1. Output of the first csv file.

File	Author	Email	Affiliation
ab-104.xml	Douwe Zeldenrust	douwe.zeldenrust@meertens.knaw.nl	Meertens Instituut (Royal Netherlands Academy of Arts and Sciences), Netherlands, The
ab-104.xml	Marc Van Oostendorp	m.van.oostendorp@hum.leidenuniv.nl	Leiden University, Netherlands, The
ab-106.xml	Md. Anwarul Islam	anwar81du@gmail.com	Univeristy of Dhaka
ab-107.xml	Willard McCarty	willard.mccarty@kcl.ac.uk	King's College London, United Kingdom
ab-108.xml	Desmond Schmidt	desmond.allan.schmidt@gmail.com	University of Queensland
ab-110.xml	Pavel Thomas Cenkl	pcenkl@sterlingcollege.edu	Sterling College, United States of America

```
from lxml import etree
import csv
import glob # find all the xml files
xml_files = glob.glob('*.xml')
print(xml_files) # check to see that what are the files
['ab-104.xml', 'ab-106.xml', 'ab-107.xml', 'ab-108.xml', 'ab-110.xml']
# now we can loop through all the files
results = []  # define an empty list to collect our data
for file_path in xml_files:
        # the etree.parse function allows us to just give it a file name
        # meaning that we don't need to open it up on our own
        tree = etree.parse(file_path)
        root = tree.getroot() # this is just part of the etree parsing formula
        ns = {'tei': 'http://www.tei-c.org/ns/1.0'}
        # namespace defined in order to have the xpath parse it
        document_id = file_path.split('/')[-1]
        # get the file name by split the slash and take the last element
        authors = root.xpath("//tei:author", namespaces = ns)
        # so we first find the author element
        for name in authors:
                # now we can grab the data we want out of each
                surname = name.xpath("tei:name/tei:surname/text()", namespaces = ns)
                forename = name.xpath("tei:name/tei:forename/text()", namespaces = ns)
                email = name.xpath("tei:email/text()", namespaces = ns)
                affiliation = name.xpath("tei:affiliation/text()", namespaces = ns)
                # all the results come back as lists
                name = forename[0] + " " + surname[0] # so use [0] to select the element
                sub_results = [document_id, name, email[0], affiliation[0]]
                results.append(sub_results)


for row in results:
        print(row)


['ab-104.xml', 'Douwe Zeldenrust', 'douwe.zeldenrust@meertens.knaw.nl', 'Meertens Instituut (Royal Neth
['ab-104.xml', 'Marc Van Oostendorp', 'm.van.oostendorp@hum.leidenuniv.nl ', 'Leiden University, Nether
['ab-106.xml', 'Md. Anwarul Islam', 'anwar81du@gmail.com', 'Univeristy of Dhaka']
['ab-107.xml', 'Willard McCarty', 'willard.mccarty@kcl.ac.uk', "King's College London, United Kingdom"]
['ab-108.xml', 'Desmond Schmidt', 'desmond.allan.schmidt@gmail.com', 'University of Queensland']
['ab-110.xml', 'Pavel Thomas Cenkl', 'pcenkl@sterlingcollege.edu', 'Sterling College, United States of

headers = ["File", "Author", "Email", "Affiliation"]
# now we have a list of lists, so we can write a CSV.
with open("author_info.csv", 'wt') as file_out:
        file_out = csv.writer(file_out)
        file_out.writerow(headers)
        file_out.writerows(results)

31
```

