

The Chronicles of

► PyXML

Name: Poonam Yadav

Company: Ansys Software

Contact Details: [LinkedIn](#) / writetowriter2@gmail.com

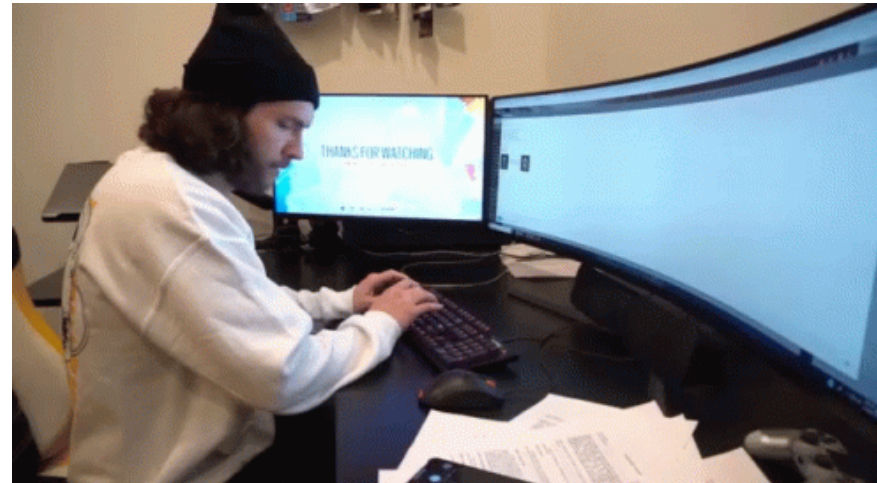
A little bit about me...



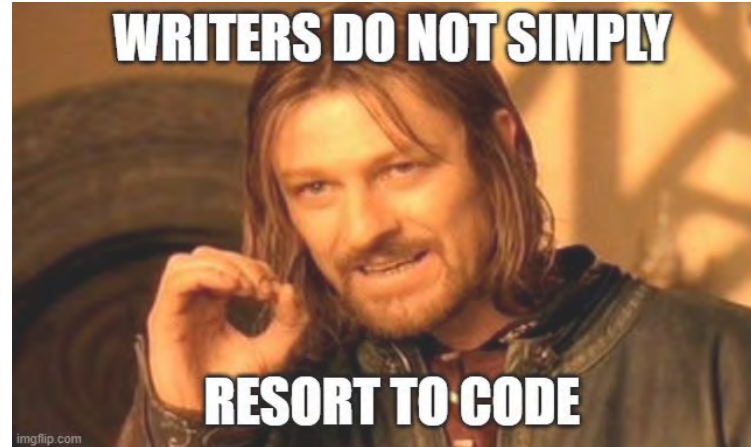
Content Writer



Technical Writer



The Requirement



- ▶ What?
 - ▶ A Reference PDF with 140+ topics/chapters
 - ▶ Around 140+ APIs which are available in a python dictionary. These APIs are well structured and have decent content around their usage, arguments, options etc.
- ▶ Why?
 - ▶ Improve Customer Experience
 - ▶ Currently, the help for these APIs are only available through the tool GUI or command prompt. The user does not have an easy-access document.
- ▶ Hindrances
 - ▶ Less time, Less resources, Complex input

