LeanDog
AN AGILE SOFTWARE STUDIO

Nokogiri Quick Reference

Parsing an XML Document

From a String

```
xml_doc = Nokogiri::XML("<root><aliens><alien><name>Alf</name></alien></aliens></root>")
```

From a File

```
f = File.open("blossom.xml")
xml_doc = Nokogiri::XML(f)
f.close
```

From the Internet

```
require 'open-uri'
xml_doc = Nokogiri::XML(open("http://somesite.com/example.xml"))
```

Parse Options

Nokogiri offers quite a few options that affect how a document is parsed. Here are a few:

- NOBLANKS - Remove blank nodes
- NOENT - Substitute entities
- NOERROR - Suppress error reports
- STRICT - Strict parsing; raise an error when parsing malformed documents

Here's how they are used:

```
doc = Nokogiri::XML(File.open("blossom.xml")) do |config|
  config.strict.noent
end
```

or

```
doc = Nokogiri::XML(File.open("blossom.xml")) do |config|
  config.options = Nokogiri::XML::ParseOptions.STRICT | Nokogiri::XML::ParseOptions.NOENT
end
```

Basic Searching

Let's suppose you have an xml document that has a list of television shows. To find all of the characters in the shows you could execute the following:

```
doc = Nokogiri::XML(File.open("shows.xml"))
chacacters = doc.xpath("//character")  # or doc.css("character")
```

The methods xpath and css return a NodeSet, which acts very much like an array and contains matching nodes from the document. To get the data within a single Node you can use the content method.

```
characters[0].to_s  # "<character>Al Bundy</character>
```

Single Results

If you know you're going to get only a single result back you can use the shortcuts at_css and at_xpath instead of having to access the first element of a NodeSet.

```
doc.css("dramas name").first   # "<name>The A-Team</name>"
doc.at_css("dramas name")     # "<name>The A-Team</name>"
```