| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Element.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/DOMStringList.html)   [**NEXT CLASS**](http://docs.google.com/org/w3c/dom/Entity.html) | [**FRAMES**](http://docs.google.com/index.html?org/w3c/dom/Element.html)    [**NO FRAMES**](http://docs.google.com/Element.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#1t3h5sf) |

## **org.w3c.dom**

Interface Element

**All Superinterfaces:** [Node](http://docs.google.com/org/w3c/dom/Node.html) **All Known Subinterfaces:** [Detail](http://docs.google.com/javax/xml/soap/Detail.html), [DetailEntry](http://docs.google.com/javax/xml/soap/DetailEntry.html), [SOAPBody](http://docs.google.com/javax/xml/soap/SOAPBody.html), [SOAPBodyElement](http://docs.google.com/javax/xml/soap/SOAPBodyElement.html), [SOAPElement](http://docs.google.com/javax/xml/soap/SOAPElement.html), [SOAPEnvelope](http://docs.google.com/javax/xml/soap/SOAPEnvelope.html), [SOAPFault](http://docs.google.com/javax/xml/soap/SOAPFault.html), [SOAPFaultElement](http://docs.google.com/javax/xml/soap/SOAPFaultElement.html), [SOAPHeader](http://docs.google.com/javax/xml/soap/SOAPHeader.html), [SOAPHeaderElement](http://docs.google.com/javax/xml/soap/SOAPHeaderElement.html) **All Known Implementing Classes:** [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html)

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

The Element interface represents an element in an HTML or XML document. Elements may have attributes associated with them; since the Element interface inherits from Node, the generic Node interface attribute attributes may be used to retrieve the set of all attributes for an element. There are methods on the Element interface to retrieve either an Attr object by name or an attribute value by name. In XML, where an attribute value may contain entity references, an Attr object should be retrieved to examine the possibly fairly complex sub-tree representing the attribute value. On the other hand, in HTML, where all attributes have simple string values, methods to directly access an attribute value can safely be used as a convenience.

