

sortierbare Tabelle im Editor

Im WYSIWYG Editor kann man mit diesem Script die Spalten in Tabellen mit einem Doppelklick auf den Header nach diesem sortieren. Die anderen Spalten werden entsprechend mitsortiert.

```
<script>

// Hook into the WYSIWYG editor setup event and add our logic once loaded
window.addEventListener('editor-tinymce::setup', event => {
  const editor = event.detail.editor;
  setupTableSort(editor);
});

// Setup the required event handler, listening for double-click on table cells.
function setupTableSort(editor) {
  editor.on('dblclick', event => {
    const target = event.target;
    const parentHeader = target.closest('table tr:first-child td, table tr:first-
child th');
    if (parentHeader) {
      // Sort out table within a transaction so this can be undone in the editor if
required.
      editor.undoManager.transact(() => {
        sortTable(parentHeader, editor);
      });
    }
  });
}

// Sort the parent table of the given header cell that was clicked.
function sortTable(headerCell) {
  const table = headerCell.closest('table');
  // Exit if the table has a header row but the clicked cell was not part of that header
  if (table.querySelector('thead') && headerCell.closest('thead') === null) {
    return;
  }
}
```

```

const headerRow = headerCell.parentNode;
const headerIndex = [...headerRow.children].indexOf(headerCell);
const tbody = table.querySelector('tbody');
const rowsToSort = [...table.querySelectorAll('tbody tr')].filter(tr => tr !==
headerRow);
const invert = headerCell.dataset.sorted === 'true';

// Sort the rows, detecting numeric values if possible.
rowsToSort.sort((a, b) => {
    const aContent = a.children[headerIndex].textContent.toLowerCase();
    const bContent = b.children[headerIndex].textContent.toLowerCase();
    const numericA = Number(aContent);
    const numericB = Number(bContent);

    if (!Number.isNaN(numericA) && !Number.isNaN(numericB)) {
        return invert ? numericA - numericB : numericB - numericA;
    }

    return aContent === bContent ? 0 : (aContent < bContent ? (invert ? 1 : -1) :
(invert ? -1 : 1));
});

// Re-append the rows in order
for (const row of rowsToSort) {
    tbody.appendChild(row);
}

// Update the sorted status for later possible inversion of sort.
headerCell.dataset.sorted = invert ? 'false' : 'true';
}
</script>

```