

Explore, Build, and Share with WordPress Playground

Facilitator Guide

Overview

Topic	WordPress Playground
Duration	60 minutes
Audience	Mixed, beginners and experienced users welcome
Environment	Browser only: no installation, account, or downloads required
Materials	Device with a modern browser (Chrome, Firefox, Edge, or Safari) and internet access

 **Version note:** Written for WordPress 7.0 (released May 20, 2026). Verify steps against the current release before running. Release notes: wordpress.org/news | Core development blog: make.wordpress.org/core. WordPress 7.0 changes relevant to this activity: the default admin color scheme is now 'Modern' — a lighter, higher-contrast palette. Navigation structure and all Playground features covered in this guide are unchanged.

Learning Objectives

By the end of this activity, participants will be able to:

- * Identify Playground's core use cases: safe testing, version testing, and environment sharing.

- * Test plugins, themes, and multi-plugin combinations without touching a live site.
 - * Load a WordPress Beta or nightly build and locate where to report findings.
 - * Use URL parameters to pre-configure a Playground environment.
 - * Save and export a Playground environment as a portable .zip file.
 - * Assess how Playground could support your own work or meetup.
-

What Participants Need

- * A device with a modern web browser (Chrome, Firefox, Edge, or Safari)
- * An internet connection
- * No WordPress account, login, or local installation required

Note: Playground uses WebAssembly to run PHP directly in the browser. First load downloads a bundle and may take 10 to 20 seconds on slower connections.

Facilitator Setup Notes

- * Test the Playground URL before your session: playground.wordpress.net.
 - * Open a browser tab to playground.wordpress.net ahead of time so you can demo without waiting for first load.
 - * Have a tab ready with the Classic Editor pre-installed for the Part 3 demo: playground.wordpress.net/?plugin=classic-editor.
 - * Have the Blueprint builder open for Part 2: playground.wordpress.net/builder/builder.html. Practice selecting a "Nightly" WordPress version and clicking Run.
 - * Export a .zip from your demo Playground ahead of time, or generate one live during Part 4.
 - * Decide in advance how participants will share files in Part 5 (AirDrop, Slack, email, USB). On managed networks, test file transfer before the session.
 - * Playground runs entirely in the browser: no downloads, no admin passwords, no account creation needed.
-

Activity Steps

Total time: 60 minutes | Parts 1 to 5 + 5 min debrief

Part 1: Launch and Explore (15 minutes)

1. Open a new browser tab and go to playground.wordpress.net.
2. When Playground loads, take a moment to look around. You are looking at a fully functioning WordPress site running entirely in your browser. No server, no account, no installation.
3. Find the **four-squares icon** in the top-right corner of the Playground toolbar and click it to open the Playground site manager.
4. Explore the **Quick Access dropdown** at the top of the screen. From here you can jump directly to Posts, Pages, Plugins, Themes, and more.
5. Navigate to **Posts > Add New** and write a short test post. Add a title, insert a Paragraph block with any content, and click **Publish**.
6. Click **View Post** to see your published post on the site front end.

 **Facilitator tip:** If anyone's Playground is slow to load, have them try a hard refresh (Cmd+Shift+R on Mac / Ctrl+Shift+R on Windows). Playground downloads a full WordPress bundle the first time, and this can take 10 to 20 seconds on slower connections.

 **Facilitator tip:** While participants write their posts, mention the Blueprint library link on the Playground home page: "This is a library of pre-built Blueprints, environments pre-configured for specific tasks. We will connect to this idea in Part 3." Plant the seed here; you'll come back to it.

