**Тестовый контроль «Плагины CMS WordPress»**



**1-15. Вопросы с выбором**

1. **Что такое хук в WordPress?**
   1. A) Функция для подключения плагина
   2. B) Специальная точка в коде, где можно вставить свой код для модификации поведения WordPress
   3. C) Шаблон для вывода данных
   4. D) Специальный файл для работы с базой данных

**Ответ:** b. B) Специальная точка в коде, где можно вставить свой код для модификации поведения WordPress

1. **Какой файл должен быть в корне плагина для его активации?**
   1. A) plugin.php
   2. B) index.php
   3. C) plugin-name.php
   4. D) functions.php

**Ответ:** c. C) plugin-name.php

1. **Какой параметр необходимо добавить в заголовок плагина, чтобы WordPress его распознал?**
   1. A) Plugin URI
   2. B) Version
   3. C) Plugin Name
   4. D) Author URI

**Ответ:** c. C) Plugin Name

1. **Как добавить действие (action) в WordPress с помощью плагина?**
   1. A) С помощью функции add\_action()
   2. B) С помощью функции add\_filter()
   3. C) С помощью функции do\_shortcode()
   4. D) С помощью функции plugin\_action()

**Ответ:** a. A) С помощью функции add\_action()

1. **Что такое фильтр в WordPress?**
   1. A) Функция для добавления нового типа записи
   2. B) Функция для изменения данных перед выводом на сайт
   3. C) Инструмент для поиска плагинов
   4. D) Метод для защиты сайта

**Ответ:** b. B) Функция для изменения данных перед выводом на сайт

1. **Какой файл обычно используется для регистрации пользовательских типов записи (Custom Post Types)?**
   1. A) functions.php
   2. B) custom-posts.php
   3. C) plugin-name.php
   4. D) register-cpt.php

**Ответ:** a. A) functions.php

1. **Как зарегистрировать собственный шорткод (shortcode) в WordPress?**
   1. A) Использовать функцию add\_shortcode()
   2. B) Использовать функцию do\_shortcode()
   3. C) Использовать функцию create\_shortcode()
   4. D) Использовать функцию register\_shortcode()

**Ответ:** a. A) Использовать функцию add\_shortcode()

1. **Как подключить внешние стили и скрипты для плагина в WordPress?**
   1. A) Использовать функцию wp\_enqueue\_style() и wp\_enqueue\_script()
   2. B) Использовать функцию wp\_add\_style()
   3. C) Добавить их прямо в HTML-шаблон
   4. D) Использовать плагин для автоматического подключения

**Ответ:** a. A) Использовать функцию wp\_enqueue\_style() и wp\_enqueue\_script()

1. **Как в плагине можно создать страницу настроек для администратора?**
   1. A) Используя функцию add\_menu\_page() и add\_submenu\_page()
   2. B) С помощью функции create\_settings\_page()
   3. C) Добавить страницу настроек через код темы
   4. D) Это невозможно в плагинах

**Ответ:** a. A) Используя функцию add\_menu\_page() и add\_submenu\_page()

1. **Что такое "опции" в WordPress и как с ними работать?**

* A) Это настройки, сохраняемые в базе данных, которые можно добавлять и изменять с помощью функций add\_option(), update\_option() и get\_option()
* B) Это функции для создания пользовательских типов записей
* C) Это плагины для безопасности
* D) Это шаблоны для вывода данных

**Ответ:** A) Это настройки, сохраняемые в базе данных, которые можно добавлять и изменять с помощью функций add\_option(), update\_option() и get\_option()

1. **Что такое REST API в WordPress?**

* A) Это способ работы с базой данных
* B) Это интерфейс для взаимодействия с другими сервисами и создания собственного API
* C) Это шаблон для вывода страниц
* D) Это система для создания плагинов

**Ответ:** B) Это интерфейс для взаимодействия с другими сервисами и создания собственного API

1. **Как в плагине подключить страницу, которая будет доступна из админ-панели?**

* A) Используя функцию add\_menu\_page() и add\_submenu\_page()
* B) Через функцию add\_dashboard\_widget()
* C) Используя функцию create\_admin\_page()
* D) Это невозможно в плагинах

**Ответ:** A) Используя функцию add\_menu\_page() и add\_submenu\_page()

1. **Как настроить безопасность плагина в WordPress?**

* A) Используя функцию nonce для защиты от CSRF
* B) Через настройку прав доступа к файлам
* C) Используя функции для защиты от SQL-инъекций
* D) Все вышеперечисленные варианты

