

Plugins no WordPress

Fazendo o Negócio Direito

Vinicius Massuchetto

WordCamp São Paulo 2012

Agosto de 2012

Download

Esta apresentação está disponível em:
<http://vinicius.soylocoporti.org.br/?p=2191>

Sobre a Palestra

- Motivação Desta Apresentação

Não encontrar o que procura



Motivos para se criar um plugin

Motivos para se criar um plugin

- Funcionalidade inexistente

Motivos para se criar um plugin

- Funcionalidade inexistente
- Diferente implementação de uma funcionalidade existente

Motivos para se criar um plugin

- Funcionalidade inexistente
- Diferente implementação de uma funcionalidade existente
- Códigos de tema portáteis

Motivos para se criar um plugin

- Funcionalidade inexistente
- Diferente implementação de uma funcionalidade existente
- Códigos de tema portáteis
- Implementações modulares para clientes

Motivos para se criar um plugin

- Funcionalidade inexistente
- Diferente implementação de uma funcionalidade existente
- Códigos de tema portáteis
- Implementações modulares para clientes
- Forks de plugins existentes

Dificuldades em se escrever um plugin

Dificuldades em se escrever um plugin

- PHP X WordPress

Dificuldades em se escrever um plugin

- PHP X WordPress
- Cultura de leitura de documentação e inspeção de código

Dificuldades em se escrever um plugin

- PHP X WordPress
- Cultura de leitura de documentação e inspeção de código
- Barreira de idioma

Dificuldades em se escrever um plugin

- PHP X WordPress
- Cultura de leitura de documentação e inspeção de código
- Barreira de idioma
- Entrega de produtos deficitários

Facilidades

Facilidades

- Código legível

Facilidades

- Código legível
- Padronização de estruturas

Facilidades

- Código legível
- Padronização de estruturas
- Melhor aprendizado e manutenção do código

Facilidades

- Código legível
- Padronização de estruturas
- Melhor aprendizado e manutenção do código
- Extensibilidade

Facilidades

- Código legível
- Padronização de estruturas
- Melhor aprendizado e manutenção do código
- Extensibilidade
- Distributividade na comunidade do software livre

Pensando Em Um Plugin

Pensando Em Um Plugin

- Definição de escopo e pesquisa de funcionalidades

Pensando Em Um Plugin

- Definição de escopo e pesquisa de funcionalidades
- Se parecer redundante, perguntar e descrever a ideia em listas e fóruns

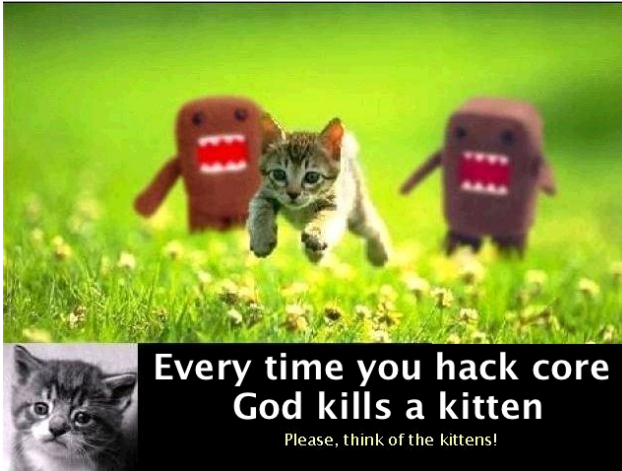
Pensando Em Um Plugin

- Definição de escopo e pesquisa de funcionalidades
- Se parecer redundante, perguntar e descrever a ideia em listas e fóruns
- Escolha de nome único e relevante

Pensando Em Um Plugin

- Definição de escopo e pesquisa de funcionalidades
- Se parecer redundante, perguntar e descrever a ideia em listas e fóruns
- Escolha de nome único e relevante
- Avaliação do uso de outras tecnologias e frameworks

Primeiro Padrão



Arquivos

Arquivos

- Nomear o diretório

Arquivos

- Nomear o diretório
- Ser coerente com nomes de arquivos

Arquivos

- Nomear o diretório
- Ser coerente com nomes de arquivos
- Incluir somente arquivos necessários e sob demanda no código

