构建系统：函数前缀与块加载

目录

- 核心理念
- 系统工作原理
  - 源代码结构
  - 构建过程
  - 构建文件位置
- 为何需要此系统
- 测试构建的 PHP 函数

Gutenberg 使用一套构建系统来自动转换 PHP 函数名称，以避免与 WordPress 核心发生冲突。该系统允许 Gutenberg 在源代码中使用标准的 WordPress 函数名，同时确保插件能与 WordPress 核心共存而不会产生命名冲突。

核心理念

源文件使用标准的 **WordPress** 函数名（例如 `block_core_navigation_link_build_css_colors`），但构建系统会在实际运行时加载的构建文件中将其转换为带前缀的版本（例如 `gutenberg_block_core_navigation_link_build_css_colors`）。

系统工作原理

源代码结构

Gutenberg 的源代码组织在 `packages/` 目录中：

```
packages/block-library/src/
├── navigation-link/
│   ├── index.php           # 包含标准函数名的源文件
│   ├── block.json
```

```
│   └── edit.js
├── post-time-to-read/
│   ├── index.php           # 包含标准函数名的源文件
│   └── block.json
└── ...
```

构建过程

构建系统在 `packages/block-library/package.json` 中配置，并转换以下内容：

- 函数名：`block_core_navigation_link_build_css_colors()` → `gutenberg_block_core_navigation_link_build_css_colors()`
- 函数调用：`wp_get_typography_font_size_value()` → `gutenberg_get_typography_font_size_value()`
- 类名：`WP_Style_Engine` → `WP_Style_Engine_Gutenberg`

构建文件位置

构建系统在 `build/` 目录中创建转换后的文件：

```
build/
├── block-library/                    # 主块库
│   ├── navigation-link.php           # 带前缀的函数
│   ├── post-time-to-read.php         # 带前缀的函数
│   └── ...
├── scripts/block-library/            # 脚本特定构建
│   ├── navigation-link.php           # 相同的带前缀函数
│   └── ...
└── scripts/style-engine/             # 带前缀类的样式引擎
    ├── class-wp-style-engine-gutenberg.php
    └── ...
```

`lib/blocks.php` 文件按优先级顺序加载构建好的块。

为何需要此系统

- 避免 **WordPress** 核心与 **Gutenberg** 插件之间的函数名冲突
- 实现向后移植兼容性 - 当函数被向后移植到核心时，Gutenberg 可以继续使用带前缀的版本

- 提供插件独立性 - Gutenberg 可以独立于 WordPress 核心发布周期进行演进
- 允许测试隔离 - 测试可以针对特定的函数版本（ Gutenberg 版与核心版）

测试构建的 PHP 函数

有关如何测试构建（带前缀）的 PHP 函数的信息，请参阅测试概述文档中的 测试带前缀的函数 部分。

该系统确保 Gutenberg 能够独立演进，同时保持与 WordPress 核心的兼容性，为区块编辑器的持续发展提供了坚实的基础。

版本号 #2
由 壳壳虫 创建于 2026-03-27 01:08:43 UTC
由 壳壳虫 更新于 2026-03-27 01:08:43 UTC