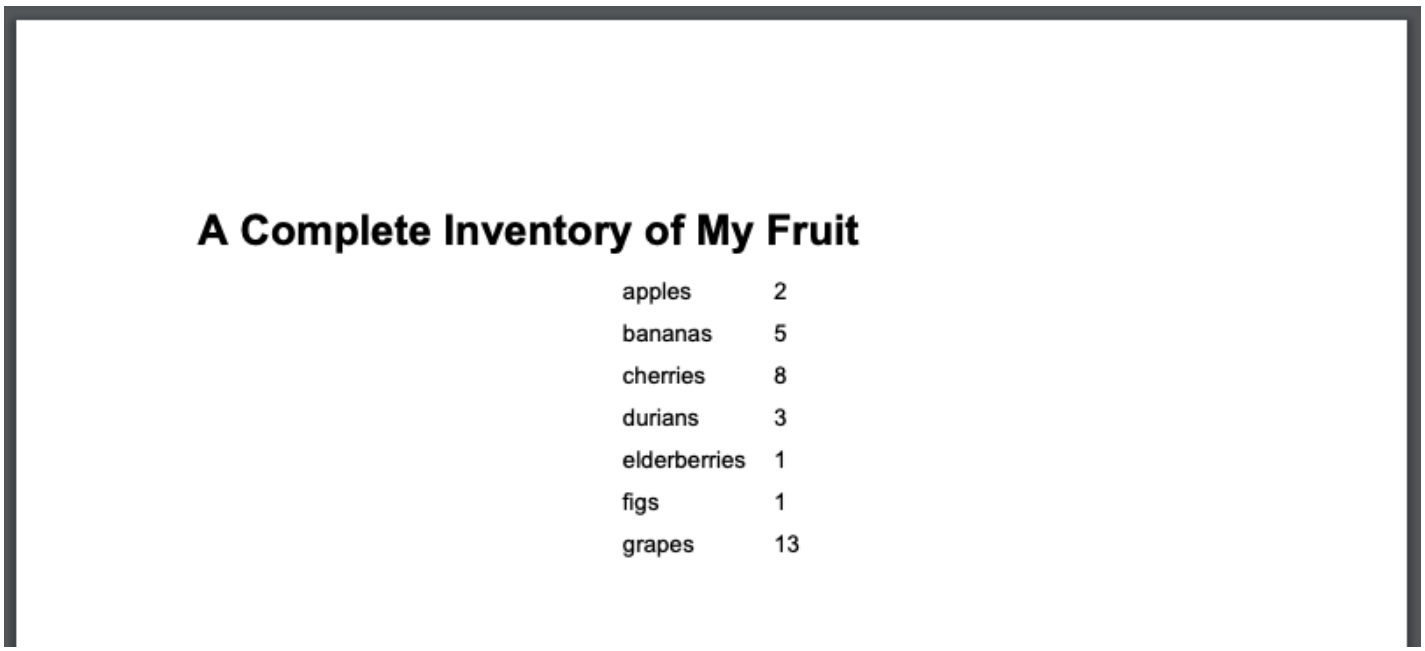
Let's spice this up by adding a ***Table***. To make a Table object, we need our data to be in a ***list-of-lists***, sometimes called a ***two-dimensional array***. We have our inventory of fruit in a dictionary. How can we convert a dictionary into a list-of-lists?

```
>>> table_data = []
>>> for k, v in fruit.items():
...   table_data.append([k, v])
...
>>> print(table_data)
[['elderberries', 1], ['figs', 1], ['apples', 2], ['durians', 3], ['bananas', 5], [
```

Great, we have the list of lists. We can now add it to our report and then generate the PDF file once again by calling the **build** method.

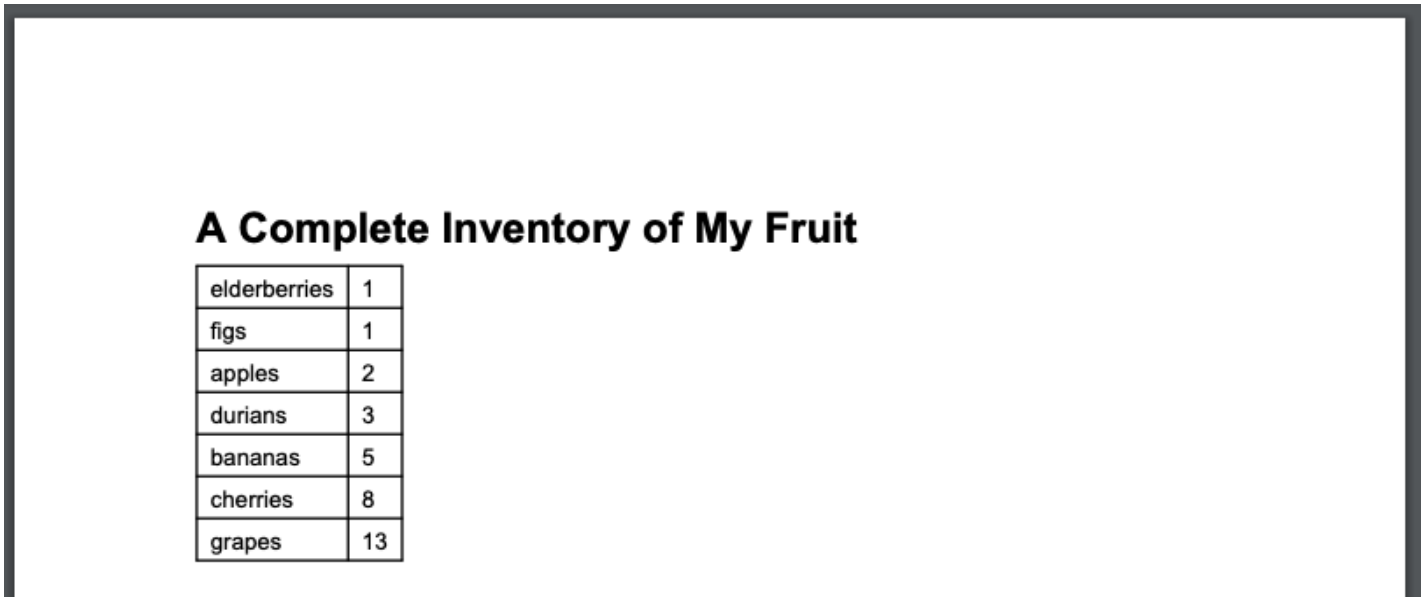
```
>>> report_table = Table(data=table_data)
>>> report.build([report_title, report_table])
```

And this is how the generated report looks now:



Okay, it worked! It's not very easy to read, though. Maybe we should add some style to **report_table**. For our example, we'll add a border around all of the cells in our table, and move the table over to the left. ***TableStyle*** definitions can get pretty complicated, so feel free to take a look at the documentation for a more complete idea of what's possible.

```
>>> from reportlab.lib import colors
>>> table_style = [('GRID', (0,0), (-1,-1), 1, colors.black)]
>>> report_table = Table(data=table_data, style=table_style, hAlign="LEFT")
>>> report.build([report_title, report_table])
```



A Complete Inventory of My Fruit

elderberries	1
figs	1
apples	2
durians	3
bananas	5
cherries	8
grapes	13

Much better! Up next, we'll look into making this more colorful by adding graphs to our reports.

✓ **Completed**