A METHOD OF DISPLAYING DATA IN A TABLE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a nonprovisional application based on my provisional application No. 60/631,152 filed on November 26, 2004, entitled "Method For Creating a Data Table with a Stationary Header in a Web Page," the priority of which is hereby claimed and the full disclosure of which is incorporated by reference herein.

BACKGROUND OF THE INVENTION

[0002] Ever since the beginning of the Internet and web pages, there has been a need to have tabular data displayed in an easily readable form. When a web page designer needs to display multiple rows of data, e.g. 20, 50, 100, or more rows, it is highly desirable for the headers or labels of each column of data row to remain "fixed" or "stationary" while a user, or person using a browser, can scroll down the rows of data while still being able to see and read the column headers or column labels at the top of the table. The browser can be any of the well known programs, such as for instance Microsoft Internet Explorer, Mozilla, Apple Safari, Opera, Firefox and the like.

[0003] Some programs allow the header rows remain stationary on top of a page, while the user scrolls vertically along the columns of in a table. One such example is Microsoft Excel, a Windows-based program extensively used in accounting. However, Microsoft Excel is not a traditional Web Page that uses HTML (Hyper Text Markup Language), or CSS (Cascading Style Sheets), that is an application suitable for displaying data on an Internet web page. Although Microsoft Excel spreadsheet can still be viewed in a browser, yet this is still not a typical web page that uses HTML or CSS; it has many limitations in its use and where it can be used.

[0004] Several attempts have been made to create a web page that has a stationary header. Some have used two HTML tables with the top table containing one row and setting the "width" of each column and/or its table attributes to a certain pixel or percentage amount so that each column width matches the column width amount of the bottom table that contains the data. The disadvantage of this method is that one has difficulty in determining the correct width amount to set each column. By setting the width to a static value, whether in pixels or percentages, the top table and its columns cannot "flow" or contract or expand width-wise as easily as if there were no width settings in the first

place. Furthermore, if the data comes from a database, the data can constantly change as well, thus the top table's columns width amounts can either be too large or too small for the data and headers to be displayed.

[0005] Other methods to display a stationary header row for a table are those that use HTML FRAMES and IFRAMEs. There are numerous web sites that list the disadvantages of FRAMES in the use of web pages. Some disadvantages are the difficulty in book marking and printing. Others are the requirements for a "clean" layout and the additional scrollbars. Still another disadvantage is the difficulty of the design and maintenance of the FRAMED or IFRAME web pages.

[0006] Complicating matters further, it is highly desirable for any of the methods of displaying a stationary header to work across multiple browsers. This cross-browser compatibility is also very difficult to achieve even without the use of stationary headers. And complicating matter still more, each browser has a text size setting that an individual can set and can override the text size settings in the browser depending up which browser is being used. Thus, even if a web programmer or web developer or web designer sets the font setting for the data, the individual users, depending up which browser they are using, can over-ride this setting so that they can make it easier for them to see the web page at the font size they wish to see it.

[0007] There are numerous factors to consider when designing a fixed header table in a web page. Some of these factors are listed below:

[0008] 1. Fixed Header tables are used to display data that is typically stored in a database of some sort. This data can constantly change. Thus, setting the <td> width attribute is very difficult.

[0009] 2. The Text as well as the Font Size and Weight (e.g. bold) of the Header information row can also change. This also affects the alignment of the columns.

[0010] 3. Images can also be placed inside the data cell. This also must be accounted for, or alignment will suffer.

[0011] 4. Images can be placed in the header (e.g. direction of sorting). Again, if this is not accounted for, the columns will be misaligned.

[0012] 5. Browser versions and browser types can display data differently. Slight differences can cause columns to be misaligned.

[0013] 6. The user's browser window size settings affect the displaying of data. A user who sets his window to be smaller or larger can affect the display of data. If the browser is maximized, this also can affect the alignment.

[0014] 7. Other browser settings, such as the Text Size Setting also affect the alignment of columns. The text size setting is set by the individual and can easily affect the alignment of the columns. Each browser's implementation of the HTML font unit type declaration is also different.

That is, certain font unit sizes can be adjusted by some browsers while other browsers cannot adjust the text. For example, IE Text Size adjustment cannot adjust font size that use the pixel (px) unit size However, the Mozilla and Opera browsers allow their users to adjust the pixel font unit type. The instant method allows a number of advantages, among those:

[0015] 1. Avoid the use of PLUG-INs or asking the user to download programs as many users are wary of downloading unknown programs (e.g. viruses, spyware) onto their computer.

