| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/StyleSheet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/text/html/ParagraphView.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/StyleSheet.BoxPainter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/StyleSheet.html)    [**NO FRAMES**](http://docs.google.com/StyleSheet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

## **javax.swing.text.html**

Class StyleSheet

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.text.StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html)  
 **javax.swing.text.html.StyleSheet**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html)

public class **StyleSheet**extends [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html)

Support for defining the visual characteristics of HTML views being rendered. The StyleSheet is used to translate the HTML model into visual characteristics. This enables views to be customized by a look-and-feel, multiple views over the same model can be rendered differently, etc. This can be thought of as a CSS rule repository. The key for CSS attributes is an object of type CSS.Attribute. The type of the value is up to the StyleSheet implementation, but the toString method is required to return a string representation of CSS value.

The primary entry point for HTML View implementations to get their attributes is the [getViewAttributes](#2lwamvv) method. This should be implemented to establish the desired policy used to associate attributes with the view. Each HTMLEditorKit (i.e. and therefore each associated JEditorPane) can have its own StyleSheet, but by default one sheet will be shared by all of the HTMLEditorKit instances. HTMLDocument instance can also have a StyleSheet, which holds the document-specific CSS specifications.

In order for Views to store less state and therefore be more lightweight, the StyleSheet can act as a factory for painters that handle some of the rendering tasks. This allows implementations to determine what they want to cache and have the sharing potentially at the level that a selector is common to multiple views. Since the StyleSheet may be used by views over multiple documents and typically the HTML attributes don't effect the selector being used, the potential for sharing is significant.

The rules are stored as named styles, and other information is stored to translate the context of an element to a rule quickly. The following code fragment will display the named styles, and therefore the CSS rules contained.

  import java.util.\*;  
   import javax.swing.text.\*;  
   import javax.swing.text.html.\*;  
     
   public class ShowStyles {  
     
   public static void main(String[] args) {  
   HTMLEditorKit kit = new HTMLEditorKit();  
   HTMLDocument doc = (HTMLDocument) kit.createDefaultDocument();  
   StyleSheet styles = doc.getStyleSheet();  
     
   Enumeration rules = styles.getStyleNames();  
   while (rules.hasMoreElements()) {  
   String name = (String) rules.nextElement();  
   Style rule = styles.getStyle(name);  
   System.out.println(rule.toString());  
   }  
   System.exit(0);  
   }  
   }  
  

The semantics for when a CSS style should overide visual attributes defined by an element are not well defined. For example, the html <body bgcolor=red> makes the body have a red background. But if the html file also contains the CSS rule body { background: blue } it becomes less clear as to what color the background of the body should be. The current implemention gives visual attributes defined in the element the highest precedence, that is they are always checked before any styles. Therefore, in the previous example the background would have a red color as the body element defines the background color to be red.

As already mentioned this supports CSS. We don't support the full CSS spec. Refer to the javadoc of the CSS class to see what properties we support. The two major CSS parsing related concepts we do not currently support are pseudo selectors, such as A:link { color: red }, and the important modifier.

Note: This implementation is currently incomplete. It can be replaced with alternative implementations that are complete. Future versions of this class will provide better CSS support.

| **Nested Class Summary** | |
| --- | --- |
| static class | [**StyleSheet.BoxPainter**](http://docs.google.com/javax/swing/text/html/StyleSheet.BoxPainter.html)            Class to carry out some of the duties of CSS formatting. |
| static class | [**StyleSheet.ListPainter**](http://docs.google.com/javax/swing/text/html/StyleSheet.ListPainter.html)            Class to carry out some of the duties of CSS list formatting. |

| **Nested classes/interfaces inherited from class javax.swing.text.**[**StyleContext**](http://docs.google.com/javax/swing/text/StyleContext.html) |
| --- |
| [StyleContext.NamedStyle](http://docs.google.com/javax/swing/text/StyleContext.NamedStyle.html), [StyleContext.SmallAttributeSet](http://docs.google.com/javax/swing/text/StyleContext.SmallAttributeSet.html) |

