Introduction

The dependencies page lists all the jars that you will need to have in your classpath.

The class com.steadystate.css.parser.CSSOMParser is a good starting point for writing your first parser.

```
InputSource source = new InputSource(new StringReader("h1 { background: #ffcc44; }"));
CSSOMParser parser = new CSSOMParser(new SACParserCSS3());
CSSStyleSheet sheet = parser.parseStyleSheet(source, null, null);
```

Processing the rules

The method getCssRules() offers all rules available.

```
CSSRuleList rules = sheet.getCssRules();
for (int i = 0; i < rules.getLength(); i++) {
    final CSSRule rule = rules.item(i);

    System.out.println(rule.getCssText());
}
```

Error handling

CSSParser error handling is based on org.w3c.css.sac.ErrorHandler and CSSParseException. You have to write your own implementation of the ErrorHandler interface and register it before start parsing.

```
InputSource source = new InputSource(new StringReader("h1 { background: #red; }"));
CSSOMParser parser = new CSSOMParser(new SACParserCSS3());

ErrorHandler errorHandler = new MyErrorHandler();
parser.setErrorHandler(errorHandler);

CSSStyleSheet sheet = parser.parseStyleSheet(source, null, null);
```

```
public static class MyErrorHandler implements ErrorHandler {

    public void warning(CSSParseException exception) throws CSSException {
        System.out.println("Warning: " + exception.getMessage());
    }

    public void error(CSSParseException exception) throws CSSException {
        System.out.println("Error: " + exception.getMessage());
    }

    public void fatalError(CSSParseException exception) throws CSSException {
        System.out.println("Fatal: " + exception.getMessage());
    }
}
```

Parsing single style declarations

It is also possible to parse only style declarations instead of a whole style sheet.

```
InputSource source = new InputSource(new StringReader("background: #ffcc44; margin-top: 4px"));
CSSOMParser parser = new CSSOMParser(new SACParserCSS3());
CSSStyleDeclaration decl = parser.parseStyleDeclaration(source);

for (int i = 0; i < decl.getLength(); i++) {
    final String propName = decl.item(i);

    System.out.println("'" + propName + "' has value: '" + decl.getPropertyValue(propName) +
 "'");
}
```