




## Objective Domains

# Web Certified Editor (WCE) for the WordPress® platform

The purpose of this exam is to assess a candidate's proficiency in using WordPress, particularly in creating and editing posts and pages on real websites. Candidates are expected to have a strong foundation in managing Posts, Pages, Comments, and the Media Library, as well as using the post and page builder and editing the CSS and HTML code of posts or pages.

Candidates are required to have at least 150 hours of combined instruction and hands-on experience with using WordPress before taking the exam. Achieving the certification demonstrates the candidate's competency at an industry-standard level, indicating that they are ready to enter the job market. The certification can also serve as a valuable tool for hiring managers when evaluating job applicants, helping to justify their hiring decisions and determine the candidate's placement within a company. WordPress Version: 6.8

### 1. Introduction to Wordpress

- 1.0.a Evolution & Key Milestones: Summarize WordPress's path from blogging tool to blockbased CMS; explain the impact of the Site Editor and Write/Design modes.
- 1.0.b Open-Source Foundations: Describe GPL licensing and project roles; locate Make/Core channels to follow proposals and releases.
- 1.0.c Release Rhythm: Differentiate major vs minor releases; read release notes; plan safe updates with backups and staging.
- 1.0.d AI in WordPress: Recognize AI is plugin/theme-driven; apply privacy/licensing checks and mandatory human review.
- 1.0.e Core vs Plugins vs Themes: Differentiate responsibilities; evaluate plugin/theme quality (updates, reviews, security); contrast block vs classic themes.
- 1.0.f Essential Vocabulary: Define posts/pages, categories/tags, blocks/patterns, Global Styles, Style Book, template parts, Query Loop, Navigation block.
- 1.0.g WordPress.com vs Self-Hosted (.org): Compare extensibility, pricing, maintenance, and data ownership; recommend the right approach for a scenario.
- 1.0.h Navigating the Interface: Navigate Dashboard/toolbar and list tables; use search/filters; operate editor List View, inspector, block toolbar, and Write  Design toggle.

- 1.0.i Site Structure Essentials: Map page hierarchies vs chronological posts; manage menus with the Navigation block; distinguish templates vs template parts; use patterns.
- 1.0.j System Requirements & Sandboxing: Check hosting requirements (PHP, DB, HTTPS, memory); set up local and staging environments; explain rollback.
- 1.0.k Maintenance & Security Basics: Apply updates/backups/least-privilege/2FA; run Site Health; note bcrypt password hashing (6.8+).
- 1.0.l Performance & Typography Tools: Optimize images and caching/CDN basics; understand Core Web Vitals; manage typography with Font Library and Global Styles/Style Book.
- 1.0.m Official Resources & Community: Use Docs, Learn. WordPress, Support Forums, Developer resources, and Make/Core; participate in meetups/WordCamps.

## 2. Organize Content with Categories & Tags

- 2.0.a Why Taxonomies Matter: Explain how categories/tags improve IA, navigation, and SEO; align on editorial governance.
- 2.0.b Creating & Editing Taxonomies: Name terms clearly; set slugs/descriptions; build category hierarchies with parent/child relationships.
- 2.0.c Assigning on the Fly: Add/remove terms in the editor sidebar, Quick Edit, and bulk actions; manage the default Uncategorized.
- 2.0.d Converting & Restructuring: Convert tags  $\square$  categories when strategy changes; merge duplicates; re-parent terms; redirect changed slugs when needed.
- 2.0.e Cleanup & Governance: Audit for tag bloat; cap top-level categories; maintain an approved tag list.
- 2.0.f Navigation & SEO: Expose term archives in menus; build taxonomy listings with Query Loop; show counts/pagination when helpful.
- 2.0.g Styling & Templates: Create term templates/patterns in the Site Editor for consistent archive layouts and intros.
- 2.0.h Performance & Accessibility Checks: Paginate long archives; write descriptive headings; keep lists keyboard-friendly.
- 2.0.i AI in Taxonomy Management: Use AI-suggested tags with human review; enforce privacy/licensing rules.

## 3. Manage Posts

- 3.0.a Creating a New Post: Compose in the Block Editor; switch Write/Design modes; set clear titles/slugs; structure with appropriate blocks/media.
- 3.0.b Post Settings & Publishing: Configure status/visibility, author, template, featured image, excerpt, categories/tags, discussion; preview; publish or schedule.