**Ответ:** D) Все вышеперечисленные варианты

1. **Что делает функция wp\_nonce\_field()?**

* A) Защищает форму от межсайтовой подделки запросов (CSRF)
* B) Обрабатывает кэширование в плагинах
* C) Создает уникальный ID для каждой формы
* D) Проверяет наличие ошибок в коде плагина

**Ответ:** A) Защищает форму от межсайтовой подделки запросов (CSRF)

1. **Как в плагине добавить пользовательскую таблицу в базу данных WordPress?**

* A) Использовать функцию dbDelta()
* B) Создать таблицу вручную через phpMyAdmin
* C) Использовать стандартный MySQL код
* D) Таблицы можно добавлять только через тему

**Ответ:** A) Использовать функцию dbDelta()

**16-25. Открытые вопросы**

1. **Опишите процесс создания простого плагина в WordPress, который добавляет новый шорткод для отображения текста.**

**Открытый ответ:**

**Процесс создания простого плагина в WordPress, который добавляет новый шорткод для отображения текста.**

1. **Создайте файл плагина**: Создайте новый файл, например, my-shortcode-plugin.php в папке wp-content/plugins.
2. **Добавьте заголовок плагина**: В начале файла добавьте информацию о плагине.
3. <?php
4. /\*
5. Plugin Name: My Shortcode Plugin
6. Description: A simple plugin to add a shortcode for displaying text.
7. Version: 1.0
8. Author: Your Name

\*/

1. **Создайте функцию для шорткода**: Определите функцию, которая будет возвращать текст.
2. function my\_shortcode\_function() {
3. return "Hello, this is my custom shortcode!";

}

1. **Зарегистрируйте шорткод**: Используйте функцию add\_shortcode для регистрации шорткода.

add\_shortcode('my\_shortcode', 'my\_shortcode\_function');

1. **Активируйте плагин**: Перейдите в админ-панель WordPress и активируйте плагин.
2. **Как в плагине можно подключить внешние скрипты и стили? Приведите пример.**

**Открытый ответ:**

**Подключение внешних скриптов и стилей в плагине.**

Используйте функции wp\_enqueue\_script и wp\_enqueue\_style для подключения скриптов и стилей.

function my\_plugin\_enqueue\_scripts() {

wp\_enqueue\_script('my-custom-script', plugin\_dir\_url(\_\_FILE\_\_) . 'js/my-script.js', array('jquery'), '1.0', true);

wp\_enqueue\_style('my-custom-style', plugin\_dir\_url(\_\_FILE\_\_) . 'css/my-style.css', array(), '1.0');

}

add\_action('wp\_enqueue\_scripts', 'my\_plugin\_enqueue\_scripts');

1. **Как бы вы создали плагин, который добавляет новую страницу в админ-панель WordPress с настройками плагина?**

**Открытый ответ:**

**Создание плагина, который добавляет новую страницу в админ-панель.**

1. **Создайте функцию для страницы настроек**:
2. function my\_plugin\_settings\_page() {
3. add\_menu\_page(
4. 'My Plugin Settings',
5. 'My Plugin',
6. 'manage\_options',
7. 'my-plugin',
8. 'my\_plugin\_settings\_page\_content'
9. );
10. }

add\_action('admin\_menu', 'my\_plugin\_settings\_page');

1. **Создайте функцию для отображения содержимого страницы**:
2. function my\_plugin\_settings\_page\_content() {
3. echo '<h1>My Plugin Settings</h1>';
4. echo '<p>Here you can configure the settings for my plugin.</p>';

}

1. **Объясните, что такое wp\_enqueue\_script() и wp\_enqueue\_style(). Как они используются при разработке плагинов?**

**Открытый ответ:**

wp\_enqueue\_script**и**wp\_enqueue\_style**.**

Эти функции используются для безопасного подключения скриптов и стилей в WordPress. Они позволяют управлять зависимостями и версиями файлов.

function my\_plugin\_enqueue\_scripts() {

wp\_enqueue\_script('my-custom-script', plugin\_dir\_url(\_\_FILE\_\_) . 'js/my-script.js', array('jquery'), '1.0', true);

wp\_enqueue\_style('my-custom-style', plugin\_dir\_url(\_\_FILE\_\_) . 'css/my-style.css', array(), '1.0');

}

add\_action('wp\_enqueue\_scripts', 'my\_plugin\_enqueue\_scripts');

