Robert Rauschenberg Foundation Archives ~Test Repository Code Examples~ & Instructions

Convert EAD XML to CSV using Python and Terminal

Project Parameters Github Repository : https://github.com/Super10veBug/rrfa-pfch-2019

This project is in response to the Robert Rauschenberg Foundation's Archives (RRFA) request for a CSV finding aid to facilitate shipping of materials for digitization and storage.

The EAD XML file used for this conversion is an export from ArchivesSpace, which is the CMS used by RRFA. The XML documents supplied by RRFA are selections of series RRFA01 – Rauschenberg Papers and RRFA02 – Audio Visual

My purpose was to supply RRFA with a reusable script, and instructions that can be followed by the archivists and future interns.

Background

Running this code requires:

Installation of Python3

Knowledge some basic command line tools and navigation

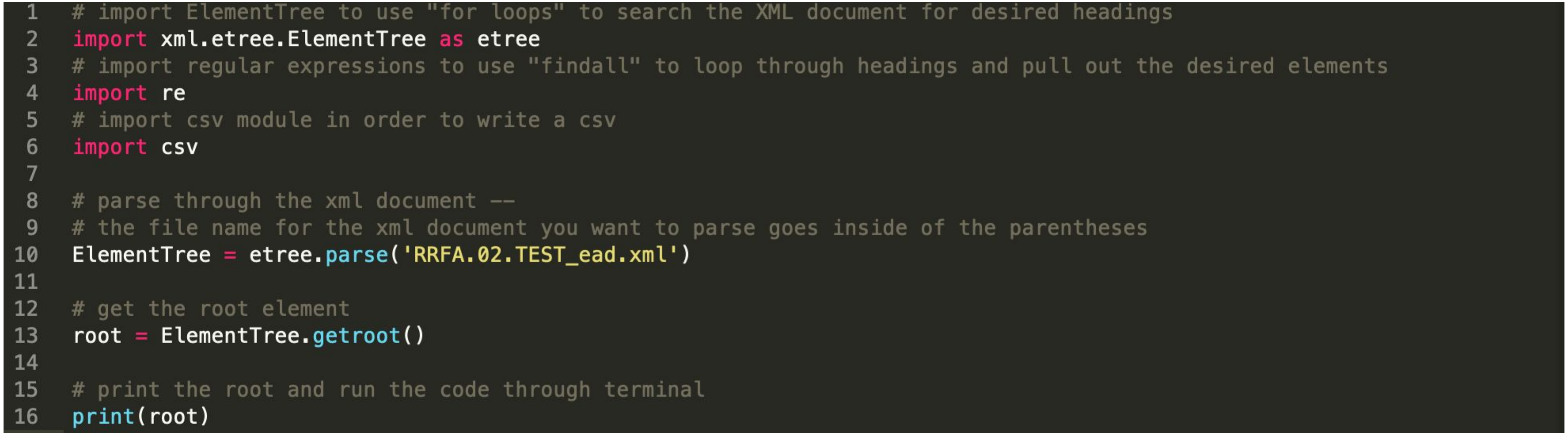
A text editor for coding

This script uses for loops, a python dictionary, and the following modules:

CSV, ElementTree, and Regular Expressions.

The rest of this document will be instructional.

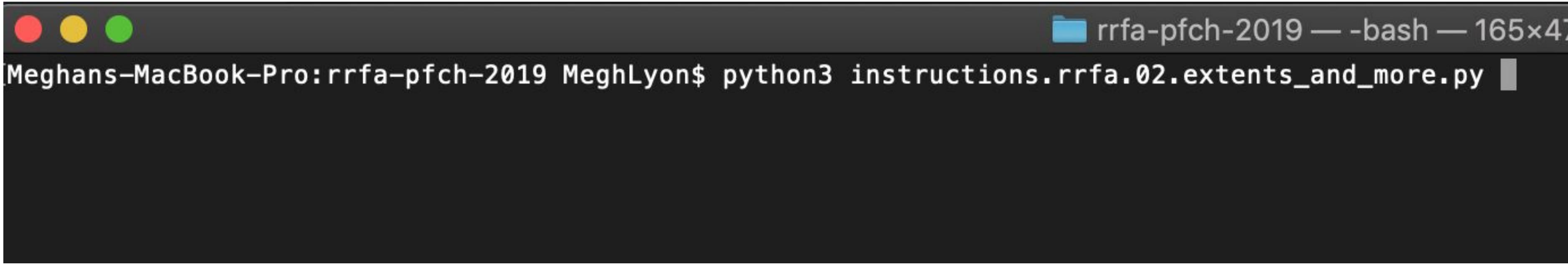
Step One - Import modules and find the root element



```python
# import ElementTree to use "for loops" to search the XML document for desired headings
import xml.etree.ElementTree as etree
# import regular expressions to use "findall" to loop through headings and pull out the desired elements
import re
# import csv module in order to write a csv
import csv

# parse through the xml document --
# the file name for the xml document you want to parse goes inside of the parentheses
ElementTree = etree.parse('RRFA.02.TEST_ead.xml')

# get the root element
root = ElementTree.getroot()

# print the root and run the code through terminal
print(root)
```

Save your XML EAD file and your Python file (code in the text editor) in the same folder on your computer. These are the first steps for the code. The yellow address in the ‘’ is the name of the xml finding aid. It must be exact. Grey text following the # sign are comments (in this case, instructions) and will not affect the code.

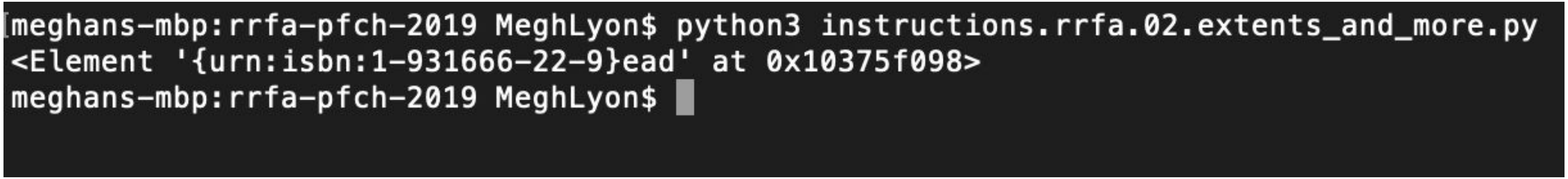
Step One - Terminal Side



```
[Meghans-MacBook-Pro:rrfa-pfch-2019 MeghLyon$ python3 instructions.rrfa.02.extents_and_more.py
```

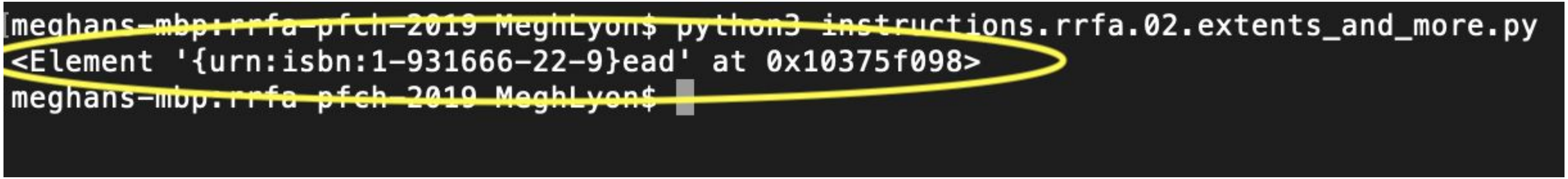
Run the code through the terminal first by navigating to the folder/directory in which you've saved your Python file—I saved my files in the directory "rrfa-pfch-2019"

