

Speicher reservieren

Reservieren ohne clearen (Null setzen)	<code>void *malloc(int Groesse);</code>
Beispiel	<code>double *t = malloc(2 * sizeof(double));</code> <code>double *v = malloc(2 * sizeof(*v));</code>
Reservieren mit clearen (reservierte Bereiche mit 0 initialisieren)	<code>void *calloc(int Anzahl_Elemente, int Groesse_eines_Elements);</code>
Beispiel	<code>double *u = calloc(2, sizeof(double));</code> <code>double *w = calloc(2, sizeof(*w));</code>

Reservierten Speicher vergrößern / verkleinern

Speicher reallozieren	<code>void *realloc(void *AlterSpeicherbereich, int NeueGroesse);</code>
Beispiel	<code>int pArray = malloc(5 * sizeof(int));</code> <code>pArray = realloc(pArray, 1000 * sizeof(int))</code>

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	<code>free(SPEICHER);</code>
Beispiel	<code>void *z = malloc(1000); //1000 Bytes Speicher reservieren</code> <code>if(z != NULL)</code> <code>{</code> <code> free(z);</code> <code>}</code>

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

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

