

etree

etree

- ▶ A “simpler” xml reader and writer
- ▶ Was elementtree (3rd party)
- ▶ Also cElementtree (for speed)
- ▶ Now xml.etree in Python 2.5
- ▶ Import to disguise origins

```
import elementtree as et
import cElementTree as et
import xml.etree as et
```

Element

- ▶ An XML node
- ▶ Acts as a list of nodes it contains
- ▶ Acts as a dictionary of attributes
- ▶ C'tor: *Element* (<tagname>, <attrib-keywords>)

```
elem = Element ("tag")
```

```
elem = Element ("tag", first="1", second="2")
```

Children

- ▶ Sub-elements (sub-nodes) are items in the Element list
- ▶ Supports list operations: indexing, slicing, append, remove etc.

```
single_elem = elem[1]
elems = elem[1:3]
elem.append (new_elem)
len (elem)
```

Attributes

- ▶ Attributes are in an `attrib` dictionary
- ▶ Can act directly on `attrib` or methods ...
- ▶ `get`, `set`, `keys`, `items`

```
>>> elem.attrib['one'] = '1'  
>>> elem.attrib['two']  
'2'  
>>> elem.set ('one', '1')  
>>> elem.get ('two')  
'2'
```

Text

Access contained (non-tag) text via `.text` member.

```
<example>some text</example>
```

```
>>> elem.text
```

```
'some text'
```

```
>>> elem.text = 'other text'
```

Building a tree

Use `SubElement` function. This looks like the `Element` constructor with an additional first argument for the parent node.

```
root = Element ("root")  
SubElement (root, "one")  
SubElement (root, "two", first="1", second="2")
```

Elementtree

- ▶ Submodule for reading files
- ▶ parse reads file
- ▶ Accepts open file or file handle

```
from etree.ElementTree import parse
file = open(filename, "r")
tree = parse(file)
elem = tree.getroot()
```

Writing elementtree

- ▶ write method produces XML
- ▶ Takes an open file or file handle
- ▶ Wrap Element to print
- ▶ No pretty-printing

```
tree = ElementTree(file=infile)
tree.write (outfile)
```

```
html = Element ("html")
body = SubElement (html, "body")
ElementTree (html).write (outfile)
```

Also ...

- ▶ Introducing Element & Elementtree
- ▶ Elementtree overview