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class javax.swing.text.**[**StyleContext**](http://docs.google.com/javax/swing/text/StyleContext.html) |
| --- |
| [DEFAULT\_STYLE](http://docs.google.com/javax/swing/text/StyleContext.html#DEFAULT_STYLE) |

| **Constructor Summary** | |
| --- | --- |
| [**StyleSheet**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#StyleSheet())()            Construct a StyleSheet |

| **Method Summary** | |
| --- | --- |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**addAttribute**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#addAttribute(javax.swing.text.AttributeSet,%20java.lang.Object,%20java.lang.Object))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old, [Object](http://docs.google.com/java/lang/Object.html) key, [Object](http://docs.google.com/java/lang/Object.html) value)            Adds an attribute to the given set, and returns the new representative set. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**addAttributes**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#addAttributes(javax.swing.text.AttributeSet,%20javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Adds a set of attributes to the element. |
| void | [**addCSSAttribute**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#addCSSAttribute(javax.swing.text.MutableAttributeSet,%20javax.swing.text.html.CSS.Attribute,%20java.lang.String))([MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) attr, [CSS.Attribute](http://docs.google.com/javax/swing/text/html/CSS.Attribute.html) key, [String](http://docs.google.com/java/lang/String.html) value)            Adds a CSS attribute to the given set. |
| boolean | [**addCSSAttributeFromHTML**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#addCSSAttributeFromHTML(javax.swing.text.MutableAttributeSet,%20javax.swing.text.html.CSS.Attribute,%20java.lang.String))([MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) attr, [CSS.Attribute](http://docs.google.com/javax/swing/text/html/CSS.Attribute.html) key, [String](http://docs.google.com/java/lang/String.html) value)            Adds a CSS attribute to the given set. |
| void | [**addRule**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#addRule(java.lang.String))([String](http://docs.google.com/java/lang/String.html) rule)            Adds a set of rules to the sheet. |
| void | [**addStyleSheet**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#addStyleSheet(javax.swing.text.html.StyleSheet))([StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html) ss)            Adds the rules from the StyleSheet ss to those of the receiver. |
| protected  [MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) | [**createLargeAttributeSet**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#createLargeAttributeSet(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Creates a large set of attributes that should trade off space for time. |
| protected  [StyleContext.SmallAttributeSet](http://docs.google.com/javax/swing/text/StyleContext.SmallAttributeSet.html) | [**createSmallAttributeSet**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#createSmallAttributeSet(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Creates a compact set of attributes that might be shared. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getBackground**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getBackground(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Takes a set of attributes and turn it into a background color specification. |
| [URL](http://docs.google.com/java/net/URL.html) | [**getBase**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getBase())()            Returns the base. |
| [StyleSheet.BoxPainter](http://docs.google.com/javax/swing/text/html/StyleSheet.BoxPainter.html) | [**getBoxPainter**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getBoxPainter(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Fetches the box formatter to use for the given set of CSS attributes. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**getDeclaration**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getDeclaration(java.lang.String))([String](http://docs.google.com/java/lang/String.html) decl)            Translates a CSS declaration to an AttributeSet that represents the CSS declaration. |
| [Font](http://docs.google.com/java/awt/Font.html) | [**getFont**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getFont(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Fetches the font to use for the given set of attributes. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getForeground**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getForeground(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Takes a set of attributes and turn it into a foreground color specification. |
| static int | [**getIndexOfSize**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getIndexOfSize(float))(float pt) |
| [StyleSheet.ListPainter](http://docs.google.com/javax/swing/text/html/StyleSheet.ListPainter.html) | [**getListPainter**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getListPainter(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Fetches the list formatter to use for the given set of CSS attributes. |
| float | [**getPointSize**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getPointSize(int))(int index)            Returns the point size, given a size index. |
| float | [**getPointSize**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getPointSize(java.lang.String))([String](http://docs.google.com/java/lang/String.html) size)            Given a string such as "+2", "-2", or "2", returns a point size value. |
| [Style](http://docs.google.com/javax/swing/text/Style.html) | [**getRule**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getRule(javax.swing.text.html.HTML.Tag,%20javax.swing.text.Element))([HTML.Tag](http://docs.google.com/javax/swing/text/html/HTML.Tag.html) t, [Element](http://docs.google.com/javax/swing/text/Element.html) e)            Fetches the style to use to render the given type of HTML tag. |
| [Style](http://docs.google.com/javax/swing/text/Style.html) | [**getRule**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getRule(java.lang.String))([String](http://docs.google.com/java/lang/String.html) selector)            Fetches the rule that best matches the selector given in string form. |
| [StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html)[] | [**getStyleSheets**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getStyleSheets())()            Returns an array of the linked StyleSheets. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**getViewAttributes**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#getViewAttributes(javax.swing.text.View))([View](http://docs.google.com/javax/swing/text/View.html) v)            Fetches a set of attributes to use in the view for displaying. |
| void | [**importStyleSheet**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#importStyleSheet(java.net.URL))([URL](http://docs.google.com/java/net/URL.html) url)            Imports a style sheet from url. |
| void | [**loadRules**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#loadRules(java.io.Reader,%20java.net.URL))([Reader](http://docs.google.com/java/io/Reader.html) in, [URL](http://docs.google.com/java/net/URL.html) ref)            Loads a set of rules that have been specified in terms of CSS1 grammar. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**removeAttribute**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#removeAttribute(javax.swing.text.AttributeSet,%20java.lang.Object))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old, [Object](http://docs.google.com/java/lang/Object.html) key)            Removes an attribute from the set. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**removeAttributes**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#removeAttributes(javax.swing.text.AttributeSet,%20javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attrs)            Removes a set of attributes. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**removeAttributes**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#removeAttributes(javax.swing.text.AttributeSet,%20java.util.Enumeration))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old, [Enumeration](http://docs.google.com/java/util/Enumeration.html)<?> names)            Removes a set of attributes for the element. |
| void | [**removeStyle**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#removeStyle(java.lang.String))([String](http://docs.google.com/java/lang/String.html) nm)            Removes a named style previously added to the document. |
| void | [**removeStyleSheet**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#removeStyleSheet(javax.swing.text.html.StyleSheet))([StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html) ss)            Removes the StyleSheet ss from those of the receiver. |
| void | [**setBase**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#setBase(java.net.URL))([URL](http://docs.google.com/java/net/URL.html) base)            Sets the base. |
| void | [**setBaseFontSize**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#setBaseFontSize(int))(int sz)            Sets the base font size, with valid values between 1 and 7. |
| void | [**setBaseFontSize**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#setBaseFontSize(java.lang.String))([String](http://docs.google.com/java/lang/String.html) size)            Sets the base font size from the passed in String. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**stringToColor**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#stringToColor(java.lang.String))([String](http://docs.google.com/java/lang/String.html) string)            Converts a color string such as "RED" or "#NNNNNN" to a Color. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**translateHTMLToCSS**](http://docs.google.com/javax/swing/text/html/StyleSheet.html#translateHTMLToCSS(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) htmlAttrSet)            Converts a set of HTML attributes to an equivalent set of CSS attributes. |

