

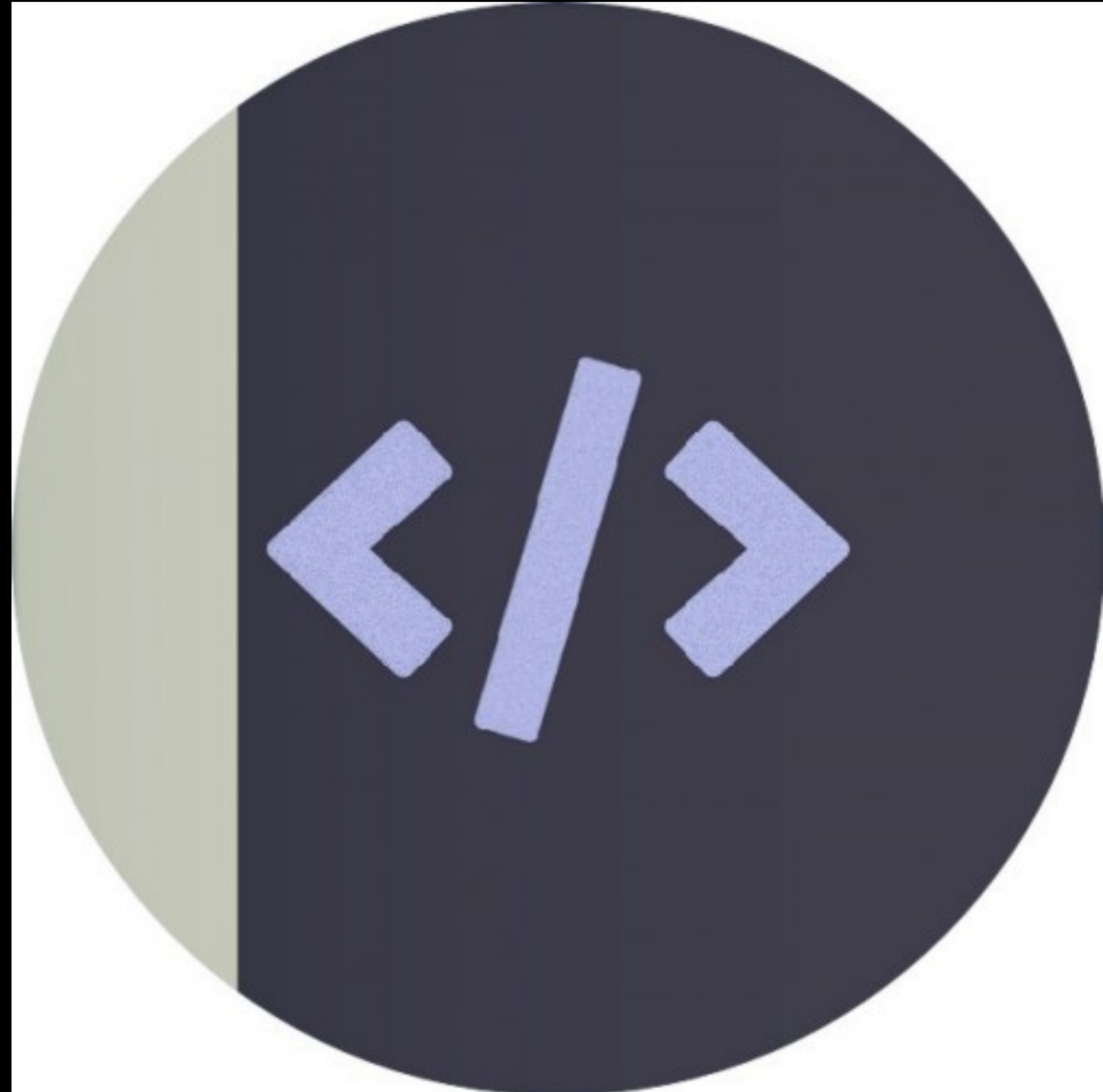
テーマレビューの現場から見た、抑えておくべきテーマ制作のセオリーと基礎知識

WordCamp Tokyo 2018 / 金井俊浩

自己紹介

金井俊浩 (mirucon)

- フリーランスの Web エンジニア
- 最近は Vue.js などのフロントエンドがメイン
- WordPress Core Contributor
- WordPress テーマ Coldbox 開発者
- WordPress テーマレビューチームモデレータ
- *Twitter: @mirucons / Facebook & GitHub etc.: mirucon*
- *<https://www.mirucon.com/>*



テーマの基礎

テーマとは

テーマとは

ウェブサイト全体の見た目からレイアウト、構成、機能まで様々な場所に影響を及ぼす、WordPress サイトの「キモ」

ディレクトリ構成

- 例えばこんな感じ：

my-theme/

├── inc/

│ ├── customizer.php

│ └── related-posts.php

├── footer.php

├── functions.php

├── header.php

├── index.php

├── readme.txt

├── screenshot.png

├── single.php

└── style.css

テーマでは何をすべき？

テーマでは何をすべき？

- ・ テーマは結局プラグインと同じただの PHP ファイルなので、やろうと思えば何だってできる
- ・ ただしテーマはプラグインと違って1つしか有効化できない
- ・ そのため WordPress.org のテーマディレクトリの要求事項では「テーマは基本的に見た目を司ることのみすべき」つまり、
=> 見た目に直接関係ない機能をテーマに入れるべきではない

