

# Fußnoten über den WYSIWYG Editor

Fußnoten können mit folgendem Code aktiviert werden:

```
<script>

// Take a footnote anchor and convert it to the HTML that would be expected
// at the bottom of the page in the list of references.
function footnoteToHtml(elem) {
    const newWrap = document.createElement('div');
    const newAnchor = document.createElement('a');
    const sup = document.createElement('sup');
    const text = document.createTextNode(' ' + elem.title.trim());
    sup.textContent = elem.textContent.trim();
    newAnchor.id = elem.getAttribute('href').replace('#', '');
    newAnchor.href = '#';
    newAnchor.append(sup);
    newWrap.append(newAnchor, text);
    return newWrap.outerHTML;
}

// Reset the numbering of all footnotes within the editor
function resetFootnoteNumbering(editor) {
    const footnotes = editor.dom.select('a[href^="#bkmrk-footnote-"]');
    for (let i = 0; i < footnotes.length; i++) {
        const footnote = footnotes[i];
        const textEl = footnote.querySelector('sup') || footnote;
        textEl.textContent = String(i + 1);
    }
}

// Update the footnotes list at the bottom of the content.
function updateFootnotes(editor) {
    // Filter out existing footnote blocks on parse
    const footnoteBlocks = editor.dom.select('body > div.footnotes');
```

```

for (const blocks of footnoteBlocks) {
  blocks.remove();
}

// Gather our existing footnote references and return if nothing to add
const footnotes = editor.dom.select('a[href^="#bkmrk-footnote-"]');
if (footnotes.length === 0) {
  return;
}

// Build and append our footnote block
resetFootnoteNumbering(editor);
const footnoteHtml = [...footnotes].map(f => footnoteToHtml(f));
editor.dom.add(editor.getBody(), 'div', {class: 'footnotes'}, '<hr/>' + footnoteHtml.join('\n'));
}

// Get the current selected footnote (if any)
function getSelectedFootnote(editor) {
  return editor.selection.getNode().closest('a[href^="#bkmrk-footnote-"]');
}

// Insert a new footnote element within the editor at cursor position.
function insertFootnote(editor, text) {
  const sup = editor.dom.create('sup', {}, '1');
  const anchor = editor.dom.create('a', {href: `#bkmrk-footnote-${Date.now()}`, title: text});
  anchor.append(sup);
  editor.selection.collapse(false);
  editor.insertContent(anchor.outerHTML + ' ');
}

function showFootnoteInsertDialog(editor) {
  const footnote = getSelectedFootnote(editor);

  // Show a custom form dialog window to edit the footnote text/label
  const dialog = editor.windowManager.open({
    title: 'Edit Footnote',
    body: {
      type: 'panel',
      items: [{type: 'input', name: 'text', label: 'Footnote Label/Text'}],
    },
  },

```

```

buttons: [
  {type: 'cancel', text: 'Cancel'},
  {type: 'submit', text: 'Save', primary: true},
],
onSubmit(api) {
  // On submit update or insert a footnote element
  const {text} = api.getData();
  if (footnote) {
    footnote.setAttribute('title', text);
  } else {
    insertFootnote(editor, text);
    editor.execCommand('RemoveFormat');
  }
  updateFootnotes(editor);
  api.close();
},
});

if (footnote) {
  dialog.setData({text: footnote.getAttribute('title')});
}
}

// Listen to pre-init event to customize TinyMCE config
window.addEventListener('editor-tinymce::pre-init', event => {
  const tinyConfig = event.detail.config;
  // Add our custom footnote button to the toolbar
  tinyConfig.toolbar = tinyConfig.toolbar.replace('italic ', 'italic footnote ');
});

// Listen to setup event so we customize the editor.
window.addEventListener('editor-tinymce::setup', event => {
  // Get a reference to the TinyMCE Editor instance
  const editor = event.detail.editor;

  // Add our custom footnote button
  editor.ui.registry.addToggleButton('footnote', {
    icon: 'footnote',
    tooltip: 'Add Footnote',
    active: false,
  });
});

```

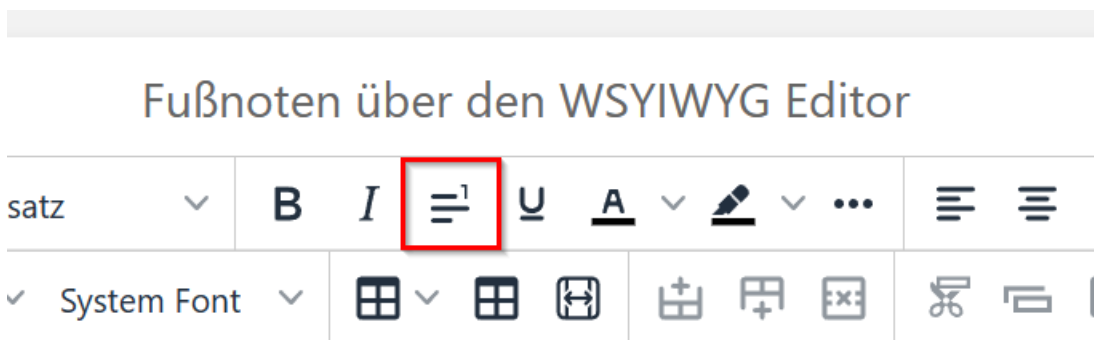
```

onAction() {
  showFootnoteInsertDialog(editor);
},
onSetup(api) {
  editor.on('NodeChange', event => {
    api.setActive(Boolean(getSelectedFootnote(editor)));
  });
},
});

// Update footnotes before editor content is fetched
editor.on('BeforeGetContent', () => {
  updateFootnotes(editor);
});
});
</script>

```

Das Ergebnis sieht dann wie folgt aus:



Dabei wird an der Stelle des Cursors eine Fußnote<sup>1</sup> erstellt, welche dann am Ende der Seite eingefügt wird.

Hier gibt es auch direkt das Beispiel<sup>2</sup> dazu.

Quelle<sup>3</sup>

<sup>1</sup> Erste Fußnote

<sup>2</sup> Beispiel

<sup>3</sup> <https://www.bookstackapp.com/hacks/wysiwyg-footnotes/>