



# Keine Angst vor der Kommandozeile - Eine Einführung in WP-CLI

Marcel Tannich | WordCamp Vienna 2020



# Marcel Tannich

WordPress Technical Supporter @ OnTheGoSystems Inc.



# Download Folien

[https://github.com/mardroid/wcvienna\\_2020](https://github.com/mardroid/wcvienna_2020)

<https://marceltannich.com>

# Was ist die Kommandozeile?

---

- Eine textbasierte Applikation zum Betrachten, Bearbeiten und Manipulieren deiner Dateien
- Auch bekannt als Konsole, Terminal, Interface, CMD, CLI oder Prompt
- Im Englischen wird sie als “command-line interface”, kurz “CLI” oder als “command prompt” (Eingabeaufforderung) bezeichnet
- Ein “>” oder “\$” Zeichen signalisiert, dass auf eine Eingabe gewartet wird

Command Prompt

Microsoft Windows [Version 10.0.18362.535]  
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Marcel>

# Was ist WP-CLI?



- WP-CLI ist eine Sammlung von Kommandozeilen-Tools zur Verwaltung deiner WordPress Installationen.
- Man kann damit verschiedenste WordPress Aktionen durchführen, ohne einen Browser zu verwenden.
- Das Ziel ist, für jede Aktion in WordPress ein Kommandozeilen-Interface als Alternative zum Admin-Bereich zur Verfügung zu stellen.

C:\WINDOWS\system32\cmd.exe - "C:\Users\Marcel\AppData\Roaming\Local by Flywheel\ssh-entry\K5VEYtm7J.bat"

```
root@a97aa8a078c1:/# wp plugin install wordpress-seo
Installing Yoast SEO (12.8)
Downloading installation package from https://downloads.wordpress.org/plugin/wordpress-seo.12.8.zip...
Unpacking the package...
Installing the plugin...
Plugin installed successfully.
Success: Installed 1 of 1 plugins.
root@a97aa8a078c1:/#
```

# Und was kann ich damit tun?

---

- WordPress Installationen verwalten (Updates, Installationen, Einstellungen, ...)
- Benutzer, Beiträge, Kategorien hinzufügen/entfernen
- Testdaten einfügen
- Suchen & Ersetzen in der Datenbank durchführen
- Passwörter zurücksetzen
- Backups erstellen
- Code Vorlagen für Plugins, Themes & Blöcke generieren
- Skripte zur Automatisierung erstellen uvm.

**Zeit sparen!!!**

# Voraussetzungen WP-CLI



- 
- Installation auf Hosting Paket / Server
  - SSH Zugang zu deinem Hosting Paket / Server
  - UNIX ähnliche Umgebung (macOS, FreeBSD, Cygwin, ...) oder Windows (eingeschränkt)
  - PHP 5.3.29 oder neuer
  - WordPress 3.7 oder neuer

# Installation

# Installation WP-CLI Windows 10

---

- Phar file herunterladen ([Klicke hier](#))
- Ordner anlegen
- Lokalen Server installieren z.B.: [XAMPP](#)
- Umgebungsvariable setzen
- Batch File anlegen
  
- <https://www.youtube.com/watch?v=sd6MWc29G84>

# Installation WP-CLI macOS / Linux

---

- `$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar (oder wget)`
- `$ php wp-cli.phar --info`
- `$ chmod +x wp-cli.phar`
- `$ mv wp-cli.phar /usr/local/bin/wp`
- `$ wp --info`
- <https://wpbeaches.com/installing-wordpress-wp-cli-macos/>
- <https://www.linode.com/docs/websites/cms/install-wordpress-using-wp-cli-on-ubuntu-18-04/>

# WP-CLI ist vorinstalliert bei:

— — —

- Siteground
- Bluehost
- Cloudways
- Dreamhost
- Synthesis
- WordPress.com VIP
- Raidboxes
- Mittwald
- 1&1
- Volle Liste im WP-CLI Handbuch abrufbar: [Klicke hier](#)

# WP-CLI Verbindung

# WP-CLI Verbindung

---

- Die Verbindung erfolgt über das Secure Shell Protokoll (SSH)
- Fragt euren Webhoster um Hilfe
- Aufbau erfolgt über Kommandozeile (Mac, Linux)
- Windows unterstützt kein SSH über CMD, verwendet z.B. Putty, GitBash Cygwin oder OpenSSH

# Kommandos

<https://developer.wordpress.org/cli/commands/>



```
$ wp plugin install wordpress-seo --activate
```

# Globale Parameter

---

- `--prompt`
- `--quiet`
- `--path`

Kommando Argumente

- `--dry-run` für `search-replace`

# Hilfe

---

```
$ wp help
```

Öffnet die allgemeine Hilfe

```
$ wp help kommando
```

Öffnet die Hilfe für das jeweilige Kommando (z.B. plugin)

# Core

---

```
$ wp core download
```

Downloaded den WordPress Core

```
$ wp core version
```

Zeigt die aktuelle WordPress Version

```
$ wp core update
```

Aktualisiert den WordPress Core auf eine neue Version

# Core

---

```
$ wp core install
```

Startet den Standard WordPress Installationsprozess

```
$ wp core install --url=meineseite.server.com --title=title  
--admin_user=you --admin_password=Password1  
--admin_email=your@email.address
```

```
$ wp core install --prompt
```

Das identische Kommando ohne Parameter. WordPress fragt nun nach jedem Wert.

