



# WordPress

## 1-15. Вопросы с выбором

### 1. Что такое хук в WordPress?

- a. A) Функция для подключения плагина
- b. B) Специальная точка в коде, где можно вставить свой код для модификации поведения WordPress
- c. C) Шаблон для вывода данных
- d. D) Специальный файл для работы с базой данных

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

### 2. Какой файл должен быть в корне плагина для его активации?

- a. A) plugin.php
- b. B) index.php
- c. C) plugin-name.php
- d. D) functions.php

**Ответ:** c. C) plugin-name.php

**3. Какой параметр необходимо добавить в заголовок плагина, чтобы WordPress его распознал?**

- a. A) Plugin URI
- b. B) Version
- c. C) Plugin Name
- d. D) Author URI

**Ответ:** c. C) Plugin Name

**4. Как добавить действие (action) в WordPress с помощью плагина?**

- a. A) С помощью функции `add_action()`
- b. B) С помощью функции `add_filter()`
- c. C) С помощью функции `do_shortcode()`
- d. D) С помощью функции `plugin_action()`

**Ответ:** a. A) С помощью функции `add_action()`

**5. Что такое фильтр в WordPress?**

- a. A) Функция для добавления нового типа записи
- b. B) Функция для изменения данных перед выводом на сайт
- c. C) Инструмент для поиска плагинов
- d. D) Метод для защиты сайта

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

**6. Какой файл обычно используется для регистрации пользовательских типов записи (Custom Post Types)?**

- a. A) `functions.php`
- b. B) `custom-posts.php`
- c. C) `plugin-name.php`
- d. D) `register-cpt.php`

**Ответ:** а. A) functions.php

**7. Как зарегистрировать собственный шорткод (shortcode) в WordPress?**

- a. A) Использовать функцию add\_shortcode()
- b. B) Использовать функцию do\_shortcode()
- c. C) Использовать функцию create\_shortcode()
- d. D) Использовать функцию register\_shortcode()

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

**8. Как подключить внешние стили и скрипты для плагина в WordPress?**

- a. A) Использовать функцию wp\_enqueue\_style() и wp\_enqueue\_script()
- b. B) Использовать функцию wp\_add\_style()
- c. C) Добавить их прямо в HTML-шаблон
- d. D) Использовать плагин для автоматического подключения

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

**9. Как в плагине можно создать страницу настроек для администратора?**

- a. A) Используя функцию add\_menu\_page() и add\_submenu\_page()
- b. B) С помощью функции create\_settings\_page()
- c. C) Добавить страницу настроек через код темы
- d. D) Это невозможно в плагинах

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

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

- А) Это настройки, сохраняемые в базе данных, которые можно добавлять и изменять с помощью функций `add_option()`, `update_option()` и `get_option()`
- В) Это функции для создания пользовательских типов записей
- С) Это плагины для безопасности
- D) Это шаблоны для вывода данных

**Ответ:** А) Это настройки, сохраняемые в базе данных, которые можно добавлять и изменять с помощью функций `add_option()`, `update_option()` и `get_option()`

### 11. Что такое REST API в WordPress?

- А) Это способ работы с базой данных
- В) Это интерфейс для взаимодействия с другими сервисами и создания собственного API
- С) Это шаблон для вывода страниц
- D) Это система для создания плагинов

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

### 12. Как в плагине подключить страницу, которая будет доступна из админ-панели?

- А) Используя функцию `add_menu_page()` и `add_submenu_page()`
- В) Через функцию `add_dashboard_widget()`
- С) Используя функцию `create_admin_page()`
- D) Это невозможно в плагинах

**Ответ:** А) Используя функцию `add_menu_page()` и `add_submenu_page()`

### 13. Как настроить безопасность плагина в WordPress?

- А) Используя функцию `nonce` для защиты от CSRF
- В) Через настройку прав доступа к файлам
- С) Используя функции для защиты от SQL-инъекций

- D) Все вышеперечисленные варианты

**Ответ:** D) Все вышеперечисленные варианты

#### **14. 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)

#### **15. 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**

#### **16. 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

\*/

9. **Создайте функцию для шорткода:** Определите функцию, которая будет возвращать текст.

```
10.function my_shortcode_function() {
```

```
11.    return "Hello, this is my custom shortcode!";
```

```
}
```

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

```
add_shortcode('my_shortcode', 'my_shortcode_function');
```

13. **Активируйте плагин:** Перейдите в админ-панель WordPress и активируйте плагин.

## 17. 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');
```

## 18. 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');
```

11. Создайте функцию для отображения содержимого страницы:

```
12. function my_plugin_settings_page_content() {  
13.     echo '<h1>My Plugin Settings</h1>';  
14.     echo '<p>Here you can configure the settings for my plugin.</p>';  
15. }
```

## 19. 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');
```

## **20. 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');
```

## **21. 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");
```

**22. 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);
```

**23. 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');
```

**24. 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);  
}
```

## 25. 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

### 26. 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**