テーマでは何をするべき？

- ・ ただしこれは WordPress.org のテーマディレクトリの話であり、他のテーマ配布サイト等では違ったりする
- ・ 結局は個々の機能をプラグイン化するのと、テーマで一元化してすべてを管理するのは便利さとのトレードオフ
- ・ また受託開発などでは機能自体に汎用性がない場合・なんらかの事情によってプラグインを使いにくい場合などもある

=> 自分の制作している目的・公開範囲などを考えて、適切なところを考えよう

ライセンスについて

WordPress はオープンソース

- WordPress は本体がオープンソース
- GPL ライセンスを使用している
- WordPress は思想として「パブリッシングの民主化 (Democratize publishing)」を掲げている
- オープンソースなので WordPress の開発・ディスカッション・翻訳などには誰でも貢献できる

GPL ライセンスとはどんなライセンスか

- **General Public License** の頭文字をとって "GPL" と呼ばれるオープンソースライセンスの一つ
- いかなる**制約なし**に無保証で**4つの自由**を認めるのが基本思想

4つの自由とは？

- ・ どんな目的にも使用する自由
- ・ ソースコードを研究し、改変する自由
- ・ 他の人に再配布する自由
- ・ 改変したものを共有する自由

最大の特徴コピーレフト

- ・ コピーレフトとは、制作物の改変されたものや派生プロダクト (derivative work) にも、もとの制作物と同一の自由を認めるべきという考え方
- ・ WordPress の場合：
 - ・ **もとの制作物** = WordPress
 - ・ **派生プロダクト** = テーマ・プラグインなど

=> つまり、WordPress が GPL である限り、配布する作ったテーマ・プラグインも GPL にする義務が発生する

配布しない場合について

- GPL は**配布する場合にのみ**適応されるライセンスであり、配布しない場合には GPL でライセンスする**必要はない**
- それでも WordPress を使っているということは GPL 製品を使っているということなので、是非皆さんに知っておいてもらいたい

テーマの始め方

スターターテーマ

スターターテーマ

- 手間のかかる最初の設定や、どんなテーマでも必要になるようなコードを設定済みにしてくれるもの
- 例えば：
 - Sass のコンパイル設定
 - index.php や single.php のループ (投稿表示部分)

_s (underscores)

- Automattic 社 (JetPack プラグインの開発などを行っている会社) の開発するスターターテーマ
- かなり中身はシンプルな PHP テンプレート + CSS
- シンプルに抑えたいテーマに特におすすめ



OUR UNDERSCORES
THEME

Theme Name

Advanced Options

he called _s, or
ke. I'm a theme
o don't use me as a
ad try turning me
awesome, WordPress
it's what I'm here for.

A just right amount of lean, well-commented, modern, HTML5 templates

A helpful 404 template.

An optional sample custom header implementation in inc/custom-he

Custom template tags in inc/template-tags.php that keep your ter
and prevent code duplication

コーディング規約

コーディング規約

- ・ コーディング規約とは、**コードの書き方**についての決まりごと
- ・ WordPress には **WordPress Coding Standards** という、WordPress 専用の規約がある
- ・ これはコードのフォーマットだけでなく、後で触れる**セキュリティ**に関することも含まれる

WordPress Coding Standards

- 例えば:

```
if (is_single() ){
```

のようなものを以下のように書くべきと規定できる:

```
if ( is_single() ) {
```

役立つとき

- ・ 複数人開発する時に、コードの書き方の癖をなくせる
- ・ 一人開発でも、アップデートの期間が空いてしまったときでもコードの質を保てる

セキュリティについて

なぜセキュリティ対策が必要なのか

なぜセキュリティ対策が必要なのか

- ・ プログラムには「特別な意味を持つ文字列」があったりする
- ・ また WordPress では HTML を扱うことが多く、**HTML を使用できる = JavaScript を使用できる** ということであり、JavaScript には色々なことができてしまうため、悪用の恐れがある

なぜセキュリティ対策が必要なのか

たとえば、HTML のこんな文字列：

< > ' "

これらを許可してしまうと、予期しないところで HTML が使われてしまう

クロスサイトスクリプティング (XSS)

- ユーザーが予期しない動作をするコード (特に JavaScript) を読み込むこと
- JavaScript で実際にできてしまうこと：
 - 勝手に他のサイト (特にウイルス配布サイトなど) に転送
 - 投稿内容を書き換え

大きく2つのセキュリティ対策

- ・ サニタイズ

=> データを**保存するとき**にデータを無害化 = 信用できない文字列を取り除く

- ・ エスケープ

=> データを**出力するとき**に特殊文字列を変換し特殊文字列としての効果を打ち消す

サニタイズ

サニタイズの例

- wp_kses() 関数
 - 許可する HTML のクラス・属性を指定し、許可されないものを削除する

例えばこんな HTML:

こんにちは、`
`

``金井``です。

普通に表示すればこうなる：

普通に表示すればこうなる：

こんにちは、
金井です。

wpkses() 関数を使うと

wpkses() 関数 を使って タグと class 属性のみを許可

wpkses() 関数を使うと：

こんにちは、金井です。

```
▼<div class="entry-  
inner">  
  "こんにちは、"  
  <span class="my-  
class">金井</span>  
  "です。"  
  ::after  
</div>
```

wpkses() 関数の使い方

```
$allowed_html = [  
    'span' => [  
        'class' => [],  
    ],  
];
```

```
$data = wp_kses( $data, $allowed_html );
```

他の WordPress サニタイズ関数

- `wp_kses_post()`
- `wp_kses_data()`
- `sanitize_email()`
- `sanitize_file_name()`
- `sanitize_html_class()`
- `sanitize_text_field()`

テーマカスタマイザーとサニタイズ

```
$wp_customize->add_setting(  
    'credit_text', [  
        'default'          => '©2018 このサイトの運営者',  
        'sanitize_callback' => 'wp_kses_post',  
    ]  
);  
$wp_customize->add_control(  
    new WP_Customize_Control(  
        $wp_customize, 'credit_text', [  
            'label'          => __( 'クレジットを編集', 'text-domain' ),  
            'section'        => 'footer',  
            'type'           => 'textarea',  
        ]  
    )  
);
```

エスケープ

エスケープの例

- `esc_html()` 関数
=> すべての HTML をプレーンテキスト化する

例えばこんな HTML:

こんにちは、`
`

``金井``です。

するとこうなる：

こんにちは、
金井です。

エスケープの動作

- HTML の特殊な文字列を、**エスケープ文字**と呼ばれる特殊な意味が無効化される文字列に置き換える

例：

< (小なり) => <

> (大なり) => >

他の WordPress エスケープ関数

- `esc_attr()`
- `esc_url()`
- `esc_textarea()`
- `esc_js()`

セキュリティ対策のコツ

セキュリティ対策のコツ

- すべてを疑うこと
- サニタイズした上でエスケープが必要なこともある
- WordPress Coding Standards を使う



The screenshot shows a code editor with a dark theme. On the left, a file explorer lists files: 'np', 'np', 'e.txt', 'shot.jpg', 'php', 'form.php', and 'php'. The main editor area shows line numbers 158 to 167. Line 161 contains the code `echo get_theme_mod('my_theme_mod');`. A yellow lightbulb icon is positioned to the left of this line, and a red squiggly line underlines the code. A tooltip box is open, displaying the message: "phpcs: All output should be run through an escaping function (see the Security sections in the WordPress Developer Handbooks), found 'get_theme_mod'". Below line 161, line 165 shows the closing PHP tag `?>`.

メンテナンスをしやすくするには

「WordPress の書き方」を覚える

- WordPress には素の PHP とは違う「WordPress 的な書き方」がある
- 例えばこのような関数など：
 - `get_template_part()`
 - `wp_enqueue_script()`, `wp_enqueue_style()`

ドキュメントを書く

- ・ クラス・関数には極力ドキュメントを書く
- ・ WordPress Coding Standards を使うとコメントが抜けていると教えてくれる
- ・ ドキュメントの書き方として **phpdoc** コメント という物がある

phpdoc コメントの書き方

```
/**
 * 指定された数の関連記事を返す.
 *
 * @param int $max_posts 表示する最大記事数.
 * @since 1.0.0
 */
function theme_get_related_posts( $max_posts ) {
    ...
}
```

テーマをチェックできるツール・プラグインを使う

- テーマユニットテスト
- Theme Check | WordPress.org
- WPTRT/theme-sniffer: Theme Sniffer plugin using sniffs.
- Debug Bar | WordPress.org

**WordPress.org テーマディレク
トリ掲載にあたって**

要求事項

- WordPress.org テーマディレクトリには **要求事項 (Requirements)** という、掲載のために沿わないといけないルールがある
- 要求事項の詳細は、Required – Theme Review Team – WordPress を参照
- 日本語訳もあります : <https://github.com/mirucon/required-ja>

テーマレビューチームは誰でも参加できる

- ・ やる気がある人なら誰でも歓迎
- ・ 人のテーマを見ることで自分のテーマ制作にとっても役に立つ
- ・ 最終的に承認できる権限は限られた人にしか与えられてないので、安心して良い

まとめ

- ・ テーマ制作には、**スターターテーマ**、**WordPress Coding Standards** など便利なツールが多い
- ・ GPL ライセンスは自由を認める、利用者にも開発者にも優しいライセンス
- ・ セキュリティ問題には主に「サニタイズ」と「エスケープ」で対策する

ありがとうございました

Twitter: @mirucons

<https://www.mirucon.com/>