

### Speicher reservieren

Reservieren ohne clearen (Null setzen)

```
void *malloc(int Groesse);
```

Beispiel

```
double *t = malloc(2 * sizeof(double));  
double *v = malloc(2 * sizeof(*v));
```

Reservieren mit clearen (reservierte Bereiche mit 0 initialisieren)

```
void *calloc(int Anzahl_Elements, int Groesse_eines_Elements);
```

Beispiel

```
double *u = calloc(2, sizeof(double));  
double *w = calloc(2, sizeof(*w));
```

### Reservierten Speicher vergrößern / verkleinern

Speicher reallozieren

```
void realloc(void AlterSpeicherbereich, int NeueGroesse);
```

Beispiel

```
int pArray = malloc(5 * sizeof(int));  
pArray = realloc(pArray, 1000 * sizeof(int))
```

Ist der neue Speicherbereich größer, werden die bisherigen Daten in den neuen Speicherbereich kopiert.

Ist der neue Speicherbereich kleiner, werden die Daten nur bis zur Größe des neuen Bereichs kopiert.

Wird als neue Größe eine 0 angegeben, funktioniert die Funktion wie free();

Wird als alter Speicherbereich der NULL-Zeiger angegeben, funktioniert die Funktion wie malloc();

### Reservierten Speicher freigeben

Funktion

```
free(SPEICHER);
```

Beispiel

```
void *z = malloc(1000); //1000Bytes Speicher reservieren  
if (z != NULL)  
{  
    free(z);  
}
```

Die free Funktion darf nur Speicherbereiche freigeben, die zuvor mit malloc oder calloc reserviert wurden.

free(NULL); bewirkt nichts und verursacht keinen Fehler;



By **TimSch**

[cheatography.com/timsch/](https://cheatography.com/timsch/)

Published 18th March, 2018.

Last updated 18th March, 2018.

Page 1 of 1.

Sponsored by **Readability-Score.com**

Measure your website readability!

<https://readability-score.com>