Arquivos

- Nomear o diretório
- Ser coerente com nomes de arquivos
- Incluir somente arquivos necessários e sob demanda no código
- Permitir que o diretório do plugin mude usando funções como:

Arquivos

- Nomear o diretório
- Ser coerente com nomes de arquivos
- Incluir somente arquivos necessários e sob demanda no código
- Permitir que o diretório do plugin mude usando funções como:
 - `plugins_url`

Arquivos

- Nomear o diretório
- Ser coerente com nomes de arquivos
- Incluir somente arquivos necessários e sob demanda no código
- Permitir que o diretório do plugin mude usando funções como:
 - `plugins_url`
 - `plugin_dir_url`

Arquivos

- Nomear o diretório
- Ser coerente com nomes de arquivos
- Incluir somente arquivos necessários e sob demanda no código
- Permitir que o diretório do plugin mude usando funções como:
 - `plugins_url`
 - `plugin_dir_url`
 - `plugin_dir_path`

Padrões de Código

Padrões de Código

- Ater-se aos padrões recomendados para código e documentação

Padrões de Código

- Ater-se aos padrões recomendados para código e documentação
- Nomear as estruturas e funções com um identificador único

Padrões de Código

- Ater-se aos padrões recomendados para código e documentação
- Nomear as estruturas e funções com um identificador único
- Clareza é melhor do que praticidade

Tag PHP

Tag PHP

Errado

```
<? /* code */ ?>
```

```
<?=$var?>
```

```
<% /* code */ %>
```

Tag PHP

Certo

```
<?php /* code */ ?>
```

```
<?php echo $var; ?>
```

Chaves

Errado

```
if(condition)
{
    action1();
    action2();
}
else
{
    action3();
    action4();
}
```

Chaves

Certo

```
if ( condition ) {  
    action1();  
    action2();  
} else {  
    action3();  
    action4();  
}
```

Funções

Errado

```
function some_function($param1='foo', $param2='bar') {  
    /* code */  
}  
  
class SomeClass {  
    /* code */  
}
```

Funções

Certo

```
function some_function( $param1 = 'foo', $param2 = 'bar' ) {  
    /* code */  
}  
  
class Some_Class {  
    /* code */  
}
```

Vetores

```
$x = $foo['bar'];    // certo
```

```
$x = $foo[ 'bar' ]; // errado
```

```
$x = $foo[ $bar ];  // certo
```

```
$x = $foo[$bar];    // errado
```


Padrões de SQL

Padrões de SQL

- Evitar escrever consultas

Padrões de SQL

- Evitar escrever consultas
- Validar os tipos de dados antes de utilizá-los

Padrões de SQL

- Evitar escrever consultas
- Validar os tipos de dados antes de utilizá-los
- Selecionar só o que for necessário

Padrões de SQL

- Evitar escrever consultas
- Validar os tipos de dados antes de utilizá-los
- Selecionar só o que for necessário
- Utilizar a wpdb

Padrões de SQL

- Evitar escrever consultas
- Validar os tipos de dados antes de utilizá-los
- Selecionar só o que for necessário
- Utilizar a `wpdb`
- Se precisar criar tabelas no banco, use `$wpdb->prefix`

Exemplo de Consulta

Errado

```
$r = $wpdb->get_results("select * from wp_posts where post_date  
> '2012-08-25 00:00:00'");
```

Exemplo de Consulta

Certo

```
$r = $wpdb->get_results(  
    "  
    SELECT ID, post_title, post_content  
    FROM {$wpdb->posts}  
    WHERE 1=1  
        AND post_date > '2012-08-25 00:00:00'  
    "  
);
```


Debug



Constantes de debug no wp-config.php

Constantes de debug no wp-config.php

- WP_DEBUG

Constantes de debug no wp-config.php

- WP_DEBUG
- WP_DEBUG_LOG

Constantes de debug no wp-config.php

- WP_DEBUG
- WP_DEBUG_LOG
- WP_DEBUG_DISPLAY

Cabeçalho

Cabeçalho

Todo plugin começa pelo começo..