1. **Как бы вы добавили пользовательский тип записи (Custom Post Type) в плагин для WordPress? Приведите пример кода.**

**Открытый ответ:**

**Добавление пользовательского типа записи (Custom Post Type).**

function my\_custom\_post\_type() {

$args = array(

'label' => 'Books',

'public' => true,

'supports' => array('title', 'editor', 'thumbnail'),

);

register\_post\_type('book', $args);

}

add\_action('init', 'my\_custom\_post\_type');

1. **Какие основные функции и методы WordPress используются для работы с базой данных в плагинах? Приведите примеры.**

**Открытый ответ:**

**Основные функции для работы с базой данных.**

* global $wpdb: Получение объекта для работы с базой данных.
* $wpdb->get\_results(): Получение результатов запроса.
* $wpdb->insert(): Вставка данных.
* $wpdb->update(): Обновление данных.
* $wpdb->delete(): Удаление данных.

global $wpdb;

$results = $wpdb->get\_results("SELECT \* FROM {$wpdb->prefix}posts");

1. **Как бы вы защитили свой плагин от SQL-инъекций? Опишите, какие меры безопасности необходимо принять.**

**Открытый ответ:**

**Защита от SQL-инъекций.**

* Используйте подготовленные запросы с $wpdb->prepare().
* Экранируйте данные с помощью esc\_sql().

global $wpdb;

$user\_id = esc\_sql($\_GET['user\_id']);

$query = $wpdb->prepare("SELECT \* FROM {$wpdb->prefix}users WHERE ID = %d", $user\_id);

$results = $wpdb->get\_results($query);

1. **Что такое "опции" в WordPress? Как вы можете использовать их в плагине для сохранения настроек?**

**Открытый ответ:**

**Опции в WordPress.**

Опции используются для хранения настроек плагина.

function my\_plugin\_settings() {

add\_option('my\_plugin\_option', 'default\_value');

register\_setting('my\_plugin\_options\_group', 'my\_plugin\_option');

}

add\_action('admin\_init', 'my\_plugin\_settings');

1. **Опишите, как добавить функциональность REST API в плагин WordPress. Как это позволит взаимодействовать с вашим плагином?**

**Открытый ответ:**

**Добавление функциональности REST API.**

function my\_plugin\_rest\_api() {

register\_rest\_route('myplugin/v1', '/data/', array(

'methods' => 'GET',

'callback' => 'my\_plugin\_get\_data',

));

}

add\_action('rest\_api\_init', 'my\_plugin\_rest\_api');

function my\_plugin\_get\_data() {

return new WP\_REST\_Response(array('data' => 'Hello, REST API!'), 200);

}

1. **Что такое хуки и фильтры в WordPress? Приведите пример их использования в плагине.**

**Открытый ответ:**

**Хуки и фильтры в WordPress.**

Хуки позволяют добавлять или изменять функциональность.

* **Действия (Actions)**: add\_action()
* **Фильтры (Filters)**: add\_filter()

function my\_custom\_action() {

echo 'This is my custom action!';

}

add\_action('init', 'my\_custom\_action');

function my\_custom\_filter($content) {

return $content . ' This is my custom filter!';

}

add\_filter('the\_content', 'my\_custom\_filter');

**Вопросы на сопоставление**

1. **Сопоставьте элементы с их описаниями.**

**Элементы:**

1. add\_action()
2. wp\_nonce\_field()
3. REST API
4. add\_menu\_page()
5. Опции

**Описания:**

A) Функция для добавления страницы в админ-панель.  
B) Функция для защиты от CSRF-атак.  
C) Специальный интерфейс для взаимодействия с внешними сервисами.  
D) Функция для добавления действия в WordPress, которая будет выполнена в определённый момент.

E) Настройки, которые сохраняются в базе данных для дальнейшего использования в плагине.

**Ответ:**

**Сопоставим элементы с их описаниями:**

1. **add\_action()** - D) Функция для добавления действия в WordPress, которая будет выполнена в определённый момент.
2. **wp\_nonce\_field()** - B) Функция для защиты от CSRF-атак.
3. **REST API** - C) Специальный интерфейс для взаимодействия с внешними сервисами.
4. **add\_menu\_page()** - A) Функция для добавления страницы в админ-панель.
5. **Опции** - E) Настройки, которые сохраняются в базе данных для дальнейшего использования в плагине.

**Таким образом, правильные сопоставления:**

* **1-D**
* **2-B**
* **3-C**
* **4-A**
* **5-E**