Step One - Terminal Side Part 2



```
[meghans-mbp:rrfa-pfch-2019 MeghLyon$ python3 instructions.rrfa.02.extents_and_more.py
<Element '{urn:isbn:1-931666-22-9}ead' at 0x10375f098>
meghans-mbp:rrfa-pfch-2019 MeghLyon$
```

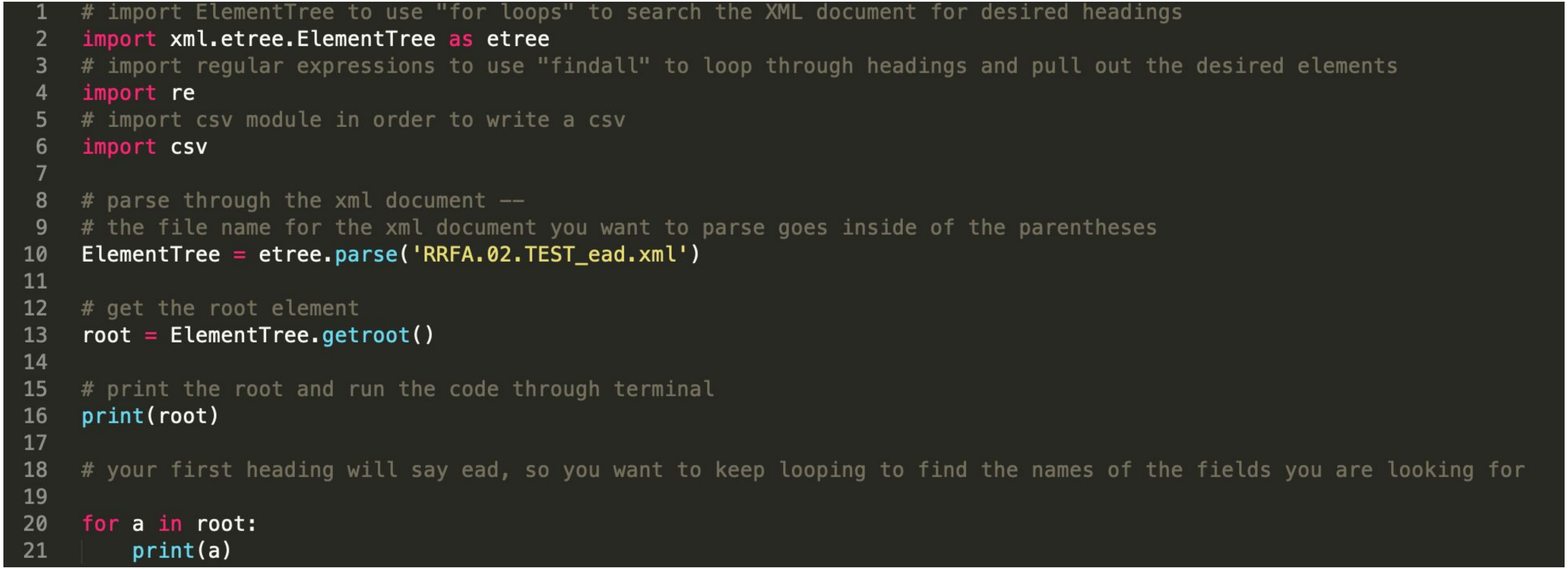
This is the result



```
[meghans-mbp:rrfa-pfch-2019 MeghLyon$ python3 instructions.rrfa.02.extents_and_more.py
<Element '{urn:isbn:1-931666-22-9}ead' at 0x10375f098>
meghans-mbp:rrfa-pfch-2019 MeghLyon$
```

Inside of this circle, you'll see we have printed the root element, which is ead.

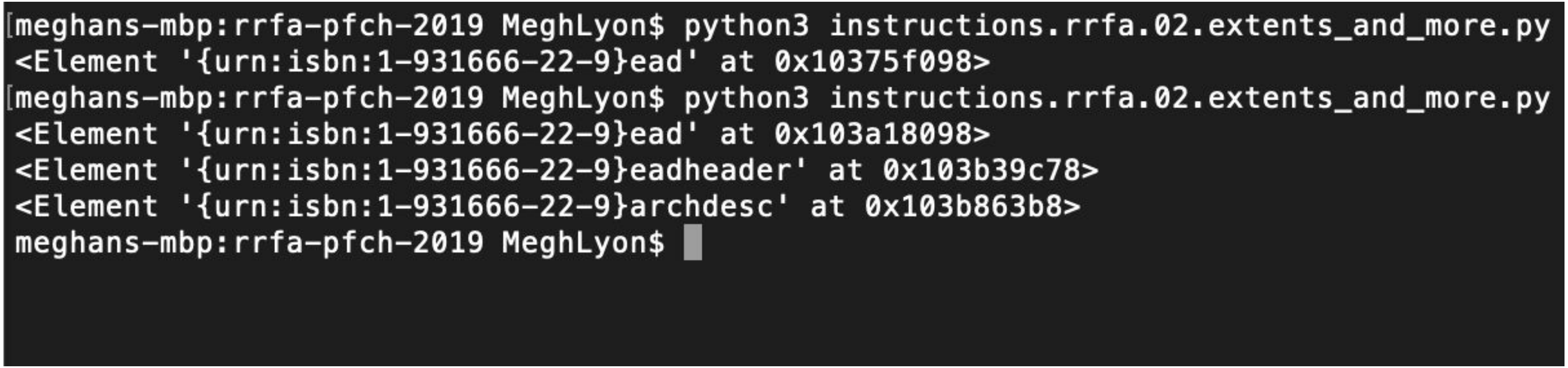
Step Two



```
1   # import ElementTree to use "for loops" to search the XML document for desired headings
2   import xml.etree.ElementTree as etree
3   # import regular expressions to use "findall" to loop through headings and pull out the desired elements
4   import re
5   # import csv module in order to write a csv
6   import csv
7
8   # parse through the xml document --
9   # the file name for the xml document you want to parse goes inside of the parentheses
10  ElementTree = etree.parse('RRFA.02.TEST_ead.xml')
11
12  # get the root element
13  root = ElementTree.getroot()
14
15  # print the root and run the code through terminal
16  print(root)
17
18  # your first heading will say ead, so you want to keep looping to find the names of the fields you are looking for
19
20  for a in root:
21      print(a)
```