Part 2: Test with Plugins, Themes, and Versions (10 minutes)

1. Open a new tab and install a plugin via URL: playground.wordpress.net/?plugin=hello-dolly. Then go to **Plugins > Installed Plugins** to confirm it is active.
2. Test a theme: open a new tab with playground.wordpress.net/?theme=twentytwentyfive and click the front-end link to see it applied.
3. Test plugin compatibility. Load two plugins at once: playground.wordpress.net/?plugin=contact-form-7&plugin=flamingo. Go to **Plugins > Installed Plugins** and confirm both are active.
4. Test a Beta build using the real-world method. Load: playground.wordpress.net/?plugin=wordpress-beta-tester. Go to **Tools > Beta Testing**. Under Update channel, select **Bleeding edge**; a Stream section will appear. Under Stream, select **Nightlies**. Click **Save Changes**. Then go to **Dashboard > Updates** and click **Update Now** to install the nightly build.
5. Once updated, explore the environment: create a post, then install a plugin you rely on. Does anything behave differently? Note anything unexpected.

 **Facilitator tip:** Frame Part 2 as a zero-risk testing lab: "You can test any plugin or theme here before installing it on a live site. If it breaks something, just close the tab and open a fresh Playground. No damage done."

 **Facilitator tip:** The plugin compatibility step shows real value: many site owners discover conflicts only after an update breaks production. Playground lets you test combinations first. Plugin slugs in URL parameters are the same slugs used on wordpress.org/plugins.

 **Facilitator tip:** The WordPress Beta Tester plugin is the same tool seasoned contributors use on staging sites before a major release. Using it here shows participants the real workflow, not a shortcut. Key message: "If a plugin you depend on breaks in the next release, you find out now, not the day of the update." During a Beta or RC cycle, participants can choose "Beta/RC Only" instead of "Nightlies" as the stream, for a more stable test target. Emphasize that Core developers read and act on bug reports; community testing is how WordPress improves.

 **Bug reporting:** Playground bugs: github.com/WordPress/wordpress-playground/issues. WordPress Core (Beta/RC) bugs: make.wordpress.org/test. Playground bugs and WordPress Core bugs are tracked separately; use the correct tracker for what you found.

Part 3: Customize with URL Parameters (10 minutes)

1. Open a **new browser tab** and paste this URL:
playground.wordpress.net/?plugin=classic-editor. This is a different plugin from the ones in Part 2: Classic Editor replaces the block editor entirely.
2. Watch Playground load. This time, the [Classic Editor plugin](#) is pre-installed and activated automatically. No manual install needed.
3. Navigate to **Posts > Add New** in this new Playground. Notice the editor interface looks different: that is the Classic Editor.
4. Now install two plugins at once. Open another new tab with:
playground.wordpress.net/?plugin=classic-editor&plugin=contact-form-7.
5. Go to **Plugins > Installed Plugins** and confirm both the [Classic Editor](#) and [Contact Form 7](#) are active.

 **Facilitator tip:** URL parameters are the building blocks of Blueprints. A Blueprint is a JSON file that can pre-install plugins, themes, import content, and run WP-CLI commands. Think of URL parameters as the simple version, and the Blueprint builder (used in Part 2) as the full-power version. Point participants to the Playground documentation for further reading:
wordpress.github.io/wordpress-playground

 **Facilitator tip:** Plugin slugs in URL parameters are the same slugs used on wordpress.org/plugins. `?plugin=classic-editor` maps directly to wordpress.org/plugins/classic-editor. Any plugin on the directory can be loaded this way. This is the same pattern participants used in Part 2, now applied to configure a specific experience.

Part 4: Save Your Work (5 minutes)

1. Go back to your **first Playground tab** (the one where you published a post in Part 1). Close any extra tabs from Parts 2 and 3.
2. Look for the “**Unsaved Playground**” label in the Playground top bar. Next to it is a **Save** button — click it.
3. Clicking **Save** stores your Playground state in the browser’s local storage; your changes will persist if you close and reopen the same tab.
4. Now export your site: from the same area, choose **Export**. This downloads a .zip containing your entire WordPress install: database, media uploads, active plugins, and theme.

