ipynb modification using python json module

Hamilton Python Users Group
Ian Stewart
2019 09 09

ipynb modification using python json module

Jupyter notebook data is saved in the ***J****ava****S****cript* ***O****bject* ***N****otation* (json) format.

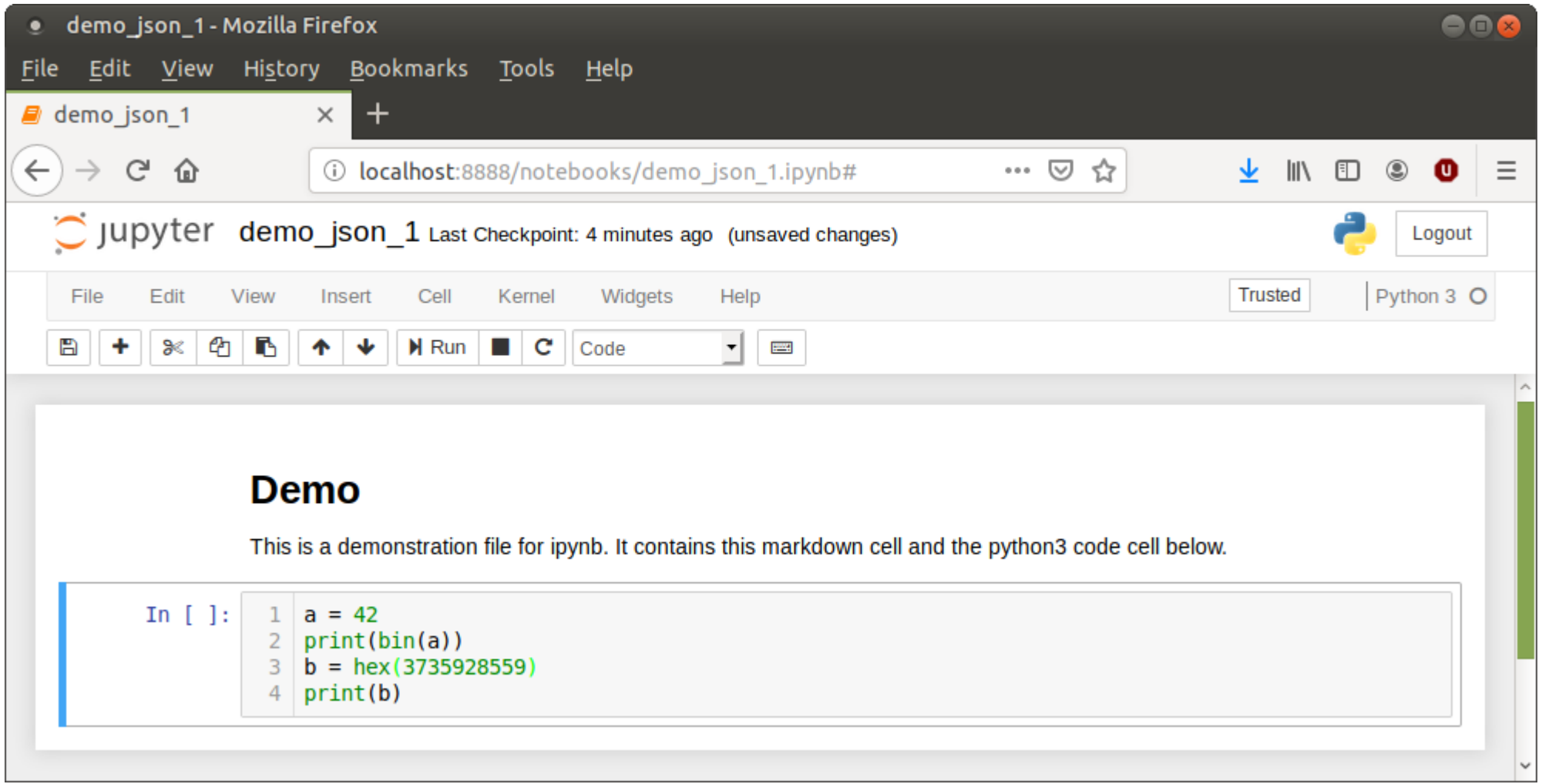
Jupyter notebook files have the extension *ipynb* (**i**nteractive **py**thon **n**ote**b**ook).

Using a python program that utilizes the json module modifications may be made to ipynb file(s).

