

## PL109 Perl-Programmierung (Unix/Linux)

### Kurzbeschreibung:

Die freie und plattformunabhängige Programmiersprache Perl ist mächtig, schnell, flexibel und rasch programmiert. Der Workshop **PL109 Perl-Programmierung (Unix/Linux)** führt Sie in die Grundlagen und die Programmierung mit Perl ein.

### Zielgruppe:

Das Training **Perl-Programmierung (Unix/Linux)** richtet sich an:

- System-Administratoren
- Datenbank-Administratoren
- Applikations-Administratoren
- Netzwerk-Administratoren

### Voraussetzungen:

Um den Lerntempo und Inhalten des Workshops **Perl-Programmierung (Unix/Linux)** gut zu folgen, benötigen Sie Kenntnisse auf Administrationsebene von Unix/Linux-Systemen.

Da beinahe die komplette Dokumentation in Englisch verfasst ist, sollten Sie als Perl-Programmierer über grundlegende Englisch-Sprachkenntnisse verfügen.

### Sonstiges:

**Dauer:** 5 Tage

**Preis:** 2490 Euro plus Mwst.

### Ziele:

Im Training **Perl-Programmierung (Unix/Linux)** lernen Sie, die Programmiersprache **PERL** überall dort sinnvoll einzusetzen, wo die klassischen, OpenSource-spezifischen Programmierertools weniger gut geeignet sind.

Perl ist eine freie und gut zu interpretierende Programmiersprache, die eine Vielzahl an Paradigmen unterstützt, und das völlig plattformunabhängig. Perl wurde in den 1980er Jahren vor allem als Mix aus C, awk und Unix-Kommandos entwickelt.

Die Vorteile von Perl sind u.a. effizientes Management, selbst bei komplexen Datenstrukturen durch die leistungsfähige Variablenstruktur, sehr hohe Geschwindigkeit in der Verarbeitung der Daten, ein mächtiges Pattern Matching und OS-Unabhängigkeit des Codes.

Nach dem Seminar PL109 Perl-Programmierung (Unix/Linux) werden Sie folgende Zeile verstehen:

```
# perl '-le(1x$)=~m|^(11+)|1+$|||print while++$<111'
```

## Inhalte/Agenda:

- - ◆ Grundlegendes zur Programmiersprache Perl
    - ◆ Die Perl-Kontexte: String, numerisch, boolean; Skalar- und Listenkontext
    - ◆ Datentypen, Dimensionen und Speicherstrukturen in Perl
    - ◆ Adressierung mehrdimensionaler Arrays und Hashes
    - ◆ Funktionen zu skalaren Variablen, Arrays und Hashes
    - ◆ Boole'sche, numerische und String-Operatoren
    - ◆ Bedingungen, Short Circuit und Ternärer Operator
    - ◆ Schleifen und Schleifenkontrolle
    - ◆ String- und arithmetische Funktionen
    - ◆ Pattern Matching
    - ◆ Translate und Substitue
    - ◆ Konvertierung von Datentypen
    - ◆ Subroutinen, Parameter-Arrays, Return-Werte
    - ◆ Namespaces, Variablen-Scope
    - ◆ Interaktion mit dem System: Dateien und System-Aufrufe, Environment
    - ◆ Module über require und use nachladen
    - ◆ Debugging in Grundzügen
    - ◆ Ausblick: CPAN
- ◆