

Merkblatt zur Programmierung in C

Mathematik-Bibliotheksfunktionen in <math.h>

Alle Argumente `x`, `y` und `n` sowie die Rückgabewerte der Mathematik-Funktionen sind in C vom Datentyp `double`.

Funktion	Beschreibung	Wertebereich
Trigonometrische Funktionen (Argument im Bogenmaß)		
<code>sin(x)</code>	Berechnet Sinus von <code>x</code> , Prototyp <code>double sin(double);</code>	- 1.0 ... 1.0
<code>cos(x)</code>	Cosinus von <code>x</code> , Prototyp <code>double cos(double);</code>	- 1.0 ... 1.0
<code>tan(x)</code>	Tangens von <code>x</code>	$-\infty \dots \infty$
<code>asin(x)</code>	Arcus Sinus von <code>x</code> (Umkehrfunktion zu <code>sin</code>).	$-\pi/2 \dots \pi/2$
<code>acos(x)</code>	Arcus Cosinus von <code>x</code> .	$0 \dots \pi$
<code>atan(x)</code>	Arcus Tangens von <code>x</code> .	$-\pi/2 \dots \pi/2$
<code>atan2(y, x)</code>	Arcus Tangens von <code>x/y</code> . Zuordnung im richtigen Quadranten	$-\pi \dots \pi$
Hyperbolische Funktionen		
<code>sinh(x)</code>	Sinus hyperbolicus von <code>x</code>	
<code>cosh(x)</code>	Cosinus hyperbolicus von <code>x</code>	
<code>tanh(x)</code>	Tangens hyperbolicus von <code>x</code>	
Exponential-Funktionen		
<code>exp(x)</code>	Exponentialfunktion von <code>x</code> , also e^x	
<code>log(x)</code>	Natürlicher Logarithmus von <code>x</code> , also $\ln(x)$ zur Basis e	
<code>log10(x)</code>	Dekadischer Logarithmus von <code>x</code>	
Potenzfunktionen		
<code>pow(x,n)</code>	Potenz von <code>x</code> hoch <code>n</code> , also x^n	
<code>sqrt(x)</code>	Quadratwurzel von <code>x</code>	
Rundungsfunktionen		
<code>ceil(x)</code>	Aufrunden zum nächsthöheren ganzzahligen Wert	
<code>floor(x)</code>	Abrunden zum nächsten kleineren ganzzahligen Wert	
<code>fabs(x)</code>	Absolutwert einer Zahl <code>x</code> bilden	
<code>fmod(x, y)</code>	Divisionsrest von <code>x/y</code> (vgl. Modulo Operator <code>%</code> für ganze Zahlen).	

Standard-Bibliotheksfunktionen in <stdlib.h>

Umwandlungsfunktionen (ermöglichen keine Kontrolle darüber, ob eine Umwandlung erfolgreich war!)		
<code>i = atoi(s)</code>	Zeichenkette <code>char *s</code> (oder <code>s[...]</code>) in Integerzahl wandeln (alpha to int)	<code>int i;</code>
<code>l = atol(s)</code>	Zeichenkette <code>char *s</code> in Long-Integerzahl umwandeln (alpha to long)	<code>long l;</code>
<code>x = atof(s)</code>	Zeichenkette <code>char *s</code> in Fließkommazahl umwandeln (alpha to float)	<code>double x;</code>
Zufallszahlen		
<code>i = rand()</code>	liefert eine Pseudo-Zufallszahl zwischen 0 und RAND_MAX (Wertebereich: 0 ... 32767)	
<code>srand(unsigned int)</code>	Initialisierung (seed) des Pseudo-Zufallszahlengenerators. Z.B. mit: <code>srand(time(NULL));</code>	

String-Bibliotheksfunktionen in <string.h>

String-Funktionen (String-Ende muss immer das <code>'\0'</code> Zeichen sein)		Parameter Def.
<code>len = strlen(s)</code>	Ermittelt die Länge der Zeichenkette <code>s</code> (Anzahl der Zeichen bis <code>'\0'</code>)	<code>int len;</code>
<code>strcat(dst, src)</code>	Quell-Zeichenkette <code>src</code> an Ziel-Zeichenkette <code>dst</code> anhängen	<code>char *src;</code>
<code>strncat(dst, src, n)</code>	Dto., wobei nur <code>n</code> Zeichen angehängt werden	<code>char *dst;</code>
<code>strcpy(dst, src)</code>	Quell-Zeichenkette <code>src</code> in Ziel-Zeichenkette <code>dst</code> kopieren	<code>int n;</code>
<code>strncpy(dst, src, n)</code>	Dto., wobei nur <code>n</code> Zeichen kopiert werden	<code>char *s1, *s2;</code>
<code>diff = strcmp(s1, s2)</code>	Lexigraphische Diff. zwischen <code>s1</code> und <code>s2</code> (< 0, = 0 (gleich) oder > 0)	<code>int diff;</code>
<code>diff = strncmp(s1, s2, n)</code>	Dto., wobei nur <code>n</code> Zeichen berücksichtigt werden	
<code>s = strstr(dst, src)</code>	Liefert Zeiger auf erstes Vorkommen von String <code>src</code> in String <code>dst</code>	<code>char *s;</code>