- 3.0.c Editing & Quick Edit: Update content/metadata in the editor; use Quick Edit for fast changes (date, status, slug, categories/tags) without breaking URLs/templates.
- 3.0.d Post Status & Trash: Differentiate draft/private/published; trash, restore, or delete permanently with awareness of impact.
- 3.0.e Bulk Operations & List-View Tools: Bulk-edit multiple posts; adjust Screen Options (columns, items per page) and row density for efficiency.
- 3.0.f Search & Filter: Find posts via search; filter by date, author, category, tag, status; apply naming that aids retrieval.
- 3.0.g Revisions & Autosave: Compare and restore revisions; rely on autosave; understand revision limits.
- 3.0.h Dynamic Post Listings: Build listings with Query Loop; surface result counts/pagination when useful.
- 3.0.i Comments on Posts: Enable/disable comments per post; access related moderation tools.
- 3.0.j Quality & Accessibility Checks: Enforce heading hierarchy, descriptive links, image alt text, and suitable reading length; run pre-publish checks.
- 3.0.k AI in Post Creation: Use AI drafting/summarising responsibly—human review, fact/licence verification, no sensitive data.

## 4. Manage Pages

- 4.0.a Creating a New Page: Draft in Write mode; switch to Design for layout; preview content-only vs template; publish or schedule.
- 4.0.b Page Settings & Metadata: Set templates, parent/order, featured image, discussion, custom fields; complete SEO plugin fields.
- 4.0.c Editing & Quick Edit: Modify parent/order/template efficiently; use Quick Edit for batch tweaks.
- 4.0.d Page Status & Trash Workflow: Handle draft/private/published; trash/restore/delete; understand effects on menus/links.
- 4.0.e Bulk Operations & List-View Tools: Apply safe bulk updates; configure Screen Options and row density for large page libraries.
- 4.0.f Search & Filter: Locate pages by parent, template, status; adopt naming conventions that help discovery.
- 4.0.g Hierarchies & Navigation: Build parent-child structures, set menu order, and add pages to Navigation.
- 4.0.h Templates & Theme Integration: Choose default/custom templates; understand block vs classic themes; know when child themes are needed.
- 4.0.i Pages vs Posts — Key Distinctions: Contrast structural vs chronological content; explain archive/feed behavior.

- 4.0.j Comments on Pages: Decide when comments/pingbacks fit; manage resulting threads.
- 4.0.k Accessibility & Quality Checks: Apply headings/link text/alt text; verify contrast and focus order.
- 4.0.l AI in Page Creation: Use AI layout/copy aids with human editing and policy compliance.

## 5. Block Editor & Site Editor (Post/Page Builder)

- 5.0.a Editing Stack Overview: Differentiate Block Editor (per post/page) vs Site Editor (templates/patterns/global design); operate Write/Design, List View, inspector/toolbar; preview content vs template.
- 5.0.b Core Blocks Toolkit: Select suitable blocks for common needs; use newer additions (Table of Contents, Details, Query Total, Gallery Lightbox).
- 5.0.c Formatting & Styling Content: Control typography/color/spacing/borders/dimensions; distinguish per-block styles vs Global Styles; preview in Style Book.
- 5.0.d Layout & Grouping Strategies: Build responsive sections with Group, Row, Stack, Columns, Cover, Media & Text; nest sensibly; manage gaps/padding/margins.
- 5.0.e Reusable Assets & Patterns: Create Reusable Blocks; convert to synced/unsynced Patterns; organize folders; update safely across instances.
- 5.0.f Dynamic & Interactive Blocks: Assemble listings with Query Loop; add Query Total when counts help; use Archives, Latest Posts/Comments, Calendar, Search, Social Icons, Tag Cloud, Navigation.
- 5.0.g Embeds & Shortcodes: Embed YouTube/Vimeo/X/Instagram/Reddit with privacy-minded settings; use the Shortcode block for legacy/plugin features.
- 5.0.h Extensibility Awareness (Concepts): Recognize plugin/theme-added blocks/styles and Block Hooks; apply theme Style Variations or custom block styles.
- 5.0.i Accessibility & Performance Checks: Maintain heading order, descriptive links, alt text, contrast, visible focus, keyboard nav; optimize media and avoid overly nested layouts.
- 5.0.j AI in Layout Building: Use AI layout/copy suggestions with human review; verify accuracy/licensing; avoid sensitive data.