Cabeçalho

Todo plugin começa pelo começo..

```
/*  
 * Plugin Name: Nome do Plugin  
 * Version: 0.01  
 * Description: Texto longo  
 * Author: Nome do Autor  
 * Contributors: Logins no WordPress.org  
 * Plugin URI: http://uma-url.com  
 */
```

Estrutura Procedural

Estrutura Procedural

```
function callback1 () { /* code */ }  
add_action( 'hook1', 'callback1' );  
  
function callback2 () { /* code */ }  
add_action( 'hook2', 'callback2' );  
  
function callback3 ($param) { /* code */ return $param; }  
add_filter( 'filter', 'callback3' );
```

Estrutura Orientada a Objetos

Estrutura Orientada a Objetos

```
class Some_Plugin {  
  
    function Some_Plugin() {  
        add_action( 'hook1', array($this, 'hook1' ));  
        add_action( 'hook1', array($this, 'hook2' ));  
    }  
  
    function hook1 { /* code */ }  
  
    function hook2 { /* code */ }  
  
}
```

Vantagens da Orientação a Objetos em Plugins

Vantagens da Orientação a Objetos em Plugins

- Organiza o código, suas funções e métodos

Vantagens da Orientação a Objetos em Plugins

- Organiza o código, suas funções e métodos
- Melhora a extensibilidade

Vantagens da Orientação a Objetos em Plugins

- Organiza o código, suas funções e métodos
- Melhora a extensibilidade
- Reduz o impacto no escopo global do PHP

Vantagens da Orientação a Objetos em Plugins

- Organiza o código, suas funções e métodos
- Melhora a extensibilidade
- Reduz o impacto no escopo global do PHP
- Ajuda a não introduzir variáveis globais

Ativação

Ativação

- `register_activation_hook`

Ativação

- `register_activation_hook`
- Criação de opções padrão

Ativação

- `register_activation_hook`
- Criação de opções padrão
- Criação de tabelas

Ativação

- `register_activation_hook`
- Criação de opções padrão
- Criação de tabelas
- Exibição de avisos para o usuário configurar o plugin

Desativação

Desativação

- `register_deactivation_hook`

Desativação

- `register_deactivation_hook`
- Em geral não deve causar nenhuma perda de dados

Desativação

- `register_deactivation_hook`
- Em geral não deve causar nenhuma perda de dados
- Desabilitar plugins dependentes

Desinstalação

Desinstalação

- `register_uninstall_hook`

Desinstalação

- `register_uninstall_hook`
- Não deve deixar nenhum dado residual no WordPress

Desinstalação

- `register_uninstall_hook`
- Não deve deixar nenhum dado residual no WordPress
- Remove opções do usuário

Desinstalação

- `register_uninstall_hook`
- Não deve deixar nenhum dado residual no WordPress
- Remove opções do usuário
- Remove tabelas

Desinstalação

- `register_uninstall_hook`
- Não deve deixar nenhum dado residual no WordPress
- Remove opções do usuário
- Remove tabelas
- Avisa o usuário sobre a remoção de dados

Inicialização

Inicialização

- `*_init()`

Inicialização

- `*_init()`
- Geralmente através de um procedimento inicializador

Inicialização

```
function someplugin_init () {  
    add_action( 'hook1', 'callback1' );  
    add_action( 'hook2', 'callback2' );  
    add_filter( 'filter', 'callback3' );  
}  
someplugin_init();
```

Inicialização

```
function someplugin_init () {  
    add_action( 'hook1', 'callback1' );  
    add_action( 'hook2', 'callback2' );  
    add_filter( 'filter', 'callback3' );  
}  
someplugin_init();
```

```
add_action( 'plugins_loaded', 'someplugin_init' );
```

Inicialização

```
class Some_Plugin {  
  
    function SomePlugin() {  
        add_action( 'hook1', array($this, 'hook1' ));  
        add_action( 'hook1', array($this, 'hook2' ));  
    }  
  
    function hook1 { /* code */ }  
  
    function hook2 { /* code */ }  
  
}  
new Some_Plugin();
```

Inicialização

```
class Some_Plugin {  
  
    function SomePlugin() {  
        add_action( 'hook1', array($this, 'hook1' ));  
        add_action( 'hook1', array($this, 'hook2' ));  
    }  
  
    function hook1 { /* code */ }  
  
    function hook2 { /* code */ }  
  
}  
new Some_Plugin();  
  
add_action( 'plugins_loaded', array('Some_Plugin', 'Some_Plugin'  
    ));
```


