```
import json
json_data = open('/data-text.json').read()
data = json.loads(json_data)
for item in data:
  print(item)
```

```
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at age 60 (years)', 'PU
    : 'Life expectancy at birth (years)', 'PUB
    : 'Life expectancy at age 60 (years)', 'PU
```

File "<tokenize>", line 31 record •••

You

Please explain the error:

```
  File "<tokenize>", line
    record[rec_key] = rec_
    ^
IndentationError: unindent
```

Colab AI

The indentation error occurred because the line `record[rec_key] =` `rec_value` is indented one level more than the previous line. To fix this, the indentation of this line should be reduced by one level.

```
# add the key/value pair
record[rec_key] = rec_va
# append the record dict
all_data.append(record)
```

Use code with caution

The corrected code:

```
"""
parses an XML file, extr
organizes the data into
"""
from xml.etree import El
# parse the XML file
```

```
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at birth (years)', 'PUB
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at birth (years)', 'PUB
: 'Life expectancy at birth (years)', 'PUB
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at birth (years)', 'PUB
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at birth (years)', 'PUB
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at birth (years)', 'PUB
: 'Life expectancy at birth (years)', 'PUB
: 'Life expectancy at age 60 (years)', 'PU
: 'Life expectancy at birth (years)', 'PUB
```

```
# parse the XML file
tree = ET.parse('/data-t
# get the root element
root = tree.getroot()
# find the 'Data' elemen
data = root.find('Data')
# create an empty list t
all_data = []
# loop through each 'obs
for observation in data:
  # create a dictionary
  record = {}
  # loop through each it
  for item in observatio
    # determine the key
    lookup_key = list(it
    # if the record is N
    if lookup_key == 'Nu
      rec_key = 'NUMERIC
      rec_value = item.a
    else:
      # if the not Numer
      rec_key = item.att
      rec_value = item.a
    # add the key/value
    record[rec_key] = re
  # append the record di
  all_data.append(record
# print the list of data
print(all_data)
```

Use code with caution

Rate this answer

```
"""
parses an XML file, extracts specific data from
organizes the data into a list of dictionaries
"""
from xml.etree import ElementTree as ET
# parse the XML file
tree = ET.parse('/data-text.xml')
# get the root element
root = tree.getroot()
# find the 'Data' element within the root
data = root.find('Data')
# create an empty list to store the data
all_data = []
# loop through each 'observation' element withi
for observation in data:
  # create a dictionary
  record = {}
  # loop through each item with the observation
  for item in observation:
    # determine the key based on the attribute
    lookup_key = list(item.attrib.keys())[0]
    # if the record is Numeric, set the record
    if lookup_key == 'Numeric':
      rec_key = 'NUMERIC'
      rec_value = item.attrib['Numeric']
    else:
      # if the not Numeric, set the record key
      rec_key = item.attrib[lookup_key]
      rec_value = item.attrib['Code']
    # add the key/value pair to the dictionary
    record[rec_key] = rec_value
  # append the record dictionary to the list of
  all_data.append(record)
# print the list of data
print(all_data)
```

```
[{'PUBLISHSTATE': 'PUBLISHED', 'YEAR': '19
```