

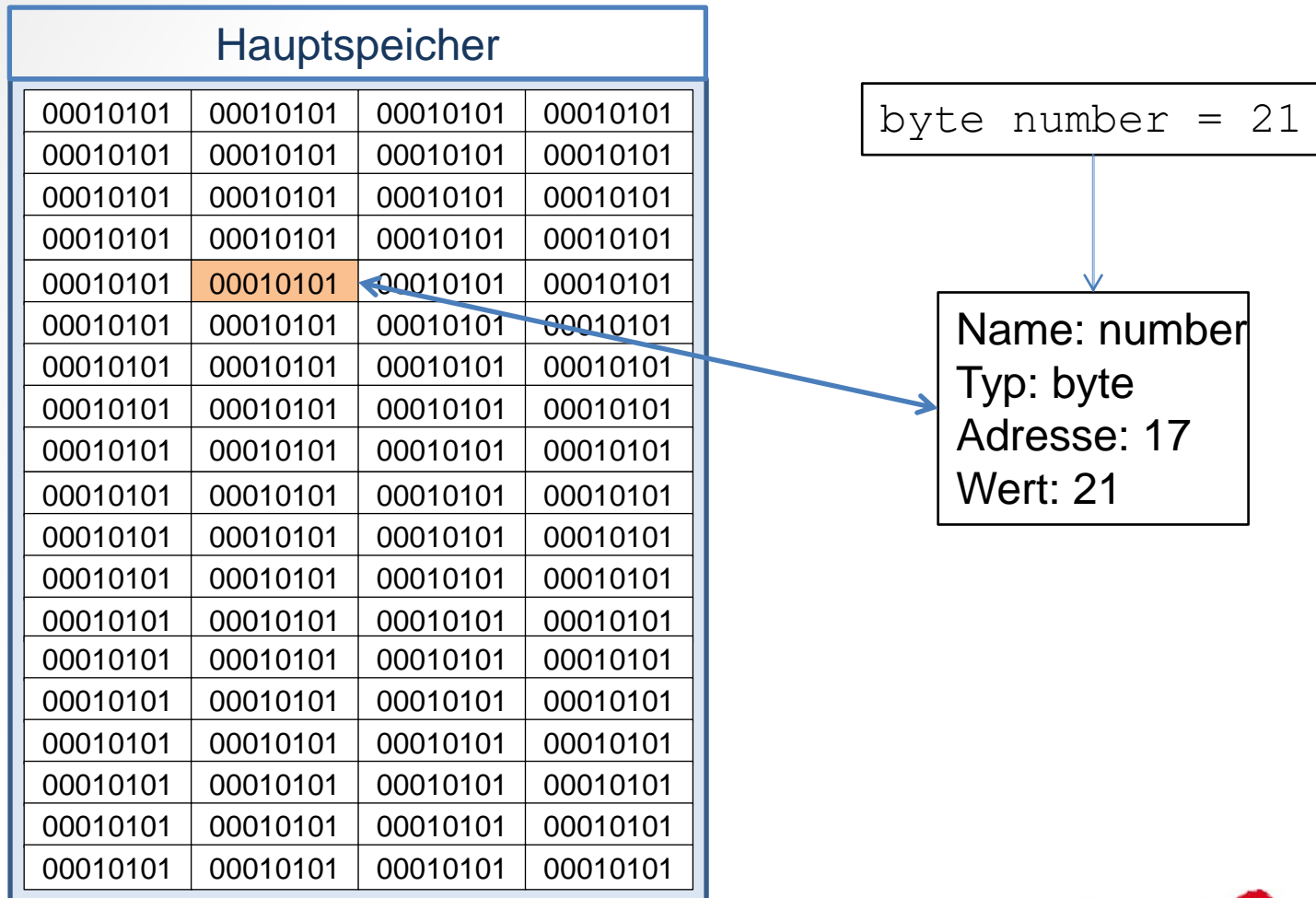
Variablen - Motivation

- Unser bisheriges „Programm“ hat bei jedem Aufruf immer „Hallo Welt“ ausgegeben.
- Wir wollen den Benutzer auffordern, seinen Namen einzugeben. Der Name soll anschließend ausgegeben werden.
- Der Benutzer soll uns den Vornamen, Nachnamen und sein Alter in Jahren mitteilen.
All diese Werte wollen wir uns im Programm merken und wieder ausgeben.
- Der Benutzer sagt uns die Kantenlänge eines Quadrates. Das Programm berechnet und merkt sich den Umfang und den Flächeninhalt. Diese Werte werden anschließend ausgegeben.

