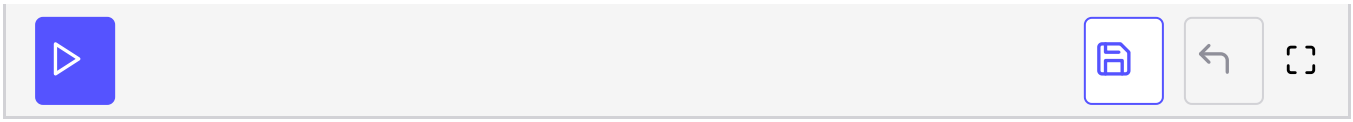
Parsing the Book Example

Well, the result of that example was kind of boring. Most of the time, you want to save the data you extract and do something with it, not just print it out to stdout. So for our next example, we'll create a data structure to contain the results. Our data structure for this example will be a list of dicts. We'll use the MSDN book example here from the earlier chapter again. Save the following XML as *example.xml*

```
<?xml version="1.0"?>
<catalog>
   <book id="bk101">
      <author>Gambardella, Matthew</author>
      <title>XML Developer's Guide</title>
      <genre>Computer</genre>
      <price>44.95</price>
      <publish_date>2000-10-01</publish_date>
      <description>An in-depth look at creating applications
      with XML.</description>
   </book>
   <book id="bk102">
      <author>Ralls, Kim</author>
      <title>Midnight Rain</title>
      <genre>Fantasy</genre>
      <price>5.95</price>
      <publish_date>2000-12-16</publish_date>
      <description>A former architect battles corporate zombies,
      an evil sorceress, and her own childhood to become queen
      of the world.</description>
   </book>
   <book id="bk103">
      <author>Corets, Eva</author>
      <title>Maeve Ascendant</title>
      <genre>Fantasy</genre>
      <price>5.95</price>
      <publish_date>2000-11-17</publish_date>
      <description>After the collapse of a nanotechnology
      society in England, the young survivors lay the
      foundation for a new society.</description>
   </book>
</catalog>
```

Now let's parse this XML and put it in our data structure!

```
from lxml import etree


def parseBookXML(xmlFile):

    with open(xmlFile) as fobj:
        xml = fobj.read()

    root = etree.fromstring(xml)

    book_dict = {}
    books = []
    for book in root.getchildren():
        for elem in book.getchildren():
            if not elem.text:
                text = "None"
            else:
                text = elem.text
            print(elem.tag + " => " + text)
            book_dict[elem.tag] = text
        if book.tag == "book":
            books.append(book_dict)
            book_dict = {}
    return books

if __name__ == "__main__":
    parseBookXML("example.xml")
```



This example is pretty similar to our last one, so we'll just focus on the differences present here. Right before we start iterating over the context, we create an empty dictionary object and an empty list. Then inside the loop, we create our dictionary like this:

```
book_dict[elem.tag] = text
```

The text is either **elem.text** or **None**. Finally, if the tag happens to be **book**, then we're at the end of a book section and need to add the dict to our list as well as reset the dict for the next book. As you can see, that is exactly what we have done. A more realistic example would be to put the extracted data into a **Book** class. I have done the latter with json feeds before.

Now we're ready to learn how to parse XML with **lxml.objectify**!