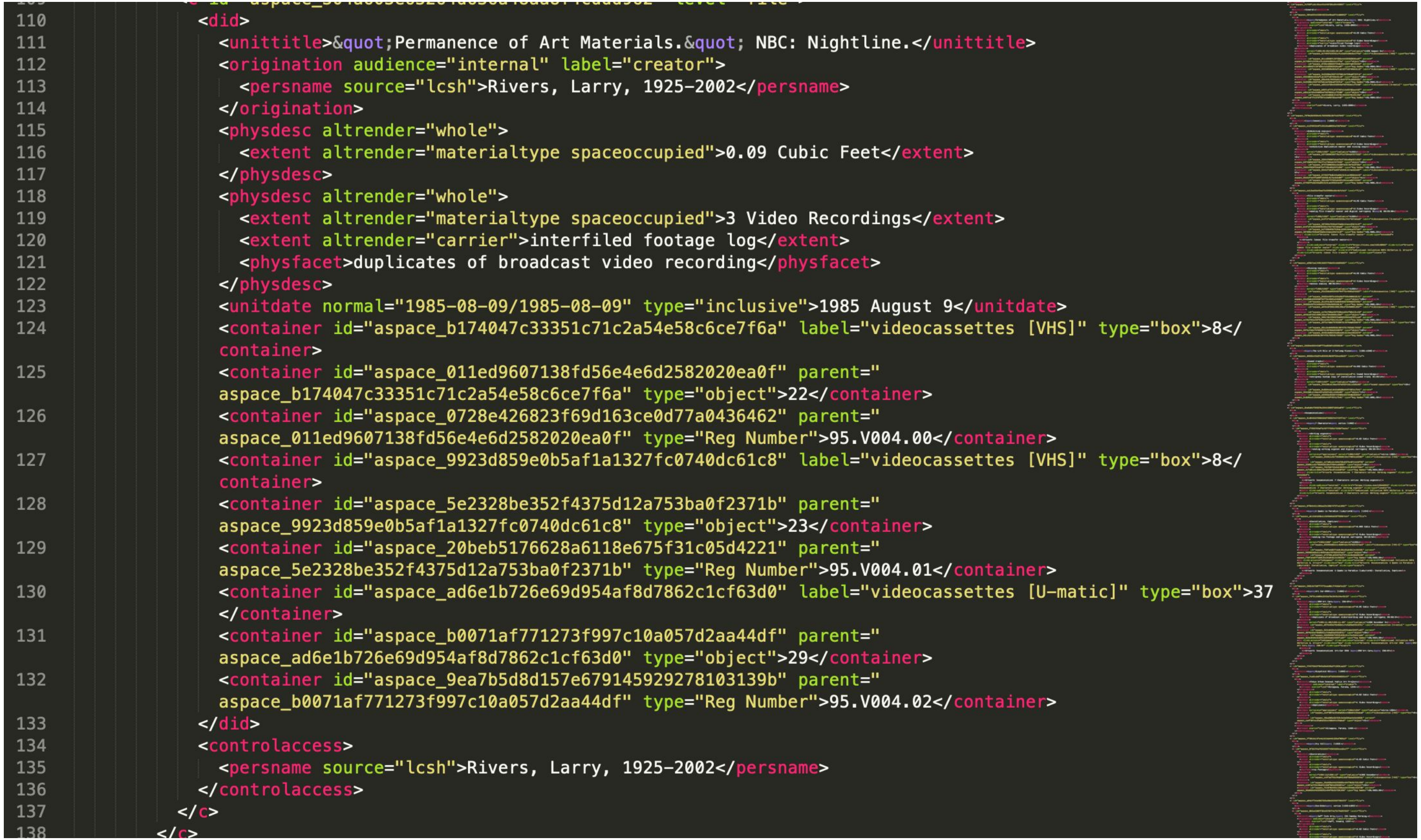
As you see from the comment on line 18 – you need to find the child elements of EAD to find the fields we want to eventually write in the CSV.

Step Two Continued



```
[meghans-mbp:rrfa-pfch-2019 MeghLyon$ python3 instructions.rrfa.02.extents_and_more.py
 <Element '{urn:isbn:1-931666-22-9}ead' at 0x10375f098>
[meghans-mbp:rrfa-pfch-2019 MeghLyon$ python3 instructions.rrfa.02.extents_and_more.py
 <Element '{urn:isbn:1-931666-22-9}ead' at 0x103a18098>
 <Element '{urn:isbn:1-931666-22-9}eadheader' at 0x103b39c78>
 <Element '{urn:isbn:1-931666-22-9}archdesc' at 0x103b863b8>
 meghans-mbp:rrfa-pfch-2019 MeghLyon$ 
```

This is the printed result of the first for loop, we are looking for container information (box numbers), titles, extent data and location -- as we can clearly see in our XML file, but we are looping to find the addresses for our code. The addresses may change depending on the output, so we need to look for each XML Finding Aid that is exported from ArchivesSpace.



```
    <did>
      <unittitle>&quot;Permanence of Art Materials.&quot; NBC: Nightline.</unittitle>
      <origination audience="internal" label="creator">
        <persname source="lcsh">Rivers, Larry, 1925-2002</persname>
      </origination>
      <physdesc altrender="whole">
        <extent altrender="materialtype spaceoccupied">0.09 Cubic Feet</extent>
      </physdesc>
      <physdesc altrender="whole">
        <extent altrender="materialtype spaceoccupied">3 Video Recordings</extent>
        <extent altrender="carrier">interfiled footage log</extent>
        <physfacet>duplicates of broadcast video recording</physfacet>
      </physdesc>
      <unitdate normal="1985-08-09/1985-08-09" type="inclusive">1985 August 9</unitdate>
      <container id="aspace_b174047c33351c71c2a54e58c6ce7f6a" label="videocassettes [VHS]" type="box">8</container>
      <container id="aspace_011ed9607138fd56e4e6d2582020ea0f" parent="aspace_b174047c33351c71c2a54e58c6ce7f6a" type="object">22</container>
      <container id="aspace_0728e426823f69d163ce0d77a0436462" parent="aspace_011ed9607138fd56e4e6d2582020ea0f" type="Reg Number">95.V004.00</container>
      <container id="aspace_9923d859e0b5af1a1327fc0740dc61c8" label="videocassettes [VHS]" type="box">8</container>
      <container id="aspace_5e2328be352f4375d12a753ba0f2371b" parent="aspace_9923d859e0b5af1a1327fc0740dc61c8" type="object">23</container>
      <container id="aspace_20beb5176628a6118e675f31c05d4221" parent="aspace_5e2328be352f4375d12a753ba0f2371b" type="Reg Number">95.V004.01</container>
      <container id="aspace_ad6e1b726e69d954af8d7862c1cf63d0" label="videocassettes [U-matic]" type="box">37</container>
      <container id="aspace_b0071af771273f997c10a057d2aa44df" parent="aspace_ad6e1b726e69d954af8d7862c1cf63d0" type="object">29</container>
      <container id="aspace_9ea7b5d8d157e679145929278103139b" parent="aspace_b0071af771273f997c10a057d2aa44df" type="Reg Number">95.V004.02</container>
    </did>
    <controlaccess>
      <persname source="lcsh">Rivers, Larry, 1925-2002</persname>
    </controlaccess>
  </c>
</c>
```

We are looping through the nested elements under <did>

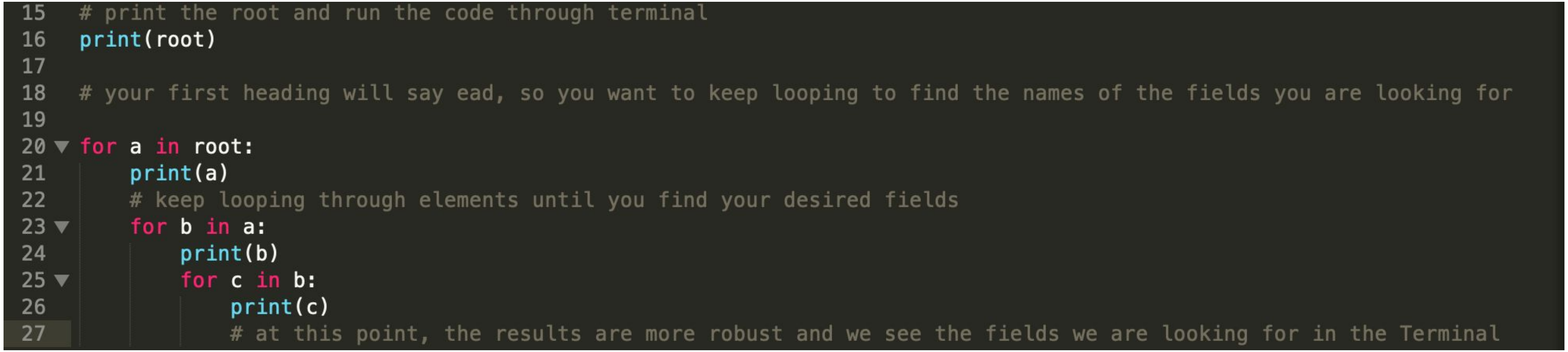
To extract text from between <extent>, <physfacet>,