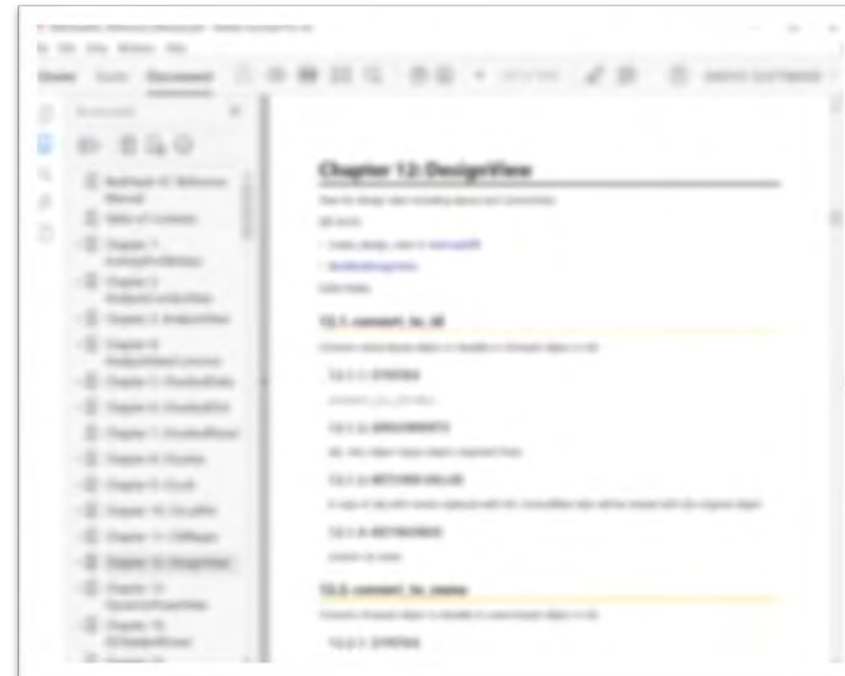
Overview



- ▶ What we had?
 - ▶ Tool help
 - ▶ GUI and console



- ▶ What we wanted?
 - ▶ Easy access document
 - ▶ PDF or HTML



How did we do it?



Devise a strategy

You could use PY source as an input, or segregate and classify the data into a text file and work further on a simple script to write XML counterparts.

Script

Start working on the algorithm. When you have decided to parse one section of the document, it kind of inspires you to write algos for all sections. You would then start improvising with bullets, formats and sub-sections.

Understand the Input

Automation:
Extract and segregate the input

Map input-structure to XML

Automation:
Text to XMLs

Collate all XMLs into Bookmap

Run and Fix Validation Errors

Publish PDF/HTML

Input

Understanding the structure, sub-modules, lists, identifying start and end of a list.

Output

Decide what XML tags would display relevant content. Have a clear idea for all sections.

Input Structure

Understand the Input

Automation:
Extract and
segregate the
input

Map input-
structure to XML

Automation:
Text to XMLs

Collate all XMLs
into Bookmap

Run and Fix
Validation Errors

Publish
PDF/HTML



Full help dictionaries

C_Element (list of dicts)

Name

Brief

Details

See Also

Functions (list of dicts)

M_Element (list of dicts)

Name

Functions (list of dicts)

Function Dictionary

-name	-short desc
-syntax	-details
-argument (list of dict)	-return value

Optional:
'example' 'see also'

Subset

Function's Argument (key)

-name
-details

Optional:
'required' 'default_value'
default_behavior 'constraint'

Input Conditioning

Understand the Input

Automation:
Extract and segregate the input

Map input-structure to XML

Automation:
Text to XMLs

Collate all XMLs into Bookmap

Run and Fix Validation Errors

Publish PDF/HTML



- Here we extract the APIs and segregate the list of C_element and M_element in separate text files.





Struct Text

```
help_all_class.txt - Notepad
File Edit Format View Help
CLASS:
A_Element
    dummy content.

SEE ALSO:
    F_Element

DETAILS:

FUNCTIONS:

>> Function__A(syntax, syntax=syntax)
    dummy content.

RETURN VALUE
    dummy content.

>> Function__B(syntax, syntax=syntax)
    dummy content.

ARGUMENTS
    arg : arg def

CLASS:
B_Element
    dummy content.

SEE ALSO:
    H_Element

DETAILS:
    dummy content.

FUNCTIONS:
```