[0016] 2. Minimal data and/or code download per web page for better system response and/or performance on the web server and the user's computer.

[0017] 3. Minimal web server CPU load to produce the fixed header code so as to serve more users and/or to serve users faster.

[0018] There are programs such as a plug-in FLASH that have been able to implement a fixed header table. But this is not standard HTML and for that matter one needs another authoring tool to work with FLASH to begin with. FLASH is reported to be slow in this respect. Moreover, downloading anything onto a computer from the Internet is just risky due to viruses and spyware. Programs like ActiveX, Java Applets are typically used by hackers to infect user's computer, and since this is so, user will set their security setting to deny such programs and will buy commercial protection software to help prevent programs like the above from ever being installed on their computer.

[0019] It is known that Microsoft Excel can be viewed in the browser. However, this is not desirable as only an Excel page can be viewed. Further, hackers have been known to exploit Excel files so that it can carry viruses and spyware. Additionally, there is also Java language that can do this as well, but again, this is not simple HTML, CSS, and JavaScript which are accepted as safe languages and are already on a user's computer to begin with.

[0020] Some software on web pages can have a person "double click" between the header columns to automatically resize the columns to fit. This additional step may not be beneficial under some circumstances as it complicates a relatively simple task. Web pages that allow this step rely heavily on JavaScript and practically no HTML. Such pages appear to be "heavy," taking time to load in a user's browser. Although not confirmed, this step may not work on the Macintosh, since the JavaScript used in such programs is rather sophisticated and complicated. As a result, the cross-browser is hard to achieve as JavaScript can have different effects on different browsers.

[0021] These and other factors should be considered when making a fixed header table (or datagrid). Any of these factors, no matter how miniscule can cause misalignment. And thus, a fixed header that can be used on a production quality level has so far been unobtainable since the inception

of web pages. A production quality fixed header table should be usable by a majority of the popular browsers (e.g. 99%) and can adjust to the many user settings as well as the data that is displayed.

[0022] The present invention contemplates elimination of drawbacks associated with the prior methods and provision of a method of displaying data in a table with the first horizontal row remaining stationary while allowing the user to vertically scroll the columns. Anyone on the Internet with almost any browser should be able to access a fixed header table designed according to the method of the present invention.

SUMMARY OF THE INVENTION

[0023] It is, therefore, an object of the present invention to provide a method of displaying data in a tabular form.

[0024] It is another object of the present invention to provide a method of displaying tabular data on a web page, with the first horizontal row of data remaining stationary.

[0025] These and other objects of the invention are achieved through a provision of a method of displaying tabulated data, while retaining the header in a "stationary position." At the same time, the data table can be vertically scrolled and still remain in alignment with the header. The Header row of the top table can contain multiple column headings corresponding to each data column below in the respective bottom Data table. In this method, the Header table contains at least two rows, and the bottom Data table contains at least two rows. One row of the top Header table contains and displays the header text that is viewable. The second "Hidden" row contains placeholder text or HTML to automatically set the width of each corresponding header column. The placeholder text or html of each "hidden" row's Header column is determined by calculating the maximum widths of each data cell and its corresponding text or HTML in the bottom data table. The text or HTML within the cell that contains the maximum width for its corresponding column is then used as the placeholder in the hidden Header rows cell. The hidden row and cell uses CSS to assist in making the cell not visible to the user.

[0026] If there are images that need to be in the hidden row, the heights of these image are set to smaller value, e.g. height = "0" or height = "1" or height = "2", etc. This causes the hidden row to reduce in height while still having the original width. Additionally, another CSS method is set the visibility property for the Hidden row or the Hidden cell to the value of the "hidden." The scrollable portion of the data table is created by the use of DIV tags and setting this DIV tag's CSS attributes to values that cause a scrollbar to appear. The CSS attributes are typically height, overflow, overflow-x and overflow-y, etc. Such information can be set via a program or directly in a web page.

[0027] Each top Header column text or HTML width is then compared to the bottom Data table width for each column. If a top Header table's cell text or HTML is wider than any of the

bottom Data table's corresponding cell, another hidden row is added to the bottom Data table. This "hidden" row for the bottom Data table will contain any corresponding HTML or text in the Header table so that the columns of the bottom Data table will resize themselves to have the same widths of the top Header table columns. The two tables, top Header and bottom Data, swap data or headers labels and place them in their "Hidden" respective columns and rows. These hidden rows use CSS properties and values like or "visibility: hidden" or "display: none" to assist in hiding the "swapped" copies data or header labels. Furthermore, the "swapped" copies of data or header labels do not have to be exact copies of the data or header labels and can be any HTML or text that will cause the width of the corresponding column to match to opposite table, e.g. the top Header table or the bottom Data table.

