

JavaScript 行列を固定したテーブル

[JavaScript]

jQuery が出てきてから、すごく楽になった。。

こんなことしなくても、jQuery プラグイン使えばよい。

インラインフレーム等を使わずに、行列を固定したテーブルを実現

title1	title2	title3	t
value(1,1)	1,2)	value(1,3)	value(
value(1,2)	2,2)	value(2,3)	value(
value(1,3)	3,2)	value(3,3)	value(

画面のリサイズ対応

HTML

```
<body onresize="resizePanel();" ...
  <div id="HEADER_PANEL" style="overflow:hidden;">
    <table styl="table-layout:fixed;" ...
  </div>
  <div id="ITEM_PANEL" style="overflow:hidden;">
    <table style="overflow:auto;" ...
  </div>
  :
```

JavaScript

```
/**
 * 画面のサイズ変更から呼び出す
 */
function resizePanel() {
  resizeTablePanel(
    80
    , 30
    , document.getElementById("HEADER_PANEL")
    , document.getElementById("ITEM_PANEL")
  );
}

/**
 * @param Description : 画面のサイズに合わせて、サイズを変更するテーブル
 * @param Arguments :
 * @param verticalMargin : 垂直マージン
 * @param horizontalMargin : 水平マージン
 * @param headerPanel : テーブルヘッダーパネル (DIV)
 * @param itemPanel : テーブルアイテムパネル (DIV)
 * @param containerHeight : 基準となるコンテナの高さ (省略時、BODY タグのクライアント領域高さ)
 * @param containerWidth : 基準となるコンテナの幅 (省略時、BODY タグのクライアント領域幅)
 */
function resizeTablePanel( verticalMargin, horizontalMargin, headerPanel, itemPanel
  , containerHeight /* document.body.clientHeight */
  , containerWidth /* document.body.clientWidth */ ) {
  // 規定値の設定
  if (containerHeight == undefined) containerHeight = document.body.clientHeight; // コンテナ高さ
  if (containerWidth == undefined) containerWidth = document.body.clientWidth; // コンテナ幅

  var SCROLL_BAR_MARGIN = 17; // スクロールバー幅 規定値
  var isVerticalScrollBar = false; // 垂直スクロールバーが表示されるか
```

```

var scrollbarMargin      = 0;          // 垂直スクロールバーマージン

// テーブル幅が変わらない設定 (overflow-y:auto or overflow-y:scroll)
var isFixWidth    = itemPanel.style.overflowY == 'scroll'
                  || itemPanel.style.overflowY == 'auto';

// コンテナの高さが、指定マージンより大きければ、テーブル明細パネルの高さを変更する
if (containerHeight > verticalMargin) {
    itemPanel.style.height = containerHeight - verticalMargin;
}
// 垂直スクロールバーが表示されるかを判定する
var adjust = 16; // 調整 (行が半分隠れている場合等正しく判定されないため)
isVerticalScrollbar = (itemPanel.style.overflow == 'scroll')          /* 常時
表示 */
                      || ((totalOffsetHeight(itemPanel)+adjust) >= itemPanel.offsetHeight) /* 子要素のほうが高い */

// コンテナの幅が、指定マージンより大きければ、
// テーブルヘッダーパネルおよびテーブル明細パネルの幅を変更する
if (containerWidth > horizontalMargin) {
    itemPanel.style.width = containerWidth - horizontalMargin;

    // 垂直スクロールバー有無により、ヘッダーの幅を調整
    if (isVerticalScrollbar) {
        scrollbarMargin = SCROLL_BAR_MARGIN;
    }

    // テーブル幅固定の場合、ヘッダーリサイズしない
    if (!isFixWidth) {
        var headerMargin = (horizontalMargin + scrollbarMargin);
        var headerWidth = containerWidth - headerMargin;
        if (containerWidth <= headerMargin) {
            headerWidth = 0;
        }
        headerPanel.style.width = headerWidth
    }
}
return;
}

/**
 * @param Description : 指定されたオブジェクトの直接の子どもの高さの合計を取得
 * @param Arguments   :
 * @param parent      : 対象となる親オブジェクト
 */
function totalOffsetHeight( parent ) {
    var children = parent.children;
    var h = 0;
    if (children == null) {
        return h;
    }

    for (var i=0; i<children.length; i++) {
        h += children[i].offsetHeight;
    }
    return h;
}

```
