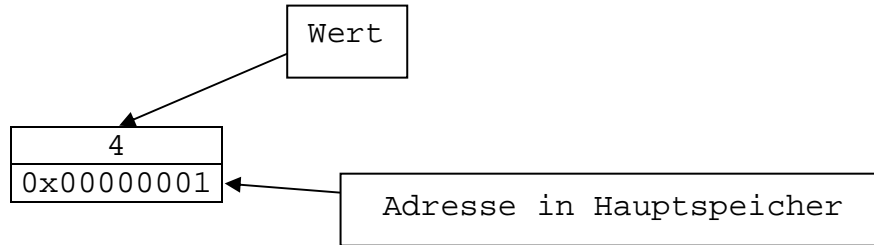


## int

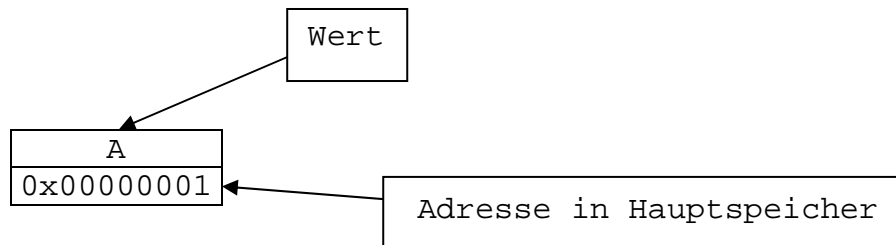
```
int zahl;  
zahl = 4;
```



`printf("%d", zahl);` → Gibt Wert von `zahl` auf `stdout`(=Bildschirm) aus. → **4**  
`scanf("%d", &zahl);` → Speichert mir beliebiges **int** von `stdin`(=Tastatur) in `zahl`.

## char

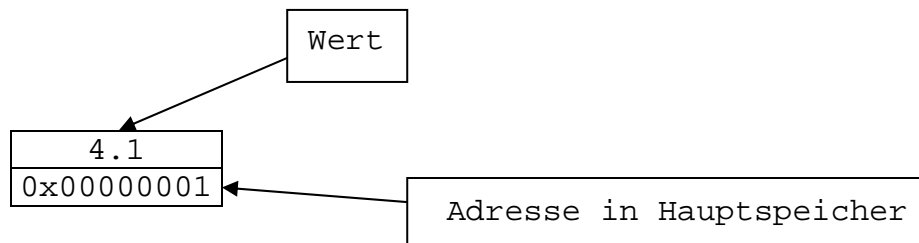
```
char zeichen;  
zeichen = 'A';
```



`printf("%c", zeichen);` → Gibt Wert von `zeichen` auf `stdout`(=Bildschirm) aus. → **A**  
`scanf("%c", &zeichen);` → Speichert mir beliebiges **char** von `stdin`(=Tastatur) in `zeichen`.

## float

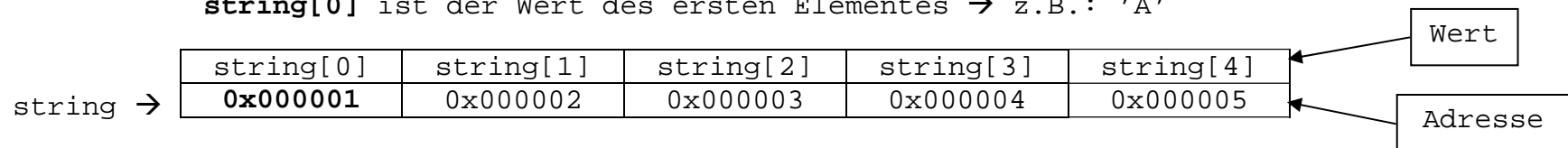
```
float kommaZahl;  
kommaZahl = 4.1;
```



printf("%f", kommaZahl); → Gibt Wert von *kommaZahl* auf *stdout*(=Bildschirm) aus. → 4.1  
 scanf("%f", & kommaZahl); → Speichert mir beliebiges **float** von *stdin*(=Tastatur) in *kommaZahl*.

