

XML DOM - ADD NODE

http://www.tutorialspoint.com/dom/xml_dom_add_node.htm

Copyright © tutorialspoint.com

This chapter discusses about appending or adding nodes in an the XML DOM. Add node operation appends the nodes to the existing element. It provides a means to:

- append new child nodes before or after the existing child nodes
- the scope to insert data within the text node
- and to add attribute node

Following methods can be used to add/append the nodes to an element in a DOM:

- appendChild
- insertBefore
- insertData

appendChild

The method appendChild adds the new child node after the existing child node.

Syntax

Syntax of appendChild method is as follows:

```
Node appendChild(Node newChild) throws DOMException
```

Where:

- *newChild* - Is the node to add
- This method returns the *Node* added.

Example

The following example *appendchildnode_example.htm* parses an XML document ("[node.xml](#)") into an XML DOM object and appends new child *PhoneNo* to the element <FirstName>.

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function loadXMLDoc(filename)
      {
        if (window.XMLHttpRequest)
        {
          xhttp = new XMLHttpRequest();
        }
        else // code for IE5 and IE6
        {
          xhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        xhttp.open("GET", filename, false);
        xhttp.send();
        return xhttp.responseXML;
      }
    </script>
  </head>
  <body>
    <script>
      xmlDoc = loadXMLDoc("/dom/node.xml");
```

```

        create_e = xmlDoc.createElement("PhoneNo");

        x = xmlDoc.getElementsByTagName("FirstName")[0];
        x.appendChild(create_e);

        document.write(x.getElementsByTagName("PhoneNo")[0].nodeName);
    </script>
</body>
</html>

```

In the above example:

- using the method `createElement`, a new element *PhoneNo* is created.
- The new element *PhoneNo* is added to the element *FirstName* using the method `appendChild`.

Execution

Save this file as *appendchildnode_example.htm* on the server path *thisfileandnode.xml* should be on the same path in your server. In the output we get the attribute value as i.e. *PhoneNo*.

insertBefore

The method *insertBefore*, inserts the new child nodes before the specified child nodes.

Syntax

Syntax of *insertBefore* method is as follows:

```
Node insertBefore(Node newChild, Node refChild) throws DOMException
```

Where:

- *newChild* - Is the node to insert
- *refChild* - Is the reference node, i.e., the node before which the new node must be inserted.
- This method returns the *Node* being inserted.

Example

The following example *insertnodebefore_example.htm* parses an XML document ("[node.xml](#)") into an XML DOM object and inserts new child *Email* before the specified element `<Email>`.

```

<!DOCTYPE html>
<html>
    <head>
        <script>
            function loadXMLDoc(filename)
            {
                if (window.XMLHttpRequest)
                {
                    xhttp = new XMLHttpRequest();
                }
                else // code for IE5 and IE6
                {
                    xhttp = new ActiveXObject("Microsoft.XMLHTTP");
                }
                xhttp.open("GET", filename, false);
                xhttp.send();
                return xhttp.responseXML;
            }
        </script>

```

```

</head>
<body>
  <script>
    xmlDoc = loadXMLDoc("/dom/node.xml");

    create_e = xmlDoc.createElement("Email");

    x = xmlDoc.documentElement;
    y = xmlDoc.getElementsByTagName("Email");

    document.write("No of Email elements before inserting was: " + y.length);
    document.write("<br>");
    x.insertBefore(create_e, y[3]);

    y=xmlDoc.getElementsByTagName("Email");
    document.write("No of Email elements after inserting is: " + y.length);
  </script>
</body>
</html>

```

In the above example:

- using the method `createElement` a new element *Email* is created.
- The new element *Email* is added before to the element *Email* using the method `insertBefore`.
- *y.length* gives the total number of elements added before and after the new element.

Execution

Save this file as *insertnodebefore_example.htm* on the server path *thisfileandnode.xml* should be on the same path in your server. In the output we get the following:

```

No of Email elements before inserting was: 3
No of Email elements after inserting is: 4

```

insertData

The method `insertData`, inserts a string at the specified 16-bit unit offset.

Syntax

The `insertData` has following syntax:

```
void insertData(int offset, java.lang.String arg) throws DOMException
```

Where:

- *offset*: is the character offset at which to insert..
- *arg*: is the key word to insert the data. It encloses the two parameters offset and string within the parenthesis separated by comma.

Example

The following example *addtext_example.htm* parses an XML document ("[node.xml](#)") into an XML DOM object and inserts new data *MiddleName* at the specified position to the element `<FirstName>`.

```

<!DOCTYPE html>
<html>
  <head>
    <script>
      function loadXMLDoc(filename)
      {
        if (window.XMLHttpRequest)

```

```

        {
            xhttp = new XMLHttpRequest();
        }
        else // code for IE5 and IE6
        {
            xhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        xhttp.open("GET", filename, false);
        xhttp.send();
        return xhttp.responseXML;
    }
</script>
</head>
<body>
    <script>
        xmlDoc = loadXMLDoc("/dom/node.xml");

        x = xmlDoc.getElementsByTagName("FirstName")[0].childNodes[0];
        document.write(x.nodeValue);
        x.insertData(6, "MiddleName");
        document.write("<br>");
        document.write(x.nodeValue);

    </script>
</body>
</html>

```

- `x.insertData6, "MiddleName"` ; : Here x holds the name of the specified child name i.e `<FirstName>`. We then insert to this text node the data `"MiddleName"` starting from position 6.

Execution

Save this file as `addtext_example.htm` on the server path `thisfileandnode.xml` should be on the same path in your server. In the output we get the following:

```

Tanmay
TanmayMiddleName
Processing math: 100%

```