GUTENBERG

(1397 - 1468) uitvinder van de boekdrukkunst in Europa

GUTENBERG

(2017-) WordPress editor

g

Gutenberg

By Gutenberg Team

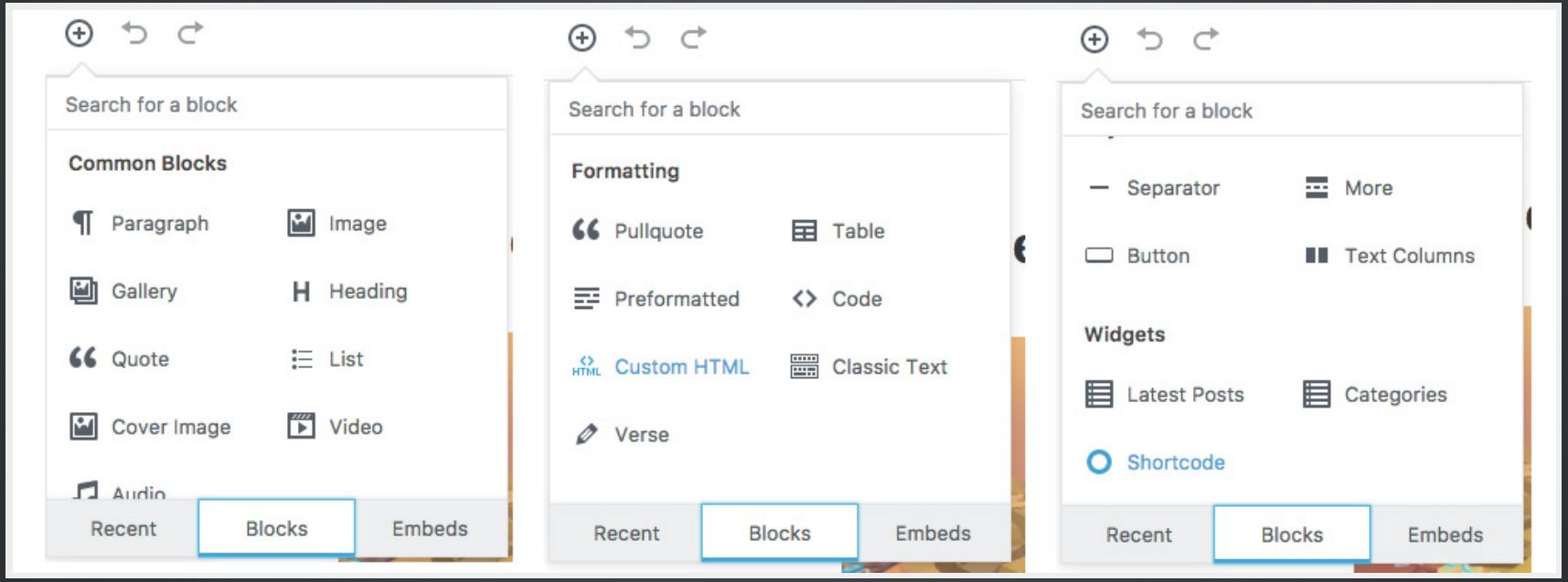
Download

INTERESSANT OMDAT:

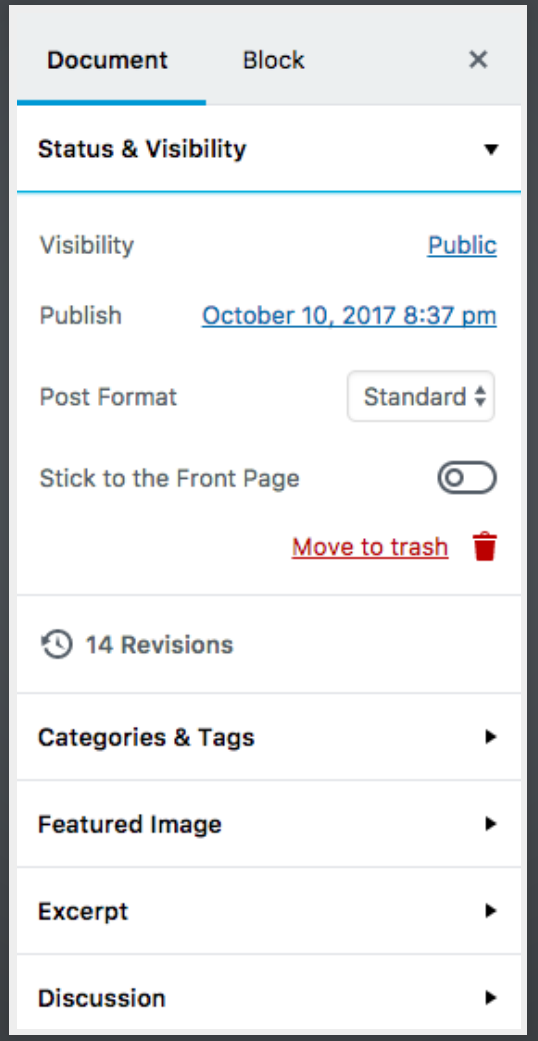
- WP5.0 pas komt als Gutenberg af is
- Niemand het er mee eens is (veranderd langzaam)
- Metaboxen oorspronkelijk zouden verdwijnen
- Dit een aardige componenten (blocks) aanpak heeft
- In react geschreven is

