

## Ein Wort in seine Buchstaben aufteilen und diese in einem Array speichern

Von Cedric Wetzel

Oft stehen die lokalen Variablen direkt am Anfang einer Methode. In diesem Eintrag werden speziell die Initialisierung sowie Deklaration erläutert.

```
static void Main(string[] args)
{
}
```

Hier haben wir die leere Main-Methode. Möchten wir beispielsweise jeden Buchstaben eines beliebigen Wortes aus einem String in einem Array speichern, so müssen logischerweise vorher an jeden Buchstaben kommen.

Fangen wir an:

Zu allererst muss überlegt werden, welche Variablen benötigt werden. Da der Benutzer etwas eingeben soll, wird also ein String benötigt. Wir nennen die Variable einfach mal *wort*. Selbstverständlich kommt noch das Array dazu. Ich nenne es *buchstabe*.

Soweit so gut.

Jetzt können die Variablen initialisiert werden.

```
static void Main(string[] args)
{
    string wort;
    string[] buchstabe;
}
```

Normalerweise initialisiert man ein Array am besten direkt mit voller Länge, heißt, dass Variablentyp sowie Arraylänge gesetzt werden.

Dies sieht folgendermaßen aus:

```
string[] buchstabe = new string[8];
```

*string[]* beschreibt dabei den Datentyp (Array, welches mit String-Elementen gefüllt wird)

*buchstabe* ist der Variablenname

*new* ist der Allokationsoperator (reserviert den Speicher)

*string[8]* für ein String-Array mit acht freien Plätzen.

An dieser Stelle wird klar, dass wir noch nicht wissen wie groß das vom Benutzer eingegebene Wort sein wird, also fällt der letzte Teil nach dem Gleichheitszeichen weg.

Im nächsten Schritt wird das vom Benutzer eingegebene Wort der Variable *wort* zugewiesen.

```
static void Main(string[] args)
{
    string wort;
    string[] buchstabe;
    wort = Console.ReadLine();
}
```

Nun, da wir wissen wie lang *wort* ist, können wir dem Array *buchstabe* eine eindeutige Länge zuweisen. Dies passiert logischerweise wie folgt:

```
buchstabe = new string[word.Length];
```

Es folgt eine for-Schleife, welche so oft wiederholt wird, wie die Variable *wort* lang ist.

```
static void Main(string[] args)
{
    string wort;
    string[] buchstabe;

    wort = Console.ReadLine();
    buchstabe = new string[wort.Length];

    for (int i = 0; i < wort.Length; i++)
    {
    }
}
```

Da *i* null ist, wird bis 24 gezählt, also insgesamt 25 Schleifendurchläufe.

Die for-Schleife muss folglich noch mit Inhalt gefüllt werden, und zwar werden dem Array endlich Werte zugewiesen (deklariert).

```
for (int i = 0; i < wort.Length; i++)
{
    buchstabe[i] = Convert.ToString(wort[i]);
}
```

*buchstabe[i]* ist die Array-Variable *buchstabe* mit der Indexierungsnummer *i*.

*wort[i]* ist die deklarierte String-Variablen mit der Indexierungsnummer *i* (auch Strings können direkt per Indexierungsnummer angesprochen werden).

Warum wird die doch ohnehin schon vom Datentyp String stammende Variable *wort* in String konvertiert?

In C# wird bei der Auswahl einzelner Buchstaben eines Strings dieser gleichzeitig in einen Charakter umgewandelt. Weil das Array *buchstabe* aber nur mit Strings deklariert werden kann, muss dieser vorher wieder in einen String konvertiert werden. Das hat den Grund, da in folgenden Artikeln das Rechnen mit diesen Variablen so einfach wie möglich machbar sein soll.

Alles verstanden?

```
static void Main(string[] args)
{
    string wort;
    string[] buchstabe;
    wort = Console.ReadLine();
    buchstabe = new string[wort.Length];

    for (int i = 0; i < wort.Length; i++)
    {
        buchstabe[i] = Convert.ToString(wort[i]);
    }
}
```