## 6. Manage Comments

- 6.0.a Purpose & Impact: State how moderation shapes reputation, trust, engagement; align on a clear comment policy.
- 6.0.b Posting & Replying: Add and reply to threaded comments; view conversation hierarchy from the Dashboard.

- 6.0.c Comment Moderation Workflow: Approve/unapprove/spam/trash/restore/delete; edit author details and comment text when necessary.
- 6.0.d Bulk Operations & List-View Tools: Moderate in bulk; tune Screen Options (columns/items) and list density for efficiency.
- 6.0.e Search & Filter: Find comments by keyword; filter by author, email, IP, date range, or status.
- 6.0.f Discussion Settings: Enable/disable per post; configure global rules in Settings ⇌
- 6.0.g Discussion (manual approval, close on older posts, disallowed keys, nesting depth).
- 6.0.h Designing the Conversation Area: Customize layouts with Comments and related blocks; style spacing/typography/borders for readability.
- 6.0.i Spam Protection & Performance: Use anti-spam tools (Akismet/antispam, blocklists,rate-limits, CAPTCHA as appropriate); consider lazy-loading avatars.
- 6.0.j Accessibility & UX Checks: Provide clear labels, good contrast, visible focus, keyboard nav; ensure messages are announced to assistive tech.
- 6.0.k AI in Comment Management: Use AI moderation/summarization with human oversight; guard privacy and bias.

## 7. Manage Media Library

- 7.0.a Purpose & Scope: Manage images/video/audio/PDFs; understand Font Library (block themes) for typography.
- 7.0.b Uploading & Adding Media: Upload/drag-drop to Library or editor; insert from URL; (optionally) source assets via Openverse integrations.
- 7.0.c Organising & Searching: Switch Grid/List views; filter by type/date/author; search by keyword; adjust list density for large libraries.
- 7.0.d Inserting & Reusing Assets: Place media in posts/pages/patterns/backgrounds; use Replace to keep URLs stable when updating.
- 7.0.e Editing Metadata & Accessibility: Edit title/caption/description/alt text; write meaningful alt text for screen readers and SEO.
- 7.0.f Basic Image Edits: Crop/rotate/flip/resize in the built-in editor; save new copies when needed.
- 7.0.g Performance Optimisation: Choose correct dimensions; compress appropriately; prefer WebP/AVIF; support Core Web Vitals.
- 7.0.h Font Library Management (Block Themes): Upload/activate/preview/remove fonts; apply via Global Styles.
- 7.0.i Deleting & Replacing Media Safely: Trash or permanently delete with awareness of impact on published content;

replace assets without breaking links.

- 7.0.j AI in Media Management: Use AI for alt-text or tagging with human review; respect copyright/privacy.

## 8. Advanced Editing & Optimization

- 8.0.a Content Quality & Structure: Write clear, scannable content with logical H1–H6 hierarchy, descriptive links, consistent voice, and focused CTAs.
- 8.0.b Accessibility Compliance: Provide meaningful alt text; meet color-contrast and keyboard-nav requirements; use ARIA sparingly/correctly; test with readers/checkers.
- 8.0.c Copyright & Licensing: Differentiate public domain/CC/commercial; attribute correctly; store license info; verify rights for fonts/images/embeds.
- 8.0.d Custom HTML & Template Tweaks: Use Code Editor for anchors/IDs/microcopy; edit template parts in Site Editor or child themes for layout-level changes.
- 8.0.e CSS & Global Styles: Prefer Global Styles tokens; add CSS in sanctioned locations; preview changes in Style Book.
- 8.0.f Shortcodes & Embeds: Insert shortcodes for legacy/plugin features; choose Embed blocks with privacy-enhanced modes when available.
- 8.0.g SEO Foundations: Craft unique titles/meta; use one H1; build internal linking/taxonomies; add schema via suitable plugins; configure Open Graph/Twitter Cards.
- 8.0.h Performance & Core Web Vitals: Optimize images (size/compression/lazy-load); limit heavy scripts; defer non-critical assets; monitor LCP/INP/CLS.
- 8.0.i Troubleshooting & Diagnostics: Use Tools - Site Health; isolate conflicts (disable plugins/switch themes); check console; enable WP\_DEBUG in staging; fix media MIME/size; flush permalinks.
- 8.0.j AI in Optimization & Code: Use AI for summaries/alt-text/CSS hints with human review; verify facts/licensing; avoid sensitive data.