| **Methods inherited from class javax.swing.text.**[**StyleContext**](http://docs.google.com/javax/swing/text/StyleContext.html) |
| --- |
| [addChangeListener](http://docs.google.com/javax/swing/text/StyleContext.html#addChangeListener(javax.swing.event.ChangeListener)), [addStyle](http://docs.google.com/javax/swing/text/StyleContext.html#addStyle(java.lang.String,%20javax.swing.text.Style)), [getChangeListeners](http://docs.google.com/javax/swing/text/StyleContext.html#getChangeListeners()), [getCompressionThreshold](http://docs.google.com/javax/swing/text/StyleContext.html#getCompressionThreshold()), [getDefaultStyleContext](http://docs.google.com/javax/swing/text/StyleContext.html#getDefaultStyleContext()), [getEmptySet](http://docs.google.com/javax/swing/text/StyleContext.html#getEmptySet()), [getFont](http://docs.google.com/javax/swing/text/StyleContext.html#getFont(java.lang.String,%20int,%20int)), [getFontMetrics](http://docs.google.com/javax/swing/text/StyleContext.html#getFontMetrics(java.awt.Font)), [getStaticAttribute](http://docs.google.com/javax/swing/text/StyleContext.html#getStaticAttribute(java.lang.Object)), [getStaticAttributeKey](http://docs.google.com/javax/swing/text/StyleContext.html#getStaticAttributeKey(java.lang.Object)), [getStyle](http://docs.google.com/javax/swing/text/StyleContext.html#getStyle(java.lang.String)), [getStyleNames](http://docs.google.com/javax/swing/text/StyleContext.html#getStyleNames()), [readAttributes](http://docs.google.com/javax/swing/text/StyleContext.html#readAttributes(java.io.ObjectInputStream,%20javax.swing.text.MutableAttributeSet)), [readAttributeSet](http://docs.google.com/javax/swing/text/StyleContext.html#readAttributeSet(java.io.ObjectInputStream,%20javax.swing.text.MutableAttributeSet)), [reclaim](http://docs.google.com/javax/swing/text/StyleContext.html#reclaim(javax.swing.text.AttributeSet)), [registerStaticAttributeKey](http://docs.google.com/javax/swing/text/StyleContext.html#registerStaticAttributeKey(java.lang.Object)), [removeChangeListener](http://docs.google.com/javax/swing/text/StyleContext.html#removeChangeListener(javax.swing.event.ChangeListener)), [toString](http://docs.google.com/javax/swing/text/StyleContext.html#toString()), [writeAttributes](http://docs.google.com/javax/swing/text/StyleContext.html#writeAttributes(java.io.ObjectOutputStream,%20javax.swing.text.AttributeSet)), [writeAttributeSet](http://docs.google.com/javax/swing/text/StyleContext.html#writeAttributeSet(java.io.ObjectOutputStream,%20javax.swing.text.AttributeSet)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### StyleSheet