**Note:** In DOM Level 2, the method normalize is inherited from the Node interface where it was moved.

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) | [**getAttribute**](http://docs.google.com/org/w3c/dom/Element.html#getAttribute(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Retrieves an attribute value by name. |
| [Attr](http://docs.google.com/org/w3c/dom/Attr.html) | [**getAttributeNode**](http://docs.google.com/org/w3c/dom/Element.html#getAttributeNode(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Retrieves an attribute node by name. |
| [Attr](http://docs.google.com/org/w3c/dom/Attr.html) | [**getAttributeNodeNS**](http://docs.google.com/org/w3c/dom/Element.html#getAttributeNodeNS(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName)            Retrieves an Attr node by local name and namespace URI. |
| [String](http://docs.google.com/java/lang/String.html) | [**getAttributeNS**](http://docs.google.com/org/w3c/dom/Element.html#getAttributeNS(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName)            Retrieves an attribute value by local name and namespace URI. |
| [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) | [**getElementsByTagName**](http://docs.google.com/org/w3c/dom/Element.html#getElementsByTagName(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns a NodeList of all descendant Elements with a given tag name, in document order. |
| [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) | [**getElementsByTagNameNS**](http://docs.google.com/org/w3c/dom/Element.html#getElementsByTagNameNS(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName)            Returns a NodeList of all the descendant Elements with a given local name and namespace URI in document order. |
| [TypeInfo](http://docs.google.com/org/w3c/dom/TypeInfo.html) | [**getSchemaTypeInfo**](http://docs.google.com/org/w3c/dom/Element.html#getSchemaTypeInfo())()            The type information associated with this element. |
| [String](http://docs.google.com/java/lang/String.html) | [**getTagName**](http://docs.google.com/org/w3c/dom/Element.html#getTagName())()            The name of the element. |
| boolean | [**hasAttribute**](http://docs.google.com/org/w3c/dom/Element.html#hasAttribute(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns true when an attribute with a given name is specified on this element or has a default value, false otherwise. |
| boolean | [**hasAttributeNS**](http://docs.google.com/org/w3c/dom/Element.html#hasAttributeNS(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName)            Returns true when an attribute with a given local name and namespace URI is specified on this element or has a default value, false otherwise. |
| void | [**removeAttribute**](http://docs.google.com/org/w3c/dom/Element.html#removeAttribute(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Removes an attribute by name. |
| [Attr](http://docs.google.com/org/w3c/dom/Attr.html) | [**removeAttributeNode**](http://docs.google.com/org/w3c/dom/Element.html#removeAttributeNode(org.w3c.dom.Attr))([Attr](http://docs.google.com/org/w3c/dom/Attr.html) oldAttr)            Removes the specified attribute node. |
| void | [**removeAttributeNS**](http://docs.google.com/org/w3c/dom/Element.html#removeAttributeNS(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName)            Removes an attribute by local name and namespace URI. |
| void | [**setAttribute**](http://docs.google.com/org/w3c/dom/Element.html#setAttribute(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) name, [String](http://docs.google.com/java/lang/String.html) value)            Adds a new attribute. |
| [Attr](http://docs.google.com/org/w3c/dom/Attr.html) | [**setAttributeNode**](http://docs.google.com/org/w3c/dom/Element.html#setAttributeNode(org.w3c.dom.Attr))([Attr](http://docs.google.com/org/w3c/dom/Attr.html) newAttr)            Adds a new attribute node. |
| [Attr](http://docs.google.com/org/w3c/dom/Attr.html) | [**setAttributeNodeNS**](http://docs.google.com/org/w3c/dom/Element.html#setAttributeNodeNS(org.w3c.dom.Attr))([Attr](http://docs.google.com/org/w3c/dom/Attr.html) newAttr)            Adds a new attribute. |
| void | [**setAttributeNS**](http://docs.google.com/org/w3c/dom/Element.html#setAttributeNS(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) qualifiedName, [String](http://docs.google.com/java/lang/String.html) value)            Adds a new attribute. |
| void | [**setIdAttribute**](http://docs.google.com/org/w3c/dom/Element.html#setIdAttribute(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) name, boolean isId)            If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . |
| void | [**setIdAttributeNode**](http://docs.google.com/org/w3c/dom/Element.html#setIdAttributeNode(org.w3c.dom.Attr,%20boolean))([Attr](http://docs.google.com/org/w3c/dom/Attr.html) idAttr, boolean isId)            If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . |
| void | [**setIdAttributeNS**](http://docs.google.com/org/w3c/dom/Element.html#setIdAttributeNS(java.lang.String,%20java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName, boolean isId)            If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . |

| **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** |
| --- |

### getTagName

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

The name of the element. If Node.localName is different from null, this attribute is a qualified name. For example, in:

<elementExample id="demo"> ...   
 </elementExample> ,

tagName has the value "elementExample". Note that this is case-preserving in XML, as are all of the operations of the DOM. The HTML DOM returns the tagName of an HTML element in the canonical uppercase form, regardless of the case in the source HTML document.

### getAttribute

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

Retrieves an attribute value by name.

**Parameters:**name - The name of the attribute to retrieve. **Returns:**The Attr value as a string, or the empty string if that attribute does not have a specified or default value.

### setAttribute

void **setAttribute**([String](http://docs.google.com/java/lang/String.html) name,  
 [String](http://docs.google.com/java/lang/String.html) value)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Adds a new attribute. If an attribute with that name is already present in the element, its value is changed to be that of the value parameter. This value is a simple string; it is not parsed as it is being set. So any markup (such as syntax to be recognized as an entity reference) is treated as literal text, and needs to be appropriately escaped by the implementation when it is written out. In order to assign an attribute value that contains entity references, the user must create an Attr node plus any Text and EntityReference nodes, build the appropriate subtree, and use setAttributeNode to assign it as the value of an attribute.

To set an attribute with a qualified name and namespace URI, use the setAttributeNS method.

**Parameters:**name - The name of the attribute to create or alter.value - Value to set in string form. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - INVALID\_CHARACTER\_ERR: Raised if the specified name is not an XML name according to the XML version in use specified in the Document.xmlVersion attribute.

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

### removeAttribute

void **removeAttribute**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Removes an attribute by name. If a default value for the removed attribute is defined in the DTD, a new attribute immediately appears with the default value as well as the corresponding namespace URI, local name, and prefix when applicable. The implementation may handle default values from other schemas similarly but applications should use Document.normalizeDocument() to guarantee this information is up-to-date.

If no attribute with this name is found, this method has no effect.