Banco de dados

Usar sempre a wpdb:

Banco de dados

Usar sempre a wpdb:

- query

Banco de dados

Usar sempre a wpdb:

- query
- prepare

Banco de dados

Usar sempre a wpdb:

- query
- prepare
- insert

Banco de dados

Usar sempre a wpdb:

- query
- prepare
- insert
- update

Banco de dados

Usar sempre a wpdb:

- query
- prepare
- insert
- update
- get_var

Tratando dados para consultas

Tratando dados para consultas

```
$wpdb->get_var( $wpdb->prepare(
    "
    SELECT foo
    FROM {$wpdb->table}
    WHERE bar = %s
    ",
    $id
) );
```


Uso de Ações e Filtros

Uso de Ações e Filtros

- Base da construção de plugins no WordPress

Uso de Ações e Filtros

- Base da construção de plugins no WordPress
- Certificar-se de agendar os eventos e tratar os dados adequadamente

Implementação de Ações e Filtros

Implementação de Ações e Filtros

- Oferecer extensibilidade aos dados gerados

Implementação de Ações e Filtros

- Oferecer extensibilidade aos dados gerados
- Possibilitar a inserção de novos procedimentos à medida que eventos relevantes acontecem

Implementação de Ações

```
function show_something() {  
    do_action( 'before_show_something' );  
    /* something */  
    do_action( 'after_show_something' );  
}
```

Implementação de Ações

```
function show_something() {  
    do_action( 'before_show_something' );  
    /* something */  
    do_action( 'after_show_something' );  
}
```

```
function show_something_header() { /* header */ }  
add_action( 'before_show_something', 'show_something_header' );
```

```
function show_something_footer() { /* footer */ }  
add_action( 'after_show_something', 'show_something_footer' );
```


Implementação de Filtros

```
function get_something ( $param ) {  
    /* get $something */  
    return apply_filters( 'get_something', $something );  
}
```

Implementação de Filtros

```
function get_something ( $param ) {  
  
    /* get $something */  
  
    return apply_filters( 'get_something', $something );  
}  
  
function parse_something ( $something ) {  
  
    return preg_replace( $regex, $replace, $something );  
}  
add_filter( 'get_something', 'parse_something' );
```

Implementação de Ações e Filtros

```
<form>
<?php foreach ( $field as $f ) : ?>
    <?php $field = apply_filters( 'display_field', $field ); ?>
    <?php do_action( 'after_display_field', $field ); ?>

    <input
        name="<?php echo $f['name']; ?>"
        type="<?php echo $f['type']; ?>"
        value="<?php echo $f['value']; ?>" />

    <?php do_action( 'after_display_field', $field ); ?>
<?php endforeach; ?>

<?php do_action( 'before_submit' ); ?>
<input type="submit" value="Enviar" />

</form>
```


Enfileiradores de scripts

Enfileiradores de scripts

Funções:

Enfileiradores de scripts

Funções:

- `wp_enqueue_script`

Enfileiradores de scripts

Funções:

- `wp_enqueue_script`
- `wp_enqueue_style`

Enfileiradores de scripts

Funções:

- `wp_enqueue_script`
- `wp_enqueue_style`
- `wp_localize_script`

Enfileiradores de scripts

Funções:

- `wp_enqueue_script`
- `wp_enqueue_style`
- `wp_localize_script`

Hooks:

Enfileiradores de scripts

Funções:

- `wp_enqueue_script`
- `wp_enqueue_style`
- `wp_localize_script`

Hooks:

- `wp_enqueue_scripts`

Enfileiradores de scripts

Funções:

- `wp_enqueue_script`
- `wp_enqueue_style`
- `wp_localize_script`

Hooks:

- `wp_enqueue_scripts`
- `admin_enqueue_scripts`

Incluindo scripts

Incluindo scripts

```
function custom_scripts() {  
    wp_enqueue_script( 'jquery' );  
    wp_enqueue_script( 'my_script', ... 'my_script.js', array(  
        'jquery' ));  
}  
add_action( 'wp_enqueue_scripts', 'custom_scripts' );
```