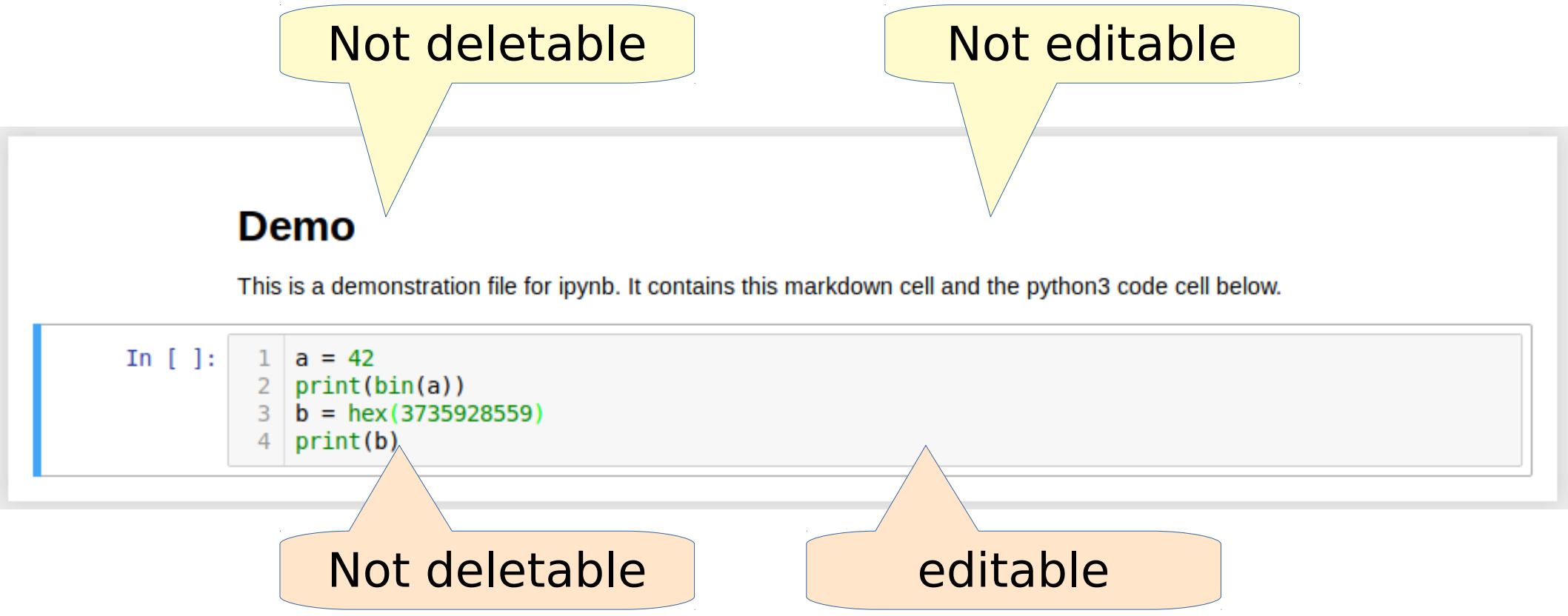
Presentation Contents

- Review a Jupyter Notebook with 2 x cells.
- Modify the metadata of the cells.
- Review the modifications in the ipynb file.
- Python program that uses the json module.
- Review the python/json code.
- Demonstration of the program modifying cell metadata.

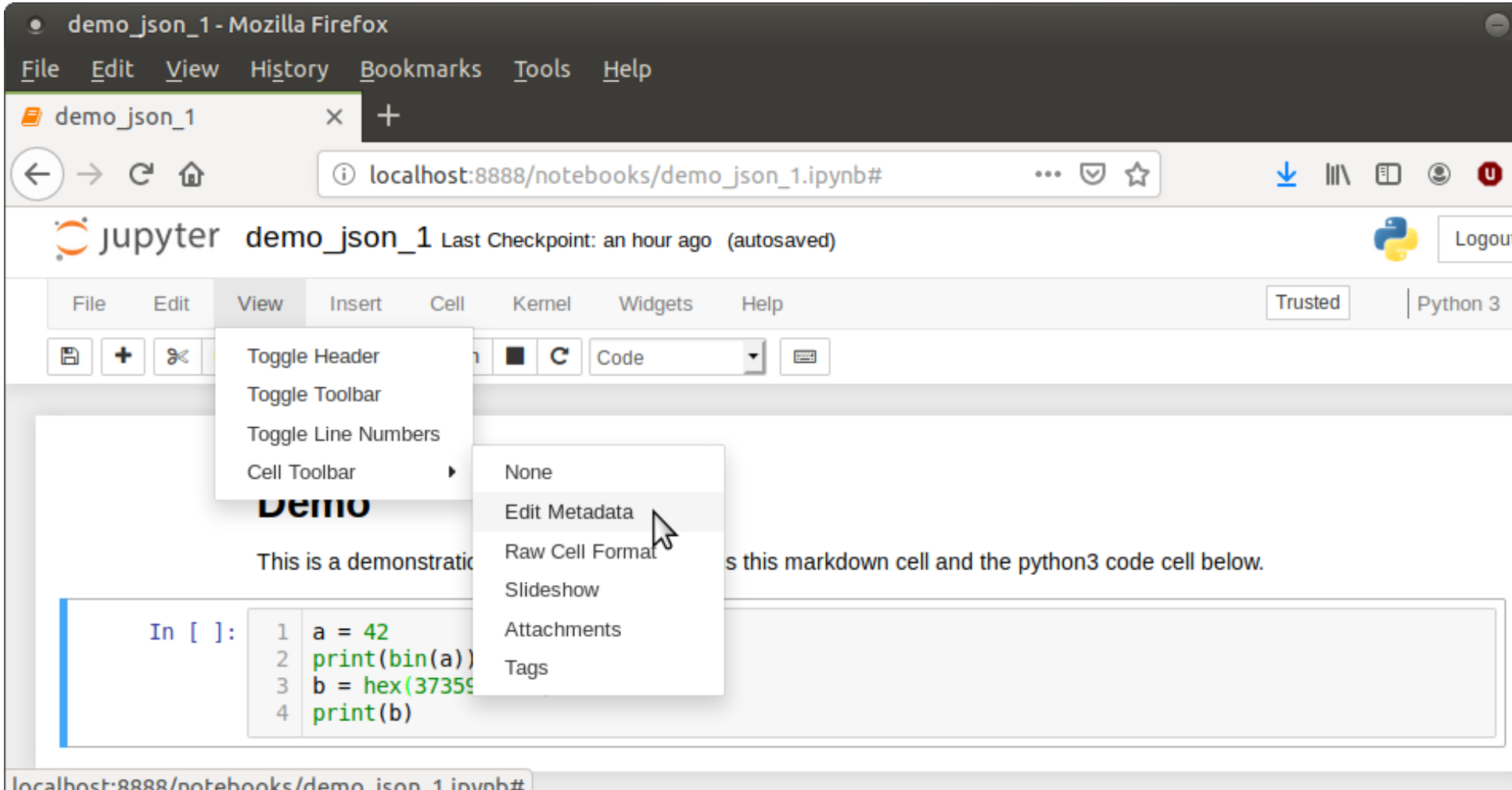
Example of Jupyter Notebook with 2 x cells