To remove an attribute by local name and namespace URI, use the removeAttributeNS method.

**Parameters:**name - The name of the attribute to remove. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

### getAttributeNode

[Attr](http://docs.google.com/org/w3c/dom/Attr.html) **getAttributeNode**([String](http://docs.google.com/java/lang/String.html) name)

Retrieves an attribute node by name.

To retrieve an attribute node by qualified name and namespace URI, use the getAttributeNodeNS method.

**Parameters:**name - The name (nodeName) of the attribute to retrieve. **Returns:**The Attr node with the specified name ( nodeName) or null if there is no such attribute.

### setAttributeNode

[Attr](http://docs.google.com/org/w3c/dom/Attr.html) **setAttributeNode**([Attr](http://docs.google.com/org/w3c/dom/Attr.html) newAttr)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Adds a new attribute node. If an attribute with that name ( nodeName) is already present in the element, it is replaced by the new one. Replacing an attribute node by itself has no effect.

To add a new attribute node with a qualified name and namespace URI, use the setAttributeNodeNS method.

**Parameters:**newAttr - The Attr node to add to the attribute list. **Returns:**If the newAttr attribute replaces an existing attribute, the replaced Attr node is returned, otherwise null is returned. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - WRONG\_DOCUMENT\_ERR: Raised if newAttr was created from a different document than the one that created the element.

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

INUSE\_ATTRIBUTE\_ERR: Raised if newAttr is already an attribute of another Element object. The DOM user must explicitly clone Attr nodes to re-use them in other elements.

### removeAttributeNode

[Attr](http://docs.google.com/org/w3c/dom/Attr.html) **removeAttributeNode**([Attr](http://docs.google.com/org/w3c/dom/Attr.html) oldAttr)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Removes the specified attribute node. If a default value for the removed Attr node is defined in the DTD, a new node immediately appears with the default value as well as the corresponding namespace URI, local name, and prefix when applicable. The implementation may handle default values from other schemas similarly but applications should use Document.normalizeDocument() to guarantee this information is up-to-date.

**Parameters:**oldAttr - The Attr node to remove from the attribute list. **Returns:**The Attr node that was removed. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

NOT\_FOUND\_ERR: Raised if oldAttr is not an attribute of the element.

### getElementsByTagName

[NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) **getElementsByTagName**([String](http://docs.google.com/java/lang/String.html) name)

Returns a NodeList of all descendant Elements with a given tag name, in document order.

**Parameters:**name - The name of the tag to match on. The special value "\*" matches all tags. **Returns:**A list of matching Element nodes.

### getAttributeNS

[String](http://docs.google.com/java/lang/String.html) **getAttributeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Retrieves an attribute value by local name and namespace URI.

Per [[XML Namespaces](http://www.w3.org/TR/1999/REC-xml-names-19990114/)] , applications must use the value null as the namespaceURI parameter for methods if they wish to have no namespace.

**Parameters:**namespaceURI - The namespace URI of the attribute to retrieve.localName - The local name of the attribute to retrieve. **Returns:**The Attr value as a string, or the empty string if that attribute does not have a specified or default value. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NOT\_SUPPORTED\_ERR: May be raised if the implementation does not support the feature "XML" and the language exposed through the Document does not support XML Namespaces (such as [[HTML 4.01](http://www.w3.org/TR/1999/REC-html401-19991224/)]).**Since:** DOM Level 2

### setAttributeNS

void **setAttributeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) qualifiedName,  
 [String](http://docs.google.com/java/lang/String.html) value)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Adds a new attribute. If an attribute with the same local name and namespace URI is already present on the element, its prefix is changed to be the prefix part of the qualifiedName, and its value is changed to be the value parameter. This value is a simple string; it is not parsed as it is being set. So any markup (such as syntax to be recognized as an entity reference) is treated as literal text, and needs to be appropriately escaped by the implementation when it is written out. In order to assign an attribute value that contains entity references, the user must create an Attr node plus any Text and EntityReference nodes, build the appropriate subtree, and use setAttributeNodeNS or setAttributeNode to assign it as the value of an attribute.

Per [[XML Namespaces](http://www.w3.org/TR/1999/REC-xml-names-19990114/)] , applications must use the value null as the namespaceURI parameter for methods if they wish to have no namespace.

