20. Structured Markup Processing Tools

Python supports a variety of modules to work with various forms of structured data markup. This includes modules to work with the Standard Generalized Markup Language (SGML) and the Hypertext Markup Language (HTML), and several interfaces for working with the Extensible Markup Language (XML).

- 20.1. `html` — HyperText Markup Language support
- 20.2. `html.parser` — Simple HTML and XHTML parser
  - 20.2.1. Example HTML Parser Application
  - 20.2.2. `HTMLParser` Methods
  - 20.2.3. Examples
- 20.3. `html.entities` — Definitions of HTML general entities
- 20.4. XML Processing Modules
  - 20.4.1. XML vulnerabilities
  - 20.4.2. The `defusedxml` and `defusedexpat` Packages
- 20.5. `xml.etree.ElementTree` — The ElementTree XML API
  - 20.5.1. Tutorial
    - 20.5.1.1. XML tree and elements
    - 20.5.1.2. Parsing XML
    - 20.5.1.3. Pull API for non-blocking parsing
    - 20.5.1.4. Finding interesting elements
    - 20.5.1.5. Modifying an XML File
    - 20.5.1.6. Building XML documents
    - 20.5.1.7. Parsing XML with Namespaces
    - 20.5.1.8. Additional resources
  - 20.5.2. XPath support
    - 20.5.2.1. Example
    - 20.5.2.2. Supported XPath syntax
  - 20.5.3. Reference
    - 20.5.3.1. Functions
    - 20.5.3.2. Element Objects
    - 20.5.3.3. ElementTree Objects
    - 20.5.3.4. QName Objects
    - 20.5.3.5. TreeBuilder Objects
    - 20.5.3.6. XMLParser Objects
    - 20.5.3.7. XMLPullParser Objects
    - 20.5.3.8. Exceptions
- 20.6. `xml.dom` — The Document Object Model API
  - 20.6.1. Module Contents
  - 20.6.2. Objects in the DOM
    - 20.6.2.1. DOMImplementation Objects
    - 20.6.2.2. Node Objects
    - 20.6.2.3. NodeList Objects

            - 20.6.2.4. DocumentType Objects
            - 20.6.2.5. Document Objects
            - 20.6.2.6. Element Objects
            - 20.6.2.7. Attr Objects
            - 20.6.2.8. NamedNodeMap Objects
            - 20.6.2.9. Comment Objects
            - 20.6.2.10. Text and CDATASection Objects
            - 20.6.2.11. ProcessingInstruction Objects
            - 20.6.2.12. Exceptions
        - 20.6.3. Conformance
            - 20.6.3.1. Type Mapping
            - 20.6.3.2. Accessor Methods
- 20.7. `xml.dom.minidom` — Minimal DOM implementation
    - 20.7.1. DOM Objects
    - 20.7.2. DOM Example
    - 20.7.3. minidom and the DOM standard
- 20.8. `xml.dom.pulldom` — Support for building partial DOM trees
    - 20.8.1. DOMEventStream Objects
- 20.9. `xml.sax` — Support for SAX2 parsers
    - 20.9.1. SAXException Objects
- 20.10. `xml.sax.handler` — Base classes for SAX handlers
    - 20.10.1. ContentHandler Objects
    - 20.10.2. DTDHandler Objects
    - 20.10.3. EntityResolver Objects
    - 20.10.4. ErrorHandler Objects
- 20.11. `xml.sax.saxutils` — SAX Utilities
- 20.12. `xml.sax.xmlreader` — Interface for XML parsers
    - 20.12.1. XMLReader Objects
    - 20.12.2. IncrementalParser Objects
    - 20.12.3. Locator Objects
    - 20.12.4. InputSource Objects
    - 20.12.5. The `Attributes` Interface
    - 20.12.6. The `AttributesNS` Interface
- 20.13. `xml.parsers.expat` — Fast XML parsing using Expat
    - 20.13.1. XMLParser Objects
    - 20.13.2. ExpatError Exceptions
    - 20.13.3. Example
    - 20.13.4. Content Model Descriptions
    - 20.13.5. Expat error constants