[0028] The instant method also determines the width of the text and/or html in each cell of the bottom Data table. In particular, the text can use different types of proportional fonts, e.g. Arial, Times, etc. that are not of stationary widths, such as Courier. The method calculates the total width of all the characters in the text for a cell of data and uses this total width to determine which cell's HTML or text should be copied and placed in the opposite table's corresponding cell.

[0029] One important part of the method is the use of an additional "placeholder" row of data obscured by cascading style sheet properties; the second part is the calculated use of the widest cell in a column to place derived or replicated contents in an adjoining table so that the columns of the data and the stationary header align properly.

[0030] In another embodiment, each column width of the table, (<TD> in HTML), in pixels or percentages can be set dynamically according the calculation of the data widths and the Headers widths and choosing the maximum widths from the Headers cells and the Data cells. If percentages are desired, each column's maximum cell width could be summed together and then percentages can be determined for each Header and Data columns. Additionally, the maximum character width of the table can also be later modified to shorten overly long data via an ellipse "..." or some other indicator to lessen the maximum width of a column or columns.

[0031] When this stationary Header table system of the instant invention is used in conjunction with another table for a third (3^{rd}) table and with certain CSS and DIV tags and smart processing, a multi-column drop down list can also be made. Most, if not almost all, drop down lists on the Internet are single column drop down lists. There are a few that have multi-columns drop down lists, however there are limitations to these types of implementations. And, as such, there are essentially no mainstream or popular Web sites that use such multi-column drop down lists.

[0032] The instant method also allows displaying a multi-column drop down list in a web page. It aligns the columns of a top row of data with the columns of the bottom rows of data while

additionally collapsing or expanding the bottom rows of data depending upon the user input. The method allows users to see in a Web page, the fixed Header rows that users now see in spreadsheet programs such as Excel when they scroll vertically through rows of data. By having a fixed header rows' columns aligned to the below rows' columns, users can easily see the columns names that are located in the fixed Header row.

[0033] Another aspect of the present invention allows a user to accomplish resizing calculations at the server and/or client and will not require an unnecessary double click by the user to auto resize the columns or any other extra user intervention to get "right" sized column widths.

BRIEF DESCRIPTION OF THE DRAWINGS

[0034] Reference will now be made to the drawings, wherein like parts are designated by like numerals, and wherein,

[0035] Figure 1 is a schematic view of a one-column data display containing a header and one or more data cells.

[0036] Figure 2 is a schematic view of a multi-column table, with each column having a header and one or more data cells.

[0037] Figure 3 is a schematic view of a one-column data display having hidden header and hidden data cell.

[0038] Figure 4 is an illustration of a fixed header table with viewable hidden rows.

[0039] Figure 5 is an illustration of a table with a hidden header; the existing header cell is the widest display of the columns.

[0040] Figure 6 is an illustration of a table with a data cell being the widest display in the column.

[0041] Figure 7 is an illustration of a table with hidden data cell contents being the widest display of the column.

[0042] Figure 8 is an illustration of a table displayed in Code Snippet of CSS style sheet used to hide hidden rows.

[0043] Figure 9 is a schematic view of a multi-column drop down box with a fixed header table in a collapsed and expanded state.

[0044] Figure 10 is an illustration of HTML table with different columns in the cell rows being the widest.

[0045] Figure 11 is an illustration of two tables, wherein in the top table the data is removed and the in bottom table - the header is removed.

[0046] Figure 12 is an illustration table demonstrating an example of character widths for "Geneva" font suitable for use in the method of the present invention.

[0047] Figure 13 is an illustration table demonstrating an example of character widths for "Times" font suitable for use in the method of the present invention.

[0048] Figure 14 is a sample calculation sheet of display width for the phrase "New Orleans Saints."

[0049] Figure 15 is an illustration of a Web page where a user can change the text size setting displaying two tables above, even while they have the same <TD> width values, will not align properly.

[0050] Figure 16 is an illustration of a fixed sidebar table.

[0051] Figure 17 illustrates a step of determining the largest height cell in the main data table.

[0052] Figure 18 illustrates a step in a process where a main data table is positioned on top of a scrollbar of a fixed sidebar table and a place holder column.

[0053] Figure 19 illustrates a view on a computer monitor screen with two fixed header tables, with one of the tables acting as a fixed sidebar table.