**Parameters:**namespaceURI - The namespace URI of the attribute to create or alter.qualifiedName - The qualified name of the attribute to create or alter.value - The value to set in string form. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - INVALID\_CHARACTER\_ERR: Raised if the specified qualified name is not an XML name according to the XML version in use specified in the Document.xmlVersion attribute.

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

NAMESPACE\_ERR: Raised if the qualifiedName is malformed per the Namespaces in XML specification, if the qualifiedName has a prefix and the namespaceURI is null, if the qualifiedName has a prefix that is "xml" and the namespaceURI is different from "<http://www.w3.org/XML/1998/namespace>", if the qualifiedName or its prefix is "xmlns" and the namespaceURI is different from "<http://www.w3.org/2000/xmlns/>", or if the namespaceURI is "<http://www.w3.org/2000/xmlns/>" and neither the qualifiedName nor its prefix is "xmlns".

NOT\_SUPPORTED\_ERR: May be raised if the implementation does not support the feature "XML" and the language exposed through the Document does not support XML Namespaces (such as [[HTML 4.01](http://www.w3.org/TR/1999/REC-html401-19991224/)]).**Since:** DOM Level 2

### removeAttributeNS

void **removeAttributeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Removes an attribute by local name and namespace URI. If a default value for the removed attribute is defined in the DTD, a new attribute immediately appears with the default value as well as the corresponding namespace URI, local name, and prefix when applicable. The implementation may handle default values from other schemas similarly but applications should use Document.normalizeDocument() to guarantee this information is up-to-date.

If no attribute with this local name and namespace URI is found, this method has no effect.

Per [[XML Namespaces](http://www.w3.org/TR/1999/REC-xml-names-19990114/)] , applications must use the value null as the namespaceURI parameter for methods if they wish to have no namespace.

**Parameters:**namespaceURI - The namespace URI of the attribute to remove.localName - The local name of the attribute to remove. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

NOT\_SUPPORTED\_ERR: May be raised if the implementation does not support the feature "XML" and the language exposed through the Document does not support XML Namespaces (such as [[HTML 4.01](http://www.w3.org/TR/1999/REC-html401-19991224/)]).**Since:** DOM Level 2

### getAttributeNodeNS

[Attr](http://docs.google.com/org/w3c/dom/Attr.html) **getAttributeNodeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Retrieves an Attr node by local name and namespace URI.

Per [[XML Namespaces](http://www.w3.org/TR/1999/REC-xml-names-19990114/)] , applications must use the value null as the namespaceURI parameter for methods if they wish to have no namespace.

**Parameters:**namespaceURI - The namespace URI of the attribute to retrieve.localName - The local name of the attribute to retrieve. **Returns:**The Attr node with the specified attribute local name and namespace URI or null if there is no such attribute. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NOT\_SUPPORTED\_ERR: May be raised if the implementation does not support the feature "XML" and the language exposed through the Document does not support XML Namespaces (such as [[HTML 4.01](http://www.w3.org/TR/1999/REC-html401-19991224/)]).**Since:** DOM Level 2

### setAttributeNodeNS

[Attr](http://docs.google.com/org/w3c/dom/Attr.html) **setAttributeNodeNS**([Attr](http://docs.google.com/org/w3c/dom/Attr.html) newAttr)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Adds a new attribute. If an attribute with that local name and that namespace URI is already present in the element, it is replaced by the new one. Replacing an attribute node by itself has no effect.

Per [[XML Namespaces](http://www.w3.org/TR/1999/REC-xml-names-19990114/)] , applications must use the value null as the namespaceURI parameter for methods if they wish to have no namespace.

**Parameters:**newAttr - The Attr node to add to the attribute list. **Returns:**If the newAttr attribute replaces an existing attribute with the same local name and namespace URI, the replaced Attr node is returned, otherwise null is returned. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - WRONG\_DOCUMENT\_ERR: Raised if newAttr was created from a different document than the one that created the element.

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

INUSE\_ATTRIBUTE\_ERR: Raised if newAttr is already an attribute of another Element object. The DOM user must explicitly clone Attr nodes to re-use them in other elements.

