

## **WordPress: Building Progressive Themes with WP-Rig**

WP Rig is a modern WordPress starter theme and build process bundled into one convenient package. It helps you create beautiful, lightning-fast WordPress themes with exciting user experiences that meet the latest web standards.

You write modern PHP, CSS, and JavaScript, WP Rig lints, optimizes, and minifies your code to make the theme—and the site it runs on—as performant as possible.

### **Chapter 1:**

Chapter 1 introduces about WP-Rig that how WP-Rig helps in building WordPress themes.

Download all the latest files from git repository of wp-rig and put them in local host folder and open in vs.

### **Chapter 2:**

Chapter 2 is about setting up development environment. Wp-rig works with node.js and composer. Composer install php which is a must for wp-rig. So first install composer and node.js

### **Chapter 3:**

Chapter 3 is about installing and configuring wp-rig. After installing node.js and composer run this command

**Npm run rig-init**

**Then configure BrowserSync in config.json file.**

```
"dev": {  
  "browserSync": {  
    "live": true,  
    "proxyURL": "wprig.local",  
    "bypassPort": "8181",  
    "https": true  
  }  
}
```

Now run the wp-rig installer by this command:

**Npm run dev**

Wp-rig uses code-editor extensions like stylelint, prettier, es-lint, phpcs. You just need to install them. Wp-rig configure them by itself and then you can automate your code with the help of these extensions. You can configure your visual studio workspace settings in settings.json like that:

```
{
  "window.zoomLevel": 0,
  "terminal.integrated.shell.windows": "C:\\\\Program Files\\Git\\bin\\bash.exe",
  "css.validate": false,
  "less.validate": false,
  "scss.validate": false, "prettier-stylelintIntegration": true, "[css]": {
    "editor.defaultFormatter": "esbenp.prettier-vscode",
    "editor.renderWhitespace": "all",
    "editor.formatOnSave": true,
  },
  "workbench.settings.editor": "json",
  "editor.minimap.enabled": false,
  "files.autoSave": "afterDelay"
}
```

To install ssl in wp-rig run this command

**Npm run generateCert**

It will generate a ssl certificate and save it in browsersync folder

## **Chapter 4:**

### **CSS in WP Rig**

WP Rig is a modern WordPress starter theme and build process bundled into one convenient package. It helps you create beautiful, lightning-fast WordPress themes with exciting user experiences that meet the latest web standards.

You write modern PHP, CSS, and JavaScript, WP Rig lints, optimizes, and minifies your code to make the theme—and the site it runs on—as performant as possible.

Custom Properties:

You can make custom properties in src folder \_custom properties file, \_typography.css, \_media.css. The benefit would be that if you want to change a rule, you will need to do it only on one place not in all files.

Changing Fonts in WP Rig:

To change font fo to php->inc->component.php and then to this function

```
protected function get_google_fonts() : array {  
    if ( is_array( $this->google_fonts ) ) {  
        return $this->google_fonts;  
    }  
}
```

```
$google_fonts = [  
    'Roboto Condensed' => [ '400', '400i', '700', '700i' ],  
    'Crimson Text'    => [ '400', '400i', '600', '600i' ],  
];
```

Change also the arrays according to google fonts and then change the fonts in \_custom properties.css.

Wp-rig is modular so if you want to change some styles in just one place e.g in a page you can do that

In component.php file add that code in In \$css\_files = [ ] **function**

```
'wp-rig-page' => [  
    'file'      => 'page.min.css',  
    'preload_callback' => function() {  
        return is_page();  
    },  
];
```

In page.php file add wp-rig-page'

Like this

```
wp_rig()->print_styles( 'wp-rig-page','wp-rig-content' );
```

Make a separate file of page.css and write your styles here e.g

```
.site-main {  
    border-top: 10px solid #000;  
}
```

Add this file to assets/css/src/page.css

## **Chapter 5**

### **PHP and WordPress Templates:**

Wp-Rig is modular base. There are different templates in optional folders and if you want you can drag them in root folder and work on them.

For ,logo and header customization there are custom\_log and custom\_header etc folders in inc folder.

All text etc are in editor folder under inc folder.

## **Chapter 6:**

### **JavaScript in WP-Rig:**

Js in wp-rig is also modular based. There are 3 files in js folder customiser.js, lazyload.js and navigation.js. These files are modular based not centralized and they depend on component.php files in inc folders. If you want to stop minifying js go to the config.default.json file and in debug function change scripts from false to true and it will stop minifying. Copy this debug function in congig.json file and turn scripts true and it will stop minifying.and in termonal write

### **Npm run gulp scripts**

And it will only run scripts

Now it will not minify and you can debug it.

After doing everything run the command

**Npm run bundle** and your theme will download.

