| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Text.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/org/w3c/dom/ProcessingInstruction.html)   [**NEXT CLASS**](http://docs.google.com/org/w3c/dom/TypeInfo.html) | [**FRAMES**](http://docs.google.com/index.html?org/w3c/dom/Text.html)    [**NO FRAMES**](http://docs.google.com/Text.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#4d34og8) |

## **org.w3c.dom**

Interface Text

**All Superinterfaces:** [CharacterData](http://docs.google.com/org/w3c/dom/CharacterData.html), [Node](http://docs.google.com/org/w3c/dom/Node.html) **All Known Subinterfaces:** [CDATASection](http://docs.google.com/org/w3c/dom/CDATASection.html), [Text](http://docs.google.com/javax/xml/soap/Text.html)

public interface **Text**extends [CharacterData](http://docs.google.com/org/w3c/dom/CharacterData.html)

The Text interface inherits from CharacterData and represents the textual content (termed [character data](http://www.w3.org/TR/2004/REC-xml-20040204#syntax) in XML) of an Element or Attr. If there is no markup inside an element's content, the text is contained in a single object implementing the Text interface that is the only child of the element. If there is markup, it is parsed into the information items (elements, comments, etc.) and Text nodes that form the list of children of the element.

When a document is first made available via the DOM, there is only one Text node for each block of text. Users may create adjacent Text nodes that represent the contents of a given element without any intervening markup, but should be aware that there is no way to represent the separations between these nodes in XML or HTML, so they will not (in general) persist between DOM editing sessions. The Node.normalize() method merges any such adjacent Text objects into a single node for each block of text.

No lexical check is done on the content of a Text node and, depending on its position in the document, some characters must be escaped during serialization using character references; e.g. the characters "<&" if the textual content is part of an element or of an attribute, the character sequence "]]>" when part of an element, the quotation mark character " or the apostrophe character ' when part of an attribute.

See also the [Document Object Model (DOM) Level 3 Core Specification](http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407).

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

| **Fields inherited from interface org.w3c.dom.**[**Node**](http://docs.google.com/org/w3c/dom/Node.html) |
| --- |
| [ATTRIBUTE\_NODE](http://docs.google.com/org/w3c/dom/Node.html#ATTRIBUTE_NODE), [CDATA\_SECTION\_NODE](http://docs.google.com/org/w3c/dom/Node.html#CDATA_SECTION_NODE), [COMMENT\_NODE](http://docs.google.com/org/w3c/dom/Node.html#COMMENT_NODE), [DOCUMENT\_FRAGMENT\_NODE](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_FRAGMENT_NODE), [DOCUMENT\_NODE](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_NODE), [DOCUMENT\_POSITION\_CONTAINED\_BY](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_CONTAINED_BY), [DOCUMENT\_POSITION\_CONTAINS](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_CONTAINS), [DOCUMENT\_POSITION\_DISCONNECTED](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_DISCONNECTED), [DOCUMENT\_POSITION\_FOLLOWING](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_FOLLOWING), [DOCUMENT\_POSITION\_IMPLEMENTATION\_SPECIFIC](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC), [DOCUMENT\_POSITION\_PRECEDING](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_PRECEDING), [DOCUMENT\_TYPE\_NODE](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_TYPE_NODE), [ELEMENT\_NODE](http://docs.google.com/org/w3c/dom/Node.html#ELEMENT_NODE), [ENTITY\_NODE](http://docs.google.com/org/w3c/dom/Node.html#ENTITY_NODE), [ENTITY\_REFERENCE\_NODE](http://docs.google.com/org/w3c/dom/Node.html#ENTITY_REFERENCE_NODE), [NOTATION\_NODE](http://docs.google.com/org/w3c/dom/Node.html#NOTATION_NODE), [PROCESSING\_INSTRUCTION\_NODE](http://docs.google.com/org/w3c/dom/Node.html#PROCESSING_INSTRUCTION_NODE), [TEXT\_NODE](http://docs.google.com/org/w3c/dom/Node.html#TEXT_NODE) |

| **Method Summary** | |
| --- | --- |
| [String](http://docs.google.com/java/lang/String.html) | [**getWholeText**](http://docs.google.com/org/w3c/dom/Text.html#getWholeText())()            Returns all text of Text nodes logically-adjacent text nodes to this node, concatenated in document order. |
| boolean | [**isElementContentWhitespace**](http://docs.google.com/org/w3c/dom/Text.html#isElementContentWhitespace())()            Returns whether this text node contains  [element content whitespace](http://www.w3.org/TR/2004/REC-xml-infoset-20040204#infoitem.character), often abusively called "ignorable whitespace". |
| [Text](http://docs.google.com/org/w3c/dom/Text.html) | [**replaceWholeText**](http://docs.google.com/org/w3c/dom/Text.html#replaceWholeText(java.lang.String))([String](http://docs.google.com/java/lang/String.html) content)            Replaces the text of the current node and all logically-adjacent text nodes with the specified text. |
| [Text](http://docs.google.com/org/w3c/dom/Text.html) | [**splitText**](http://docs.google.com/org/w3c/dom/Text.html#splitText(int))(int offset)            Breaks this node into two nodes at the specified offset, keeping both in the tree as siblings. |