NOT\_SUPPORTED\_ERR: May be raised if the implementation does not support the feature "XML" and the language exposed through the Document does not support XML Namespaces (such as [[HTML 4.01](http://www.w3.org/TR/1999/REC-html401-19991224/)]).**Since:** DOM Level 2

### getElementsByTagNameNS

[NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) **getElementsByTagNameNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Returns a NodeList of all the descendant Elements with a given local name and namespace URI in document order.

**Parameters:**namespaceURI - The namespace URI of the elements to match on. The special value "\*" matches all namespaces.localName - The local name of the elements to match on. The special value "\*" matches all local names. **Returns:**A new NodeList object containing all the matched Elements. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NOT\_SUPPORTED\_ERR: May be raised if the implementation does not support the feature "XML" and the language exposed through the Document does not support XML Namespaces (such as [[HTML 4.01](http://www.w3.org/TR/1999/REC-html401-19991224/)]).**Since:** DOM Level 2

### hasAttribute

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

Returns true when an attribute with a given name is specified on this element or has a default value, false otherwise.

**Parameters:**name - The name of the attribute to look for. **Returns:**true if an attribute with the given name is specified on this element or has a default value, false otherwise.**Since:** DOM Level 2

### hasAttributeNS

boolean **hasAttributeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Returns true when an attribute with a given local name and namespace URI is specified on this element or has a default value, false otherwise.

Per [[XML Namespaces](http://www.w3.org/TR/1999/REC-xml-names-19990114/)] , applications must use the value null as the namespaceURI parameter for methods if they wish to have no namespace.

**Parameters:**namespaceURI - The namespace URI of the attribute to look for.localName - The local name of the attribute to look for. **Returns:**true if an attribute with the given local name and namespace URI is specified or has a default value on this element, false otherwise. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NOT\_SUPPORTED\_ERR: May be raised if the implementation does not support the feature "XML" and the language exposed through the Document does not support XML Namespaces (such as [[HTML 4.01](http://www.w3.org/TR/1999/REC-html401-19991224/)]).**Since:** DOM Level 2

### getSchemaTypeInfo

[TypeInfo](http://docs.google.com/org/w3c/dom/TypeInfo.html) **getSchemaTypeInfo**()

The type information associated with this element.

**Since:** DOM Level 3

### setIdAttribute

void **setIdAttribute**([String](http://docs.google.com/java/lang/String.html) name,  
 boolean isId)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . This affects the value of Attr.isId and the behavior of Document.getElementById, but does not change any schema that may be in use, in particular this does not affect the Attr.schemaTypeInfo of the specified Attr node. Use the value false for the parameter isId to undeclare an attribute for being a user-determined ID attribute.

To specify an attribute by local name and namespace URI, use the setIdAttributeNS method.

**Parameters:**name - The name of the attribute.isId - Whether the attribute is a of type ID. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

NOT\_FOUND\_ERR: Raised if the specified node is not an attribute of this element.**Since:** DOM Level 3

### setIdAttributeNS

void **setIdAttributeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName,  
 boolean isId)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . This affects the value of Attr.isId and the behavior of Document.getElementById, but does not change any schema that may be in use, in particular this does not affect the Attr.schemaTypeInfo of the specified Attr node. Use the value false for the parameter isId to undeclare an attribute for being a user-determined ID attribute.

**Parameters:**namespaceURI - The namespace URI of the attribute.localName - The local name of the attribute.isId - Whether the attribute is a of type ID. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

NOT\_FOUND\_ERR: Raised if the specified node is not an attribute of this element.**Since:** DOM Level 3

### setIdAttributeNode

void **setIdAttributeNode**([Attr](http://docs.google.com/org/w3c/dom/Attr.html) idAttr,  
 boolean isId)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . This affects the value of Attr.isId and the behavior of Document.getElementById, but does not change any schema that may be in use, in particular this does not affect the Attr.schemaTypeInfo of the specified Attr node. Use the value false for the parameter isId to undeclare an attribute for being a user-determined ID attribute.

**Parameters:**idAttr - The attribute node.isId - Whether the attribute is a of type ID. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

NOT\_FOUND\_ERR: Raised if the specified node is not an attribute of this element.**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/Element.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/DOMStringList.html)   [**NEXT CLASS**](http://docs.google.com/org/w3c/dom/Entity.html) | [**FRAMES**](http://docs.google.com/index.html?org/w3c/dom/Element.html)    [**NO FRAMES**](http://docs.google.com/Element.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#1t3h5sf) |

[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).