public **StyleSheet**()

Construct a StyleSheet

| **Method Detail** |
| --- |

### getRule

public [Style](http://docs.google.com/javax/swing/text/Style.html) **getRule**([HTML.Tag](http://docs.google.com/javax/swing/text/html/HTML.Tag.html) t,  
 [Element](http://docs.google.com/javax/swing/text/Element.html) e)

Fetches the style to use to render the given type of HTML tag. The element given is representing the tag and can be used to determine the nesting for situations where the attributes will differ if nesting inside of elements.

**Parameters:**t - the type to translate to visual attributese - the element representing the tag; the element can be used to determine the nesting for situations where the attributes will differ if nested inside of other elements **Returns:**the set of CSS attributes to use to render the tag

### getRule

public [Style](http://docs.google.com/javax/swing/text/Style.html) **getRule**([String](http://docs.google.com/java/lang/String.html) selector)

Fetches the rule that best matches the selector given in string form. Where selector is a space separated String of the element names. For example, selector might be 'html body tr td''

The attributes of the returned Style will change as rules are added and removed. That is if you to ask for a rule with a selector "table p" and a new rule was added with a selector of "p" the returned Style would include the new attributes from the rule "p".

### addRule

public void **addRule**([String](http://docs.google.com/java/lang/String.html) rule)

Adds a set of rules to the sheet. The rules are expected to be in valid CSS format. Typically this would be called as a result of parsing a <style> tag.

### getDeclaration

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **getDeclaration**([String](http://docs.google.com/java/lang/String.html) decl)

Translates a CSS declaration to an AttributeSet that represents the CSS declaration. Typically this would be called as a result of encountering an HTML style attribute.

