ipynb-extractor

A python program that will:
• Read a Jupyter ipynb file using simplejson module.
• Use json to locate the markdown and code cells.
• Create a unique folder based on the ipynb filename.
• Extract cell markdown data to .md files in the folder
• Extract cell code data to .py files in the folder.

Recommended prerequisite presentations:
ipynb modification using python json module
... to better understand the json involved.

Ian Stewart
2019-08-09

ipynb-extractor

Presentation Contents:

- Introduction to structure of an ipynb file.
- Logic to the python program that extracts from ipynb files.
- Example of extracting from a ipynb file.
- ipynb-extractor -h and --help
- Arguments when launching ipynb-extractor program

Ian Stewart
2019-08-09

ipynb highlevel layout.

```
{
    "cells": [                          ← 4 x Dictionary keywords
        {cell 1 dict}, {cell 2 dict}, {cell ... dict}
    ]                                   ← List of dictionaries
    "metadata": { ~15 overall keyword:value pairs}
    "nbformat": 4,
    "nbformat_minor": 2
}
```

- A dictionary with 4 x keywords: cells, metadata, nbformat and nbformat_minor.
- For "cells" keyword the value is a list.
- The list items are dictionaries for each Jupyter cell.
- **Markdown** cell dictionaries have 3 keywords
- **Code** cell dictionaries have 5 x keywords.

"Code" cell ipynb data layout.

```json
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {},
 "outputs": [],
 "source": [
  "# hello_world\n,print(\"hello world\")\n,\n"
 ]
},
```

5 x Dictionary keywords: cell_type, execution_count, metadata, outputs, source.

Contents of source list is to be extracted

Jupyter renders this code cell as...

```
In [1]:  1  # hello_world
         2  print("hello world")
         3
         4

         hello world
```

Logic to the python program

- Select ipynb file(s).
- Based on the filename create a folder
- Import simplejson
- Using simplejson module load a selected ipynb file.
- Iterate through each cell
- Determine if the cell is markdown or code
- Extract cell source list and write to a file in the folder.
- Markdown is written as a .md file.
- Code is written as a .py file.
- As required, repeat above step until all cells have been written

ipynb-extractor program logic

Import simplejson module

```
30 try:
31     import simplejson as json
32 except ImportError:
33     import json
```

Open ipynb file for read and then json loads as data

```
# Use json to load the ipynb file as "data"
data = json.load(fin)
cell_total = len(data["cells"])
print("ipynb file: {}, total cells: {}".format(file_name, cell_total))
```
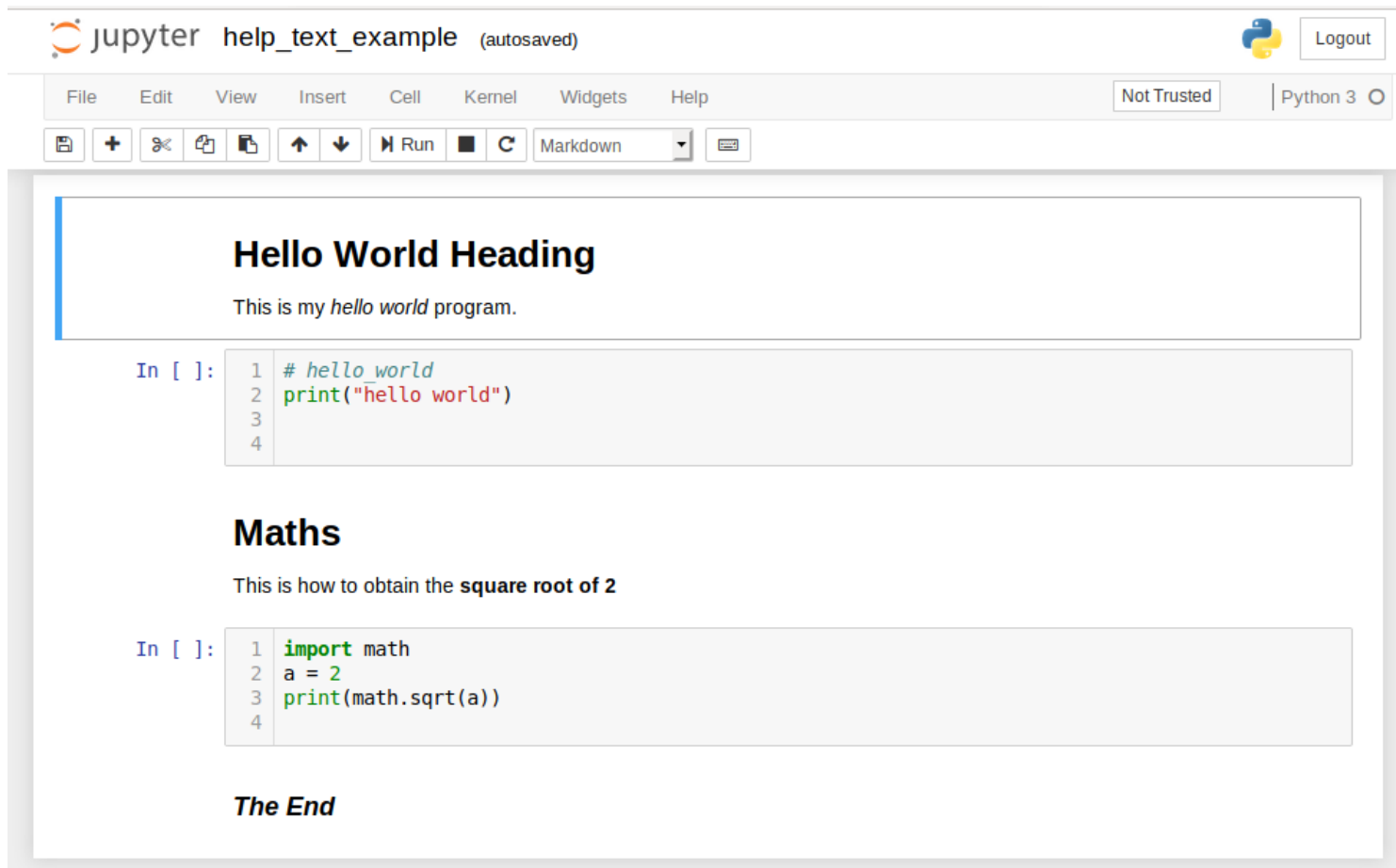
ipynb-extractor program logic

