

PHP-Praktikum: Alternative Entwicklungsumgebungen

Da die virtuellen Linux-Server vorübergehend nicht erreichbar sind, können Sie PHP-Aufgaben mit einer der folgenden Alternativen bearbeiten.

Option 1: Laragon (Windows, empfohlen)

Laragon ist eine schlanke und schnelle Entwicklungsumgebung für Windows. Im Vergleich zu XAMPP startet Laragon deutlich schneller, benötigt weniger Ressourcen und lässt sich einfacher konfigurieren. Mehrere PHP-Versionen können parallel installiert und per Klick umgeschaltet werden.

Hinweis: Laragon ist nur für Windows verfügbar. macOS-Nutzerinnen und -Nutzer verwenden bitte Option 2 (MAMP).

Installation und Einrichtung

1. Laden Sie Laragon herunter: <https://laragon.org/download/> (Full-Version empfohlen, enthält Apache, PHP und MySQL).
2. Installieren Sie Laragon (Standardeinstellungen sind ausreichend).
3. Starten Sie Laragon. Im Hauptfenster klicken Sie auf **Start All**.
4. Legen Sie Ihre PHP-Dateien im Ordner `C:\laragon\www\` ab, z.B. in einem Unterordner `C:\laragon\www\praktikum\`.
5. Klicken Sie im Laragon-Fenster auf **www** oder öffnen Sie den Browser und rufen Sie `http://localhost/praktikum/` auf.

Tipp: Laragon erkennt neue Unterordner in `www\` automatisch und erstellt dafür eine lokale Domain (z.B. `http://praktikum.test`). Diese Funktion kann in den Einstellungen deaktiviert werden, falls sie nicht benötigt wird.

Option 2: XAMPP / MAMP (lokale Webserverumgebung)

XAMPP (Windows) und MAMP (macOS) installieren Apache, PHP und MySQL in einem Paket und funktionieren auf beiden Plattformen zuverlässig.

Windows (XAMPP)

1. Laden Sie XAMPP herunter: <https://www.apachefriends.org>
2. Installieren Sie XAMPP (Standardeinstellungen sind ausreichend).
3. Starten Sie das XAMPP Control Panel und klicken Sie bei **Apache** auf **Start**.

4. Legen Sie Ihre PHP-Dateien im Ordner `C:\xampp\htdocs\` ab, z.B. in einem Unterordner `C:\xampp\htdocs\praktikum\`.
5. Öffnen Sie den Browser und rufen Sie `http://localhost/praktikum/` auf.

macOS (MAMP)

1. Laden Sie MAMP herunter (kostenlose Version): <https://www.mamp.info>
2. Installieren Sie MAMP und starten Sie die Anwendung.
3. Klicken Sie auf **Start Servers**.
4. Legen Sie Ihre PHP-Dateien im Ordner `/Applications/MAMP/htdocs/` ab, z.B. in einem Unterordner `/Applications/MAMP/htdocs/praktikum/`.
5. Öffnen Sie den Browser und rufen Sie `http://localhost:8888/praktikum/` auf.

Hinweis: MAMP verwendet standardmäßig Port 8888. Den Port können Sie in den MAMP-Einstellungen auf 80 ändern, dann funktioniert auch `http://localhost/praktikum/`.

Option 3: PHP Built-in-Webserver

PHP bringt einen einfachen Webserver mit, der sich direkt aus dem Terminal starten lässt. Keine zusätzliche Software wie Apache oder MAMP nötig. Geeignet für Aufgaben ohne Datenbank.

Windows

Schritt 1: PHP installieren

1. Laden Sie PHP von <https://windows.php.net/download/> herunter (aktuelle "Thread Safe"-Version als ZIP).
2. Entpacken Sie das Archiv, z.B. nach `C:\php\`.
3. Fügen Sie `C:\php\` zur PATH-Umgebungsvariable hinzu:
 - Startmenü öffnen, nach "Umgebungsvariablen" suchen
 - "Systemumgebungsvariablen bearbeiten" öffnen
 - Unter "Benutzervariablen" die Variable `Path` auswählen und `C:\php\` hinzufügen
 - Mit OK bestätigen und die Eingabeaufforderung neu starten
4. Prüfen Sie die Installation mit:

```
php --version
```

Schritt 2: Server starten

1. Öffnen Sie die Eingabeaufforderung (`cmd`) oder PowerShell.

2. Navigieren Sie zum Ordner mit Ihren PHP-Dateien:

```
cd C:\Benutzer\IhrName\praktikum
```

3. Starten Sie den Server:

```
php -S localhost:8000
```

4. Öffnen Sie den Browser und rufen Sie `http://localhost:8000/` auf.