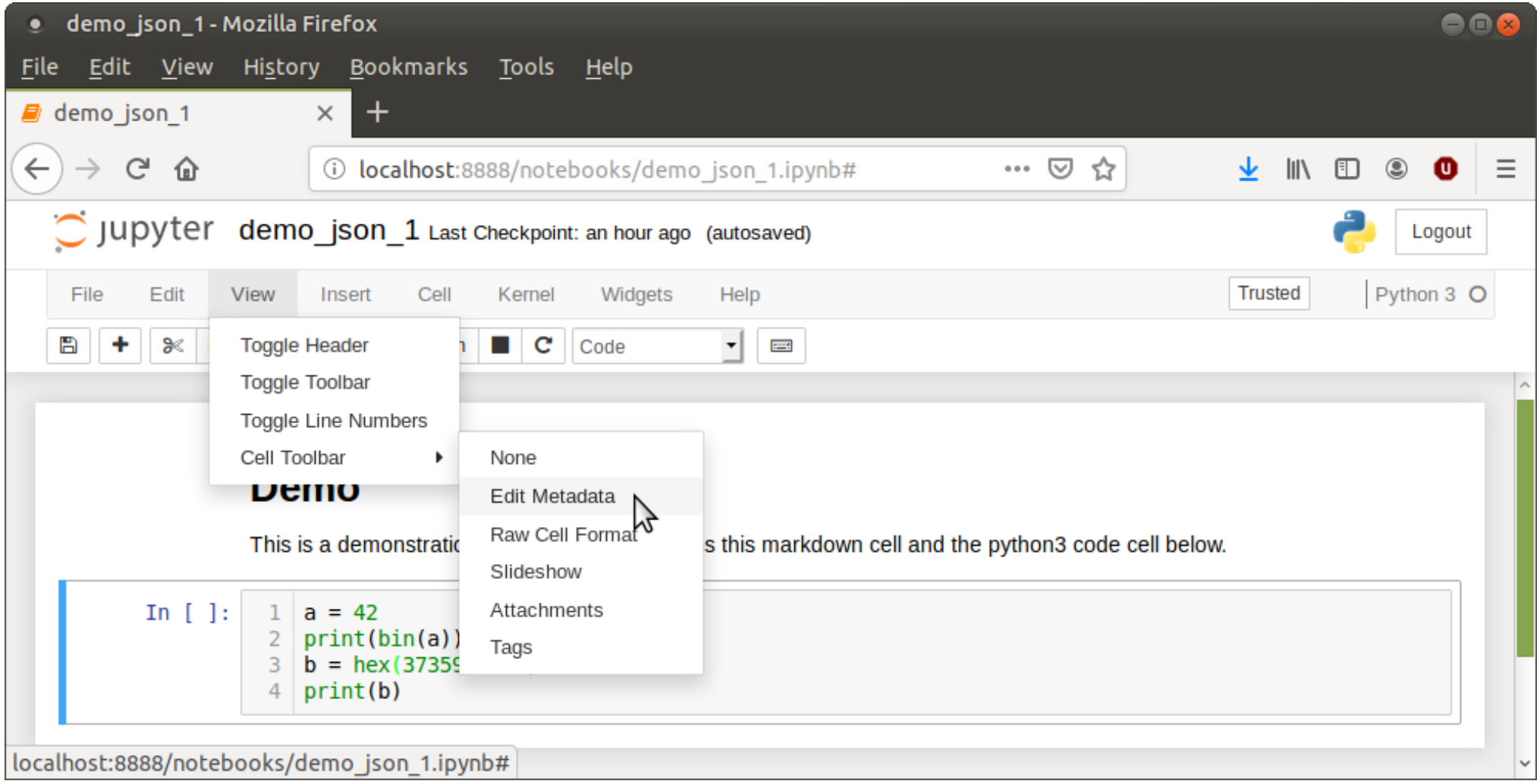
Modifications that may be desirable...