Get json's source data list and output line by line to file.

```python
# Write the cells "source" to the file.
with open(extracted_file, "w") as fout:
    for item in data["cells"][i]["source"]:
        fout.write(item)
```

ipynb-extractor example 1/3

Jupyter notebook ipynb file...

example 2/3

```
$ python3 ipynb-extractor.py help_text_example.ipynb

ipynb-extractor version: 0.1

Read a Jupyter notebook (.ipynb) file and extract to a
folder
the markdown and code cells as markdown.md and python.py
files.


Created folder `help_text_example`.
ipynb file: help_text_example.ipynb, total cells: 5
ipynb file: `help_text_example.ipynb` cells extracted to
folder `help_text_example`
Markdown files: 3, Python files: 2
```

```
$ ls -1 help_text_example/
help_text_example_01.md
help_text_example_02.py
help_text_example_03.md
help_text_example_04.py
help_text_example_05.md

$ cat help_text_example/help_text_example_04.py
import math
a = 2
```

ipynb-extractor -h and --help

```
$ python3 ipynb-extractor.py -h
ipynb-extractor version: 0.1

Read a Jupyter notebook (.ipynb) file and extract to a folder
the markdown and code cells as markdown.md and python.py files.
Usage: ipynb-extractor [OPTION]... [FILE]...

From Jupyter notebook ipynb file(s) extract the markdown and code cells.
Create a folder based on the ipynb filename. Extract to the folder the
code cells as python (.py) files or as markdown (.md) files.

[OPTION]...
Options and arguments:
   -h      print this brief help message and exit.
   --help   print a full help message and exit.
```

ipynb_creator --help text has example

```
$ python3 ipynb-extractor.py help_text_example.ipynb

ipynb-extractor version: 0.1

Read a Jupyter notebook (.ipynb) file and extract to a folder
the markdown and code cells as markdown.md and python.py files.

Created folder `help_text_example`.
ipynb file: help_text_example.ipynb, total cells: 5
ipynb file: `help_text_example.ipynb` cells extracted to folder
`help_text_example`
Markdown files: 3, Python files: 2
```

ipynb-extractor – launching arguments

Examples of the arguments that may be passed to
ipynb-extractor on launching

```
$ python3 ipynb-extractor.py
$ python3 ipynb-extractor.py example.ipynb
$ python3 ipynb-extractor.py example1.ipynb example2.ipynb
$ python3 ipynb-extractor.py *.ipynb
```

If ipynb-extractor moved to /local/bin/ then...

```
$ ipynb-extractor
$ ipynb-extractor example.ipynb
$ ipynb-extractor example1.ipynb example2.ipynb
$ ipynb-extractor *.ipynb
```

ipynb-extractor

Questions?
Demos?