NAME
>   XML::Parser::Style::Tree – Tree style parser

SYNOPSIS
```
use XML::Parser;
my $p = XML::Parser->new(Style => 'Tree');
my $tree = $p->parsefile('foo.xml');
```

DESCRIPTION
>   This module implements XML::Parser's Tree style parser.

>   When parsing a document, `parse()` will return a parse tree for the document. Each node in the tree takes the form of a tag, content pair. Text nodes are represented with a pseudo-tag of ''0'' and the string that is their content. For elements, the content is an array reference. The first item in the array is a (possibly empty) hash reference containing attributes. The remainder of the array is a sequence of tag-content pairs representing the content of the element.

>   So for example the result of parsing:

```
<foo><head id="a">Hello <em>there</em></head><bar>Howdy<ref/></bar>do</foo>
```

>   would be:
```
      Tag   Content
    ==================================================================
  [foo, [{}, head, [{id => "a"}, 0, "Hello ",  em, [{}, 0, "there"]],
         bar, [     {}, 0, "Howdy",  ref, [{}]],
           0, "do"
       ]
  ]
```

>   The root document ''foo'', has 3 children: a ''head'' element, a ''bar'' element and the text ''do''. After the empty attribute hash, these are represented in it's contents by 3 tag-content pairs.