Mapping Docbook

Understand the Input

Automation:
Extract and
segregate the
input

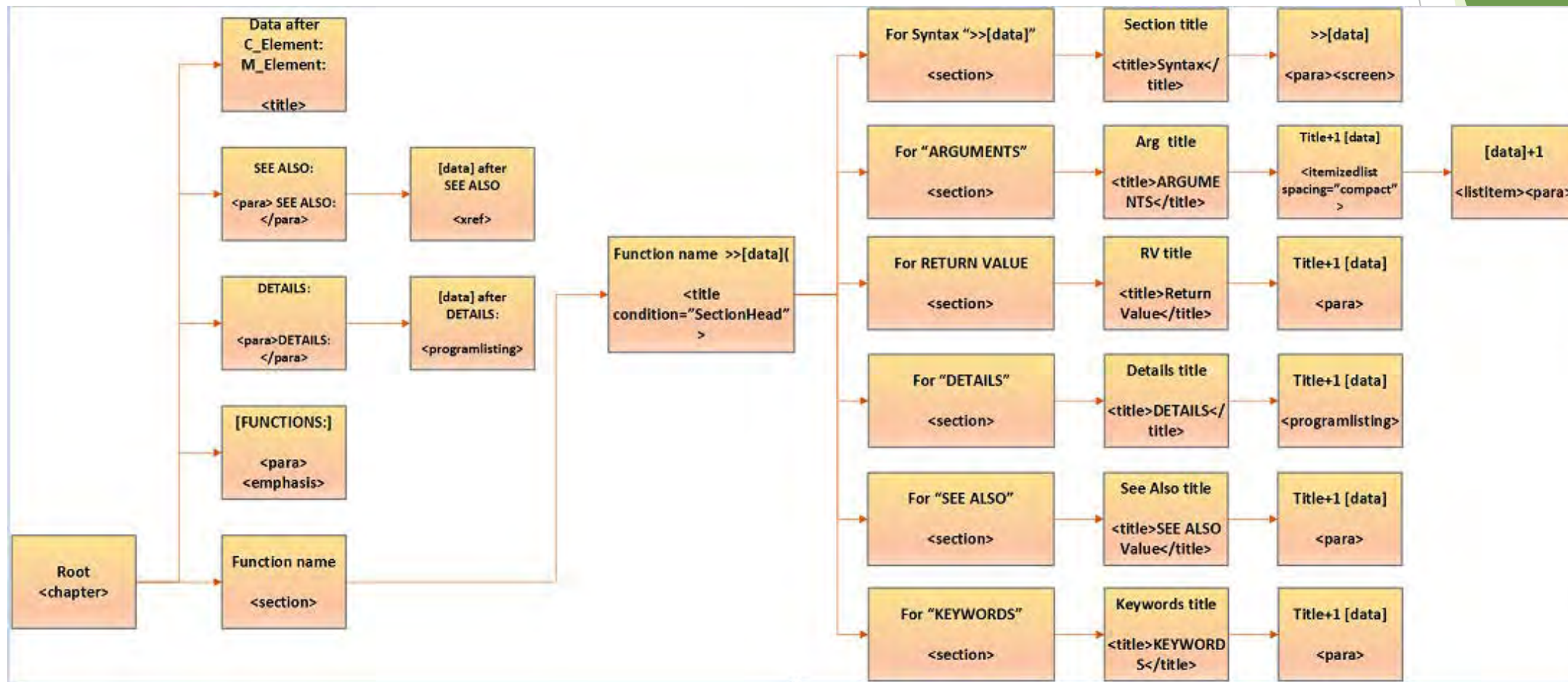
Map input-
structure to XML

Automation:
Text to XMLs

Collate all XMLs
into Bookmap

Run and Fix
Validation Errors

Publish
PDF/HTML



Automation

Understand the
Input

Automation:
Extract and
segregate the
input

Map input-
structure to XML

Automation:
Text to XMLs

Collate all XMLs
into Bookmap

Run and Fix
Validation Errors

Publish
PDF/HTML



- ▶ Written in Jupyter Notebook - PyAnsys Lab
- ▶ Key opensource module used - `xml.etree.ElementTree`
- ▶ Learn more about ElementTree on <https://docs.python.org/3/>
 - ▶ Navigate to Global Module Index.
 - ▶ Under the 'X' index, click `xml.etree.ElementTree`