DETAIL DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0054] Turning now to the drawings in more detail, Figure 10 illustrates an HTML table having a header 12, rows 14 and 16 and a plurality of vertical columns A, B, C, and D. HTML Tables will naturally collapse around the widest cell in a columns if the columns width attribute is not set (or declared) to some pixel width or percentage width. For example, ColumnA, the cell in row #2 is the widest. In ColumnB, the cell in row #1 is the widest. However, in ColumnC, the cell that contains the text, "ColumnC" in the header row is the widest of all rows, including row #1 and row #2. In this example, an HTML table without any width attributes that are declared will naturally collapse or shrink around the widest cell for each column.

[0055] This invention provides a method of organizing a display wherein a single datagrid is designed with individual tags and controls to make the fixed header section. Figure 11 illustrates such a datagrid containing a header and another datagrid directly underneath the datagrid contain the header with two rows and a plurality of vertical columns. In this illustration, since the data is separated from the datagrid containing the header, it is natural for the table to shrink to the minimum column width if no other table and cell width settings are declared. Browsers, such as IE, Mozilla, Opera, etc, each allow the individual user to set the Text Size of the contents of the web page to their own setting. This allows those with poor vision or those web pages that have the font size too small be customized by the individual user. This additional setting must also be accounted for in developing a fixed header table.

[0056] The method of displaying tabular data of the present invention can be accomplished by "swapping" and "hiding" steps. First, the software creator or web page designer determines the maximum cell width for each column on the scrollable data table, which will become a visible bottom table. In effect, the designer determines what information to swap between the tables. Then, a copy of the contents of each cell that is the maximum width for its column is placed in a non-visible row in the fixed Header table, which is positioned above the Data table. Then, the designer determines if the header content for each cell in the fixed Header table (top table) is larger in width (text or HTML) than the maximum cell width of the scrollable Data table (bottom table). If the header content is wider, then the designer places a copy of the header in an extra non-visible row in the scrollable Data table. The designer makes the swapped information non-visible in the holder-type rows, hiding the swapped copies in each table. Optionally, the designer adjusts any swapped images (if any) to have the height value of 1 (or to be more general a smaller height) bearing in mind that certain browsers need to have at least a height of one (1) to be able to sense that the image is there for the column to adjust. Some browsers will still respond to a zero (0) height.

[0057] As a result of the "swapping," there are two tables with the same number of columns. If the widest cell in the same column of one table is larger than that in the opposite table, than a copy of that text is placed in the opposite table thereby increasing the width of the opposite table to be that from the originating table. The same steps are repeated, except from the opposite table to the originating table. This process balances and equalizes the widths of text or HTML for each column for the two tables, the Header table and the Data table. As a result, the widths of the vertical columns of both tables are aligned with each other. If desired, the order of the above algorithm can be re-arranged in the code, but the overall method and concept remain the same.

[0058] In an example shown in Figure 1, a table 20 is formed with a Header table 22 and a Data table 24 positioned below the Header table 22. The Header table 22 contains a visible Header Cell A designated by numeral 26 and a Hidden Header Cell A designated by numeral 28. The bottom Data table 24 contains a first visible Data Cell A1 designated by numeral 30 and a second visible Data Cell A2 designated by numeral 32. A Hidden Data Cell A designated by numeral 34 is positioned below the second Data Cell 32. The Hidden Data Cell 34 is used as a place holder from data derived from Header table 22 and the Header Cell 26. The Hidden Header Cell 28 expands or contracts width-wise depending on the type of information placed in the visible Data Cells 30 and 32. Typically, the wider of first or second Data Cells 30, 32 is chosen as a guide, and a copy of that information is placed in the Hidden Header Cell 28. Similarly, if the contents of the visible Header cell 26 are wider than any of the Data cells 30, 32, a copy or representative of the Header cell 26 is placed in the Hidden Data cell 34.

[0059] As a result of "swapping" and "hiding," the designer creates two tables, a top Header table 22 and a bottom Data table 24 that have the same display width and accordingly the same width column-wise resulting in alignment of the borders of the two tables, table 22 and table 24.

[0060] Figure 2 illustrates application of the above-described steps in a multi-column environment. Here, a Header table 40 has three visible Header cells A, B and C for the respective columns A, B and C, as well as Hidden Header cells A, B, and C for columns A, B, and C. The Data table 42 positioned below the Header table 40 has one or more data cells A1, B1, C1 and optionally Data cells A2, B2 and C2. Hidden Data cells for each respective column are positioned below the Data cells A2, B2 and C2. For each column A, B, and C, the widest display width of a column of Data Cells in the Data table 42 is determined. For example, the widest width in text or HTML of Data cell A1 or A2 is determined and then contents of the Hidden Header cell A is filled with the wider value of Data cell A1 or A2 to achieve the same resulting width equivalent to the widest width of column A of the Data table.