| **Methods inherited from interface org.w3c.dom.**[**CharacterData**](http://docs.google.com/org/w3c/dom/CharacterData.html) |
| --- |
| [appendData](http://docs.google.com/org/w3c/dom/CharacterData.html#appendData(java.lang.String)), [deleteData](http://docs.google.com/org/w3c/dom/CharacterData.html#deleteData(int,%20int)), [getData](http://docs.google.com/org/w3c/dom/CharacterData.html#getData()), [getLength](http://docs.google.com/org/w3c/dom/CharacterData.html#getLength()), [insertData](http://docs.google.com/org/w3c/dom/CharacterData.html#insertData(int,%20java.lang.String)), [replaceData](http://docs.google.com/org/w3c/dom/CharacterData.html#replaceData(int,%20int,%20java.lang.String)), [setData](http://docs.google.com/org/w3c/dom/CharacterData.html#setData(java.lang.String)), [substringData](http://docs.google.com/org/w3c/dom/CharacterData.html#substringData(int,%20int)) |

| **Methods inherited from interface org.w3c.dom.**[**Node**](http://docs.google.com/org/w3c/dom/Node.html) |
| --- |
| [appendChild](http://docs.google.com/org/w3c/dom/Node.html#appendChild(org.w3c.dom.Node)), [cloneNode](http://docs.google.com/org/w3c/dom/Node.html#cloneNode(boolean)), [compareDocumentPosition](http://docs.google.com/org/w3c/dom/Node.html#compareDocumentPosition(org.w3c.dom.Node)), [getAttributes](http://docs.google.com/org/w3c/dom/Node.html#getAttributes()), [getBaseURI](http://docs.google.com/org/w3c/dom/Node.html#getBaseURI()), [getChildNodes](http://docs.google.com/org/w3c/dom/Node.html#getChildNodes()), [getFeature](http://docs.google.com/org/w3c/dom/Node.html#getFeature(java.lang.String,%20java.lang.String)), [getFirstChild](http://docs.google.com/org/w3c/dom/Node.html#getFirstChild()), [getLastChild](http://docs.google.com/org/w3c/dom/Node.html#getLastChild()), [getLocalName](http://docs.google.com/org/w3c/dom/Node.html#getLocalName()), [getNamespaceURI](http://docs.google.com/org/w3c/dom/Node.html#getNamespaceURI()), [getNextSibling](http://docs.google.com/org/w3c/dom/Node.html#getNextSibling()), [getNodeName](http://docs.google.com/org/w3c/dom/Node.html#getNodeName()), [getNodeType](http://docs.google.com/org/w3c/dom/Node.html#getNodeType()), [getNodeValue](http://docs.google.com/org/w3c/dom/Node.html#getNodeValue()), [getOwnerDocument](http://docs.google.com/org/w3c/dom/Node.html#getOwnerDocument()), [getParentNode](http://docs.google.com/org/w3c/dom/Node.html#getParentNode()), [getPrefix](http://docs.google.com/org/w3c/dom/Node.html#getPrefix()), [getPreviousSibling](http://docs.google.com/org/w3c/dom/Node.html#getPreviousSibling()), [getTextContent](http://docs.google.com/org/w3c/dom/Node.html#getTextContent()), [getUserData](http://docs.google.com/org/w3c/dom/Node.html#getUserData(java.lang.String)), [hasAttributes](http://docs.google.com/org/w3c/dom/Node.html#hasAttributes()), [hasChildNodes](http://docs.google.com/org/w3c/dom/Node.html#hasChildNodes()), [insertBefore](http://docs.google.com/org/w3c/dom/Node.html#insertBefore(org.w3c.dom.Node,%20org.w3c.dom.Node)), [isDefaultNamespace](http://docs.google.com/org/w3c/dom/Node.html#isDefaultNamespace(java.lang.String)), [isEqualNode](http://docs.google.com/org/w3c/dom/Node.html#isEqualNode(org.w3c.dom.Node)), [isSameNode](http://docs.google.com/org/w3c/dom/Node.html#isSameNode(org.w3c.dom.Node)), [isSupported](http://docs.google.com/org/w3c/dom/Node.html#isSupported(java.lang.String,%20java.lang.String)), [lookupNamespaceURI](http://docs.google.com/org/w3c/dom/Node.html#lookupNamespaceURI(java.lang.String)), [lookupPrefix](http://docs.google.com/org/w3c/dom/Node.html#lookupPrefix(java.lang.String)), [normalize](http://docs.google.com/org/w3c/dom/Node.html#normalize()), [removeChild](http://docs.google.com/org/w3c/dom/Node.html#removeChild(org.w3c.dom.Node)), [replaceChild](http://docs.google.com/org/w3c/dom/Node.html#replaceChild(org.w3c.dom.Node,%20org.w3c.dom.Node)), [setNodeValue](http://docs.google.com/org/w3c/dom/Node.html#setNodeValue(java.lang.String)), [setPrefix](http://docs.google.com/org/w3c/dom/Node.html#setPrefix(java.lang.String)), [setTextContent](http://docs.google.com/org/w3c/dom/Node.html#setTextContent(java.lang.String)), [setUserData](http://docs.google.com/org/w3c/dom/Node.html#setUserData(java.lang.String,%20java.lang.Object,%20org.w3c.dom.UserDataHandler)) |