<container>,

and

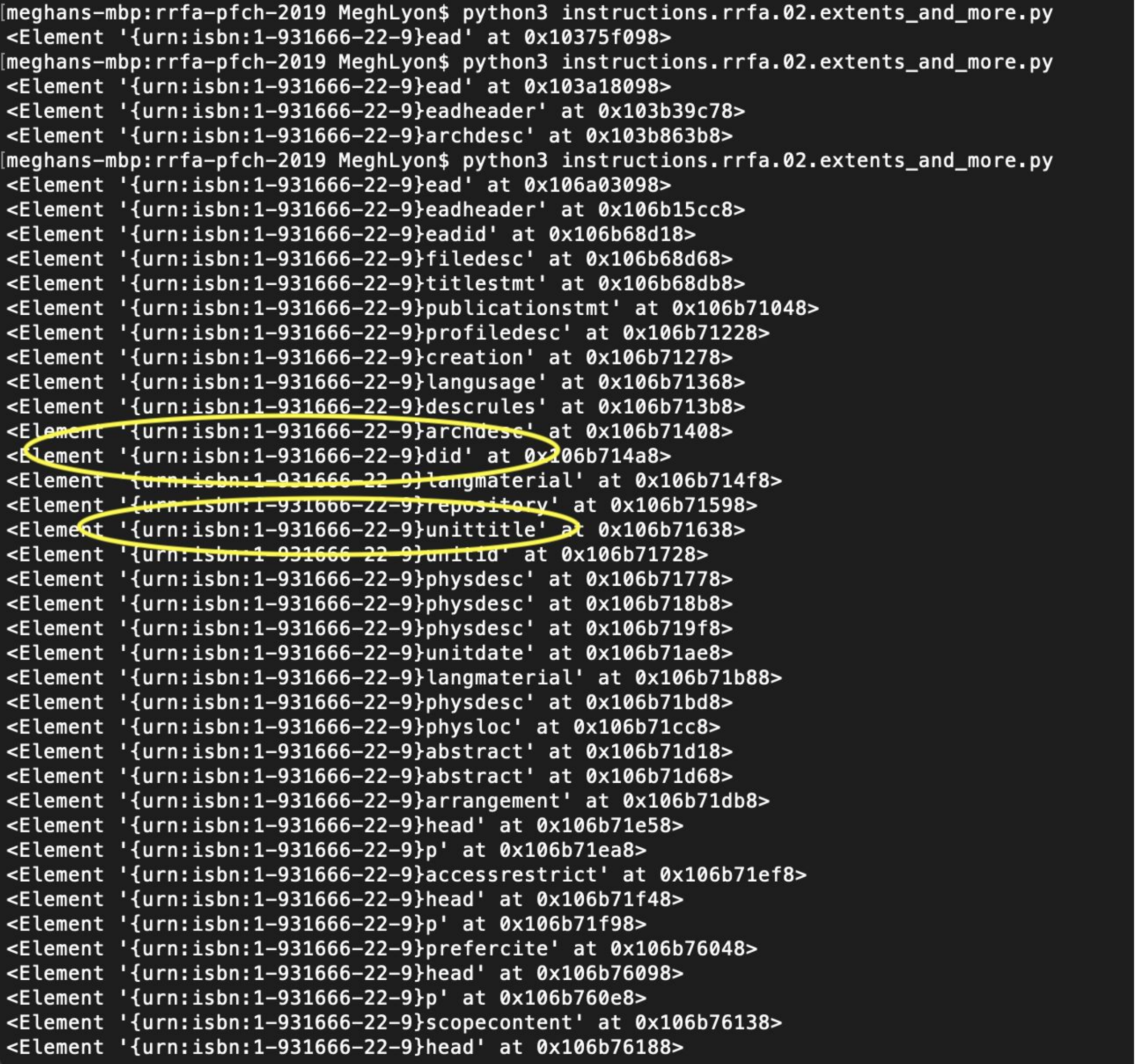
<unittitle>

Step Three - Looping through the elements



```
# print the root and run the code through terminal
print(root)

# your first heading will say ead, so you want to keep looping to find the names of the fields you are looking for

for a in root:
    print(a)
    # keep looping through elements until you find your desired fields
    for b in a:
        print(b)
        for c in b:
            print(c)
            # at this point, the results are more robust and we see the fields we are looking for in the Terminal
```

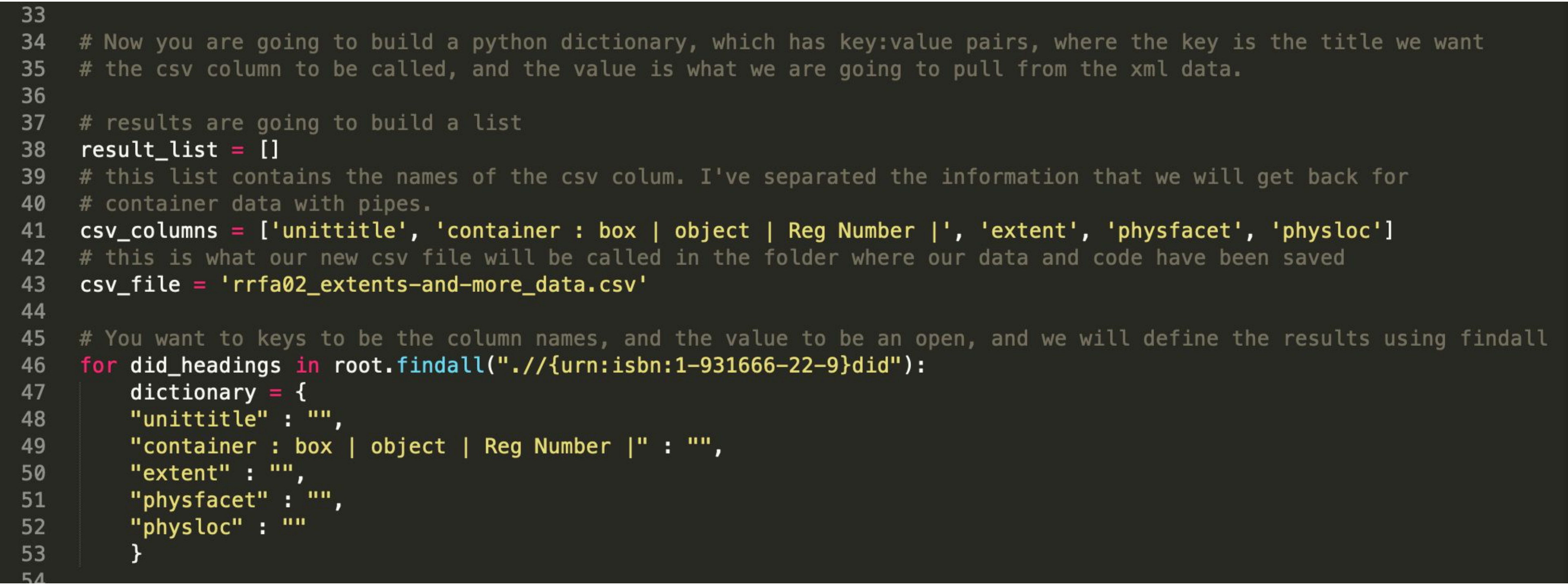
(a, b, and c are variables)