[0061] Figure 3 illustrates an example of a situation where the widest display from both the Header table and the Data table is replicated in the opposite table's Hidden row/cell. In the method of designing a stationary Header, the designer first determines if Data cell A1 in the Data table 44 has a display width that is greater than in the Data cell A2. the "display width" is the total character width and image width in a line. If there are multiple lines, then the line with the largest total character widths and images is chosen. A character width's units can be in pixels, points, em's, or any other absolute or relative measurement unit. Figures 12 – 14 show examples for the size for each letter and the calculations for a phrase that can be in a data or header cell.

[0062] If the Data cell A2 is wider in text or HTML, then a copy of the contents or representation of the contents of Data cell A2 is placed in the Hidden Header cell A of the Header table 46. Conversely, if the contents of the Header cell A occupy more display width than Data cell A1 or Data cell A2, a copy of the Header cell A or a representation of the contents of the Header cell A is placed in Hidden data cell A below the Data cell A2. Both Hidden Header cell A and Hidden Data cell A use the cascading style sheet property: visibility: hidden; *and line-height; 1px*;

[0063] If there are images that will be placed in Hidden Header cell A or Hidden Data cell B, then the height of the images will be modified to be height = "1" or height = "0" The steps of comparing the widths of the display and then copying the contents in the Hidden Header or Hidden Data cells are repeated for the next column. As a result, the Header Table 46 and the Data table 44 will have equal widths and accordingly the same vertical column alignment, widths and border alignments.

[0064] Figure 4 shows an example of a table with a fixed or stationary Header with Hidden rows viewable on a user's screen. The Hidden row area in the Hidden Header row 48 is partially duplicated from the visible Header and data areas, the areas that have the largest "display widths." The first visible row 50 contains a first column 52 identified as "Order # ID," a second column 54 identified as "Customer ID, a third column 56 identified as "Ship Name", a fourth column 58 identified as "Order Date" and a fifth column 60 identified as "ShipCountry." The widest text in the third column 56 "Vins et alcools Chevalier" visible in the row across from Order # 10248 in the visible Data Table 68 controls the width of the Hidden Header cell 62 and Hidden Data cell 64 in the Hidden Data row 66.

[0065] In the illustration shown in Figure 5, no scrollbar is shown. In this illustration, the Header cell 70 is the widest display of the respective column 72. The Header cell 70 is also wider than the text in the Data table 74 for the column 72. The Hidden Header cell 76 contains data replicated from the Header cell 70. The image was reduced to a height of "1" so that it will be hidden in the future. The same text, being widest for the column is reproduced in the Hidden Data cell 78, which occupies the row at the very bottom of the column. However, it should be noted that the Hidden Data cell may be positioned in the middle of the column or at the very top of the column, depending on the designer's preference.

[0066] Figure 6 presents an illustration wherein one of the Data cells is the widest in the column. In this illustration, Data cell 80 presents the largest display width for column 82. The same data is replicated in the Hidden Header 86 and the Hidden Data cell 88. Although the Hidden Header and Data table cells 86, 88 can be seen in Figure 6, it will be understood that a CSS style property will be used to "hide" these contents. For instance, CSS properties visibility: hidden; *and line-height: 1px*; can be used to hide the Hidden Header cell 86 and the Hidden Data cell 88.

[0067] Figure 7 illustrates a display where one of the Data cells, in this case Data cell 90, is the widest in the column. The text in the Hidden Data cell 90 has been derived from the text, which appears somewhere in the column 92 and is not visible in the illustration since the user used a scrollbar 94 to move down along the columns 92 with respect to the text "Centro comercial Moctezuma." If there are several pages of data in the table, the widest data value could be in the pages not currently visible within the browser even if the user moves down the scrollbar. The Hidden Header cell 96 is adjusted to have the same width as the Hidden Data cell 90. The scrollbar 94 can be created with <DIV> tags and the CSS attributes: height and overflow-x and overflow-y.

[0068] Figure 8 illustrates a Code Snippet 98 of CSS style sheet that may be used to hide the hidden rows (place holder rows). In this illustration, the hidden rows of the Header cells and Data cells have been hidden and are not visible to the user.

[0069] Figure 9 illustrates a multi-column dropdown box with a Stationary Header in a collapsed and expanded state. As the multi-column dropdown box used more <DIV> tags that will make the Data table and its Header row be hidden or disappear. As a result, the display on the user's computer terminal will look like the Collapsed State view in Figure 9. An extra <DIV> tag surrounds the entire Data table as well as the Header table. Another program method or function controls the <DIV> attributes, making it hidden or not hidden.

