

CSSParser can work with different kinds of input by setting up an instance of class `org.w3c.css.sac.InputSource` in different ways.

## String input

Working with strings as input is the simplest case because there is no need to take care of any encoding.

```
InputSource source = new InputSource(new StringReader("h1 { background: #ffcc44; }"));
CSSOMParser parser = new CSSOMParser(new SACParserCSS3());

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

## Using an InputStreamReader

When using an `InputStreamReader` the reader itself is responsible for the correct conversion from the bytes (e.g. taken from a file) into characters. Therefore you have to provide the correct encoding as second parameter of the `InputStreamReader` constructor.

```
InputStream inStream = new FileInputStream("input.css");
try {
    InputSource source = new InputSource(new InputStreamReader(inStream, "UTF-8"));

    CSSOMParser parser = new CSSOMParser(new SACParserCSS3());
    ....
} finally {
    inStream.close();
}
```

## Using an InputStream

If you like to use an `InputStream` as parser input, you have to inform the `InputSource` about the correct encoding of your input stream.

```
InputStream inStream = new FileInputStream("input.css");
try {
    InputSource source = new InputSource();
    source.setByteStream(inStream);
    source.setEncoding("UTF-8");

    CSSOMParser parser = new CSSOMParser(new SACParserCSS3());
    ....
} finally {
    inStream.close();
}
```