Das Programm muss sich Dinge merken können!
Die Lösung dafür sind Variablen.

Variablen

Daten, die im Programm mehrmals benötigt werden, können als Variablen gespeichert werden.



Deklaration

Bevor eine Variable verwendet werden kann muss diese bekanntgegeben (=deklariert) werden.

Die Variable heißt
„vorname“ und ist vom
Datentyp „string“
(Zeichenkette)

```
string vorname;
```

Die Variable heißt „tage“
und ist vom Datentyp
„int“ (Ganzzahl)

```
int tage;
```

Deklaration mit Initialisierung

Bei der Deklaration einer Variable kann auch gleichzeitig ein Wert festgelegt werden:

```
string nachname = "Mustermann";  
string wohnort = Console.ReadLine();  
int tage = 9;  
int minuten = 24 * 60;
```

Alle Variablen müssen vor der Benutzung initialisiert werden (aber nicht zwangsläufig sofort bei der Deklaration)

Datentypen - Übersicht

- `string` Texte aus einem oder mehreren Zeichen
- `char` einzelnes Zeichen
- `int` Ganzzahl zwischen
 -2.147.483.648 und
 2.147.483.647
- `bool` Wahrheitswert `true` und `false`
- `double` Gleitkommazahl

Einfache Datentypen in C#

Schlüsselwort	abgebildet auf	Wertebereich
sbyte	System.SByte	-128 .. 127
short	System.Int16	-32768 .. 32767
int	System.Int32	-2 147 483 648 .. 2 147 483 647
long	System.Int64	$-2^{63} .. 2^{63}-1$
byte	System.Byte	0 .. 255
ushort	System.UInt16	0 .. 65535
uint	System.UInt32	0 .. 4294967295
ulong	System.UInt64	$0 .. 2^{64}-1$
float	System.Single	$\pm 1.5E-45 .. \pm 3.4E38$ (32 Bit)
double	System.Double	$\pm 5E-324 .. \pm 1.7E308$ (64 Bit)
decimal	System.Decimal	$\pm 1E-28 .. \pm 7.9E28$ (128 Bit)
bool	System.Boolean	true, false
char	System.Char	Unicode-Zeichen

Verwendung (Evaluation)

Der aktuelle Wert einer Variablen kann jederzeit bestimmt werden.

```
Console.WriteLine(vorname);  
Console.WriteLine(tage + 7);
```

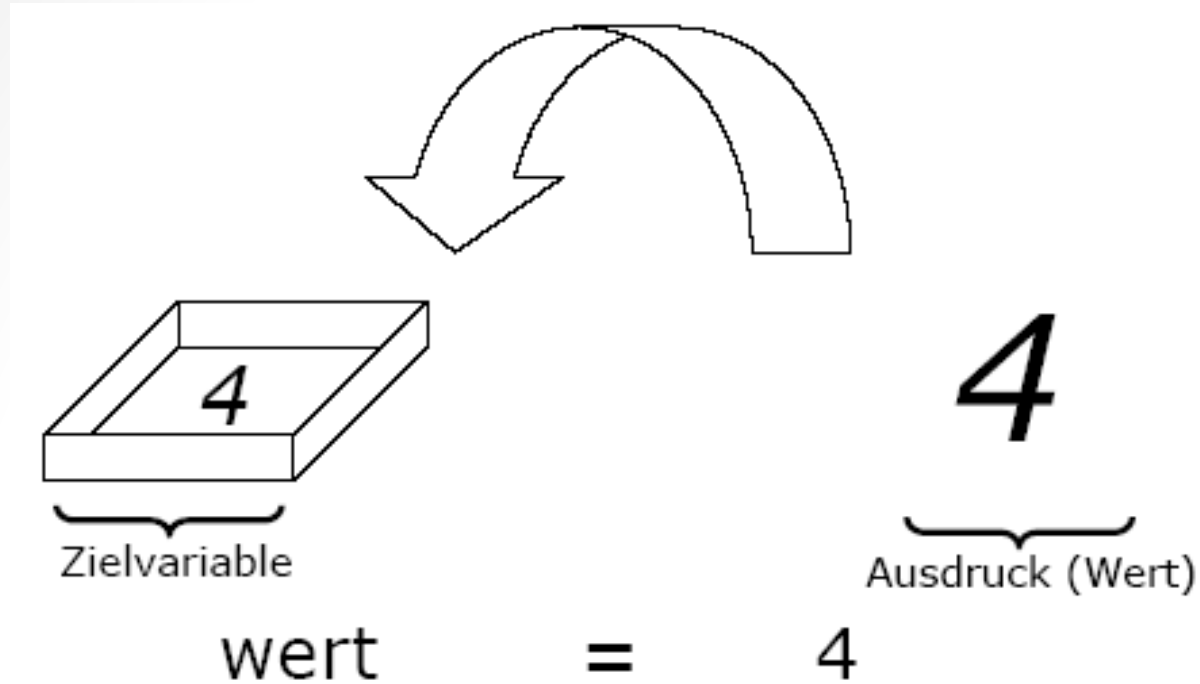
Zuweisung (Assignment)