[0070] Figures 12 and 13 provide a sample of possible character widths for "Geneva" and "Times" fonts, providing units in Pixels for conventional font sizes. Different fonts have different widths as well. Also, within the same font, the point sizes can have different relative widths when compared to other point sizes. Because there are so many different sizes and variations, it is preferred to make a hidden copy and let the user's browser perform the sizing operation. Figure 14 illustrates a sample calculation of display width for the phrase "New Orleans Saints," with the total size being 94.21223 pixels.

[0071] Figures 12 and 13 also show that many characters have the same width within the same font group or matches character widths from other font groups. Accordingly, the present invention can substitute other characters or a combination of other characters for the original characters in the Hidden Header cell 86 and the Hidden Data cell 88 to adjust the width of the column. Encrypted or "representative" data can be used to also adjust the widths of the columns. Thus, instead of copying data to the Hidden Header cell 86, an association can be made between the Hidden Header cell 86 and the maximum data cell in a column and then "representative" data is then copied into the Hidden Header cell 86. Likewise, the same association can be done for the Hidden Data cell 88 and Header cell where representative data is copied into the Hidden Data cell 88.

[0072] The present invention also provides an alternative method of creating display tables which have "stationary" headers. According to the alternative method the software program first determines the maximum cell width for each column on the scrollable Data table, or bottom table. The program then determines whether the Header contents for each cell in the fixed top table is larger in width than the maximum cell width of the scrollable table (bottom table). At the next step, the program calculates the minimum column widths and sets each column in both tables to the same minimum column width, thereby aligning the fixed top table with the scrollable bottom table.

[0073] Since setting the width of the table cell <td> can also control the width of the column, the program, instead of putting place holder data inside the hidden cell, as discussed in the first embodiment of the present invention, can simply calculate and determine the cell with maximum width of each column and then calculate what the width attribute of the <td> cell should be,

<td width="maximum calculated number in pixels">

[0074] It is envisioned that the alternative method may require certain in calling for setting of the table cell's width, <td width="in pixels">. For instance, if a programmer wishes to set the table to be 100% width, the fixed <td>'s may not flow and stretch as easily and cleanly as not setting the width of the <td> to begin with. Such situation may be encountered when for instance, a cell has a lot of text that needs to wrap to the next line down. Sometimes it is desirable for the website to have the table to be 100% width and take up as much room as possible width wise. And sometimes it's desirable for the table to shrink to the smallest possible width to save space.

[0075] There are numerous ways, by which a program can set the column width. For instance, the column width may be determined by setting the <td> width attribute to: (1) a pixel value, or (2) a percentage value, or (3) by setting no values and letting the contents inside the cell determine the width of the cell. Furthermore, adjacent column contents and width attribute value can also affect the column width, especially the percentage widths. Even then, a programmer may encounter situations where tabular data to be displayed has multiple lines, some short, some combined with images, with the headers having images, or text, or a combination thereof. The calculated font size of each character can change from browser to browser. The windows size that the user set his/her browser, the resolution, etc. Each font type has subtle changes between one program to another program, including those main stream and standards compliant browser programs. These slight differences can be seen in a program or browser when the columns of the headers do not align perfectly with the data columns.

[0076] The tables below and the illustration in Figure 15 have the same TD values for the width attribute. The cell code and width values are illustrated in the table below:

```
<td width="50"> </td>
<td width="45"> </td>
<td width="65"> </td>
```


asf	**12345**	abcdefg

[0077] As one can see in the two tables above, their columns are in alignment with each other. For illustration purposes, a space has been placed between the two tables so one can see that they are actually separate. But in practice, the two tables are closer together, vertically, and even in contact with each other. However, it is possible that should a user change the text size setting in the table above via the browser option shown in Figure 15, the two tables shown above will start to expand at different rates as the user changes the text size setting in their browser. Thus, the two tables shown above will have their columns out of alignment with each other. If the fonts inside the fixed header are fixed to a pixel setting, as opposed to an adjustable setting, it is envisioned that the table columns will align in at least a browser entitled "Internet Explorer" and perhaps when using other browsers. Thus, instead of setting the width of each column, the program should determine the maximum cell width for each column, if the program wishes all the columns to still align properly.