 **Facilitator tip:** Save uses the browser’s IndexedDB; it’s local to that browser and device. The Export .zip is fully portable: it can be imported into a new Playground instance, a local dev environment (MAMP, Studio), or even a live production site.

 **Going further: launching as a real site:** Yes, an exported Playground .zip can become a real production site. Playground uses SQLite by default, so you cannot upload the .zip directly to most shared hosting (which requires MySQL). Here are the two main paths:

Via WordPress.com: import the .zip into Studio by Automattic (free desktop app at developer.wordpress.com/studio), then deploy to WordPress.com with one click.

Part 5: Partner Exchange (10 minutes)

1. Find a partner. If the group has an odd number, form a trio: you will all still import one environment each.
2. Share your exported .zip with your partner. Options: AirDrop (Mac/iOS), Slack, email, or a shared USB drive; use whatever works in your space.
3. While you wait for your partner’s file, open a fresh Playground in a new tab: playground.wordpress.net.
4. Once you have your partner’s .zip, click the **four-squares icon** (top right) to open the Playground site manager. Choose **Import** and select your partner’s .zip file.
5. Explore what your partner built. You are now running their entire Playground environment inside your own browser.

 **Facilitator tip:** Odd groups: A shares with B, B shares with C, C shares with A. Everyone gives and receives, same time as pairs.

 **Facilitator tip:** Import can take 15 to 30 seconds depending on the .zip size. Use this wait time: “What might you use this import/export workflow for in your own work?”

Debrief and Discussion (5 minutes)

Discussion questions:

1. What surprised you most about Playground? Was there anything it could do that you did not expect?
2. Where could you see yourself using Playground: for testing a plugin, teaching a concept, or sharing an environment with a client or teammate?
3. How might you use the partner exchange workflow at a future meetup or in your own work?

The win: Today you launched a fully functioning WordPress site in your browser with no setup, tested plugins and themes safely, loaded a WordPress Beta build and found where to report what you discover, customized Playground with URL parameters, saved and exported your work, and imported someone else's entire environment, all without installing a single thing. That is the power of Playground.

Next steps for participants:

- * [Try the Playground Blueprint builder](#)
- * [Explore the Blueprint library](#)
- * [Read the Playground documentation](#)
- * [Keep learning on Learn WordPress](#)
- * [Join the #playground channel in WordPress Slack](#)
- * [Studio by Automattic \(deploy to WordPress.com\)](#)

Sources

All URLs verified as of April 2026. Check before running the activity.

1. [WordPress Playground documentation](#): Official documentation maintained by the WordPress open source project.
2. [WordPress Playground GitHub repository](#): Source code, issue tracker, and release history.
3. [WordPress release notes](#): Official WordPress release announcements. Verify the current stable version before running.
4. [Blueprint demos library](#): Curated collection of pre-built Playground Blueprints.
5. [Hello Dolly plugin](#): Used in Part 2 plugin installation demo.
6. [Twenty Twenty-Five theme](#): Used in Part 2 theme testing demo.
7. [Contact Form 7 plugin](#): Used in Part 2 and Part 3 multi-plugin URL parameter demos.
8. [Flamingo plugin](#): Used in Part 2 plugin compatibility demo alongside Contact Form 7.
9. [WordPress Beta Tester plugin](#): Used in Part 2 Beta testing step. The standard tool for switching a site to a nightly or Beta build.
10. [WordPress Playground issue tracker](#): Where to report bugs found in Playground itself during Beta testing.
11. [make.wordpress.org/test](#): WordPress Core test team. Report issues found in WordPress Beta/RC builds here.
12. [Classic Editor plugin](#): Used in Part 3 URL parameter demo.
13. [Learn WordPress](#): Official WordPress learning platform. Suggested next step for participants.
14. [make.wordpress.org/core](#): Core development blog. Track changes to the block editor and Playground in upcoming releases.
15. [Studio by Automattic](#): Free desktop app for local WordPress development. Imports Playground zips and deploys to WordPress.com.