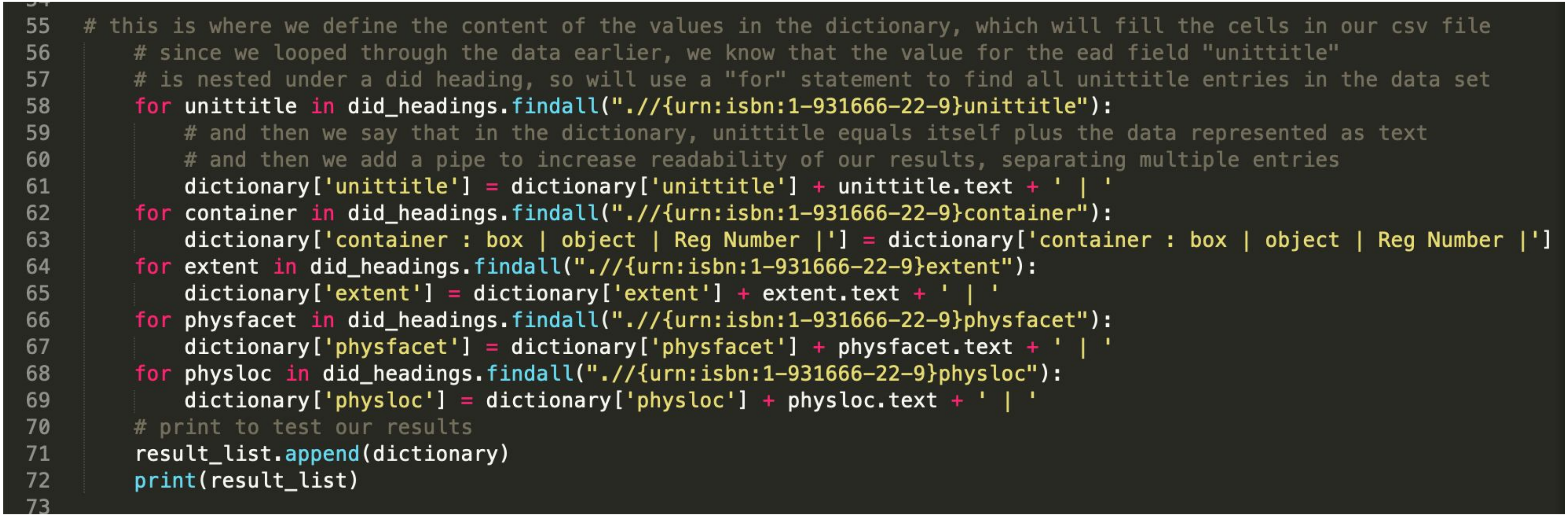
```
meghans-mbp:rrfa-pfch-2019 MeghLyon$ python3 instructions.rrfa.02.extents_and_more.py
 <Element '{urn:isbn:1-931666-22-9}ead' at 0x10375f098>
[meghans-mbp:rrfa-pfch-2019 MeghLyon$ python3 instructions.rrfa.02.extents_and_more.py
 <Element '{urn:isbn:1-931666-22-9}ead' at 0x103a18098>
 <Element '{urn:isbn:1-931666-22-9}eadheader' at 0x103b39c78>
 <Element '{urn:isbn:1-931666-22-9}archdesc' at 0x103b863b8>
[meghans-mbp:rrfa-pfch-2019 MeghLyon$ python3 instructions.rrfa.02.extents_and_more.py
 <Element '{urn:isbn:1-931666-22-9}ead' at 0x106a03098>
 <Element '{urn:isbn:1-931666-22-9}eadheader' at 0x106b15cc8>
 <Element '{urn:isbn:1-931666-22-9}eadid' at 0x106b68d18>
 <Element '{urn:isbn:1-931666-22-9}filedesc' at 0x106b68d68>
 <Element '{urn:isbn:1-931666-22-9}titlestmt' at 0x106b68db8>
 <Element '{urn:isbn:1-931666-22-9}publicationstmt' at 0x106b71048>
 <Element '{urn:isbn:1-931666-22-9}profiledesc' at 0x106b71228>
 <Element '{urn:isbn:1-931666-22-9}creation' at 0x106b71278>
 <Element '{urn:isbn:1-931666-22-9}langusage' at 0x106b71368>
 <Element '{urn:isbn:1-931666-22-9}descrules' at 0x106b713b8>
 <Element '{urn:isbn:1-931666-22-9}archdesc' at 0x106b71408>
 <Element '{urn:isbn:1-931666-22-9}did' at 0x106b714a8>
 <Element '{urn:isbn:1-931666-22-9}langmaterial' at 0x106b714f8>
 <Element '{urn:isbn:1-931666-22-9}repository' at 0x106b71598>
 <Element '{urn:isbn:1-931666-22-9}unittitle' at 0x106b71638>
 <Element '{urn:isbn:1-931666-22-9}unitid' at 0x106b71728>
 <Element '{urn:isbn:1-931666-22-9}physdesc' at 0x106b71778>
 <Element '{urn:isbn:1-931666-22-9}physdesc' at 0x106b718b8>
 <Element '{urn:isbn:1-931666-22-9}physdesc' at 0x106b719f8>
 <Element '{urn:isbn:1-931666-22-9}unitdate' at 0x106b71ae8>
 <Element '{urn:isbn:1-931666-22-9}langmaterial' at 0x106b71b88>
 <Element '{urn:isbn:1-931666-22-9}physdesc' at 0x106b71bd8>
 <Element '{urn:isbn:1-931666-22-9}physloc' at 0x106b71cc8>
 <Element '{urn:isbn:1-931666-22-9}abstract' at 0x106b71d18>
 <Element '{urn:isbn:1-931666-22-9}abstract' at 0x106b71d68>
 <Element '{urn:isbn:1-931666-22-9}arrangement' at 0x106b71db8>
 <Element '{urn:isbn:1-931666-22-9}head' at 0x106b71e58>
 <Element '{urn:isbn:1-931666-22-9}p' at 0x106b71ea8>
 <Element '{urn:isbn:1-931666-22-9}accessrestrict' at 0x106b71ef8>
 <Element '{urn:isbn:1-931666-22-9}head' at 0x106b71f48>
 <Element '{urn:isbn:1-931666-22-9}p' at 0x106b71f98>
 <Element '{urn:isbn:1-931666-22-9}prefercite' at 0x106b76048>
 <Element '{urn:isbn:1-931666-22-9}head' at 0x106b76098>
 <Element '{urn:isbn:1-931666-22-9}p' at 0x106b760e8>
 <Element '{urn:isbn:1-931666-22-9}scopecontent' at 0x106b76138>
 <Element '{urn:isbn:1-931666-22-9}head' at 0x106b76188>
```

The number in the curly brackets {} is really all we need to do our regular expression. But we can find the other elements just to check.