[0078] The multi-column drop down list described in this application shows that it can finally be done on a production quality level that is cross-browser compliant. The multi-column drop down list uses practically the same swapping and synchronizing for the selected data rows as that described in the stationary Header table above. This third (3^{rd}) table has hidden rows via CSS that are still populated with the correct data such that the columns width will still be aligned with the other data table columns and header table columns. Additionally, when a user selects an item in the Data table, that item is populated in this third (3^{rd}) table.

[0079] It is envisioned that many computer languages can be used to implement the instant method of creating a data table with a stationary header. It should be noted that to implement the instant method, the user may employ the server of a "client/server" configuration, a client's browser via a language like JavaScript or Java or VBScript. Furthermore, parts of this method can take place on both the web server and the client.

[0080] With regards to the server-side languages, there are many to choose from. For example, Java, J2EE, Perl, PHP, ColdFusion, Python, Visual Basic, ASP.NET, C#, J#, JSP, HTML, machine, or assembly language can be used to implement the method described here. The data for these tables will typically come from a database such as MS Access, SQL Server, IBM DB2, Oracle, MySQL, etc. Nevertheless, it can also come from an XML file, flat file, JavaScript Object, or some customized data format. The data retrieval can also use a method referred to as AJAX, asynchronous JavaScript and XML to improve performance and response time to between the server and the client browser. Again, the variations are different in regards to languages and databases used, but the same method of creating a stationary header and aligning the columns can and is still used.

[0081] An additional feature of the present invention is a method of making a fixed side bar of a cell that adjusts to the data cell row height but also scrolls up and down and stays in vertical

synchronization with the main fixed header table while this main header table is scrolled left or right to see hidden columns. This task may appear somewhat easier than creating a fixed header as there typically can be only 1, 2, 3, etc. lines in height. Still, if a full spreadsheet program such is Excel, with its large feature set, were to be implemented using this invention, the program may get rather sophisticated.

[0082] In forming a fixed side bar, the same concept of swapping for the place holder column is used. The program looks for the number of line breaks and performs the data cell swap accordingly. Thus, there are two side-by-side Fixed header tables where, for example, the entire left fixed header table acts also as a Fixed Sidebar. The scrollbar for the left header table is either removed or somehow hidden, but, through some programming language, the Fixed Sidebar Table scrollbar position remains in sync with the main fixed header table's scrollbars position.

[0083] Referring to Figures 16 and 17, a place fixed sidebar table 100 comprising a place holder column 102 and a fixed sidebar column 108. The place holder column 102 has a header "Customer ID" and a plurality of data cells 104. One of the data cells 105 has the largest amount of information and consequently has the largest height value.

[0084] A sliding scrollbar 106 is shown on the right of the place holder column 102. The fixed sidebar column 108 is shown to the left of the place holder column 100. The main data table 110 may contain additional columns, such as shown in Figure 17, where a "Ship Name" column 112, "Order Date" column 114 and "Ship Country" column 116 are added. Of course, the main table may contain only one column, depending on the nature of the displayed data.

[0085] In the process of designing a fixed sidebar, one of the steps provides for hiding the place holder column 102 and the scrollbar 106 of the fixed sidebar table by the main data table 110 using a CSS property, for instance CSS z-index layers, while still leaving the fixed sidebar column 108 visible. The place holder column 102 is created by determining the cell in the main data table 110 or sidebar table 100 that has the largest height value. In this illustration, it is cell 105. Similarly, if the cell has only text, the value is determined based on the number of lines, carriage returns or HTML <BP> or <P> tags.

[0086] As can be seen in Figure 18, the main data table 110 partially covers (is placed "on top of") the fixed sidebar table 100, that the place holder column 102 and the scrollbar 106 of the fixed sidebar table 100. Thus, when the main table indicia covers the scrollbar 106 of the fixed sidebar table 100, both table 100 and 110 may appear as one table.

[0087] Figure 19 illustrates another variation of the application of the instant invention to a display page, where two fixed header tables are presented, with one of the tables acting as a fixed sidebar table. In this illustration, a large table 112 has a fixed sidebar table 114 and a plurality of data

columns 115. The fixed sidebar table 114 acts as fixed column that does not change when a horizontal scrollbar 116 is used or moved left or right. The horizontal scrollbar 116 is placed on the bottom of the table, allowing the user to move the main data table 118 horizontally to see other columns of data to the left or right. A vertical scrollbar 120 allows the user to scroll the main table up and down, while the header may or may not remain fixed, depending on the choice of the program designer.

[0088] The present invention allows a program designer to create a column sidebar with two fixed header tables that use the same data but shown using two side by side tables. The main table "overlaps" or covers the fixed sidebar table so that the place holder column and the scrollbar of the fixed sidebar table are hidden via some CSS Property (or other similar means) that can be used to hide the fixed sidebar table's place holder column and scrollbar.