- ▶ YouTube source: <https://youtu.be/j0xr0-lAqyk>



Demo Video



The screenshot shows a Jupyter Notebook interface. On the left is a file explorer with a search bar and a list of files. The main area is a code editor showing the content of 'help_all_class.txt'. The status bar at the bottom indicates the current position is Line 28, Column 10, with 4 spaces, in the file 'help_all_class.txt'.

```
1 CLASS:
2 A_Element
3 dummy content.
4
5 SEE ALSO:
6 F_Element
7
8
9 DETAILS:
10
11
12 FUNCTIONS:
13
14 >> Function__A(syntax, syntax=syntax)
15 dummy content.
16
17 RETURN VALUE
18 dummy content.
19
20 >> Function__B(syntax, syntax=syntax)
21 dummy content.
22
23 ARGUMENTS
24 arg : arg def
25
26
27 CLASS:
28 B_Element
29 dummy content.
30
31 SEE ALSO:
32 H_Element
33
34
35 DETAILS:
```

Bookmap (could be automated)

Understand the
Input

Automation:
Extract and
segregate the
input

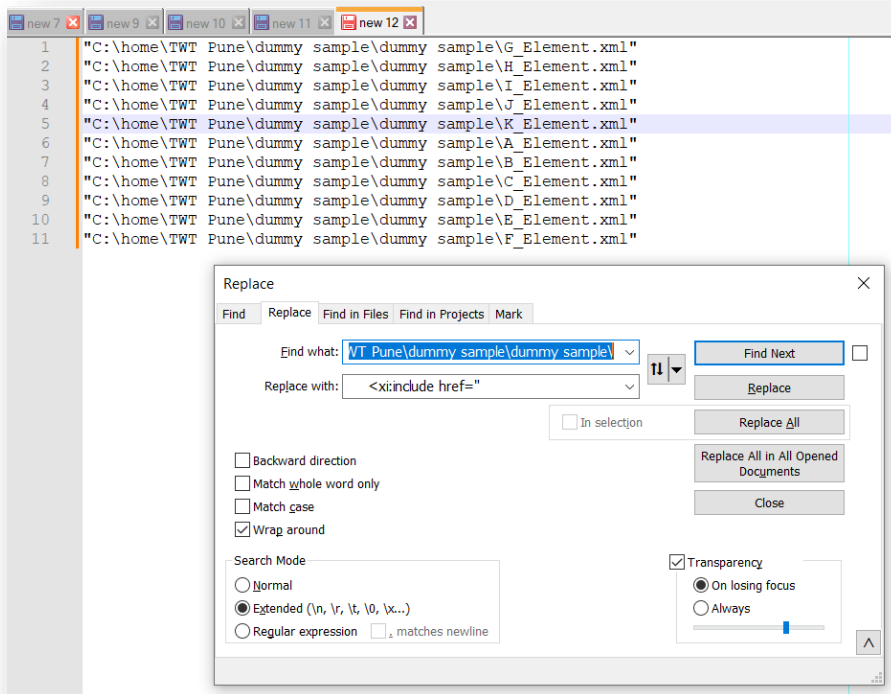
Map input-
structure to XML

Automation:
Text to XMLs

Collate all XMLs
into Bookmap

Run and Fix
Validation Errors

Publish
PDF/HTML



```
42 <toc/>
43 <xi:include href="A_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
44 <xi:include href="B_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
45 <xi:include href="C_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
46 <xi:include href="D_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
47 <xi:include href="E_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
48 <xi:include href="F_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
49 <xi:include href="G_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
50 <xi:include href="H_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
51 <xi:include href="I_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
52 <xi:include href="J_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
53 <xi:include href="K_Element.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/>
54 </book>
```