Incluindo scripts e variáveis

Incluindo scripts e variáveis

```
function custom_scripts() {  
    wp_enqueue_script( 'jquery' );  
    wp_enqueue_script( 'my_script', ... 'my_script.js', array(  
        'jquery' ));  
    $params = array(  
        'key1' => 'val1',  
        'key2' => 'val2'  
    );  
    wp_localize_script( 'my_plugin', 'params', $params );  
}  
add_action( 'wp_enqueue_scripts', 'custom_scripts' );
```


Incluindo scripts e variáveis

Incluindo scripts e variáveis

```
<script .. jquery.js .. ></script>
```

```
<script type="text/javascript">
```

```
/* <![CDATA[ */
```

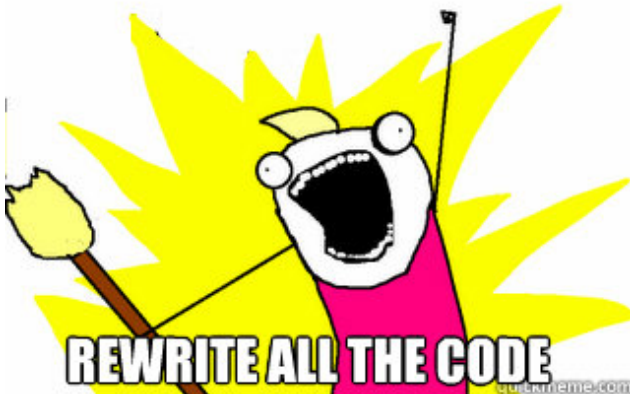
```
var params = {"key1":"val1","key2":"val2"};
```

```
/* ]]> */
```

```
</script>
```

```
<script .. my_script.js .. ></script>
```


Não precisa reinventar a roda



Internacionalização

Internacionalização

- Usar funções `__()` e `_e()`

Internacionalização

- Usar funções `__()` e `_e()`
- Carregar o arquivo MO de idioma

Tratamento de erros

Tratamento de erros

- WP_Error

Morrer elegantemente

Morrer elegantemente

- wp_die

Checar compatibilidade com versão

Checar compatibilidade com versão

```
if (!function_exists) ..
```

Formatação

Formatação

- `is_email`

Formatação

- `is_email`
- `remove_accents`

Formatação

- `is_email`
- `remove_accents`
- `sanitize_title`

Formatação

- `is_email`
- `remove_accents`
- `sanitize_title`
- `sanitize_email`

Formatação

- `is_email`
- `remove_accents`
- `sanitize_title`
- `sanitize_email`
- `seems_utf8`

Formatação

- `is_email`
- `remove_accents`
- `sanitize_title`
- `sanitize_email`
- `seems_utf8`
- `zeroise`

Formatação

- `is_email`
- `remove_accents`
- `sanitize_title`
- `sanitize_email`
- `seems_utf8`
- `zeroise`
- `wptexturize`

Transients API

Transients API

- `set_transient`

Transients API

- `set_transient`
- `get_transient`

Transients API

- `set_transient`
- `get_transient`
- `delete_transient`

HTTP API

HTTP API

- `wp_remote_get`

HTTP API

- `wp_remote_get`
- `wp_remote_retrieve_body`

HTTP API

- `wp_remote_get`
- `wp_remote_retrieve_body`
- `wp_remote_retrieve_headers`

Object Cache

Object Cache

- `wp_cache_add`

Object Cache

- `wp_cache_add`
- `wp_cache_set`

Object Cache

- `wp_cache_add`
- `wp_cache_set`
- `wp_cache_get`

Object Cache

- `wp_cache_add`
- `wp_cache_set`
- `wp_cache_get`
- `wp_cache_delete`

Object Cache

- `wp_cache_add`
- `wp_cache_set`
- `wp_cache_get`
- `wp_cache_delete`
- `wp_cache_flush`

Classes Úteis

Classes Úteis

- SimplePie

Classes Úteis

- SimplePie
- PHPMailer

Funções Úteis

Funções Úteis

- `wp_mail`

Funções Úteis

- `wp_mail`
- `fetch_feed`

Funções Úteis

- `wp_mail`
- `fetch_feed`
- `human_time_diff`

Referências

- Codex: Writing a Plugins
http://codex.wordpress.org/Writing_a_Plugin
- WordPress Answers
<http://wordpress.stackexchange.com/questions/715/objective-best-practices-for-plugin-development>