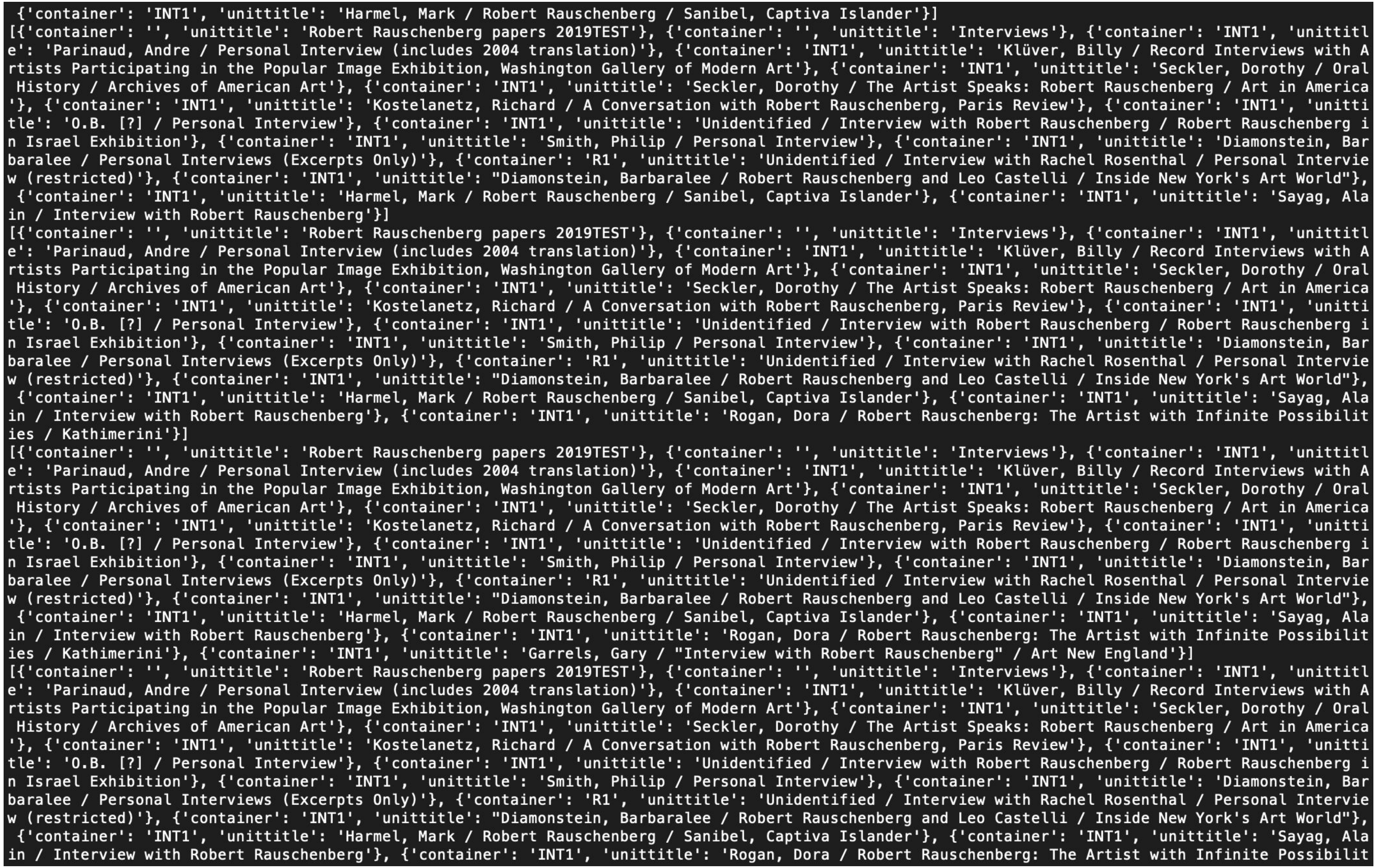
Step Four - “Findall” and Python Dictionary



```python

# Now you are going to build a python dictionary, which has key:value pairs, where the key is the title we want
# the csv column to be called, and the value is what we are going to pull from the xml data.

# results are going to build a list
result_list = []
# this list contains the names of the csv colum. I've separated the information that we will get back for
# container data with pipes.
csv_columns = ['unittitle', 'container : box | object | Reg Number |', 'extent', 'physfacet', 'physloc']
# this is what our new csv file will be called in the folder where our data and code have been saved
csv_file = 'rrfa02_extents-and-more_data.csv'

# You want to keys to be the column names, and the value to be an open, and we will define the results using findall
for did_headings in root.findall(".//{urn:isbn:1-931666-22-9}did"):
    dictionary = {
    "unittitle" : "",
    "container : box | object | Reg Number |" : "",
    "extent" : "",
    "physfacet" : "",
    "physloc" : ""
    }
```