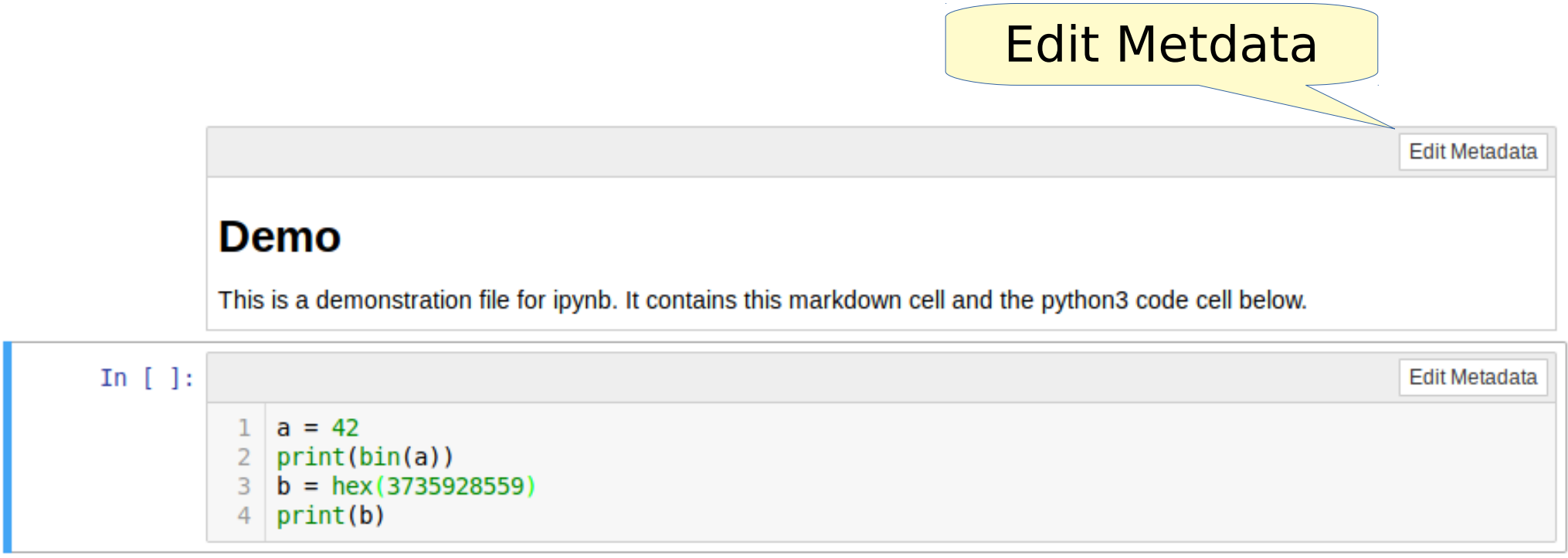
Set cells to *Edit Metadata*



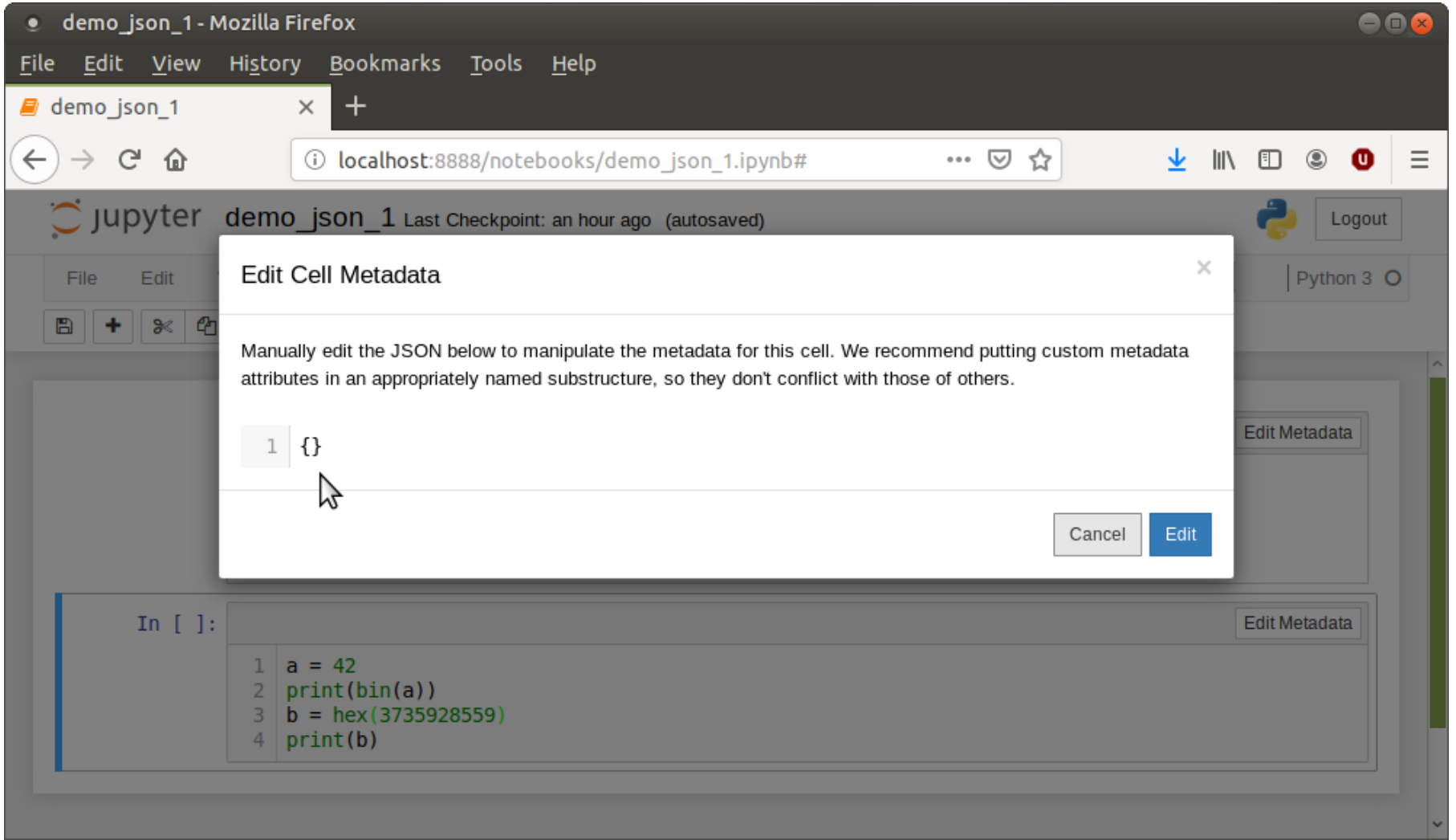
Set cells to *Edit Metadata*



Cells have *Edit Metadata* capability



Editing first cell metadata



Cell metadata keys

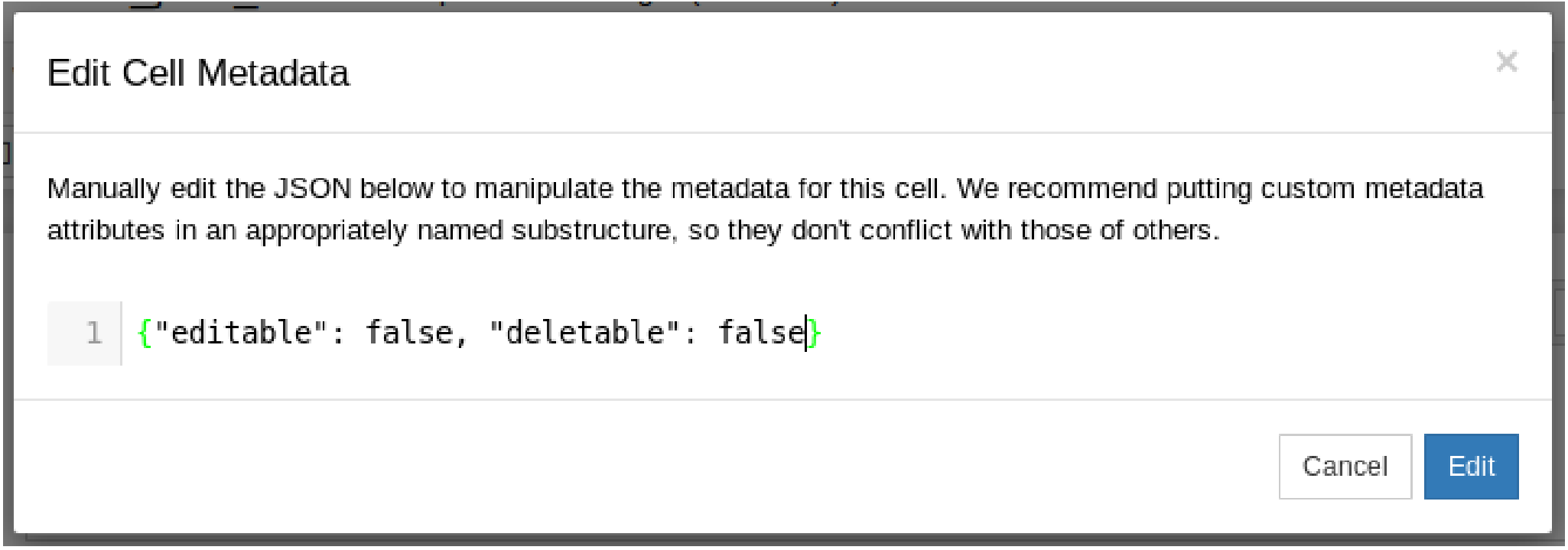
Cell metadata

The following metadata keys are defined at the cell level:

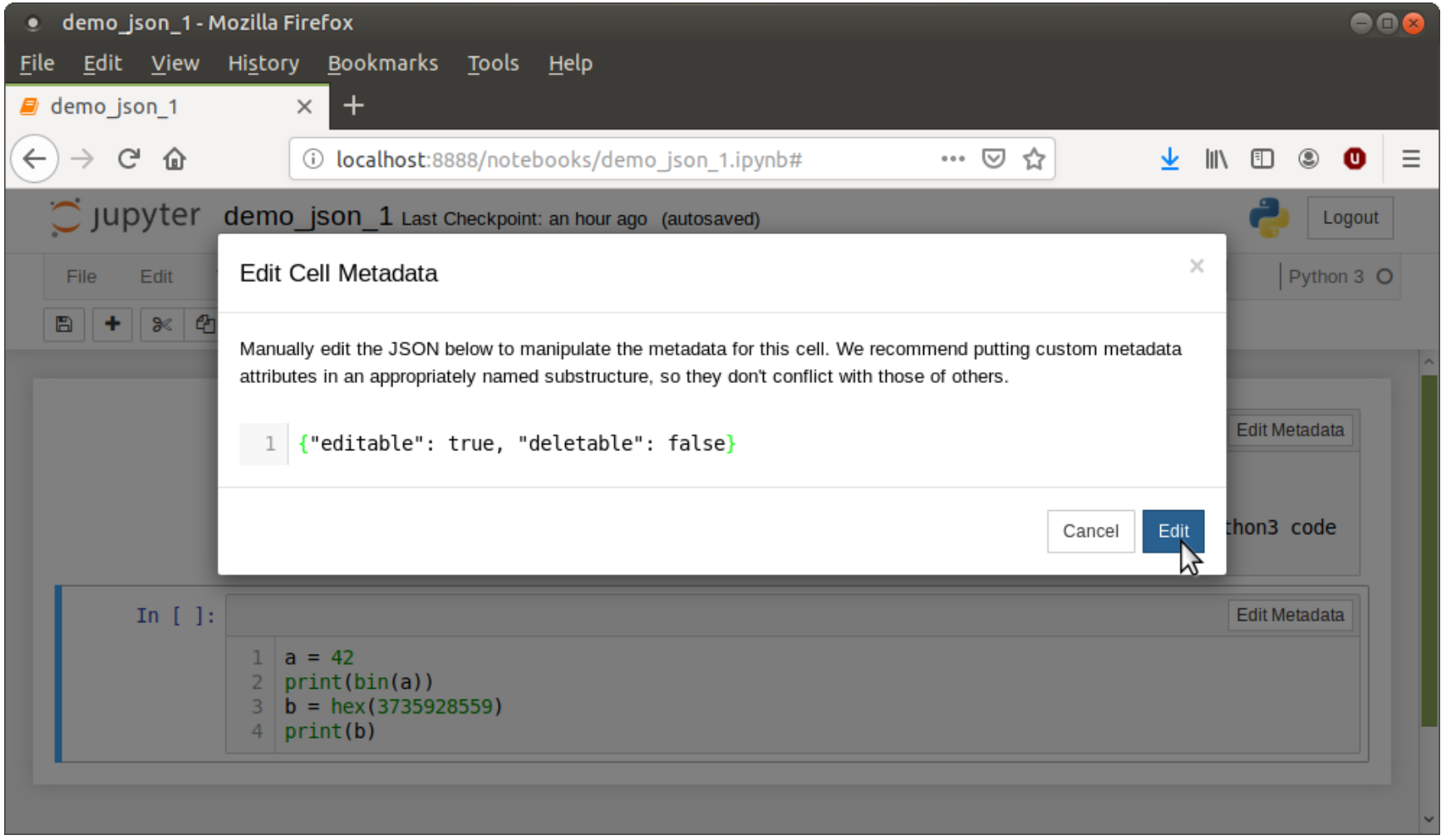
Key	Value	Interpretation
collapsed	bool	Whether the cell's output container should be collapsed
autoscroll	bool or 'auto'	Whether the cell's output is scrolled, unscrolled, or autoscrolled
deletable	bool	If False, prevent deletion of the cell
format	'mime/type'	The mime-type of a Raw NBConvert Cell
name	str	A name for the cell. Should be unique
tags	list of str	A list of string tags on the cell. Commas are not allowed in a tag

Above table from:
https://ipython.org/ipython-doc/dev/notebook/nbformat.html
Not listed: "editable" – new in version 5