An einer beliebigen Stellen im Programm kann der Wert einer Variable verändert werden

```
vorname = "Max";  
tage = 7;  
nachname = Console.ReadLine();
```

- Eine Zuweisung erfolgt durch das Zeichen =
- Links vom = steht der Variablenname.
- Rechts steht der neue Wert (=Term).

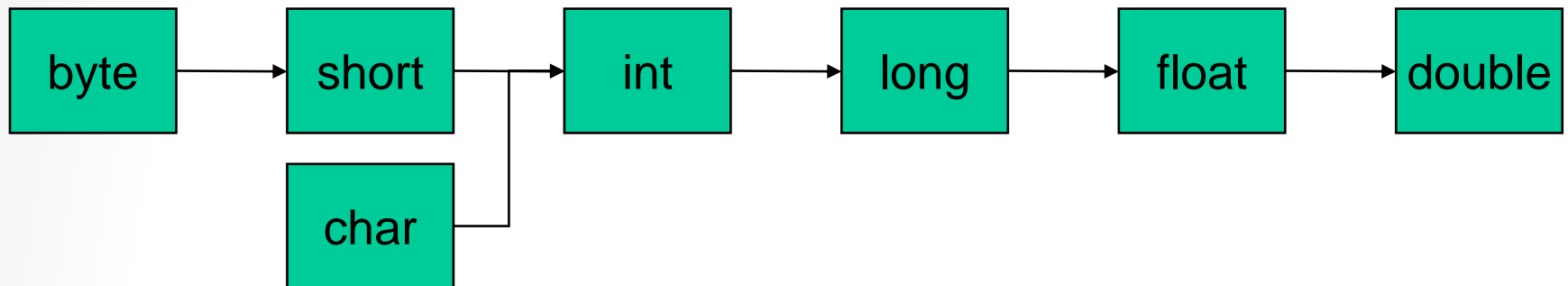
Zuweisungsoperator



Zuweisungsoperator

Vorsicht

- Zugewiesener Wert muss zum Typ der Variablen kompatibel sein (im verfügbaren Speicher Platz haben)
 - Klein in Groß funktioniert, aber nicht umgekehrt!
 - Zuweisung kann aber erzwungen werden



Variablen - Beispiele

```
string message = "Important text";
```

```
int errorCount = 42;
```

```
bool isValid = true;
```

Punkt oder Komma?

```
double piValue = 3.141592653;
```

```
string firstName = "Hansi";
```

```
string lastName = "Hinterseer";
```

```
Console.WriteLine($"{firstName} {lastName}");
```

```
int summand1 = 17;
```

```
int summand2 = 42;
```

```
int sum = summand1 + summand2;
```

```
Console.WriteLine($"Die Summe beträgt {sum}");
```

```
int count = 0;
```

```
count = count + 1;
```

Bezeichner – Namen von Variablen

- Kombination aus Zeichen, Ziffern, `_`, `@`
- Case-sensitiv
- Schlüsselwörter dürfen nicht verwendet werden
 - Siehe <https://msdn.microsoft.com/de-de/library/x53a06bb%28v=vs.120%29.aspx>
- Richtlinien für die Namenswahl:
 - Sprechende Bezeichner verwenden – der Name beschreibt, was in der Variablen gespeichert ist!
 - Nur englische Namen verwenden
 - „CamelCase“
 - Beginnen mit kleinem Anfangsbuchstaben
 - Jedes innere Teilwort groß beginnen
 - Bsp: *myVariableInCamelCaseNotation*



Konstanten

- Schlüsselwort: **const**
- Verwendung:
 - Fixe Werte, die sich während des Programmablaufs nie ändern, und im Programm mehrmals benötigt werden.
- Richtlinien:
 - Immer groß schreiben
 - Am Beginn des Programmcodes sammeln
- Beispiele:

```
const int GROWN_UP_AGE = 19;  
const float TAX_RATE = 20.0;
```