doesn’t even throw any errors or give you a clue as to why the rows are not displayed even though they are appended to the table.

In this case, the workaround is simple. Internet Explorer will allow \texttt{tr} elements to be added to a \texttt{tbody} element, just not directly to the \texttt{table} element. For example, if you defined an empty table like this:

\begin{verbatim}
<table id="myTable">
  <tbody id="myTableBody"></tbody>
</table>
\end{verbatim}

the correct way to add a row to the table would be to add the row to the table body instead of to the table, like so:

\begin{verbatim}
var cell = document.createElement("td").appendChild(document.createTextNode("foo"));
var row = document.createElement("tr").appendChild(cell);
document.getElementById("myTableBody").appendChild(row);
\end{verbatim}

Fortunately, this method works in all modern browsers, including Internet Explorer. Stay in the habit of always using table bodies within your tables, and you’ll never have to worry about this issue.

### Setting an Element’s Style via JavaScript

Ajax techniques allow developers to create Web applications that can seamlessly communicate with the server without a complete page refresh. Users of such applications will expect to see the page flicker whenever data is sent to or from the server. This flicker does not occur during Ajax requests, so the user may not know when data has been updated on the page. You’ll likely change the style of some elements to indicate that some data on the page has changed; for example, you may highlight the name of a stock whose price was seamlessly updated via an Ajax request.

You should set the style of an element through JavaScript using the \texttt{setAttribute} method of the element. For example, to change the text within a \texttt{span} element to be displayed in bold and in red, use the \texttt{setAttribute} method as follows:

\begin{verbatim}
var spanElement = document.getElementById("mySpan");
spanElement.setAttribute("style", "font-weight:bold; color:red;";)
\end{verbatim}

This works well in all modern browsers except Internet Explorer. The workaround for Internet Explorer is to use the nonstandard but widely supported \texttt{cssText} property of the element’s \texttt{style} object to set the desired style, as follows:

\begin{verbatim}
var spanElement = document.getElementById("mySpan");
spanElement.style.cssText = "font-weight:bold; color:red;";
\end{verbatim}

This method works well in Internet Explorer and most other browsers except Opera. To make the code portable across all modern browsers, use the \texttt{setAttribute} method in addition to the \texttt{cssText} property of the element’s \texttt{style} object, like so:

\begin{verbatim}
var spanElement = document.getElementById("mySpan");
spanElement.setAttribute("style", "font-weight:bold; color:red;";)
spanElement.style.cssText = "font-weight:bold; color:red;";
\end{verbatim}