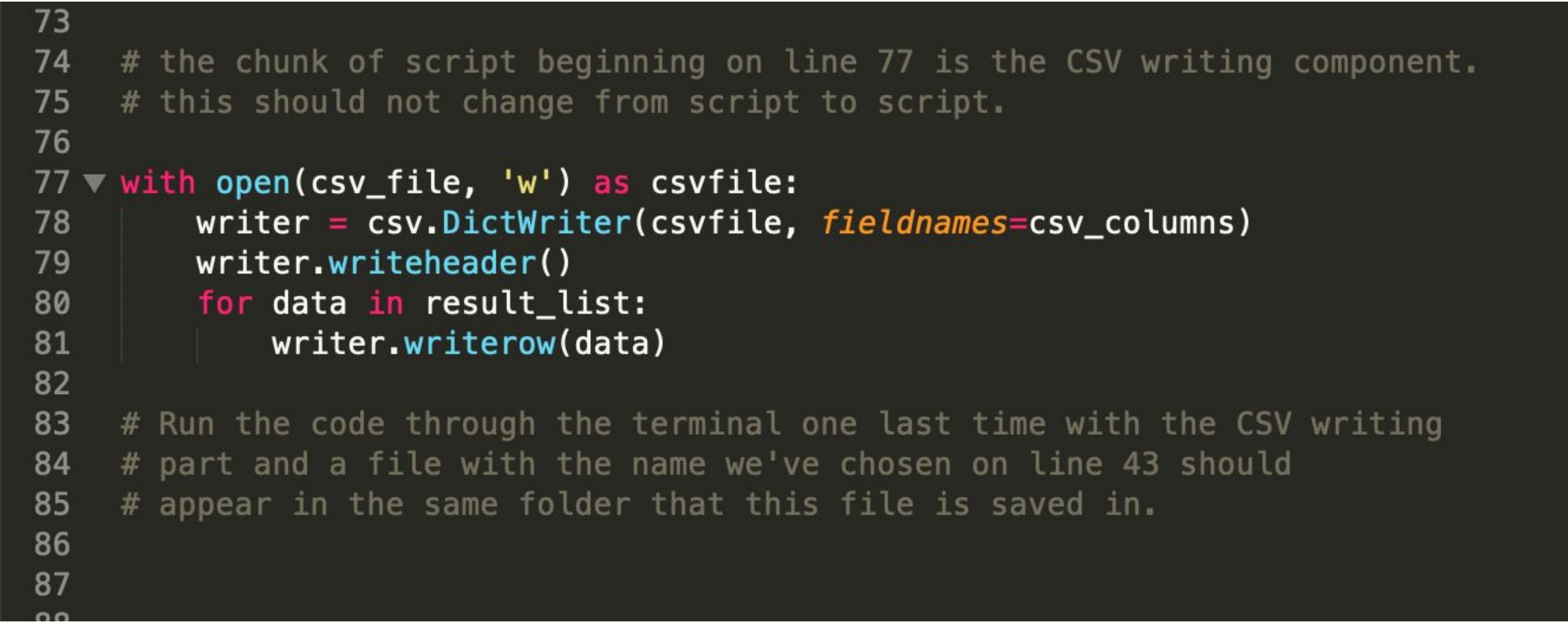
```
# this is where we define the content of the values in the dictionary, which will fill the cells in our csv file
  # since we looped through the data earlier, we know that the value for the ead field "unittitle"
  # is nested under a did heading, so will use a "for" statement to find all unittitle entries in the data set
  for unittitle in did_headings.findall(".//{urn:isbn:1-931666-22-9}unittitle"):
    # and then we say that in the dictionary, unittitle equals itself plus the data represented as text
    # and then we add a pipe to increase readability of our results, separating multiple entries
    dictionary['unittitle'] = dictionary['unittitle'] + unittitle.text + ' | '
  for container in did_headings.findall(".//{urn:isbn:1-931666-22-9}container"):
    dictionary['container : box | object | Reg Number |'] = dictionary['container : box | object | Reg Number |']
  for extent in did_headings.findall(".//{urn:isbn:1-931666-22-9}extent"):
    dictionary['extent'] = dictionary['extent'] + extent.text + ' | '
  for physfacet in did_headings.findall(".//{urn:isbn:1-931666-22-9}physfacet"):
    dictionary['physfacet'] = dictionary['physfacet'] + physfacet.text + ' | '
  for physloc in did_headings.findall(".//{urn:isbn:1-931666-22-9}physloc"):
    dictionary['physloc'] = dictionary['physloc'] + physloc.text + ' | '
  # print to test our results
  result_list.append(dictionary)
  print(result_list)
```



```
 {'container': 'INT1', 'unittitle': 'Harmel, Mark / Robert Rauschenberg / Sanibel, Captiva Islander'}]
[{'container': '', 'unittitle': 'Robert Rauschenberg papers 2019TEST'}, {'container': '', 'unittitle': 'Interviews'}, {'container': 'INT1', 'unittitl
e': 'Parinaud, Andre / Personal Interview (includes 2004 translation)'}, {'container': 'INT1', 'unittitle': 'Klüver, Billy / Record Interviews with A
rtists Participating in the Popular Image Exhibition, Washington Gallery of Modern Art'}, {'container': 'INT1', 'unittitle': 'Seckler, Dorothy / Oral
 History / Archives of American Art'}, {'container': 'INT1', 'unittitle': 'Seckler, Dorothy / The Artist Speaks: Robert Rauschenberg / Art in America
'}, {'container': 'INT1', 'unittitle': 'Kostelanetz, Richard / A Conversation with Robert Rauschenberg, Paris Review'}, {'container': 'INT1', 'uniti
tle': 'O.B. [?] / Personal Interview'}, {'container': 'INT1', 'unittitle': 'Unidentified / Interview with Robert Rauschenberg / Robert Rauschenberg i
n Israel Exhibition'}, {'container': 'INT1', 'unittitle': 'Smith, Philip / Personal Interview'}, {'container': 'INT1', 'unittitle': 'Diamonstein, Bar
baralee / Personal Interviews (Excerpts Only)'}, {'container': 'R1', 'unittitle': 'Unidentified / Interview with Rachel Rosenthal / Personal Intervie
w (restricted)'}, {'container': 'INT1', 'unittitle': "Diamonstein, Barbaralee / Robert Rauschenberg and Leo Castelli / Inside New York's Art World"},
 {'container': 'INT1', 'unittitle': 'Harmel, Mark / Robert Rauschenberg / Sanibel, Captiva Islander'}, {'container': 'INT1', 'unittitle': 'Sayag, Ala
in / Interview with Robert Rauschenberg'}]
[{'container': '', 'unittitle': 'Robert Rauschenberg papers 2019TEST'}, {'container': '', 'unittitle': 'Interviews'}, {'container': 'INT1', 'unittitl
e': 'Parinaud, Andre / Personal Interview (includes 2004 translation)'}, {'container': 'INT1', 'unittitle': 'Klüver, Billy / Record Interviews with A
rtists Participating in the Popular Image Exhibition, Washington Gallery of Modern Art'}, {'container': 'INT1', 'unittitle': 'Seckler, Dorothy / Oral
 History / Archives of American Art'}, {'container': 'INT1', 'unittitle': 'Seckler, Dorothy / The Artist Speaks: Robert Rauschenberg / Art in America
'}, {'container': 'INT1', 'unittitle': 'Kostelanetz, Richard / A Conversation with Robert Rauschenberg, Paris Review'}, {'container': 'INT1', 'uniti
tle': 'O.B. [?] / Personal Interview'}, {'container': 'INT1', 'unittitle': 'Unidentified / Interview with Robert Rauschenberg / Robert Rauschenberg i
n Israel Exhibition'}, {'container': 'INT1', 'unittitle': 'Smith, Philip / Personal Interview'}, {'container': 'INT1', 'unittitle': 'Diamonstein, Bar
baralee / Personal Interviews (Excerpts Only)'}, {'container': 'R1', 'unittitle': 'Unidentified / Interview with Rachel Rosenthal / Personal Intervie
w (restricted)'}, {'container': 'INT1', 'unittitle': "Diamonstein, Barbaralee / Robert Rauschenberg and Leo Castelli / Inside New York's Art World"},
 {'container': 'INT1', 'unittitle': 'Harmel, Mark / Robert Rauschenberg / Sanibel, Captiva Islander'}, {'container': 'INT1', 'unittitle': 'Sayag, Ala
in / Interview with Robert Rauschenberg'}, {'container': 'INT1', 'unittitle': 'Rogan, Dora / Robert Rauschenberg: The Artist with Infinite Possibilit
ies / Kathimerini'}]
[{'container': '', 'unittitle': 'Robert Rauschenberg papers 2019TEST'}, {'container': '', 'unittitle': 'Interviews'}, {'container': 'INT1', 'unittitl
e': 'Parinaud, Andre / Personal Interview (includes 2004 translation)'}, {'container': 'INT1', 'unittitle': 'Klüver, Billy / Record Interviews with A
rtists Participating in the Popular Image Exhibition, Washington Gallery of Modern Art'}, {'container': 'INT1', 'unittitle': 'Seckler, Dorothy / Oral
 History / Archives of American Art'}, {'container': 'INT1', 'unittitle': 'Seckler, Dorothy / The Artist Speaks: Robert Rauschenberg / Art in America
'}, {'container': 'INT1', 'unittitle': 'Kostelanetz, Richard / A Conversation with Robert Rauschenberg, Paris Review'}, {'container': 'INT1', 'uniti
tle': 'O.B. [?] / Personal Interview'}, {'container': 'INT1', 'unittitle': 'Unidentified / Interview with Robert Rauschenberg / Robert Rauschenberg i
n Israel Exhibition'}, {'container': 'INT1', 'unittitle': 'Smith, Philip / Personal Interview'}, {'container': 'INT1', 'unittitle': 'Diamonstein, Bar
baralee / Personal Interviews (Excerpts Only)'}, {'container': 'R1', 'unittitle': 'Unidentified / Interview with Rachel Rosenthal / Personal Intervie
w (restricted)'}, {'container': 'INT1', 'unittitle': "Diamonstein, Barbaralee / Robert Rauschenberg and Leo Castelli / Inside New York's Art World"},
 {'container': 'INT1', 'unittitle': 'Harmel, Mark / Robert Rauschenberg / Sanibel, Captiva Islander'}, {'container': 'INT1', 'unittitle': 'Sayag, Ala
in / Interview with Robert Rauschenberg'}, {'container': 'INT1', 'unittitle': 'Rogan, Dora / Robert Rauschenberg: The Artist with Infinite Possibilit
ies / Kathimerini'}, {'container': 'INT1', 'unittitle': 'Garrels, Gary / "Interview with Robert Rauschenberg" / Art New England'}]
[{'container': '', 'unittitle': 'Robert Rauschenberg papers 2019TEST'}, {'container': '', 'unittitle': 'Interviews'}, {'container': 'INT1', 'unittitl
e': 'Parinaud, Andre / Personal Interview (includes 2004 translation)'}, {'container': 'INT1', 'unittitle': 'Klüver, Billy / Record Interviews with A
rtists Participating in the Popular Image Exhibition, Washington Gallery of Modern Art'}, {'container': 'INT1', 'unittitle': 'Seckler, Dorothy / Oral
 History / Archives of American Art'}, {'container': 'INT1', 'unittitle': 'Seckler, Dorothy / The Artist Speaks: Robert Rauschenberg / Art in America
'}, {'container': 'INT1', 'unittitle': 'Kostelanetz, Richard / A Conversation with Robert Rauschenberg, Paris Review'}, {'container': 'INT1', 'uniti
tle': 'O.B. [?] / Personal Interview'}, {'container': 'INT1', 'unittitle': 'Unidentified / Interview with Robert Rauschenberg / Robert Rauschenberg i
n Israel Exhibition'}, {'container': 'INT1', 'unittitle': 'Smith, Philip / Personal Interview'}, {'container': 'INT1', 'unittitle': 'Diamonstein, Bar
baralee / Personal Interviews (Excerpts Only)'}, {'container': 'R1', 'unittitle': 'Unidentified / Interview with Rachel Rosenthal / Personal Intervie
w (restricted)'}, {'container': 'INT1', 'unittitle': "Diamonstein, Barbaralee / Robert Rauschenberg and Leo Castelli / Inside New York's Art World"},
 {'container': 'INT1', 'unittitle': 'Harmel, Mark / Robert Rauschenberg / Sanibel, Captiva Islander'}, {'container': 'INT1', 'unittitle': 'Sayag, Ala
in / Interview with Robert Rauschenberg'}, {'container': 'INT1', 'unittitle': 'Rogan, Dora / Robert Rauschenberg: The Artist with Infinite Possibilit
```