### loadRules

public void **loadRules**([Reader](http://docs.google.com/java/io/Reader.html) in,  
 [URL](http://docs.google.com/java/net/URL.html) ref)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Loads a set of rules that have been specified in terms of CSS1 grammar. If there are collisions with existing rules, the newly specified rule will win.

**Parameters:**in - the stream to read the CSS grammar fromref - the reference URL. This value represents the location of the stream and may be null. All relative URLs specified in the stream will be based upon this parameter. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html)

### getViewAttributes

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **getViewAttributes**([View](http://docs.google.com/javax/swing/text/View.html) v)

Fetches a set of attributes to use in the view for displaying. This is basically a set of attributes that can be used for View.getAttributes.

### removeStyle

public void **removeStyle**([String](http://docs.google.com/java/lang/String.html) nm)

Removes a named style previously added to the document.

**Overrides:**[removeStyle](http://docs.google.com/javax/swing/text/StyleContext.html#removeStyle(java.lang.String)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**nm - the name of the style to remove

### addStyleSheet

public void **addStyleSheet**([StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html) ss)

Adds the rules from the StyleSheet ss to those of the receiver. ss's rules will override the rules of any previously added style sheets. An added StyleSheet will never override the rules of the receiving style sheet.

**Since:** 1.3

### removeStyleSheet

public void **removeStyleSheet**([StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html) ss)

Removes the StyleSheet ss from those of the receiver.

**Since:** 1.3

### getStyleSheets

public [StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html)[] **getStyleSheets**()

Returns an array of the linked StyleSheets. Will return null if there are no linked StyleSheets.

**Since:** 1.3

### importStyleSheet

public void **importStyleSheet**([URL](http://docs.google.com/java/net/URL.html) url)

Imports a style sheet from url. The resulting rules are directly added to the receiver. If you do not want the rules to become part of the receiver, create a new StyleSheet and use addStyleSheet to link it in.

**Since:** 1.3

### setBase

public void **setBase**([URL](http://docs.google.com/java/net/URL.html) base)

Sets the base. All import statements that are relative, will be relative to base.

**Since:** 1.3

### getBase

public [URL](http://docs.google.com/java/net/URL.html) **getBase**()

Returns the base.

**Since:** 1.3

### addCSSAttribute

public void **addCSSAttribute**([MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) attr,  
 [CSS.Attribute](http://docs.google.com/javax/swing/text/html/CSS.Attribute.html) key,  
 [String](http://docs.google.com/java/lang/String.html) value)

Adds a CSS attribute to the given set.

**Since:** 1.3

### addCSSAttributeFromHTML

public boolean **addCSSAttributeFromHTML**([MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) attr,  
 [CSS.Attribute](http://docs.google.com/javax/swing/text/html/CSS.Attribute.html) key,  
 [String](http://docs.google.com/java/lang/String.html) value)

Adds a CSS attribute to the given set.

**Since:** 1.3

### translateHTMLToCSS

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **translateHTMLToCSS**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) htmlAttrSet)

Converts a set of HTML attributes to an equivalent set of CSS attributes.

**Parameters:**htmlAttrSet - AttributeSet containing the HTML attributes.

### addAttribute

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **addAttribute**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old,  
 [Object](http://docs.google.com/java/lang/Object.html) key,  
 [Object](http://docs.google.com/java/lang/Object.html) value)

Adds an attribute to the given set, and returns the new representative set. This is reimplemented to convert StyleConstant attributes to CSS prior to forwarding to the superclass behavior. The StyleConstants attribute has no corresponding CSS entry, the StyleConstants attribute is stored (but will likely be unused).

