

# Complete WordPress Technical Interview Preparation

## 1. How do you safely update WordPress core, themes, and plugins?

- Use a staging site
- Backup files and database
- Update via WP-CLI or Dashboard
- Test all functions/UI
- Use security plugins like Wordfence
- Disable auto-updates for production

## 2. How do you troubleshoot a broken website after an update?

- Enable WP\_DEBUG in wp-config.php
- Use browser dev tools
- Check PHP error logs
- Deactivate plugins/themes one by one
- Use default theme (Twenty Twenty-Four)

## 3. How do you integrate Microsoft Dynamics CRM with WordPress forms?

- Use Zapier/Make or custom API integration
- Authenticate using OAuth2
- Use wp\_remote\_post for API call
- Map form fields to Dynamics
- Handle API logs and errors

## 4. How do you handle Google Ads Pixel / Meta Pixel installations and issues?

- Use Google Tag Manager
- Add GTM code in wp\_head
- Verify with Pixel Helper or Tag Assistant
- Use cookie plugins for consent mode (V2)
- Load scripts only after consent

## 5. What's your approach to ACF management and updates?

- Use local JSON (acf-json folder)
- Export groups as PHP when needed

- Use `get_field()` with fallback
- Test fields after ACF updates
- Sync between environments

## **6. How do you keep Gravity Forms integrated with HubSpot?**

- Use official HubSpot Add-On
- Authenticate using OAuth
- Use `gform_after_submission` for advanced logic
- Map fields and verify leads in HubSpot

## **7. How do you debug JS conflicts in a WordPress theme or plugin?**

- Use browser console
- Disable plugins one by one
- Use `noConflict` with jQuery
- Use `wp_enqueue_script` with dependencies

## **8. How do you develop or modify a WordPress widget via code?**

- Extend `WP_Widget` class
- Define `widget()`, `form()`, and `update()` methods
- Register widget using `widgets_init` action

## **9. How do you handle Geo IP and Cookie Consent per country?**

- Use `GeoTargetingWP` or Cloudflare geolocation
- Implement Google Consent Mode V2
- Use `Complianz` or `CookieYes` for cookie banner

## **10. What's your strategy for WordPress emergency support (Out of Hours)?**

- Use monitoring tools (Uptime Robot, Pingdom)
- Slack/Telegram alerts
- Keep access to FTP/cPanel/SSH
- Deploy hotfix or restore backup if needed

## **11. How do you optimize WordPress for speed?**

- Use caching (page, object, browser)
- Optimize images (WebP, compression)
- Use a CDN like Cloudflare or BunnyCDN

- Minify and combine CSS/JS
- Use lightweight themes and limit plugins
- Optimize database regularly
- Disable unnecessary WP features (emoji, embeds)
- Enable GZIP, OPcache, PHP 8.x
- Lazy load images and videos
- Monitor with GTmetrix or PageSpeed Insights

## **12. How do you work with ACF forms?**

- Use `acf_form()` to display frontend forms
- Use `'post_id' => 'new_post'` to create or `get_the_ID()` to update
- Limit fields using `'field_groups'` parameter
- Always use `acf_form_head()` before `get_header()`
- Validate user permissions before allowing form actions
- Use `acf/save_post` for custom handling after form submission

## **13. Basics of WordPress Theme Development?**

- Minimum files: `style.css` and `index.php`
- Common files: `header.php`, `footer.php`, `functions.php`, `page.php`, `single.php`
- Use `get_header()`, `get_footer()`, `get_sidebar()`
- Enqueue styles/scripts using `wp_enqueue_scripts` hook in `functions.php`
- Use template hierarchy for structure

## **14. Basics of Plugin Development?**

- Create a folder inside `wp-content/plugins/`
- Add a main `.php` file with plugin header comment
- Hook custom functions using `add_action()` or `add_shortcode()`
- Activate from admin Plugins panel
- Example: add a shortcode or modify WP behavior via hooks

## **15. Working with \$wpdb in WordPress?**

- Use global `$wpdb` to interact with DB
- Safe query using `$wpdb->prepare()`
- Examples:
  - SELECT: `$wpdb->get_results()`

- INSERT: `$wpdb->insert()`
- UPDATE: `$wpdb->update()`
- DELETE: `$wpdb->delete()`
- Always validate and sanitize inputs before use

## 16. What is `add_action()` in WordPress?

- `add_action()` hooks a custom function to a specific event.
- Commonly used to enqueue styles, run logic on init, etc.

Example:

```
add_action('init', 'my_custom_function');  
function my_custom_function() {  
    error_log("Init triggered");  
}
```

## 17. What is `add_filter()` in WordPress?

- `add_filter()` modifies data before it's output.
- Useful for changing titles, content, excerpts, etc.

Example:

```
add_filter('the_title', 'prefix_title');  
function prefix_title($title) {  
    return '-> ' . $title;  
}
```

## 18. ACF Form Frontend Usage Example

- To show a form on frontend with ACF fields:

1. At the top of the template:

```
acf_form_head();
```

2. Inside the page template:

```
acf_form([  
    'post_id' => 'new_post',
```

```

'post_title' => true,
'post_content' => true,
'new_post' => [
    'post_type' => 'post',
    'post_status' => 'draft'
],
'submit_value' => 'Submit Your Post'
]);

```

- Use 'field\_groups' to load only specific ACF fields
- Validate users with current\_user\_can() if needed

## 19. Difference Between Plugin and Widget in WordPress?

- A **Plugin** is a piece of software that adds functionality to WordPress.
- A **Widget** is a content block that appears in widget-ready areas (like sidebars).
- Widgets can be created via plugins or themes, but plugins have a broader scope (can add widgets, shortcodes, APIs, etc.)

Plugin = System-level feature

Widget = UI block shown in a specific area

## 20. How to Enqueue CSS and JavaScript in WordPress?

Use wp\_enqueue\_style() and wp\_enqueue\_script() in functions.php:

Example:

```

add_action('wp_enqueue_scripts', 'load_my_assets');
function load_my_assets() {
    wp_enqueue_style('my-style', get_stylesheet_uri());
    wp_enqueue_script('my-script', get_template_directory_uri() . '/main.js', array('jquery'), null, true);
}

```

- Always use hooks to enqueue
- Use get\_stylesheet\_uri() for style.css
- Use get\_template\_directory\_uri() for custom files