5. Den Server beenden Sie mit `Strg + C`.

macOS

Schritt 1: PHP installieren

Neuere macOS-Versionen (ab Monterey) enthalten kein PHP mehr. Installieren Sie PHP über Homebrew:

1. Falls Homebrew noch nicht installiert ist, führen Sie im Terminal aus:

```
bash  
  
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD"
```

2. PHP installieren:

```
bash  
  
brew install php
```

3. Prüfen Sie die Installation:

```
bash  
  
php --version
```

Schritt 2: Server starten

1. Öffnen Sie das Terminal.

2. Navigieren Sie zum Ordner mit Ihren PHP-Dateien:

```
bash
```

```
cd ~/praktikum
```

3. Starten Sie den Server:

```
bash  
  
php -S localhost:8000
```

4. Öffnen Sie den Browser und rufen Sie `http://localhost:8000/` auf.

5. Den Server beenden Sie mit `Ctrl + C`.

Option 4: VS Code mit PHP lokal

Wenn Sie bereits VS Code verwenden, können Sie direkt darin entwickeln. VS Code selbst führt PHP nicht aus, Sie benötigen dafür PHP auf Ihrem System (siehe Option 3 für die Installation). Der Built-in-Server aus Option 3 läuft dabei im integrierten Terminal von VS Code.

Einrichtung

1. Installieren Sie PHP wie in **Option 3** beschrieben.
2. Installieren Sie die Extension **PHP Intelephense** in VS Code:
 - Extensions-Panel öffnen (`Strg+Shift+X`) / (`Cmd+Shift+X`)
 - Nach "PHP Intelephense" suchen und installieren
 - Diese Extension bietet Autovervollständigung, Fehlererkennung und Navigation im Code.
3. Öffnen Sie Ihren Praktikumsordner in VS Code: `Datei > Ordner öffnen`.

Server aus VS Code heraus starten

1. Öffnen Sie das integrierte Terminal: `Terminal > Neues Terminal` (oder `Strg+``).
2. Starten Sie den PHP-Server direkt im Terminal:

```
bash  
  
php -S localhost:8000
```

3. Öffnen Sie `http://localhost:8000/` im Browser.

Tipp: Mit der Extension **PHP Debug** (benötigt Xdebug) können Sie Breakpoints setzen und den Code schrittweise debuggen. Für einfache Aufgaben ist das jedoch nicht nötig.

Option 5: Online-IDE (ohne Laptop / überall nutzbar)

Diese Lösung funktioniert vollständig im Browser und erfordert keine lokale Installation. Geeignet für Studierende ohne eigenen Laptop oder als schnelle Backup-Lösung.

Replit (empfohlen für den schnellen Einstieg)

1. Rufen Sie <https://replit.com> auf und legen Sie einen kostenlosen Account an.
2. Klicken Sie auf + **Create Repl**.
3. Wählen Sie als Template **PHP** aus.
4. Ein Editor öffnet sich mit einer Beispieldatei `index.php`.
5. Schreiben Sie Ihren PHP-Code in den Editor.
6. Klicken Sie auf **Run**: Replit startet automatisch einen Webserver und zeigt die Ausgabe im eingebetteten Browser-Fenster rechts an.
7. Weitere PHP-Dateien können Sie über das Datei-Panel links anlegen.

Hinweis zur Datensicherung: Kostenlose Replit-Projekte sind öffentlich sichtbar. Legen Sie keine Passwörter oder sensible Daten dort ab. Laden Sie Ihre fertigen Dateien am Ende herunter (Rechtsklick auf die Datei > Download).

GitHub Codespaces (für Studierende mit GitHub-Account)

Codespaces bietet eine vollständige Linux-Umgebung im Browser, die der ursprünglichen Serverumgebung am nächsten kommt.

1. Melden Sie sich bei <https://github.com> an.
2. Erstellen Sie ein neues Repository (z.B. `php-praktikum`) oder öffnen Sie ein bestehendes.
3. Klicken Sie auf den grünen Button **Code** > **Codespaces** > **Create codespace on main**.
4. Nach kurzer Ladezeit öffnet sich ein vollständiges VS-Code-Interface im Browser.
5. Öffnen Sie das Terminal im unteren Bereich und starten Sie PHP:

```
bash
```

```
php -S 0.0.0.0:8000
```

6. Codespaces erkennt den Port automatisch und fragt, ob er geöffnet werden soll. Bestätigen Sie mit **Open in Browser**.
7. Ihre PHP-Seite ist nun im Browser erreichbar.

Hinweis: GitHub Codespaces ist für Studierende im Rahmen des GitHub Student Developer Pack kostenlos nutzbar (<https://education.github.com/pack>).