[0089] Additionally, the main table's vertical scrollbar is synchronized with the fixed sidebar's scrollbar so that the rows in both tables are aligned with respect to each other. This can be done via the swapping method of data and looking for the cell with the greatest height and introducing the data in the fixed sidebar table's option as well as hidden 2nd column. Both tables have the same vertical position scrollbar position with respect to each other when scrolling up or down. This can be done via a client side JavaScript script in the web page in a variety of ways.

[0090] By horizontally scrolling left or right in the main data table, the fixed header table's header and data from a hidden column are swapped into next visible column's location while a visible column in the main data table (left or right, depending upon the direction of scrolling) is made hidden.

[0091] Many changes and modifications can be made in the method of the present invention without departing from the spirit thereof. I, therefore, pray that my rights to the present invention be limited only by the scope of the appended claims.

I Claim:

1. A method of displaying on a computer-controlled screen, records arranged in a table, said table having columns, rows and a plurality of cells defined by the columns and rows, the method comprising the steps of:

forming a first non-scrollable sub-table for displaying headings of the columns;

forming a second scrollable sub-table for displaying records corresponding to the headings in the first sub-table;

defining maximum size parameters for each column in the first sub-table and the second sub-table based on parameters of a maximum width or height cell within each column;

defining a visible part of the first sub-table to be visible on the computer-controlled screen and a hidden part of the first sub-table that is non-visible on the computer-controlled screen; and

copying data of the maximum parameter cell into a cell within a corresponding column in the hidden part of the first sub-table, thereby defining the width or height parameters of the corresponding column.

2. The method of Claim 1, wherein the records to be copied into the hidden first sub-table have a pre-determined height adjusted to a pre-determined geometric attribute value prior to copying the records into the hidden part of the first sub-table.

3. The method of Claim 2, wherein the geometric attribute value has the height value of 1 or less.

4. The method of Claim 1, wherein a cell width is determined by setting the <td> width attribute to a pixel value of the record within the cell.

5. The method of Claim 1, wherein a cell width is determined by determining a percentage value of the corresponding column in relation to remainder data table.

6. The method of Claim 1, wherein the non-scrollable first sub-table defines a fixed header of the table.

7. The method of Claim 1, wherein the first sub-table defines a fixed sidebar of the table.

8. Computer program code embodied in a computer-readable medium, comprising:

computer-executable program code for defining a set of parameters for a record of the database arranged in a data table having columns, rows and a plurality of cells defined by the columns and rows;

computer-executable program code for determining maximum width or height parameters for each column or row in the data table based on the records to be displayed in a cell having the maximum width or height parameters;

computer-executable program code for establishing a first sub-table defining a non-scrollable part of each column;

computer-executable program code for establishing a second sub-table defining scrollable part of each column;

computer-executable program code for establishing a visible part of the first sub-table and a hidden part of the first sub-table;

computer-executable program code for associating records from the maximum parameter data cell into a cell within a corresponding column in the hidden part of the first sub-table, to thereby define width or height parameters of the corresponding columns or rows in the first sub-table and the second sub-table.

9. The computer program code of Claim 8, wherein the records to be associated into the hidden first sub-table have a pre-determined height adjusted to a pre-determined geometric attribute value prior to copying the records into the hidden part of the first sub-table.

10. The computer program code of Claim 9, wherein the geometric attribute value has the height value of 1 or less.

11. The computer program code of Claim 8, wherein a cell width is determined by setting the <td> width attribute to a pixel value of the record within the cell.

12. The computer program code of Claim 8, wherein a cell width is determined by determining a percentage value of the corresponding column in relation to remainder data table.

13. The computer program code of Claim 8, wherein the non-scrollable first sub-table defines a fixed header of the table.

14. The computer program code of Claim 8, wherein the non-scrollable first sub-table defines a fixed sidebar of the table.

15. The computer program code of Claim 8, wherein the records to be associated with the hidden first sub-table are copied into hidden part of the first sub-table.

16. The method of Claim 1, wherein the data of the maximum parameter cell copied into a cell within a corresponding column in the hidden part of the first sub-table is representative data of the maximum parameter cell.

17. A method for creating stationary headers rows in a web page for a table of data that is scrollable, the method comprising:

associating to the stationary header rows of at least one placeholder from the table of data;

placing a placeholder in a corresponding table to cause a column to automatically size to same width as a corresponding originating table cell and column; and

calculating the maximum text width for each cell of data.