XML Validation

Understand the
Input

Automation:
Extract and
segregate the
input

Map input-
structure to XML

Automation:
Text to XMLs

Collate all XMLs
into Bookmap

Run and Fix
Validation Errors

Publish
PDF/HTML



Code running	 Panik
Code successful	 Kalm
XML validation errors	 Panik

Observations:

- ▶ MetaData, directory structure should be correct before running validation.
- ▶ Common XML validation errors:
 - ▶ Broken <xref> links
 - ▶ Empty element tags
 - ▶ List having only one item
- ▶ Try solving these errors by modifying the code.
- ▶ Try manipulating the input or the code as per your requirement.
- ▶ Focus on generating a clean book through script. Avoid manual intervention on output.

Publish

Understand the
Input

Automation:
Extract and
segregate the
input

Map input-
structure to XML

Automation:
Text to XMLs

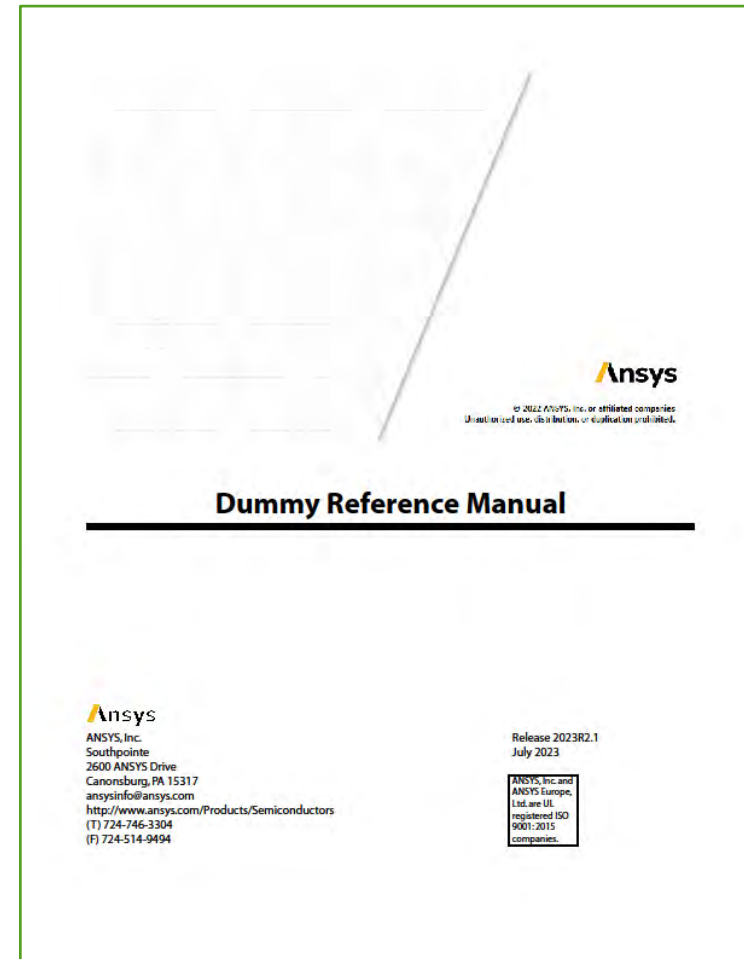
Collate all XMLs
into Bookmap

Run and Fix
Validation Errors

Publish
PDF/HTML



- Apply transformation scenarios as applicable.





Future

- ▶ Improving current documentation processes with the help of AI
 - ▶ Oxygen AI Positron Assistant Add-on
 - ▶ GPT Plugins – Code interpreter
 - ▶ OCR any document in any language and analyze
 - ▶ Clean data : Examples with csv files
 - ▶ Create tested and working python code
 - ▶ Generate charts super easy
- ▶ Doc-as-Code, DocOps, Minimalism, Content Consumption
- ▶ Career Advancements
 - ▶ Product
 - ▶ Customer Experience
 - ▶ Sales
 - ▶ And of course; Managerial.



► Thank You

Q&A