PHP Command-line interpreter

START HERE

`php <file>`
Run a script

`php -a`
Interactive shell

`php -r '<code>'`
Run one-liner code

`php -v`
Show version

`php -l <file>`
Syntax check only

`php -m`
List extensions

`php -S localhost:8000`
Dev web server

`php --ini`
Which php.ini is loaded

php.ini main config → `scan dir` **conf.d/*.ini** per extension → `-d` **-d key=value** this run only

CONCEPTS

The CLI is its own build with its own php.ini: settings you change for FPM or Apache may not apply to php on the command line.

Later sources win: a -d flag overrides conf.d files, which override php.ini.

php --ini lists the files; -n skips all of them for a clean baseline.

RUN CODE

`php <file> [args]`
Run a script, args land in $argv

`php -r '<code>'`
Run code without <?php tags

`php -r '<code>' -- -x`
Pass args that look like options

`php -f <file>`
Parse and execute a file

`php < <file>`
Run a script from stdin

`php -d <key>=<value> <file>`
Override an ini setting for this run

`php -d memory_limit=-1 <file>`
Lift the memory cap for one run

`php -c <path> <file>`
Use another php.ini file or dir

`php -n <file>`
Ignore php.ini entirely

`php -w <file>`
Show source without comments

`php -s <file>`
Output HTML-highlighted source

CHECK AND INSPECT

`php -l <file>`
Lint: parse only, do not run
Catches syntax errors only, not runtime fatals such as an undefined function.

`php -v`
Show PHP version and SAPI

`php -m`
List loaded modules

`php -i`
Print phpinfo()

`php --ini`
Show loaded ini files

`php --ini=diff`
Show ini values changed from default
Needs PHP 8.5 or newer.

`php --ri <ext>`
Show one extension's settings

`php --ri main`
Show core settings

`php -h`
List all CLI options

BUILT-IN SERVER

`php -S localhost:8000`
Serve the current directory

`php -S localhost:8000 -t <dir>`
Serve a given docroot

`php -S 0.0.0.0:8000`
Reachable from other machines

`php -S localhost:8000 <router.php>`
Run a router script per request
Return false from the router to serve the requested file as-is.

`PHP_CLI_SERVER_WORKERS=4 php -S localhost:8000`
Handle requests in parallel
Single-threaded by default, so one slow request blocks the rest. Not on Windows.

BUILT-IN SERVER (cont.)

The built-in server is for development and demos only; use nginx with php-fpm in production.

Without a router it serves index.php or index.html for unknown paths, walking up to the docroot.

CONFIGURATION

`display_errors=1`
Show errors on screen (dev)

`error_reporting=E_ALL`
Report every error level

`memory_limit=512M`
Raise the memory cap

`max_execution_time=0`
No time limit (CLI default)

`date.timezone=UTC`
Silence timezone warnings

`opcache.enable_cli=1`
Enable the opcache for CLI runs
Off by default in the CLI; only helps long-running or repeated processes.

`php -d <key>=<value> --ri <ext>`
Verify a setting took effect

Put these lines in php.ini or a conf.d/*.ini file, or pass them as -d key=value on the command line.

REFLECTION

`php --rf <function>`
Show a function's signature

`php --rc <class>`
Show a class's methods and constants

`php --re <ext>`
Show an extension's functions and classes

`php --rz <zend-ext>`
Show a Zend extension's config

PROCESS INPUT LINES

`php -R '<code>'`
Run code per stdin line ($argn)

`php -B '<code>' -R '<code>' -E '<code>'`
Begin, per line, end

`php -F <file>`
Run a file for each stdin line

INTERACTIVE

`php -a`
Open the interactive shell
Needs the readline extension; PHP 8.1 and later refuse to start without it.

`#cli.pager=less`
Page long shell output

`#cli.prompt=<text>`
Change the shell prompt

In the shell write code without <?php, end each statement with ; and press Tab twice to list completions.

RECIPES

Lint every file in a tree

```
find src -name '*.php' -print0
| xargs -0 -n1 php -l
```

Quick data one-liners

```
php -r 'echo PHP_VERSION,
PHP_EOL;'
php -r 'echo json_encode(
array_slice(
get_loaded_extensions(), 0,
3));'
echo 'a b' | php -R 'echo
strtoupper($argn), PHP_EOL;'
```

Dev server with a front controller

```
php -S localhost:8000 -t public
public/index.php
# open http://localhost:8000
```

Debug a missing extension

```
php -m | grep -i <ext>
php --ini
php --ri <ext>
```