**Specified by:**[addAttribute](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#addAttribute(javax.swing.text.AttributeSet,%20java.lang.Object,%20java.lang.Object)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html)**Overrides:**[addAttribute](http://docs.google.com/javax/swing/text/StyleContext.html#addAttribute(javax.swing.text.AttributeSet,%20java.lang.Object,%20java.lang.Object)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**old - the old attribute setkey - the non-null attribute keyvalue - the attribute value **Returns:**the updated attribute set**See Also:**[MutableAttributeSet.addAttribute(java.lang.Object, java.lang.Object)](http://docs.google.com/javax/swing/text/MutableAttributeSet.html#addAttribute(java.lang.Object,%20java.lang.Object))

### addAttributes

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **addAttributes**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)

Adds a set of attributes to the element. If any of these attributes are StyleConstants attributes, they will be converted to CSS prior to forwarding to the superclass behavior.

**Specified by:**[addAttributes](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#addAttributes(javax.swing.text.AttributeSet,%20javax.swing.text.AttributeSet)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html)**Overrides:**[addAttributes](http://docs.google.com/javax/swing/text/StyleContext.html#addAttributes(javax.swing.text.AttributeSet,%20javax.swing.text.AttributeSet)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**old - the old attribute setattr - the attributes to add **Returns:**the updated attribute set**See Also:**[MutableAttributeSet.addAttribute(java.lang.Object, java.lang.Object)](http://docs.google.com/javax/swing/text/MutableAttributeSet.html#addAttribute(java.lang.Object,%20java.lang.Object))

### removeAttribute

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **removeAttribute**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old,  
 [Object](http://docs.google.com/java/lang/Object.html) key)

Removes an attribute from the set. If the attribute is a StyleConstants attribute, the request will be converted to a CSS attribute prior to forwarding to the superclass behavior.

**Specified by:**[removeAttribute](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#removeAttribute(javax.swing.text.AttributeSet,%20java.lang.Object)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html)**Overrides:**[removeAttribute](http://docs.google.com/javax/swing/text/StyleContext.html#removeAttribute(javax.swing.text.AttributeSet,%20java.lang.Object)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**old - the old set of attributeskey - the non-null attribute name **Returns:**the updated attribute set**See Also:**[MutableAttributeSet.removeAttribute(java.lang.Object)](http://docs.google.com/javax/swing/text/MutableAttributeSet.html#removeAttribute(java.lang.Object))

### removeAttributes

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **removeAttributes**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old,  
 [Enumeration](http://docs.google.com/java/util/Enumeration.html)<?> names)

Removes a set of attributes for the element. If any of the attributes is a StyleConstants attribute, the request will be converted to a CSS attribute prior to forwarding to the superclass behavior.

**Specified by:**[removeAttributes](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#removeAttributes(javax.swing.text.AttributeSet,%20java.util.Enumeration)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html)**Overrides:**[removeAttributes](http://docs.google.com/javax/swing/text/StyleContext.html#removeAttributes(javax.swing.text.AttributeSet,%20java.util.Enumeration)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**old - the old attribute setnames - the attribute names **Returns:**the updated attribute set**See Also:**[MutableAttributeSet.removeAttributes(java.util.Enumeration)](http://docs.google.com/javax/swing/text/MutableAttributeSet.html#removeAttributes(java.util.Enumeration))

### removeAttributes

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **removeAttributes**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attrs)

Removes a set of attributes. If any of the attributes is a StyleConstants attribute, the request will be converted to a CSS attribute prior to forwarding to the superclass behavior.

**Specified by:**[removeAttributes](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#removeAttributes(javax.swing.text.AttributeSet,%20javax.swing.text.AttributeSet)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html)**Overrides:**[removeAttributes](http://docs.google.com/javax/swing/text/StyleContext.html#removeAttributes(javax.swing.text.AttributeSet,%20javax.swing.text.AttributeSet)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**old - the old attribute setattrs - the attributes **Returns:**the updated attribute set**See Also:**[MutableAttributeSet.removeAttributes(java.util.Enumeration)](http://docs.google.com/javax/swing/text/MutableAttributeSet.html#removeAttributes(java.util.Enumeration))

### createSmallAttributeSet

protected [StyleContext.SmallAttributeSet](http://docs.google.com/javax/swing/text/StyleContext.SmallAttributeSet.html) **createSmallAttributeSet**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Creates a compact set of attributes that might be shared. This is a hook for subclasses that want to alter the behavior of SmallAttributeSet. This can be reimplemented to return an AttributeSet that provides some sort of attribute conversion.