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

### splitText

[Text](http://docs.google.com/org/w3c/dom/Text.html) **splitText**(int offset)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Breaks this node into two nodes at the specified offset, keeping both in the tree as siblings. After being split, this node will contain all the content up to the offset point. A new node of the same type, which contains all the content at and after the offset point, is returned. If the original node had a parent node, the new node is inserted as the next sibling of the original node. When the offset is equal to the length of this node, the new node has no data.

**Parameters:**offset - The 16-bit unit offset at which to split, starting from 0. **Returns:**The new node, of the same type as this node. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - INDEX\_SIZE\_ERR: Raised if the specified offset is negative or greater than the number of 16-bit units in data.

NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

### isElementContentWhitespace

boolean **isElementContentWhitespace**()

Returns whether this text node contains  [element content whitespace](http://www.w3.org/TR/2004/REC-xml-infoset-20040204#infoitem.character), often abusively called "ignorable whitespace". The text node is determined to contain whitespace in element content during the load of the document or if validation occurs while using Document.normalizeDocument().

**Since:** DOM Level 3

### getWholeText

[String](http://docs.google.com/java/lang/String.html) **getWholeText**()

Returns all text of Text nodes logically-adjacent text nodes to this node, concatenated in document order.

For instance, in the example below wholeText on the Text node that contains "bar" returns "barfoo", while on the Text node that contains "foo" it returns "barfoo".

+-----+  
 | <p> |  
 +-----+  
 /\  
 / \  
 /-----\ +-------+  
 | bar | | &ent; |  
 \-----/ +-------+  
 |  
 |  
 /-----\  
 | foo |  
 \-----/

*Figure: barTextNode.wholeText value is "barfoo"*

**Since:** DOM Level 3

### replaceWholeText

[Text](http://docs.google.com/org/w3c/dom/Text.html) **replaceWholeText**([String](http://docs.google.com/java/lang/String.html) content)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Replaces the text of the current node and all logically-adjacent text nodes with the specified text. All logically-adjacent text nodes are removed including the current node unless it was the recipient of the replacement text.

This method returns the node which received the replacement text. The returned node is:

* null, when the replacement text is the empty string;
* the current node, except when the current node is read-only;
* a new Text node of the same type ( Text or CDATASection) as the current node inserted at the location of the replacement.

For instance, in the above example calling replaceWholeText on the Text node that contains "bar" with "yo" in argument results in the following:

+-----+  
 | <p> |  
 +-----+  
 |  
 |  
 /-----\  
 | yo |  
 \-----/

*Figure: barTextNode.replaceWholeText("yo") modifies the textual content of barTextNode with "yo"*

Where the nodes to be removed are read-only descendants of an EntityReference, the EntityReference must be removed instead of the read-only nodes. If any EntityReference to be removed has descendants that are not EntityReference, Text, or CDATASection nodes, the replaceWholeText method must fail before performing any modification of the document, raising a DOMException with the code NO\_MODIFICATION\_ALLOWED\_ERR.

For instance, in the example below calling replaceWholeText on the Text node that contains "bar" fails, because the EntityReference node "ent" contains an Element node which cannot be removed.

**Parameters:**content - The content of the replacing Text node. **Returns:**The Text node created with the specified content. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised if one of the Text nodes being replaced is readonly.**Since:** DOM Level 3

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Text.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/org/w3c/dom/ProcessingInstruction.html)   [**NEXT CLASS**](http://docs.google.com/org/w3c/dom/TypeInfo.html) | [**FRAMES**](http://docs.google.com/index.html?org/w3c/dom/Text.html)    [**NO FRAMES**](http://docs.google.com/Text.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#4d34og8) |

[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).