# TYPO3 Core - Bug #93302

## Pre-formatted text in RTE gets re-formatted when inside an ordered / unordered list

2021-01-17 10:46 - Joschi Kuphal

| Status: | New | | Start date: | 2021-01-17 |
|---|---|---|---|---|
| Priority: | Should have | | Due date: | |
| Assignee: | | | % Done: | 0% |
| Category: | RTE (rtehtmlarea + ckeditor) | | Estimated time: | 0.00 hour |
| Target version: | next-patchlevel | | | |
| TYPO3 Version: | 10 | | Complexity: | easy |
| PHP Version: | 7.4 | | Is Regression: | |
| Tags: | | | Sprint Focus: | |

**Description**

When a pre-formatted text (pre element) gets entered into an RTE enabled field in the backend, the formatting might get destroyed during persistence depending on the context of the pre element:

- The formatting stays in order as long as the pre element is a top-level element or a child of particular other elements
- The formatting might get destroyed as soon as the pre element is a child of an ordered or unordered list (ol / ul) and potentially other elements as well

Due to the nature of the problem it's difficult / impossible to give a representative RTE value example that would survive the format mangling done by Forge ... Just imagine a pre-formatted code block inside a ul > li.

The reason for the problem is that only particular HTML elements are parsed recursively in RteHtmlParser::TS_transform_db (TYPO3 10.4, taken from line 416 onwards):

```
        // Traverse the blocks
        foreach ($blockSplit as $k => $v) {
            if ($k % 2) {
                // Inside block:
                // Init:
                $tag = $this->getFirstTag($v);
                $tagName = strtolower($this->getFirstTagName($v));
                // Process based on the tag:
                switch ($tagName) {
                    case 'blockquote':
                    case 'dd':
                    case 'div':
                    case 'header':
                    case 'section':
                    case 'footer':
                    case 'nav':
                    case 'article':
                    case 'aside':
                        $blockSplit[$k] = $tag . $this->TS_transform_db($this->removeFirstAndLastT
ag($blockSplit[$k])) . '</' . $tagName . '>';
                        break;
                    case 'pre':
                        break;
                    default:
                        // usually <hx> tags and <table> tags where no other block elements are wi
thin the tags
                        // Eliminate true linebreaks inside block element tags
                        $blockSplit[$k] = preg_replace('/[' . LF . ']+/', ' ', $blockSplit[$k]);
                }
            } else {
```

Right now, the default case gets triggered for ordered and unordered lists, eliminating linebreaks within the tag content and thus effectively destroying the formatting of pre-formatted text (even though pre elements are treated individually). It seems lists should be treated just like the other block level elements (i.e. parsed recursively). Adding ul and ol (and potentially other elements?) to the list of

recursively transformed elements seems to solve the problem.