

LUCA PAGLIARO x WordPress Meetup Catania

CREARE UN BLOCCO GUTENBERG

From zero to Hero

CHI SONO

Luca Pagliaro

Web Developer @ TheWaveStudio.it

- A 18 anni collaboro con una agenzia di Padova
- Entro in **YITH** dove realizzo temi e plugin per **WordPress** da vendere su **ThemeForest** e **Yithemes**.
- Sviluppo e disegno il frontend di **LovePedia**, un sito di incontri con 1.7+M di utenti.
- Ho fondato una startup per applicare alla ristorazione un algoritmo basato sui big data.
- Entro in **Motorsquare** dove sviluppo la Web Application
- Collaboro con **The Wave Studio** in cui coordino l'area web frontend



Roadmap

- Cos'è Gutenberg e cosa sono i blocchi
- Anatomia di un blocco
- Creare blocchi statici
- Creare blocchi dinamici

mashroom5

PROLOGO

COS'È GUTENBERG

Il nuovo editor di WordPress

Gutenberg è il nuovo **editor** basato su **React** prende il nome da **Johannes Gutenberg**, che **inventò** una macchina per **la stampa a caratteri mobili** più di 500 anni fa.

L'obiettivo è rendere semplice la creazione dei contenuti tramite l'uso di "piccoli blocchi" per raggiungere pagine complesse **senza ricorrere a Shortcodes & HTML**.

Gutenberg in pratica

The screenshot shows the Gutenberg editor interface. At the top, there's a toolbar with a plus icon, the text "Write your story", and three icons: a text "T", a single image "img", and a double image "img". Below the toolbar is a yellow header bar with the text "The name of the game is blocks". The main content area contains a rich text block with the following text:

The goal of this new editor is to make writing simple and enjoyable. This whole process is somewhat similar to LEGO bricks—just stack them together. Move your cursor around and move blocks around with outlines and arrows. Press `ctrl`+`z` to undo and `ctrl`+`y` to redo.

A context menu is open over the text, with the "More" option highlighted. The menu items are:

- Hide Block Settings
- Duplicate
- Insert Before
- Insert After

On the right side of the editor, there are two vertical columns of text snippets:

- ss
- t—
- eract
- ight
- ickly,

Vantaggi

- Semplicità di uso
- Uniformare come il contenuto viene aggiunto: Addio visual composer?
- Estendibile
- Risparmio di tempo per lo sviluppo
- Grande spazio di manovra per editori e creativi



CODICE

COME È FATTO IL CODICE DI UN BLOCCO?

Anatomia di un blocco

```
const {registerBlockType} = wp.blocks;

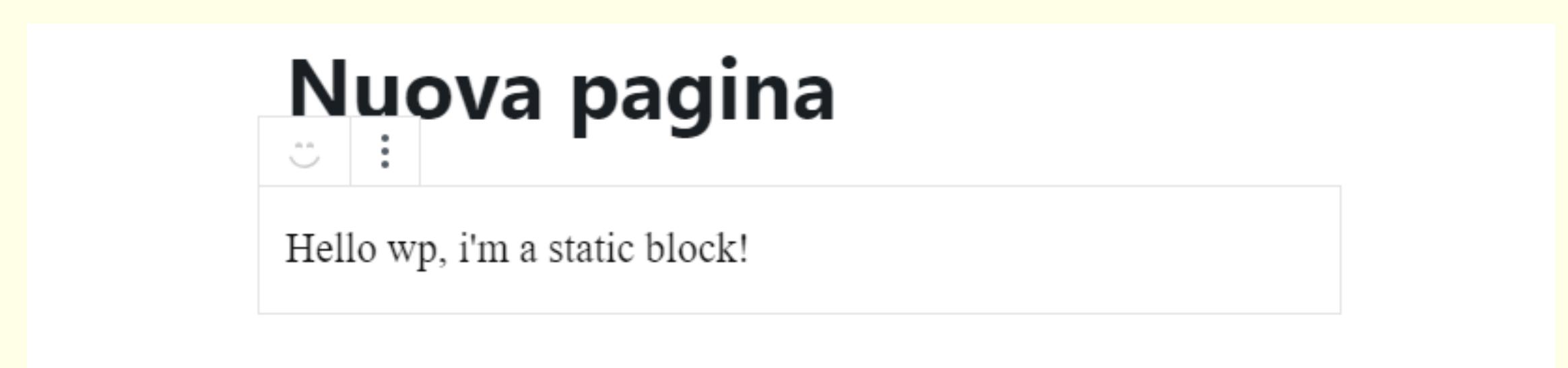
registerBlockType('tw/sample-gutenberg-block', {
    title: 'TW Hello World',
    icon: 'smiley',
    category: 'layout',

    // Variabili del nostro blocco
    attributes: {},

    // HTML & Logica del pannello
    edit: () => <div>Hello wp, i'm a static block!</div>

    // HTML che verrà salvato nella pagina
    save: () => <div>Hello, wp!</div>,
});

});
```