This is what our dictionary should look like when we run the code through the terminal.

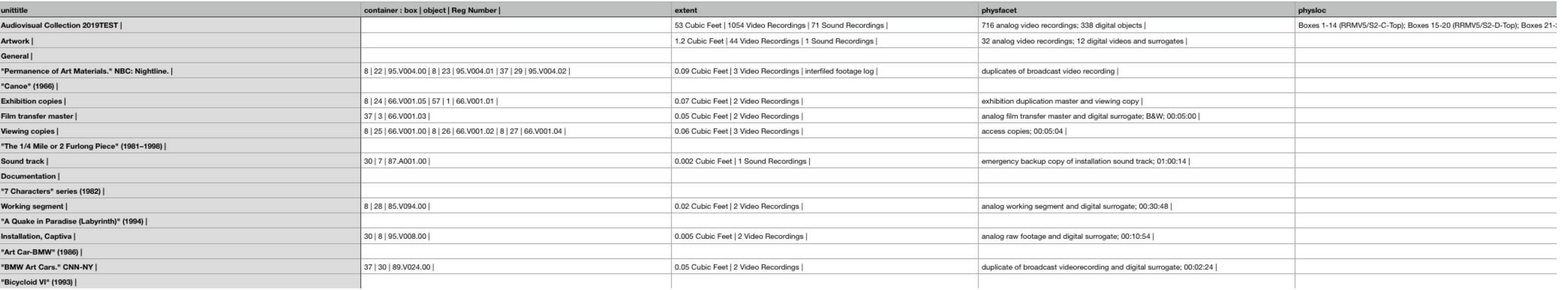
Step Five - CSV Writer



```
73
74  # the chunk of script beginning on line 77 is the CSV writing component.
75  # this should not change from script to script.
76
77  with open(csv_file, 'w') as csvfile:
78      writer = csv.DictWriter(csvfile, fieldnames=csv_columns)
79      writer.writeheader()
80      for data in result_list:
81          writer.writerow(data)
82
83  # Run the code through the terminal one last time with the CSV writing
84  # part and a file with the name we've chosen on line 43 should
85  # appear in the same folder that this file is saved in.
86
87
88
```

Last step

Run the code through the terminal one last time, and look for your CSV in the folder where the code and the data are saved.



<table>
<tr><th>unittitle</th><th>container : box | object | Reg Number |</th><th>extent</th><th>physfacet</th><th>physloc</th></tr>
<tr><td>Audiovisual Collection 2019TEST |</td><td></td><td>53 Cubic Feet | 1054 Video Recordings | 71 Sound Recordings |</td><td>716 analog video recordings; 338 digital objects |</td><td>Boxes 1-14 (RRMV5/S2-C-Top); Boxes 15-20 (RRMV5/S2-D-Top); Boxes 21-</td></tr>
<tr><td>Artwork |</td><td></td><td>1.2 Cubic Feet | 44 Video Recordings | 1 Sound Recordings |</td><td>32 analog video recordings; 12 digital videos and surrogates |</td><td></td></tr>
<tr><td>General |</td><td></td><td></td><td></td><td></td></tr>
<tr><td>"Permanence of Art Materials." NBC: Nightline. |</td><td>8 | 22 | 95.V004.00 | 8 | 23 | 95.V004.01 | 37 | 29 | 95.V004.02 |</td><td>0.09 Cubic Feet | 3 Video Recordings | interfiled footage log |</td><td>duplicates of broadcast video recording |</td><td></td></tr>
<tr><td>"Canoe" (1966) |</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Exhibition copies |</td><td>8 | 24 | 66.V001.05 | 57 | 1 | 66.V001.01 |</td><td>0.07 Cubic Feet | 2 Video Recordings |</td><td>exhibition duplication master and viewing copy |</td><td></td></tr>
<tr><td>Film transfer master |</td><td>37 | 3 | 66.V001.03 |</td><td>0.05 Cubic Feet | 2 Video Recordings |</td><td>analog film transfer master and digital surrogate; B&W; 00:05:00 |</td><td></td></tr>
<tr><td>Viewing copies |</td><td>8 | 25 | 66.V001.00 | 8 | 26 | 66.V001.02 | 8 | 27 | 66.V001.04 |</td><td>0.06 Cubic Feet | 3 Video Recordings |</td><td>access copies; 00:05:04 |</td><td></td></tr>
<tr><td>"The 1/4 Mile or 2 Furlong Piece" (1981–1998) |</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Sound track |</td><td>30 | 7 | 87.A001.00 |</td><td>0.002 Cubic Feet | 1 Sound Recordings |</td><td>emergency backup copy of installation sound track; 01:00:14 |</td><td></td></tr>
<tr><td>Documentation |</td><td></td><td></td><td></td><td></td></tr>
<tr><td>"7 Characters" series (1982) |</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Working segment |</td><td>8 | 28 | 85.V094.00 |</td><td>0.02 Cubic Feet | 2 Video Recordings |</td><td>analog working segment and digital surrogate; 00:30:48 |</td><td></td></tr>
<tr><td>"A Quake in Paradise (Labyrinth)" (1994) |</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Installation, Captiva |</td><td>30 | 8 | 95.V008.00 |</td><td>0.005 Cubic Feet | 2 Video Recordings |</td><td>analog raw footage and digital surrogate; 00:10:54 |</td><td></td></tr>
<tr><td>"Art Car-BMW" (1986) |</td><td></td><td></td><td></td><td></td></tr>
<tr><td>"BMW Art Cars." CNN-NY |</td><td>37 | 30 | 89.V024.00 |</td><td>0.05 Cubic Feet | 2 Video Recordings |</td><td>duplicate of broadcast videorecording and digital surrogate; 00:02:24 |</td><td></td></tr>
<tr><td>"Bicycloid VI" (1993) |</td><td></td><td></td><td></td><td></td></tr>
</table>

Result should look something like this. Values are separated with pipes and hierarchies are represented in unittitle. Don't change the column order!!

Github!!

All files related to this project, including XML files, working Python codes, clean-finished codes, and codes with instructions are located ...

https://github.com/Super10veBug/rrfa-pfch-2019

Thank you!

Feel free to e-mail questions or concerns