# Plugins

---

```
$ wp plugin install wordpress-seo --activate
```

Installiert Yoast SEO und aktiviert das Plugin.

Tipp: Für eine bestimmte Versionsnummer:

```
--version=version Nummer
```

# Plugins

---

```
$ wp plugin update wordpress-seo
```

Aktualisiert Yoast SEO

```
$ wp plugin update --all
```

Aktualisiert alle Plugins

# Themes

---

```
$ wp theme install TwentyTwenty --activate
```

Installiert & aktiviert das TwentyTwenty Theme

```
$ wp theme activate Astra
```

Aktiviert ein bereits installiertes Theme

```
$ wp theme update Astra
```

Aktualisiert ein Theme

# Themes

---

```
$ wp theme TwentyTwenty delete
```

Löscht ein Theme

# Search & Replace

```
---  
$ wp search-replace "alter string" "neuer string"
```

Durchsucht alle Zeilen nach einem String und ersetzt diesen

```
$ wp search-replace "dev.test" "meinewebseite.com"
```

Praktisch bei einer Datenbankmigration um die Site URL zu ändern

```
$ wp search-replace "dev.test" "meinewebseite.com" --dry-run
```

Tipp: Verwende zuerst immer `--dry-run`. Dies zeigt die Auswirkung ohne etwas in die Datenbank zu schreiben

# Wartungsmodus

---

```
$ wp maintenance-mode activate
```

Aktiviert den WordPress Wartungsmodus

```
$ wp maintenance-mode status
```

Zeigt den aktuellen Status

```
$ wp maintenance-mode deactivate
```

Deaktiviert diesen

# Cache

---

```
$ wp cache flush
```

Leert den WordPress Object Cache

# Medien

---

```
$ wp media import
```

Importiert eine Mediendatei

```
$ wp media regenerate
```

Regeneriert Thumbnails

```
$ seq 500 1000 | xargs wp media regenerate
```

Regeneriert Thumbnails der IDs 500 - 1000

# Benutzerverwaltung

---

```
$ wp user create
```

Erstellt einen neuen WordPress User und zeigt alle verfügbaren Parameter

```
$ wp user create --prompt
```

Tipp: Verwende den globalen Parameter `--prompt` um dir die genauen Parameter nicht merken zu müssen

# Benutzerverwaltung

---

```
$ wp user add-role 5 administrator
```

Weist dem Benutzer mit ID "5" Adminrechte zu

```
$ wp user get 5
```

Gibt Details über den User mit ID "5" aus

```
$ wp user delete 5 --reassign=1
```

Löscht den User mit ID "5" und weist alle Posts dem User mit ID "1" zu

# Posts

---

```
$ wp post generate
```

Generiert 100 Dummy Posts

```
$ wp post generate --count=10 --post_date=2020_02_15
```

Generiert 10 Posts mit Datum 2020-02-15

```
$ curl -N http://loripsum.net/api/5 | wp post generate  
--post_content --count=10
```

Generiert 10 Posts mit 5 Absätzen Lorem Ipsum Text

# Posts

---

```
$ wp post list
```

Zeigt alle posts

# Kommentare

---

```
$ wp comment approve 1
```

Genehmigt den Kommentar mit ID "1"

```
$ wp comment list
```

Zeigt eine Liste aller Kommentare

```
$ wp comment spam 1
```

Markiert den Kommentar mit ID "1" als Spam

# Menüs

---

```
$ wp menu create "Footer"
```

Erstellt ein Menü mit dem Namen "Footer"

```
$ wp menu location list
```

Zeigt alle Menüs

# Datenbank

---

```
$ wp db tables
```

Zeigt alle Datenbank-Tabellen

```
$ wp db check
```

Zeigt den aktuellen Status der Tabellen

```
$ wp db optimize
```

Startet mysqlcheck mit --optimize=true

# Datenbank

---

```
$ wp db repair
```

Startet mysqlcheck mit --repair=true

```
$ wp db clean --yes
```

Führt ein DROP\_TABLE aus

# Datenbank

---

```
$ wp db export
```

Exportiert die Datenbank als .sql File

```
$ wp db import datenbank.sql
```

Importiert die Datenbank

# Sprachen

---

```
$ wp language core list
```

Zeigt alle verfügbaren Sprachen

```
wp language core install es_ES --activate
```

Installiert Spanisch und aktiviert die Sprache

# Scaffold

---

```
$ wp scaffold child-theme --parent_theme=TwentyTwenty
```

Erstellt ein Child Theme auf Basis des aktuellen Themes

```
$ wp scaffold plugin mein-plugin
```

Erstellt Startcode für ein Plugin

```
$ wp scaffold block team --title="Team" --theme=TwentyTwenty
```

Erstellt einen neuen "Team" Gutenberg Block für das "TwentyTwenty" Theme

**WP-CLI ist erweiterbar!**  
**\$ wp package install**

**WP-CLI ist per Skripting automatisierbar!**

**[https://make.wordpress.org/cli/handbook/  
commands-cookbook/](https://make.wordpress.org/cli/handbook/commands-cookbook/)**

**Vielen Dank für die  
Aufmerksamkeit!**

**Fragen? [hello@marceltannich.com](mailto:hello@marceltannich.com)  
[@MarcelTannich](https://www.instagram.com/MarcelTannich)**