## 9. Profile and Screen Options

- 9.0.a Editing Your User Profile: Update display name, bio, profile picture, language, contact fields; set Admin Color Scheme and toolbar preference.
- 9.0.b Password & Security Controls: Generate strong passwords; “log out everywhere else”; reset via “Lost your password?”; manage Application Passwords; enable 2FA per policy.
- 9.0.c Customising the Dashboard: Show/hide/reorder widgets (Welcome, Quick Draft, Activity, Site Health) via Screen Options and drag-and-drop.
- 9.0.d Screen Options for List Views: Choose columns, set items per page, and adjust row density for Posts, Pages, Media, Comments, Categories, Tags.
- 9.0.e Posts & Pages Specifics: Reveal columns like Author,

Categories, Tags, Comments, Slug, Sticky, Template; filter by status/category/date/author.

- 9.0.f Categories & Tags Screens: Show/hide Description, Slug, Count; set items per page; manage parent relationships (categories).
- 9.0.g Comments Screen: Show/hide In Response To, Submitted On, IP; approve/unapprove/spam/trash/restore; bulk-moderate.
- 9.0.h Global User Preferences: Toggle Top Toolbar, Write mode, List View, pre-publish checks, Block Manager visibility; set preferred default context.
- 9.0.i AI in Profile & Screen Customisation: Use AI helpers carefully; avoid sensitive data; keep human editorial control. Copyright & Licensing: Differentiate public domain/CC/commercial; attribute correctly; store license info; verify rights for fonts/images/embeds.

## 10. Working with AI in WordPress

- 10.0.a Why Prompt Craft Matters: Write precise prompts to reduce rework; set scope/tone/format and constraints that fit brand and audience.
- 10.0.b Anatomy of an Effective Prompt: Specify role/goal, audience/tone, format (bullets/HTML/JSON), inputs, constraints, acceptance criteria, and “do-not” rules.
- 10.0.c Iterative Workflow: Draft ⇌ critique ⇌ refine; request outlines first; version prompts/outputs; test in staging.
- 10.0.d Evaluating AI Output: Fact-check names/dates/links; run plagiarism/licence checks; validate HTML/schema/accessibility/SEO; review code for security/performance.
- 10.0.e Ethical & Legal Guardrails: Exclude PII/secrets; respect copyright/brand rules; follow org data policies; disclose AI assistance where required.
- 10.0.f SEO & Performance Considerations: Avoid bloated markup; ensure unique titles/meta; compress images; check impact on LCP/INP/CLS.
- 10.0.g Accessibility Compliance: Generate accurate alt text; keep semantic structure; ensure contrast and keyboard operability.
- 10.0.h Common Prompt Patterns for WP Tasks: Request SEO titles/meta, alt-text options, block-pattern outlines, taxonomy plans, and publish QA checklists.
- 10.0.i Privacy & Data-Handling Checklist: Minimize/anonymize data; use staging URLs; store outputs in approved repos.
- 10.0.j When to Escalate to Human Expertise: Escalate legal/compliance copy, accessibility audits, security incidents, payments, migrations, custom production code, and highstakes SEO.

## 11. Web Design Foundations

- 11.0.a Understanding the Purpose of Web Design: Define

audience/goals/metrics; translate into content hierarchy, page types, and user journeys; balance aesthetics with accessibility, performance, and SEO.

- 11.0.b UI vs. UX Design Basics: Differentiate UI components/ states from UX research/flows; apply heuristics (consistency, feedback, error prevention).
- 11.0.c Wireframing and Storyboarding: Sketch low-fi grids and CTAs; map wireframes to WordPress blocks/patterns; iterate with stakeholders before visual polish.
- 11.0.d Visual Design Principles: Apply alignment, balance, contrast, hierarchy, proximity, repetition, and white space; use predictable grids and 8-pt spacing.
- 11.0.e Color Theory and Branding: Build accessible palettes; meet WCAG contrast; encode brand tokens in Global Styles/theme JSON.
- 11.0.f Typography and Readability: Pair heading/body faces; manage webfonts via Font Library; subset and limit weights.
- 11.0.g Image Use and Layout: Select purposeful imagery; preserve aspect ratios/focal points; avoid text-in-images; optimize (WebP/AVIF), lazy-load; write descriptive alt text; compose layouts with Group/Columns/Cover.
- 11.0.h Introduction to Responsive Design: Design mobile-first with fluid grids and flexible media; test common breakpoints; ensure nav/tables/media scale gracefully using block layout controls and Preview.