# 216: Official WP Block Dev

Block.Json is new

@wordpress/scripts has more Utility

Interactivity API is new

Summary: introduction to the new Wordpress tools

## 217: How to setup Multi. Blocks

### New File Structure : SRC

- SRC is needed to have @wordpress/scripts detect each Block.Json File inside

```
∨ src
  ∨ footer
      block.json
      index.js
      render.php
```

### Inside block.Json

```json
{
  "$schema": "https://schemas.wp.org/trunk/block.json",
  "apiVersion": 3,
  "name": "ourblocktheme/footer",
  "title": "Fictional University Footer",
  "editorScript": "file:./index.js",
  "render": "file:./render.php"
}
```

—This is the Blocks config File

### Inside function.Php

- Setup Function to create new blocks

```php
// Register our new blocks
function our_new_blocks() {
    register_block_type_from_metadata(__DIR__ . '/build/footer');
}
```

### Inside package.Json

— Add "blocks" to the scripts

```
"blocks": "wp-scripts start --experimental-modules",
```

### Inside render.php

— Insert HTML that needs to be displayed in Front end

- this file is used to
display block in admin

```js
import { registerBlockType } from "@wordpress/blocks"
import metadata from "./block.json"

function Edit() {
  return <div>Hello, I am the footer block in the editor.</div>
}

registerBlockType(metadata.name, {
  edit: Edit
})
```

Files Changed

SRC/Footer/
Block.JSON
Functions.PhP
Render.PhP
Index.PhP

themename/
Package.Json

29

## 218: Practice makes Perfect

Start by running 2 commands
- npm run Start
- npm run build

## Inside index.JS

Removed edit Function
edit.JS File
into its own

## Inside edit.JS

— Add useBlockProps so element Can be selected
                                    in Admin editor

```js
import { useBlockProps } from "@wordpress/block-editor"

export default function Edit() {
  const blockProps = useBlockProps()

  return (
    <div {...blockProps}>
      <div className="our-placeholder-block">University Footer Place
    </div>
  )
}
```

NOTE: Remember for each new block
to create its own Folder in SRC and
register it within functions.php

### Files Changed

SRC/Footer/
index.JS
edit.JS

29.

# 219: Our Banner Block

- Making Banner Block Interactive

**NOTE:** An old Banner Block already exists so we are Just moving components

## Inside edit.JS

- We add the imports from the old Banner block

```
import apiFetch from "@wordpress/api-fetch"
import { Button, PanelBody, PanelRow } from "@wordpress/components"
import {
  useBlockProps,
  InnerBlocks,
  InspectorControls,
  MediaUpload,
  MediaUploadCheck
} from "@wordpress/block-editor"
import { registerBlockType } from "@wordpress/blocks"
import { useEffect } from "@wordpress/element"
```
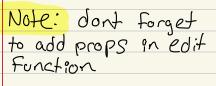
## Inside block.Json

- We add any "Supports" and "attributes" to the existing JSON

**NOTE:** We will address background image later

```
{
  "$schema": "https://schemas.wp.org/trunk/block.json",
  "apiVersion": 3,
  "name": "ourblocktheme/banner",
  "title": "Fictional University Banner",
  "supports": {
    "align": ["full"]
  },
  "attributes": {
    "align": { "type": "string", "default": "full" },
    "imgID": { "type": "number" },
    "imgURL": { "type": "string" }
  },
  "editorScript": "file:./index.js",
  "render": "file:./render.php"
}
```

Add JSX & functions from old Banner

Note: dont forget to add props in edit function

```
useEffect(
  function () {
    if (props.attributes.imgID) {
      async function go() {
)}
```

Extra Code

```
function onFileSelect(x) {
  props.setAttributes({ imgID: x.id })
}
```

Copy and paste these two Blocks of Code

```
<InspectorControls>
  <PanelBody title="Background" initialOpen={true}>
    <PanelRow>
      <MediaUploadCheck>
```

Extra Code

```
    <InnerBlocks allowedBlocks={["
  </div>
</div>
```

## Inside index.JS

to save content for the block you need to adjust the save portion of the Block

Note: Add

import Save from "./save"

and add a new Save.JS File to keep Consistent

```
import { registerBlockType } from "@wordpress/blocks"
import metadata from "./block.json"
import Edit from "./edit"

registerBlockType(metadata.name, {
  edit: Edit,
  save: Save
})
```

• Make a new Save function inside Save.JS and add the return Content

Inside Save.JS

```
import { InnerBlocks } from "@wodpress/block-editor"
```

Extra Code

```
return <InnerBlocks.Content />
```

# Solving Fallback Background Image

```
// Register our new blocks
function our_new_blocks() {
    wp_localize_script(a, b, c);
```

within new block registration
add localize script code

A: `'wp-editor'` → Script to attach to

B: `'ourThemeData'` → Variable name

C:
```
array('themePath' => get_stylesheet_directory_uri())
```

## Inside Edit.JS

add a new useEffect within
the Edit function

```
useEffect(function() {
    if (!props.attributes.imgURL) {
        props.setAttributes({imgURL: ourThemeData.themePath + ""})
    }
}, [])
```

• Add url for image

→ `"/images/library-hero.jpg"`

## Files Changed

# 220: Slide & Slideshow Blocks