CODICE

**CREIAMO
IL NOSTRO
BLOCCO**

Setup Progetto

The image shows a file explorer on the left and a code editor on the right. The file explorer displays a project structure for a Gutenberg block named 'tw-gutenberg-block'. The structure includes a 'build' folder, a 'node_modules' folder (marked as 'library root'), and a 'src' folder containing '_commons' and 'components' subfolders. The '_commons' folder contains 'tw-media.js' and 'tw-text.js'. The 'components' folder contains 'addable-fields.js', 'dynamic-block.js', 'hello-world.js', 'media-fields.js', 'text-fields.js', and 'index.js'. Also listed are '.gitignore', 'package.json', 'package-lock.json', and 'tw-gutenberg-block.php'. The code editor on the right shows the 'package.json' file with the following content:

```
1  {
2      "name": "tw-gutenberg-block",
3      "version": "1.0.0",
4      "description": "Gutenberg sample block by TheWaveStudio.it",
5      "main": "index.js",
6      "scripts": {
7          "start": "wp-scripts start",
8          "build": "wp-scripts build",
9          "lint:js": "wp-scripts lint-js ."
10     },
11     "author": "",
12     "license": "ISC",
13     "devDependencies": {
14         "@wordpress/scripts": "3.1.0"
15     }
16 }
17 
```

Come Registrare un blocco

```
// register Block
function tw_register_block(){
    wp_register_script(
        'tw-gutenberg-block-js',
        plugins_url('build/index.js', __FILE__),
        ['wp-blocks', 'wp-element', 'wp-editor', 'wp-i18n', 'wp-components']
    );

    register_block_type('tw/sample-gutenberg-block');
}

add_action('init', 'tw_register_block');

// Enqueue JS for Gutenberg block
function tw_gutenberg_enqueue()
{
    wp_enqueue_script('tw-gutenberg-block-js');
}

add_action('enqueue_block_editor_assets', 'tw_gutenberg_enqueue');

// Add admin css for blocks
function tw_gutenberg_stylesheet()
{
    wp_enqueue_style(
        'tw-gutenberg-block-style',
        plugins_url('build/style.css', __FILE__)
    );
}

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

CODICE

BLOCCO CON CAMPI DI TESTO

Risultato finale

Nuova pagina

Hello wp, i'm a static block!

TW Text Sample

Titolo della section

Text

Contenuto testuale

Text

Gli attributi

```
import TwText from "../../_commons/tw-text";
const {registerBlockType} = wp.blocks;

registerBlockType('tw/text-gutenberg-block', {
    title: 'TW Text Field',
    icon: 'smiley',
    category: 'layout',
    attributes: {
        title: {
            type: 'string',
            source: "text",
            selector: ".title",
            default: ''
        },
        text: {
            type: 'string',
            source: "text",
            selector: ".text",
            default: ''
        }
    },
    edit: () => ...,
    save: () => ...
});
```

Edit: cosa vediamo nel pannello