Markdown cell: "editable": false, "deletable": false



Code cell: "editable": true, "deletable": false



View of Jupyter ipynb script. Before modification

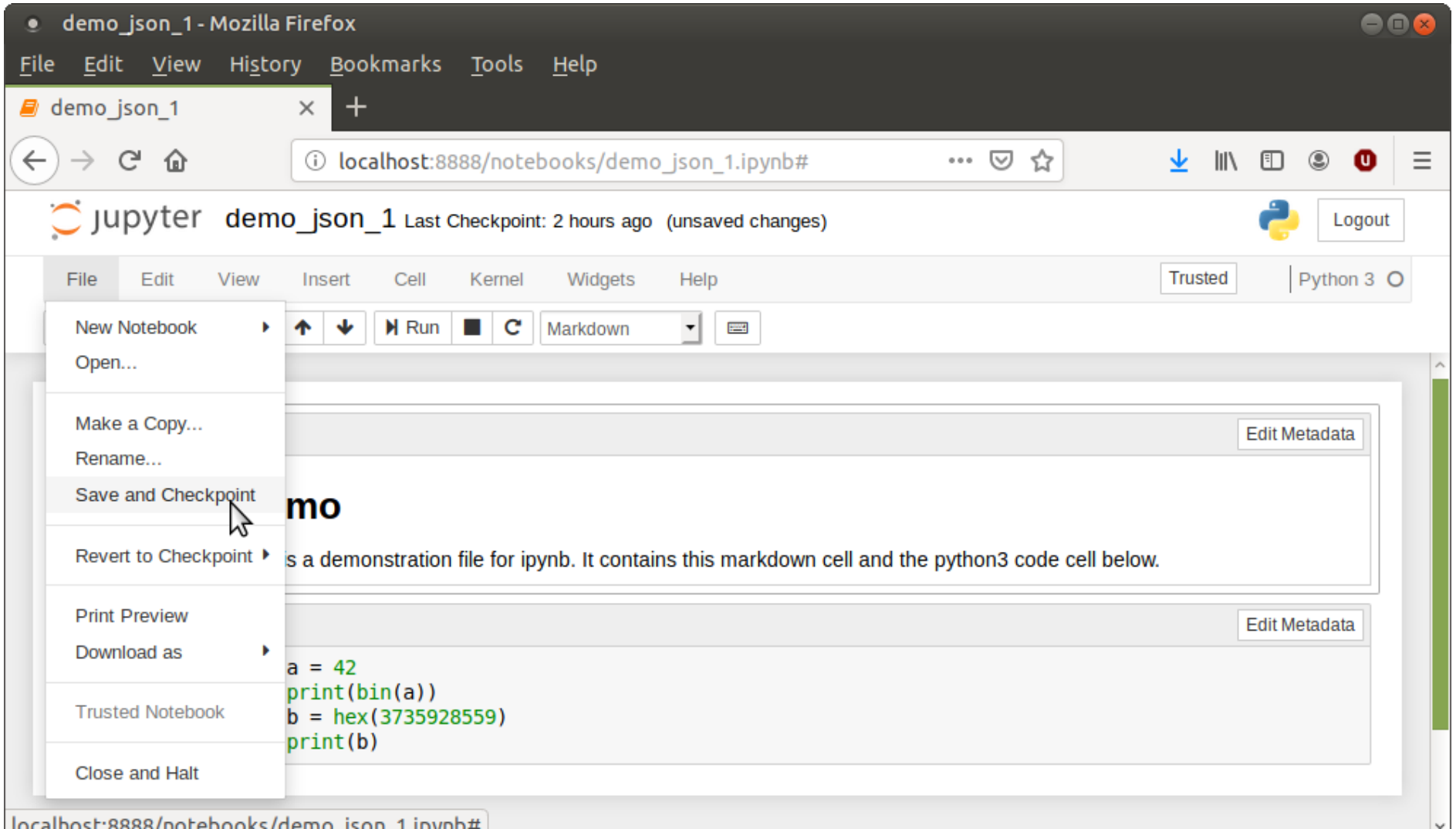
demo_json_1.ipynb

```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "# Demo\n",
    "\n",
    "This is a demonstration file for ipynb. It contains this markdown cell and the python3 code cell below. "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "a = 42\n",
    "print(bin(a))\n",
    "b = hex(3735928559)\n",
    "print(b)"
   ]
  }
```

Markdown cell. No metadata

Code cell. No metadata

Save Jupyter Notebook so ipynb file is updated



Save Jupyter Notebook so ipynb file is updated

demo_json_1.ipynb

```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "deletable": false,
    "editable": false
   },
   "source": [
    "# Demo\n",
    "\n",
    "This is a demonstration file for ipynb. It contains this markdown cell and the python3 code cell below. "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "deletable": false
   },
   "outputs": [],
   "source": [
```

Markdown cell.
With metadata added

Code cell.
With metadata
"Editable": true
Default? So not added?

Any Issue with editing metadata for each cell?

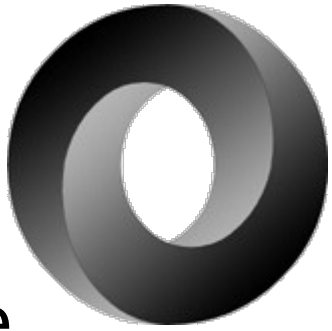
- What if each ipynb file has, say, 20 cells, and your project has 20 ipynb files.
- Then that’s 400 manual edits required which will take a lot of time.

Solution?

Solution

- A python program that imports the simplejson module.
- Poll user for the ipynb files to change.
- Poll user for cell metadata parameters to modify.
- json is used to load each file.
- Iterate over each cell and modify the metadata.
- json is used to dump the data back to the file.
- Jupyter-notebook re-loads the modified file.

json



- JSON is a text format that is completely language independent. It was derived from JavaScript.
- Website: https://www.json.org/
- Wiki: https://en.wikipedia.org/wiki/JSON
- ECMA-404, ISO/IEC 21778:2017, RFC 8259
- IPYNB notebook documents are stored in the JSON plain text format

json module

- Linux distro may include json and simplejson.
- Recommended to use simplejson
- https://github.com/simplejson/simplejson
- Docs: https://simplejson.readthedocs.io/en/latest/
- PyPi: https://pypi.org/project/simplejson/
- simplejson version 1.0 released 2005-12-25
- simplejson version 3.16.1 released 2018-09-07
- ```
  try:
      import simplejson as json
  except ImportError:
      import json
  ```
  ```

Python program flow

- Get the ipynb files in the current working directory.
- User selects the ipynb files to modify.
- User selects which metatdata parameters to apply.
- Create a backup folder to copy files to before modification.
- **If metadata to be cleared** then loop through for each file:
  - backup ipynb file to backup folder.
  - open ipynb file as r+
  - Use json module to load the file to data variable.
  - clear the metadata: data["cells"][i]['metadata'] = {}
  - f.seek(0) # to reset file position to the beginning.
  - json.dump(data, fin, indent=1) # write data bck to file.

Python program flow

- **If metadata to be selectively modified:**
  - Answer queries to select editable and deletable parameters for code and/or markdown cells.
- Loop through for each file:
  - backup ipynb file to backup folder.
  - open ipynb file as r+ (r/w positioned at beginning of file)
  - Use json module to load the file to data variable.
  - Set the metadata. E.g.:
    - data["cells"][i]['metadata']['editable'] = True
  - f.seek(0) # reset file position to the beginning.
  - json.dump(data, fin, indent=1) # write data back to file.
  - f.truncate() # remove remaining part

ipynb highlevel layout.

