

PHP Development Professional

Prüfungscode IFW-PHP-300-01 · Fassung 1.0 · Stand 18.09.2026

PHP für anspruchsvolle Projekte: Objektorientierung und Typsystem, Fehlerbehandlung und Generatoren, Sicherheit sowie Architektur, Standards und Performance.

Die Prüfung im Überblick

Bearbeitungszeit 2 Stunden	Fragen 70
Bestehensgrenze 70 %	Fragetypen Einfach- und Mehrfachauswahl
Prüfungssprache Deutsch	Zertifikat gültig 36 Monate
Durchführung Online, zeitlich überwacht	Preis 99,00 €

Themenblöcke

Block	Thema	Gewichtung	Fragen
1	Objektorientierung, Typsystem und SPL	30 %	21
2	Fehlerbehandlung und Generatoren	20 %	14
3	Sicherheit	20 %	14
4	Architektur, Standards und Performance	30 %	21
Gesamt		100 %	70

Die Gewichtung gibt an, welchen Anteil ein Block an den Prüfungsfragen hat. Die Fragen werden beim Start zufällig und anteilig je Block aus dem Fragenpool gezogen.

Objektorientierung, Typsystem und SPL

Moderne Sprachmittel von PHP 8 gezielt einsetzen und ihr Verhalten in Randfällen vorhersagen.

- 1.1** Unveränderliche Objekte entwerfen
readonly-Eigenschaften, final-Klassen, Constructor Promotion
- 1.2** Methoden als Closures weitergeben
First-Class-Callable-Syntax `$this->methode(...)`, Sichtbarkeit beim späteren Aufruf
- 1.3** Statische Eigenschaften und Late Static Binding beurteilen
geteilter Zustand in Klassenhierarchien, `static::` gegenüber `self::`
- 1.4** Enums einsetzen
Fälle als Singletons, Konstanten, Interfaces und Traits; keine Eigenschaften, keine Vererbung
- 1.5** Varianz von Typen beurteilen
kontravariante Parametertypen und kovariante Rückgabetypen bei Interfaces
- 1.6** Objektreferenzen und SPL-Datenstrukturen nutzen
`WeakMap`, `SplObjectStorage`, `SplQueue`, `SplPriorityQueue`, `SplFixedArray`

Fehlerbehandlung und Generatoren

Den Kontrollfluss bei Ausnahmen und in Generatoren sicher vorhersagen – auch in Randfällen, in denen beides zusammenkommt.

- 2.1** Die Fehlerhierarchie von PHP 8 kennen
`Throwable`, `Exception`, `Error` und ihre Unterklassen
- 2.2** `finally` in Randfällen beurteilen
`return` in `finally` überschreibt Rückgabewerte und verwirft Exceptions
- 2.3** Generatoren einsetzen
`yield`, `yield from`, Schlüssel und Rückgabewerte
- 2.4** Ressourcen in Generatoren sauber freigeben
`try/finally` im Generator, vorzeitiger Abbruch der Iteration

Sicherheit

Angriffe erkennen, die an PHP-typischen Eigenheiten ansetzen, und wirksame Gegenmaßnahmen wählen.

- 3.1** Gefahren durch lose Vergleiche erkennen
numerische Strings wie `'0e...'`, `hash_equals()`, `password_verify()`

- 3.2** Passwörter und Signaturen sicher behandeln
password_hash(), password_needs_rehash(), hash_hmac(), zeitkonstante Vergleiche
- 3.3** Unsichere Deserialisierung vermeiden
unserialize() auf Benutzerdaten, Object Injection, json_decode() als Alternative
- 3.4** Sitzungen absichern
session.use_strict_mode, session_regenerate_id(), Cookie-Flags
- 3.5** Die Grenzen von Prepared Statements kennen
Platzhalter nur für Werte, nicht für Tabellen- oder Spaltennamen; Whitelists

BLOCK 4

30 % · 21 Fragen

Architektur, Standards und Performance

Anwendungen so strukturieren, dass sie testbar, austauschbar und schnell bleiben.

- 4.1** Autoloading nach PSR-4 einrichten
Namespace-Präfix, Verzeichnis und Dateiname, composer dump-autoload
- 4.2** PSR-Standards zuordnen
PSR-3 Logger, PSR-4 Autoloading, PSR-7 HTTP-Nachrichten, PSR-11 Container, PSR-15 Middleware
- 4.3** Abhängigkeiten injizieren
Konstruktor-Injektion gegen Interfaces statt new im Konstruktor, Testbarkeit
- 4.4** OPcache verstehen
vorkompilierter Bytecode im Shared Memory, opcache.validate_timestamps, Preloading

Institut für Webentwicklung · unabhängiges Lernangebot ohne Verbindung zu den genannten Technologieanbietern. Maßgeblich ist die Fassung auf der Zertifizierungsseite.