```
...,
edit: (props) => {

  const {attributes} = props;
  const handleFieldUpdate = (update) => {
    const newObject = {...attributes, ...update};
    props.setAttributes(newObject);
  };

  return (<section className='tw-block'>

    <h4>TW Text Sample</h4>

    <TwText label="Titolo della section" field='title'
            current={attributes.title} onChange={handleFieldUpdate}/>

    <TwText label="Contenuto testuale" field='text'
            current={attributes.text} onChange={handleFieldUpdate}/>

    </section>
  },
  save: ({attributes}) => ...
}
```

Save: cosa si vede nella pagina finale

```
● ● ●  
...,  
  save: ({attributes}) => {  
    return (<section>  
      <h2 className="title">{attributes.title}</h2>  
      <div className="text">{attributes.text}</div>  
    </section>);  
  },
```

Titolo

Contenuto testuale

EXTRA

UN COMPONENTE PER UN CODICE ELEGANTE

Rendere semplice il codice

```
class TwText extends React.Component {  
  
    // Get this.props  
    constructor(props) {  
        super(props);  
    }  
  
    // When the text is selected then send the content to parent (onTextChange)  
    handleChange(content) {  
  
        const {field = 'content'} = this.props;  
        const obj = {};  
  
        obj[field] = content || '';  
  
        this.props.onChange(obj);  
    }  
  
    // Return JSX  
    render() {  
        const {current = '', placeholder = 'Text',  
              classes = 'content-plain-text', label = ''} = this.props;  
  
        return (<label>  
            {label && <span className="label">{label}</span>}  
            <PlainText className={classes} style={{height: 58}}  
            placeholder={placeholder} value={current}  
            onChange={content => this.handleChange(content)}>  
            </label>);  
    }  
}
```



mashroom5

CODICE

BLOCCHI DINAMICI

Il caso

Ogni blocco genera un HTML statico, ma se volessimo creare un blocco con del contenuto dinamico?

es: Vogliamo stampare l'ultimo articolo scritto sul blog

Blocco dinamico: PHP

```
function tw_register_block() {
    wp_register_script(
        'tw-gutenberg-block-js',
        plugins_url('build/index.js', __FILE__),
        ['wp-blocks', 'wp-element', 'wp-editor', 'wp-i18n', 'wp-component'],
    );

    register_block_type('tw/dynamic-gutenberg-block', array(
        'render_callback' => 'tw_dynamic_render_callback'
    ));
}

function tw_dynamic_render_callback($attributes, $content)
{
    $recent_posts = wp_get_recent_posts(array(
        'numberposts' => 1,
        'post_status' => 'publish',
    ));

    if (count($recent_posts) === 0) {
        return 'No posts';
    }

    $post = $recent_posts[0];
    $post_id = $post['ID'];

    return sprintf(
        '<a class="wp-block-tw-latest-post" href="%1$s">%2$s</a>',
        esc_url(get_permalink($post_id)),
        esc_html(get_the_title($post_id))
    );
}
```

Blocco dinamico: JS

```
const {registerBlockType} = wp.blocks;
const {ServerSideRender} = wp.components;

registerBlockType( 'tw/dynamic-gutenberg-block', {
    title: 'TW Dynamic Block',
    icon: 'smiley',
    category: 'layout',

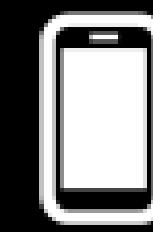
    edit: (props) => {
        return (
            <ServerSideRender block="tw/dynamic-gutenberg-block"
                attributes={props.attributes}/>
        );
    },
    save: () => null
});
```

Nuova pagina

[Ciao mondo! Test!](#)

Grazie

Passiamo alle domande :)



Scan me