```
{
  "cells": [
    {cell 1 dict}, {cell 2 dict}, {cell ... dict}
    ]
  "metadata": { ~15 overall attribute:value pairs}
  "nbformat": 4,
  "nbformat_minor": 2
}
```

4 x Dictionary keywords

List of dictionaries

Each cell has a "metadata":{} dictionary

- After modification of every cells metadata in {cell ... dict}:

```
"metadata": {
    "deletable": false,
    "editable": false
   },
```

Attribute – value pairs

Code to clear cell metadata:

```
if status[0]:
    print("\nClearing metadata of all cells in files:")
    for file_name in modify_list:
        shutil.copy(file_name, folder + file_name)
        print(file_name)
        with open(file_name, "r+") as f:
            data = json.load(f)
            total_cells = len(data["cells"])
            for i in range(total_cells):
                data["cells"][i]['metadata'] = {}

            f.seek(0)
            json.dump(data, f, indent=1)
            f.truncate() # remove remaining part

    sys.exit("Completed clearing metadata on all cells.")
```

dict keywd, list pointer, dict keywd

Code to set cell metadata:

```
    # Setting matadata for each markdown and code cell.
    print("\nSetting metadata for ipynb files:")
    #print(status)
    for file_name in modify_list:
        shutil.copy(file_name, folder + file_name)
        print(file_name)
        with open(file_name, "r+") as f:
            data = json.load(f)
            total_cells = len(data["cells"])
            for i in range(total_cells):
                if data["cells"][i]['cell_type'] == "markdown":
                    data["cells"][i]['metadata']['editable'] = status[1]
                    data["cells"][i]['metadata']['deletable'] = status[3]
                if data["cells"][i]['cell_type'] == "code":
                    data["cells"][i]['metadata']['editable'] = status[2]
                    data["cells"][i]['metadata']['deletable'] = status[4]
            f.seek(0)  # reset file positi
            json.dump(data, f, indent=1)
            f.truncate()  # remove remaini
```

dict kw, list pointer,
dict kw, key-value pair

Run program. Clear metadata of all cells

$ **python3 modify_notebook_metadata.py**
Modify ipynb files by changing the metadata of each cell.
Choose the modification option you wish to apply to the ipynb file(s):

In the directory /home/ian/python/ipynb_modification_using_json_module, there are 1 ipynb files:
  1. demo_json_1.ipynb

Enter the number of the file to modify or * for all files: **1**
The file selected is: demo_json_1.ipynb
Are you sure you want to modify this file? [Y/n]:
Are there more files you wish to modify? [N/y]:
Clear all cell metadata [N/y]: **y**
Cell metadata will be cleared. Proceed? [Y/n]: **y**
A backup of the files before modification is in the folder:
/home/ian/python/ipynb_modification_using_json_module/backup-20190804-085836/

Clearing metadata of all cells in files:
demo_json_1.ipynb
Completed clearing metadata on all cells.

Each cells metadata is cleared

```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "# Demo\n",
    "\n",
    "This is a demonstration file for ipynb. It contains this markdown cell and the python3 code cell below. "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "a = 42\n",
```

Previously:

```
"metadata": {
    "deletable": false,
    "editable": false
  },
```

Run Program. Set cells metadata

In the directory /home/ian/python/ipynb_modification_using_json_module, these are 1 ipynb files:
  1. demo_json_1.ipynb
Enter the number of the file to modify or * for all files: **1**
The file selected is: demo_json_1.ipynb
Are you sure you want to modify this file? [Y/n]:
Are there more files you wish to modify? [N/y]:
Clear all cell metadata [N/y]:
Set Markdown cell to be editable? [Y/n]: **n**
Set Code cell to be editable? [Y/n]: **y**
Set Markdown cell to be deletable? [Y/n]: **n**
Set Code cell to be deletable? [Y/n]: **n**
Markdown cells: Editable:False and Deletable:False
Code cells: Editable:True and Deletable:False
Proceed? [Y/n]: **y**
A backup of the files before modification is in the folder:
/home/ian/python/ipynb_modification_using_json_module/backup-20190804-102327/
Setting metadata for ipynb files:
demo_json_1.ipynb
Completed setting metadata on cells.

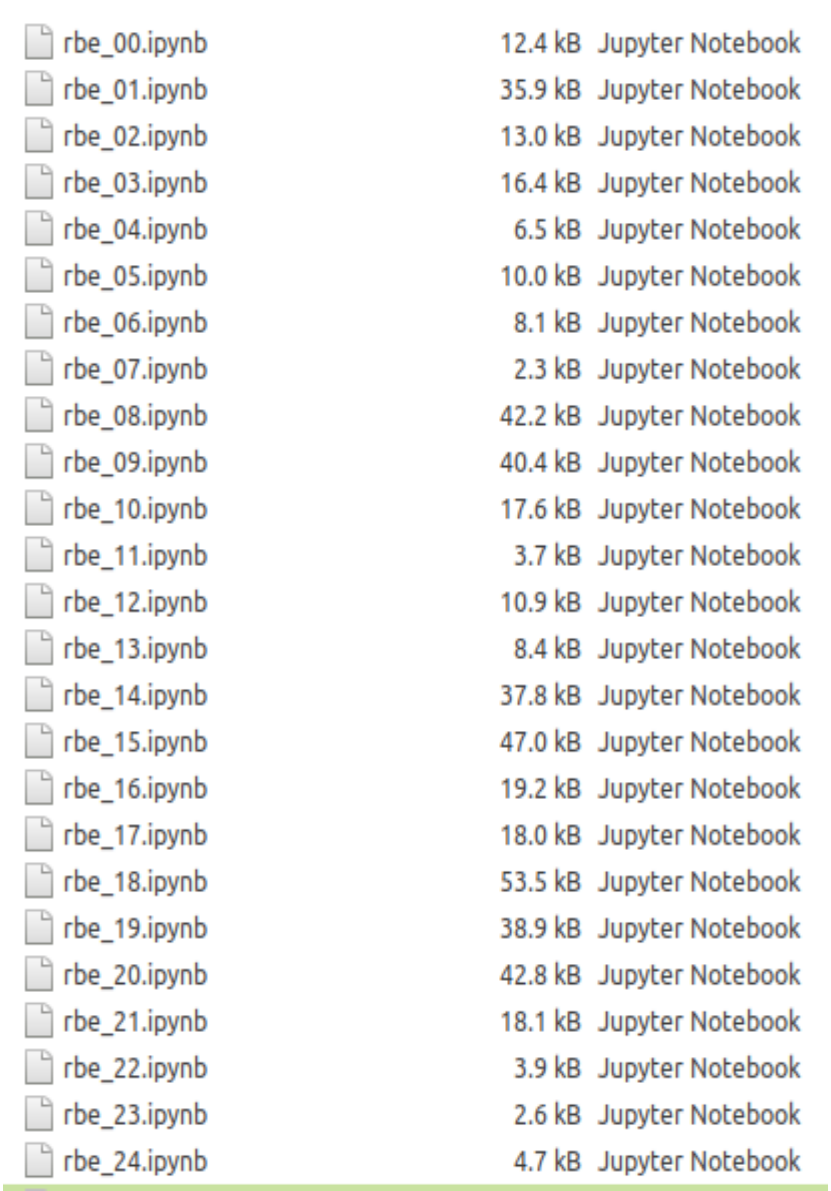
Cell metadata includes “editable” and “deletable”