EDITOR COMPONENTEN

- .wp-block
- Block tabje in sidebar, custom properties per block
- Echte wysiwyg
- Verplaatsbaar



SIDEBAR


Document
Block
Status & Visibility
Visibility
Public
Publish
October 10, 2017 8:37 pm
Post Format
Standard
Stick to the Front Page
Move to trash
14 Revisions
Categories & Tags
Featured Image
Excerpt
Discussion

Demo

ZELF BLOKKEN MAKEN

Leuk en aardig die standaard blokken, maar niet genoeg

- Voorheen: Html in de editor
- Nu: een lijst van objecten en attributen
- Html van verschillende blokken kan hetzelfde zijn, maar totaal anders eruit zien

```
[
    {
        type: 'core/cover-image',
        attributes: {
            url: 'my-hero.jpg',
            align: 'full',
            hasParallax: false,
            hasBackgroundDim: true
        },
        children: [
            "Gutenberg posts aren't HTML"
        ]
    },
    {
        type: 'core/paragraph',
        children: [
            "Lately I've been hearing plen…"
        ]
```

- Gutenberg code blijft in het geheugen en is niet zichtbaar aan de front-end / back-end text view. (Like a printed page has no trace of the structure of the letters that originated it in the press)
- `post_content` bevat geserializeerde output
- HTML comments worden gebruikt voor de logica in de back-end, en worden niet geoutput in de front-end

```
<!-- wp:core/cover-image {"url":"http://localhost/turbolinks/wp-content/uploa
<section class="wp-block-cover-image has-background-dim" style="background-ima
    <h2>Dit is tekst over de foto</h2>
</section>
<!-- /wp:core/cover-image -->
```

BLOCKS API

- block.js - component
- editor.css - back-end styles
- style.css - front-end styles
- index.php - enqueue assets logic

CSS

Spreekt voor zich...

PHP

block.js & editor.css

```
add_action( 'enqueue_block_editor_assets', 'iv_custom_block_editor_assets' );
```

style.css

```
add_action( 'enqueue_block_assets', 'iv_custom_block_assets' );
```

JS

```
const { __ } = wp.i18n;
const { registerBlockType } = wp.blocks;
registerBlockType( 'iv/slider', {
        title: __( 'Slider', 'ivaldi_theme' ), // Block title.
        icon: 'shield', (Dashicons of custom svg)
        category: 'common',
        edit: function( props ) { },
        save: function( props ) { }
});
```

yarn, webpack, node_modules (en dus ook Sass)

```
//package.json   (yarn build, yarn dev)
"scripts": {
    "build": "NODE_ENV=production webpack",
    "dev": "webpack --watch"
}
```

```
//Webpack.congif.js
module.exports = {
        entry: './block.js', // Webpack
        output: {
                path: __dirname,
                filename: 'block.build.js',
        },
        module: {
                loaders: [
                        {
                                test: /.js$/,
                                loader: 'babel-loader',
                                exclude: /node_modules/,
                        },
                ],
        },
};
```

AANPASBARE BLOKKEN

```
const { el } = wp.element.createElement;
const { Editable } = wp.blocks.Editable;
const { children } = wp.blocks.query.children;
```

```
registerBlockType( 'iv/editable-block', {
        title: __( 'Editable', 'ivaldi_theme' ),
        attributes: {
                content: children( 'p' ),
        },
        edit: function( props ) {
                var content = props.attributes.content;

                function onChangeContent( newContent ) {
                        props.setAttributes( { content: newContent } );
                }

                return el(
                        Editable,
                        { // Creates <div class="wp-block-iv-editable-block"
                                tagName: 'p',
                                className: props.className,
```

IVALDI COMPONENTEN ALS PLUGIN

Ivaldi components Docs