

# Entwurf eines einfachen Programms mit Perl (Getränkeautomat)

## Was soll das Programm tun?

1. Getränkeangebot und Preise zeigen
2. Getränkeauswahl des Kunden entgegennehmen und prüfen
3. Zahlung verlangen
4. Münzen entgegennehmen, prüfen, Restbetrag anzeigen
5. Abbruch mit Geldrückgabe ermöglichen
6. Getränk ausgeben
7. Rückgeld berechnen und ausgeben

## 1. Wie sag ich's meinem Computer?

**Computer verstehen keine natürlichen Sprachen.**

**Also brauchen wir spezielle Programmiersprachen.**

In unserem Beispiel:

**Perl**

Jext (Editor)

Beide sind Freeware.

Literatur: Schröter, Jürgen: Perl. Grundlagen und effektive Strategien.  
München, Wien: Oldenbourg 2003

## 2. Variablennamen, Wertzuweisung

**Variablennamen** beginnen mit \$,  
unterscheiden Groß- und Kleinschreibung  
z.B. \$Betrag, \$Getraenk; \$Betrag und \$betrag sind verschieden.

### **Wertzuweisung**

Variablenname1 = Variablenname2; oder Variablenname1 = Konstante;

Numerische Variable (Konstante ohne Anführungszeichen):

`$Betrag = 100;`

Nichtnum. / String- / Textvariable (Konstante mit Anführungszeichen):

`$Getraenk = "Erdbeermilch";`

Befehle werden mit einem Strichpunkt ; abgeschlossen.

### 3. Eingabe, Ausgabe

```
print "Eingabe: ";  
$Getraenk = <STDIN>; # Standard-Input  
chomp ($Getraenk);   # schneidet NewLine ab
```

```
print Konstante, Variablenname, "\n" (neue Zeile)  
print "Zu zahlen: ", $Betrag, "\n";
```

## 4. Ablaufsteuerung – Schleife

while (Bedingung) {Schleifenkörper}: abweisend, kopfgesteuert

```
while ($Ende != 9)
{
  print "Beenden? Ende = 9 ";
  $Ende = <STDIN>;
  chomp ($Ende);
} # endwhile ($Ende != 9)
```

## 5. Vergleichsoperatoren

Numerisch: ==, !=, <, <=, >, >=  
String: eq, ne, lt, le, gt, ge

## Verknüpfung von Vergleichsbedingungen

and

or

## 6. Ablaufsteuerung – Abfragen

### Einseitige Abfrage

```
if (Bedingung)  
    {Ja-Zweig}
```

### Zweiseitige Abfrage

```
if (Bedingung)  
    {Ja-Zweig}  
else  
    {Nein-Zweig}
```

## 6. Ablaufsteuerung – Fallunterscheidung

```
if (Fall1-Bedingung)
    {Fall1-Zweig}
else {if (Fall2-Bedingung)      # Fall2
    {Fall2-Zweig}
else {if(Fall3-Bedingung)      # Fall3
    {Fall3-Zweig}
else {if (Fall4-Bedingung)     # Fall4
    {Fall4-Zweig}
else
    {SonstigerFall-Zweig}
}                               # Fall4
}                               # Fall3
}                               # Fall2
```