```
  {
   "cell_type": "markdown",
   "metadata": {
    "editable": false,
    "deletable": false
   },
   "source": [
    "# Demo\n",
    "\n",
    "This is a demonstration file for ipynb. It contains this markdown cell and the python3 code cell below. "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "editable": true,
    "deletable": false
   },
```

Yeah, big deal – so what?

Let’s say you have 25 x ipynb files and each file has 10+ cells.

...Might be handy?



Set metadata in all cells in 25 x files

$ **modify_notebook_metadata**

Modify ipynb files by changing the metadata of each cell.
Choose the modifiction option you wish to apply to the ipynb file(s):

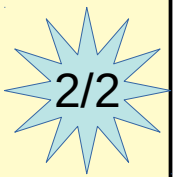
In the directory /home/ian/rust/rbe, these are 25 ipynb files:
  1. rbe_00.ipynb
  2. rbe_01.ipynb
...snip...
 24. rbe_23.ipynb
 25. rbe_24.ipynb

Enter the number of the file to modify or * for all files: *****
Selected all files
Clear all cell metadata [N/y]:
Set Markdown cell to be editable? [Y/n]: **n**
Set Code cell to be editable? [Y/n]: **y**
Set Markdown cell to be deletable? [Y/n]: **n**
Set Code cell to be deletable? [Y/n]: **n**

Set metadata in all cells in 25 x files

Markdown cells: Editable:False and Deletable:False
Code cells: Editable:True and Deletable:False
Proceed? [Y/n]: **y**
A backup of the files before modification is in the folder:
/home/ian/rust/rbe/backup-20190804-104541/

Setting metadata for ipynb files:
rbe_00.ipynb
rbe_01.ipynb
rbe_02.ipynb
...snip...
rbe_23.ipynb
rbe_24.ipynb
Completed setting metadata on cells.




```
"cells": [
 {
  "cell_type": "markdown",
  "metadata": {
   "editable": false,
   "deletable": false
  },
```

```
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "editable": true,
   "deletable": false
  },
```

...And if you do want to “restore” the files so that each cell has no metadata...

(Although these no metadata files already exist in a backup folder)

...then...

Clear cell metadata in 25 x files

$ **modify_notebook_metadata**

Modify ipynb files by changing the metadata of each cell.
Choose the modification option you wish to apply to the ipynb file(s):

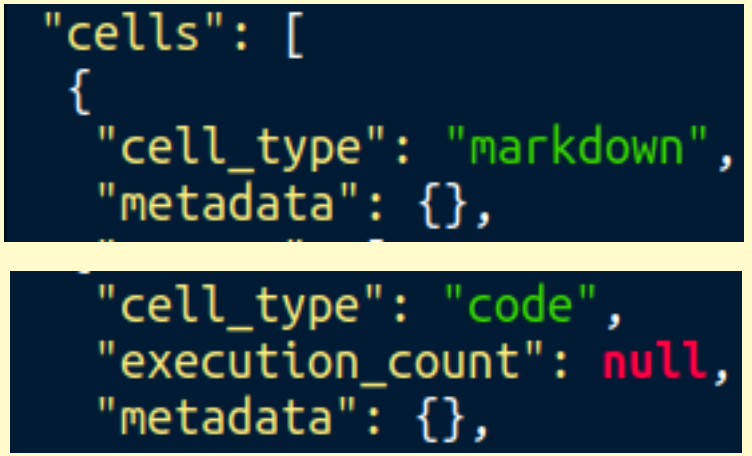
In the directory /home/ian/rust/rbe, these are 25 ipynb files:
  1. rbe_00.ipynb
  2. rbe_01.ipynb
...snip...
 24. rbe_23.ipynb
 25. rbe_24.ipynb

Enter the number of the file to modify or * for all files: *****
Selected all files
Clear all cell metadata [N/y]: **y**
Cell metadata will be cleared. Proceed? [Y/n]: **y**

Clear cell metadata in 25 x files

A backup of the files before modification is in the folder:
/home/ian/rust/rbe/backup-20190804-104118/

Clearing metadata of all cells in files:
rbe_00.ipynb
rbe_01.ipynb
rbe_02.ipynb
...snip...
rbe_23.ipynb
rbe_24.ipynb
Completed clearing metadata on all cells.



```
your_query = input("Questions? ")
```

```
the.__end__(".")
```