

Creating a **Gutenberg Custom Block** from scratch requires knowledge of JavaScript, React, and WordPress development. Below is a structured roadmap to help you through the process.

---

## Roadmap to Creating a Custom Gutenberg Block (from Scratch)

### 1 Prerequisites

- Basic knowledge of **JavaScript (ES6+), React, and JSX**
  - Familiarity with **Node.js, npm/yarn, and WordPress development**
  - Understanding of **WordPress filters, actions, and PHP basics**
- 

### 2 Setting Up the Development Environment

#### ◆ Install Node.js & npm

Make sure you have Node.js installed. You can check with:

```
node -v
npm -v
```

If not installed, download it from [Node.js official site](https://nodejs.org/).

#### ◆ Create a WordPress Plugin for Your Block

Inside the WordPress `wp-content/plugins/` directory, create a new folder:

```
mkdir gutenberg-custom-block
cd gutenberg-custom-block
```

Create a main plugin file:

```
touch gutenberg-custom-block.php
```

Edit `gutenberg-custom-block.php`:

```
<?php
/**
 * Plugin Name: Gutenberg Custom Block
 * Description: A custom Gutenberg block from scratch.
 * Version: 1.0.0
 * Author: Hawana
```

```
* License: GPL2+
*/

if (!defined('ABSPATH')) exit;

// Load block assets
function load_gutenberg_custom_block() {
    wp_enqueue_script(
        'gutenberg-custom-block-js',
        plugins_url('build/index.js', __FILE__),
        array('wp-blocks', 'wp-editor', 'wp-element', 'wp-components'),
        filetime(plugin_dir_path(__FILE__) . 'build/index.js')
    );

    wp_enqueue_style(
        'gutenberg-custom-block-css',
        plugins_url('build/style.css', __FILE__),
        array(),
        filetime(plugin_dir_path(__FILE__) . 'build/style.css')
    );
}
add_action('init', 'load_gutenberg_custom_block');
```

---

### 3 Initialize the Project

Inside `gutenberg-custom-block`, run:

```
npm init -y
```

This creates `package.json`, which stores dependencies.

---

### 4 Install Required Dependencies

Run the following command to install WordPress Gutenberg dependencies:

```
npm install @wordpress/scripts @wordpress/block-editor @wordpress/blocks
@wordpress/components @wordpress/i18n --save-dev
```

---

### 5 Create the Block Structure

Inside the plugin folder, create:

```
mkdir src
touch src/index.js
```

---

## 6 Register the Custom Block

Edit src/index.js:

```
import { registerBlockType } from '@wordpress/blocks';
import { useBlockProps } from '@wordpress/block-editor';
import { __ } from '@wordpress/i18n';

registerBlockType('custom/block-example', {
  title: __('Custom Block', 'custom-block'),
  icon: 'smiley',
  category: 'common',
  attributes: {
    content: { type: 'string', default: 'Hello, Gutenberg!' }
  },
  edit: ({ attributes, setAttributes }) => {
    const blockProps = useBlockProps();
    return (
      <div {...blockProps}>
        <input
          type="text"
          value={attributes.content}
          onChange={(e) => setAttributes({ content: e.target.value
        ))}
      </div>
    );
  },
  save: ({ attributes }) => {
    return <p>{attributes.content}</p>;
  }
});
```

---

## 7 Compile the JavaScript & CSS

Modify package.json to include a build script:

```
"scripts": {
  "build": "wp-scripts build",
  "start": "wp-scripts start"
}
```

Then, run:

```
npm run build
```

This generates the build/ directory with the compiled block.

---

## 8 Activate the Plugin in WordPress

Go to **WordPress Dashboard** → **Plugins**, activate "Gutenberg Custom Block," and check if the block appears in the Gutenberg editor.

---

## 9 Enhance the Block (Optional)

- Add **Inspector Controls** (Sidebar settings)
  - Include **block styles & custom classes**
  - Use **Server-side Rendering (SSR)**
  - Register multiple blocks
- 

## 10 Final Steps

- Test your block in different environments
  - Optimize for **performance & security**
  - Submit to the **WordPress Plugin Repository** if needed
- 

💡 **Congratulations!** 🎉 You've successfully created a custom Gutenberg block from scratch. 🚀