

Fragen zur Wiederholung:

- Was macht `chomp`?
→ entfernt den `Input_Record_Separator` (\$/) → normalerweise `\n`
- Was ist der Unterschied zwischen `>` (redirecting Operator) und `|` (Pipe)?
→ `>` leitet die Standardausgabe weiter z.B. in Datei, `|` leitet die Ausgabe eines Befehls an einen anderen Befehl weiter
- Was ist der Unterschied zwischen `$` und `@` bei Variablen?
→ `$` für Skalare (Zahlen, Strings, etc.), `@` für Listen
- Was für Arten von Schleifen gibt es?
→ `while`, `do while`, `do until`, `for`, `foreach`
- Was macht die `while`-Schleife?
→ wiederholt so lange wie eine Bedingung wahr ist, prüft vor der Ausführung
- Was macht die `do while` Schleife?
→ wiederholt so lange wie eine Bedingung wahr ist, prüft erst nach der Ausführung
- Was macht die `do until` Schleife?
→ wiederholt so lange wie eine Bedingung falsch ist, prüft erst nach der Ausführung
- Was macht die `foreach` Schleife?
→ wiederholt für jedes Element einer Liste
- Was macht die `for` Schleife?
→ wiederholt mit speziellen Anweisungen
- Schreibe ein Programm das bis 10 zählt für jede Schleife, `$i = 0`:
→ `while($i <= 10) { print $i; $i++; }`
→ `do { print $i; $i++; } while ($i <= 10);`
→ `do { print $i; $i++; } until ($i > 10);`
→ `@liste` mit Zahlen 0 bis 10: `foreach $element (@liste) { print $element; }`
→ `for ($i = 0; $i <= 10; $i++) { print $i; }`
- Wie kann ich eine Bedingung prüfen?
→ `if`, `else` Anweisung
- Wie greife ich auf das erste Element von `@liste` zu?
→ `$liste[0]`
- Wie greife ich auf das letzte Element von `@liste` zu, wenn `@liste` 5 Elemente hat?
→ `$liste[4]`

- Wie kann ich den Inhalt von \$name mit „Hallo ...“ auf der Konsole ausgeben?
→ `print „Hallo $name \n“;`
- Wie vergleiche ich zwei Zahlen auf gleich und ungleich?
→ `$zahl1 == | != $zahl2`
- Wie vergleiche ich zwei Strings auf gleich und ungleich?
→ `$name1 ne | eq $name2`
- Welche fünf Fehler hat folgendes Programm?

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm mit Fehlern

use strict
{

$name1 = "Nicola";
print "Bitte geben Sie einen Namen ein:>>>\n";
$name2 = <>;

if ($name1 == $name2) {
    print "Du heißt genau wie ich!\n";
}

else {
    print "Du hast aber einen tollen Namen!\n";
}
```

- 1) kein Strichpunkt nach `use strict`;
- 2) geschweifte Klammer geht nicht wieder zu
- 3) kein Chomp für \$name2 → wird nie gleich sein
- 4) falscher Vergleichsoperator
- 5) Anführungszeichen gehen nicht wieder zu