**Overrides:**[createSmallAttributeSet](http://docs.google.com/javax/swing/text/StyleContext.html#createSmallAttributeSet(javax.swing.text.AttributeSet)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**a - The set of attributes to be represented in the the compact form.

### createLargeAttributeSet

protected [MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) **createLargeAttributeSet**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Creates a large set of attributes that should trade off space for time. This set will not be shared. This is a hook for subclasses that want to alter the behavior of the larger attribute storage format (which is SimpleAttributeSet by default). This can be reimplemented to return a MutableAttributeSet that provides some sort of attribute conversion.

**Overrides:**[createLargeAttributeSet](http://docs.google.com/javax/swing/text/StyleContext.html#createLargeAttributeSet(javax.swing.text.AttributeSet)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**a - The set of attributes to be represented in the the larger form.

### getFont

public [Font](http://docs.google.com/java/awt/Font.html) **getFont**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Fetches the font to use for the given set of attributes.

**Overrides:**[getFont](http://docs.google.com/javax/swing/text/StyleContext.html#getFont(javax.swing.text.AttributeSet)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**a - the attribute set **Returns:**the font

### getForeground

public [Color](http://docs.google.com/java/awt/Color.html) **getForeground**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Takes a set of attributes and turn it into a foreground color specification. This might be used to specify things like brighter, more hue, etc.

**Overrides:**[getForeground](http://docs.google.com/javax/swing/text/StyleContext.html#getForeground(javax.swing.text.AttributeSet)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**a - the set of attributes **Returns:**the color

### getBackground

public [Color](http://docs.google.com/java/awt/Color.html) **getBackground**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Takes a set of attributes and turn it into a background color specification. This might be used to specify things like brighter, more hue, etc.

**Overrides:**[getBackground](http://docs.google.com/javax/swing/text/StyleContext.html#getBackground(javax.swing.text.AttributeSet)) in class [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **Parameters:**a - the set of attributes **Returns:**the color

### getBoxPainter

public [StyleSheet.BoxPainter](http://docs.google.com/javax/swing/text/html/StyleSheet.BoxPainter.html) **getBoxPainter**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Fetches the box formatter to use for the given set of CSS attributes.

### getListPainter

public [StyleSheet.ListPainter](http://docs.google.com/javax/swing/text/html/StyleSheet.ListPainter.html) **getListPainter**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Fetches the list formatter to use for the given set of CSS attributes.

### setBaseFontSize

public void **setBaseFontSize**(int sz)

Sets the base font size, with valid values between 1 and 7.

### setBaseFontSize

public void **setBaseFontSize**([String](http://docs.google.com/java/lang/String.html) size)

Sets the base font size from the passed in String. The string can either identify a specific font size, with legal values between 1 and 7, or identifiy a relative font size such as +1 or -2.

### getIndexOfSize

public static int **getIndexOfSize**(float pt)

### getPointSize

public float **getPointSize**(int index)

Returns the point size, given a size index.

### getPointSize

public float **getPointSize**([String](http://docs.google.com/java/lang/String.html) size)

Given a string such as "+2", "-2", or "2", returns a point size value.

### stringToColor

public [Color](http://docs.google.com/java/awt/Color.html) **stringToColor**([String](http://docs.google.com/java/lang/String.html) string)

Converts a color string such as "RED" or "#NNNNNN" to a Color. Note: This will only convert the HTML3.2 color strings or a string of length 7; otherwise, it will return null.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/StyleSheet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/text/html/ParagraphView.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/StyleSheet.BoxPainter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/StyleSheet.html)    [**NO FRAMES**](http://docs.google.com/StyleSheet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).