



2. Variablen-Konzept

Programmiersprachen kommen ohne Variablen nicht aus. Variablen sind "Platzhalter" für Werte.

2.1 Vorstellung

Man stelle sich eine Schachtel vor, etwa eine Zigarrenschachtel. Diese habe außen auf dem Deckel einen eindeutigen Namen stehen. In der Zigarrenschachtel liegt ein Zettel, auf dem etwas steht. Genau so funktioniert eine Variable. Sie hat einen eindeutigen Namen (Bezeichner) und einen Wert.

2.2 Variablen-Deklaration in PHP

In vielen Programmiersprachen muss man eine Variable deklarieren, d.h. bevor wir mit ihr arbeiten können müssen wir festlegen, von welchem Typ (Zahl oder Wort) die Variable ist. Nicht so in PHP.

Man erkennt Variablen in PHP an dem "\$"-Zeichen.

Ohne Deklaration kann man sofort eine Variable verwenden, indem man ihr mit dem "="-Zeichen einen Wert zuweist.

Beispiele:

```
$MeinName="Friedrich";  
$MeinAlter=34;  
$Irendeinname="Müller";
```

Merke: Zahlen haben keine Anführungs- und Schlusszeichen. Alle Worte schon.

Der Bezeichner der Variable kann beliebig gewählt werden. Um nicht allzu viel schreiben zu müssen verwendet man oft nur kurze Worte, z.B. \$i, \$b, \$B.

Beachte, dass Groß- und Kleinschreibung unterschieden wird.

2.3 Wertzuweisung

In obigen Beispielen ist als \$MeinName der Bezeichner und "Friedrich" der Wert der Variablen.

Im Programmverlauf kann der Wert einer Variablen verändert werden, indem ihr einfach ein neuer Wert zugewiesen wird, auch hier wieder mit dem "="-Zeichen. Der Zettel in der Schachtel wird also ersetzt durch einen neuen Zettel mit einem neuen Wert.



Beispiele:

```
$Name="Meier";  
$Name="Huber";
```

Der Wert der Variablen Huber ist also "Huber". Die erste Zuweisung geht durch die zweite verloren.

Welchen Wert hat die Variable \$i am Ende des Programms?

```
$i = 10;  
$i = 15;  
$i = $i + 10;  
$i = $i * 10;
```

An diesem Beispiel sieht man sehr schön, dass das **"="-Zeichen** kein **"ist gleich"-Zeichen** ist, sondern eine **Wertzuweisung** darstellt. Man sollte besser sagen, \$i "erhält den neuen Wert". Der Rechner arbeitet dabei immer von rechts nach links, d.h. er kennt denn alten Wert von \$i noch, bevor er ihn neu vergibt. So ergibt $\$i = \$i + 10$ keine Fehlermeldung. Der alte Wert von \$i von 15 wird um 10 erhöht und dann erst der Variablen \$i zugeordnet.

Wir haben also am Ende des Programms eine Variable \$i mit dem Wert von 250. In unserer "Vorstellung" wurde am Anfang eine Schachtel mit \$i bezeichnet und der Zettel darin hatte den Wert 10, dann den Wert 15, dann 25 und zuletzt 250.

2.4 Verwendung von Variablen

Variablen lassen sich verwenden, um den Inhalt auszugeben oder mit ihnen zu rechnen. Sie stellen Platzhalter dar.

Beispiel:

```
$Name="Hans Mueller";  
$Klasse="6c";  
$Note1 = 1; $Note2 = 4; $Note3 = 1;
```

```
$Durchschnittsnote = ($Note1 + $Note2 + $Note3) / 3;
```

```
echo "Hallo $Name, du besuchst die Klasse $Klasse. ";  
echo "Die Durchschnittsnote in den Kernfächern betraegt $Durchschnittsnote";
```



2.5 Konkatenation

Darunter versteht man das Zusammensetzen von Zeichenketten.

Beispiel:

```
$Vorname="Hans";
```

```
$Nachname="Müller";
```

Beide Variablen sollen zu eine Variablen \$Name verbunden werden. Dabei benötigen wir noch ein Leerzeichen zwischen den beiden Bestandteilen:

```
$Name=$Vorname." ".$Nachname.
```

In vielen Programmiersprachen verwendet man ein "+"-Zeichen, in PHP den "." als Konkatenations-Operator.