2. Output of the second csv file.

File	Publisher	Distributor	Address	Publication Place
ab-104.xml	University of Nebraska-Lincoln	Center for Digital Research in the Humanities	319 Love Library University of NebraskaLincoln Lincoln, NE 68588-4100	Lincoln, Nebraska
ab-106.xml	University of Nebraska-Lincoln	Center for Digital Research in the Humanities	319 Love Library University of NebraskaLincoln Lincoln, NE 68588-4100	Lincoln, Nebraska
ab-107.xml	University of Nebraska-Lincoln	Center for Digital Research in the Humanities	319 Love Library University of NebraskaLincoln Lincoln, NE 68588-4100	Lincoln, Nebraska
ab-108.xml	University of Nebraska-Lincoln	Center for Digital Research in the Humanities	319 Love Library University of NebraskaLincoln Lincoln, NE 68588-4100	Lincoln, Nebraska
ab-110.xml	University of Nebraska-Lincoln	Center for Digital Research in the Humanities	319 Love Library University of NebraskaLincoln Lincoln, NE 68588-4100	Lincoln, Nebraska

```
results = [] # redefine an empty list to collect our data
for file_path in xml_files:
        tree = etree.parse(file_path)
        root = tree.getroot()
        ns = {'tei': 'http://www.tei-c.org/ns/1.0'}
        document_id = file_path.split('/')[-1]
        # for this xpath expression, I want to get information on publication,
        # so I find the publicationStmt element.
        publications = root.xpath("//tei:publicationStmt", namespaces = ns)
        for r in publications:
                publisher = r.xpath("tei:publisher/text()", namespaces = ns)
                distributor = r.xpath("tei:distributor/tei:name/text()", namespaces = ns)
                address = r.xpath("tei:distributor/tei:address/tei:addrLine/text()", namespaces = ns)
                # When I try to join the address, a error message returns "UnicodeEncodeError: 'ascii'
                # code can't encode character u'\u2013' in position 50: ordinal not in range(128)"
                # so i add .encode('ascii', 'ignore').decode('ascii')
                # fulladdress returns a string, others are list so need to add [0]
                fulladdress = ' '.join(address[0:3]).encode('ascii', 'ignore').decode('ascii')
                pubplace = r.xpath("tei:pubPlace/text()", namespaces = ns)
                sub_results = [document_id, publisher[0], distributor[0], fulladdress, pubplace[0]]
                results.append(sub_results)


for row in results:
        print(row)
```

```
['ab-104.xml', 'University of Nebraska-Lincoln', 'Center for Digital Research in the Humanities', '319
['ab-106.xml', 'University of Nebraska-Lincoln', 'Center for Digital Research in the Humanities', '319
['ab-107.xml', 'University of Nebraska-Lincoln', 'Center for Digital Research in the Humanities', '319
['ab-108.xml', 'University of Nebraska-Lincoln', 'Center for Digital Research in the Humanities', '319
['ab-110.xml', 'University of Nebraska-Lincoln', 'Center for Digital Research in the Humanities', '319
```

```
headers = ["File", "Publisher", "Distributor", "Address", "Publication Place"]
with open("publication_info.csv", 'wt') as file_out:
        file_out = csv.writer(file_out)
        file_out.writerow(headers)
        file_out.writerows(results)
```

```
54
```

3. Output of the third csv file

File	Title	Category	Subcategory	Keywords	Topics
ab-104.xml	Combining tailor made research solu	Paper	Short Paper	Infrastructures, Virtual Research Environments, Phonology	archives, r
ab-106.xml	Reading Habits & Attitude in the Di	Paper	Short Paper	reading habits, attitudes, online reading, digital environment, Bangladesh	digital hum
ab-107.xml	Becoming interdisciplinary	Paper	Long Paper	interdisciplinary, research, ethnography, disciplines	user studie
ab-108.xml	Collation on the Web	Paper	Short Paper	collation web-application	scholarly e
ab-110.xml	A New Ecological Model for Learning	Paper	Short Paper	writing, ecology, systems, liberal arts, environment	digital hum

```
results = [] # redefine an empty list to collect our data
for file_path in xml_files:
        tree = etree.parse(file_path)
        root = tree.getroot()
        ns = {'tei': 'http://www.tei-c.org/ns/1.0'}
        document_id = file_path.split('/')[-1]
        title = root.xpath("//tei:titleStmt/tei:title/text()", namespaces = ns)
        # except the title, others are under textClass element under profileDesc elements
        profileDesc = root.xpath("//tei:profileDesc/tei:textClass", namespaces = ns)
        for r in profileDesc:
                # since there are "n" attributes have different values under keywords elements,
                # I have to use predicates to find the node that contains a specific value.
                # Selects all the keywords elements that have a "n" attribute with a value of "category"
                category = r.xpath("tei:keywords[@n='category']/tei:term/text()", namespaces = ns)
                subcategory = r.xpath("tei:keywords[@n='subcategory']/tei:term/text()", namespaces = ns)
                keyword = r.xpath("tei:keywords[@n='keywords']/tei:term/text()", namespaces = ns)
                keywords = ', '.join(keyword).encode('ascii', 'ignore').decode('ascii')
                topic = r.xpath("tei:keywords[@n='topic']/tei:term/text()", namespaces = ns)
                topics = ', '.join(topic).encode('ascii', 'ignore').decode('ascii')
                sub_results = [document_id, title[0], category[0], subcategory[0], keywords, topics]
                results.append(sub_results)


for row in results:
        print(row)
```

```
['ab-104.xml', 'Combining tailor made research solutions with big infrastructures: The speaking map of the
['ab-106.xml', 'Reading Habits & Attitude in the Digital Environment: A Study on Dhaka University Students'
['ab-107.xml', 'Becoming interdisciplinary', 'Paper', 'Long Paper', 'interdisciplinary, research, ethnograp
['ab-108.xml', 'Collation on the Web', 'Paper', 'Short Paper', 'collation web-application', 'scholarly edit
['ab-110.xml', 'A New Ecological Model for Learning', 'Paper', 'Short Paper', 'writing, ecology, systems, l
```

```
headers = ["File", "Title", "Category", "Subcategory", "Keywords", "Topics"]
with open("paper_info.csv", 'wt') as file_out:
        file_out = csv.writer(file_out)
        file_out.writerow(headers)
        file_out.writerows(results)
```