## char Arrays - 1 Dimension

**char string[5];** **string** zeigt auf die Startadresse von **char string[5]** → **0x000001**  
**string[0]** ist der Wert des ersten Elementes → z.B.: 'A'



**string[5]** kann 4 chars speichern, also von **string[0]** bis **string[3]** adressiert werden.  
**string[4]** ist für '\0' reserviert !

```
void einlesenStringEinzeln()
{
    char string[5];

    for ( int i = 0; i < 4; i++ ) // Zeilenindex
    {
        printf( "Geben Sie ein Zeichen ein !\n" );
        fflush(stdin);
        scanf( "%c", &string[i] ); // Lese einzelne Zeichen ein.
    }
    string[i] = '\0';
}
```

```
void einlesenStringGesamt()
{
    char string[5];

    printf( "Geben Sie maximal 4 Zeichen ein !\n" );
    scanf( "%s", string ); // Lese den String auf einmal ein.
}
```

## char Arrays - 2 Dimensional (Matrix)

`char stringListe[3][5];` `stringListe[0]` zeigt auf die Startadresse der ersten **Zeile**.

<code>stringListe[0]</code> →	<code>stringListe[0][0]</code> 0x000001	<code>stringListe[0][1]</code> 0x000002	<code>stringListe[0][2]</code> 0x000003	<code>stringListe[0][3]</code> 0x000004	<code>stringListe[0][4]</code> 0x000005
<code>stringListe[1]</code> →	<code>stringListe[1][0]</code> 0x000006	<code>stringListe[1][1]</code> 0x000007	<code>stringListe[1][2]</code> 0x000008	<code>stringListe[1][3]</code> 0x000009	...
<code>stringListe[2]</code> →	<code>stringListe[2][0]</code> 0x0000011	<code>stringListe[2][1]</code> 0x0000012	<code>stringListe[2][2]</code> 0x0000013	....	...

`void einlesenStringListe()` // liest 3 Strings(=charArrrays) ein und speichert sie in **stringListe**.

```
{
  char string[5]; // temporäre, lokale Variable.

  for ( int i = 0; i < 3; i++ ) // i ist Zeilenindex.
  {
    printf( "Geben Sie einen String ein !\n" );
    scanf( "%s", string ); // Lese den String in die Startadresse von string.
    strcpy( stringListe[i], string ); // Kopiere den String von string in die Startadresse von
                                     stringListe[i].
  }
}
```

Es wird 3 mal "abcd" über `stdin`(=Tastatur) eingegeben.

a	b	c	d	\0
a	b	c	d	\0
a	b	c	d	\0

→ letztes Byte muss immer \0 sein, um den String abzuschließen - das macht `strcpy()` automatisch.

```

void printStringListe() // gibt den Inhalt von stringListe zeilenweise aus.
{
    printf( "\n-----\n Folgende Matrix wurde eingegeben ! \n" );
    for ( int i = 0; i < 3; i++ )
    {
        printf( "%s", stringListe[i] ); // es wird zeilenweise ausgegeben !
        printf( "\n" );
    }
}

```

**Ausgabe am Bildschirm nach Aufruf von:** `einlesenStringListe();`  
`printStringListe();`

```

Geben Sie einen String ein !
abcd
Geben Sie einen String ein !
abcd
Geben Sie einen String ein !
abcd

```

```

-----
Folgende Matrix wurde eingegeben !
abcd
abcd
abcd

```

## Beliebter Fehler bei Eingabe:

Es wird 1 mal "abcde" über stdin eingegeben - also ein Zeichen zuviel und 2 mal "abcd" eingegeben.

a	b	c	d	e
a	b	c	d	\0
a	b	c	d	\0

## Ausgabe am Bildschirm:

```
Geben Sie einen String ein !
abcde
Geben Sie einen String ein !
abcd
Geben Sie einen String ein !
abcd
```

-----  
Folgende Matrix wurde eingegeben !  
abcdeabcd  
abcd  
abcd

Es wurde der erste String nicht mit \0 abgeschlossen da an der Stelle s[4] ein **e** reingeschrieben wurde.  
Der String **string** wurde ohne abschließendes \0 mittels **strcpy()** in stringListe[0] kopiert.  
Bei der Ausgabe wird zeilenweise aus stringListe (stringListe[0], stringListe[1]...) der jeweilige String ausgegeben.  
Da die erste Zeile aber kein \0 hat, geht er weiter bis ans Ende der nächsten Zeile.