

C

Dateihandhabung

<code>.c</code>	Sourcefile (Code)
<code>.o</code>	Objectfile (Compiliert)

Grundprogramm

<code>#include ...</code>	Import von Bibliotheken (header files)	<code>#include <stdio.h></code> <code>#include "myheader.h"</code>	Standard-Bibliothek persönliches File
<code>#define ...</code>	Konstanten	<code>#define NUMBER 123</code>	
<code>int ...</code>	globale/statische Variablen		
<code>main()</code> { } }	main / Hauptfunktion lokale Variablen		

Kommentar

<code># Kommentar</code>	einzeiliger Kommentar
<code>/* Kommentar */</code>	mehrzeiliger Kommentar

Operanden

<code>+</code>	<code>-</code>	<code>*</code>	Grundoperationen (Klammer setzen)	<code>2+5 // =7</code>	<code>5-2 // =3</code>	<code>2*5 // =10</code>
<code>%</code>			Modulo (Rest einer Division)	<code>10%3 // 1</code>		
<code>/</code>			Ganzzahldivision bei Ganzzahl normale Division bei Fließzahl	<code>10/6 // 1</code> <code>10.0/6 // 1.666666</code>		
<code>==</code>	<code>!=</code>	<code>=</code>	Vergleich, Ungleich, Zuweisung	<code>1 == 2 // false</code>	<code>1 != 2 // true</code>	<code>x = 1</code>
<code><</code>	<code>></code>	<code><=</code>	<code>>=</code>	kleiner, grösser, kleiner gleich, grösser gleich <code>2<4</code> <code>4>3</code> <code>2<=2</code> <code>3>=2</code>		
<code>&&</code>	<code> </code>	<code>!</code>	<code>^</code>	Negation, AND, OR (! vor && vor), XOR <code>x && y;</code> <code>x y;</code> <code>x ! y</code>		
<code>++</code>	<code>--</code>	<code>~</code>	Inkrement, Dekrement, Einerkomplement	<code>1++ // 2</code>	<code>5-- // 4</code>	<code>~(0010) // 1101</code>
<code>+=</code>	<code>-=</code>	<code>*=</code>	<code>/=</code>	<code>%=</code>	Operation mit direkter Zuweisung <code>x+=5 // x=x+5</code>	

Variablen, Typen, Deklarationen

	Datentyp	Bits
Zeichen	'a' <code>char</code>	8 (ASCII)
Ganzzahlen	<code>short</code>	16
	<code>short int</code>	
	<code>int</code> (→ Standard)	32
	<code>long</code>	64
Gleitkomma	<code>long int</code>	
	<code>float</code>	32
	<code>long float</code>	32
	<code>double</code> (→ Standard)	64

Deklaration	<code>int i,j;</code> <code>unsigned int k;</code>
Deklaration und Initialisation	<code>int i = 0;</code> <code>char ch = 'c';</code> <code>double d = 5.0;</code>
Zuweisung	<code>b = 2;</code> <code>b = a;</code>
Abfrage	<code>b;</code>
externe Variable	<code>extern int i;</code>

Cast Operatoren

<code>int</code> in <code>byte</code>	<code>b = (byte) i;</code>
<code>int + short</code> in <code>double</code>	<code>d = (double) (i + s);</code>
<code>char</code> in <code>int</code>	<code>i = (int) ch;</code>

Output / Input

Eingabe	<code>int getchar(void);</code>	<code>scanf ("%2d", &var_a);</code>	int wert von Tastatur
	<code>prog <infile</code>		int wert von File
Ausgabe	<code>int putchar(int);</code>		int wert zum Bildschirm
	<code>prog >outfile</code>		int wert zum File
	<code>printf("%d %6.1f\n", fahr, celsius);</code>		Formatierte Ausgabe

printf-Befehl

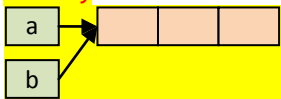
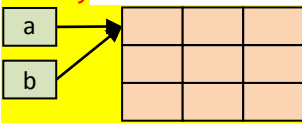
<code>%d</code>	<code>%3d</code>	dezimale ganze Zahl	+ Breite der Anzeige						
<code>%f</code>	<code>%.2f</code>	<code>%3.1f</code>	Gleitpunktzahl + Nachkommastellen +Breite der Anzeige						
<code>%o</code>	<code>%x</code>	<code>%c</code>	<code>%s</code>	<code>%%</code>	oktale	hexadezimale	Zeichen	Zeichenkette	selbst %

Steuerzeichen

<code>\a</code>	<code>\n</code>	<code>\t</code>	<code>\b</code>	<code>\"</code>	<code>\\</code>	Ton	Zeilentrenner	Tabulator	Backspace	Doppelanführungszeichen	Gegenschrägstrich
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----	---------------	-----------	-----------	-------------------------	-------------------

Variablen, Array

	0D	1D	2D
	<code>a</code> <code>1</code>	<code>a</code> <code>a[0]</code> <code>2</code> <code>a</code> <code>a[1]</code> <code>3</code> <code>a</code> <code>a[2]</code> <code>5</code>	<code>a</code> <code>a[0]</code> <code>2</code> <code>a[][0]</code> <code>a[][1]</code> <code>a[][2]</code> <code>a</code> <code>a[1]</code> <code>4</code> <code>8</code> <code>1</code>
	<code>b</code> <code>2</code>	<code>a.length;</code> <code>->3</code>	<code>a.length;</code> <code>->2</code> <code>a[0].length;</code> <code>->3</code>

Deklaration	<code>int a,b;</code> <code>int a = 1;</code>	<code>int a[3];</code> <code>static int a[] = {2,3,5};</code> (nur bei static und extern)	<code>int a[2][3];</code> <code>static int a[2][3] = {{2,3,5},{4,8,1}};</code> (nur bei static und extern)
Zuordnung	<code>b = 2;</code>	<code>a[1] = 3;</code>	<code>a[1][2] = 1;</code>
Abfrage	<code>b; ->2</code>	<code>a[2]; ->5</code>	<code>a[0][2]; ->5</code>
Zuweisung	<code>a = b;</code>	<code>a = b;</code> 	<code>a = b;</code> 
Speicherproblem			

Pointer

```
char arrayname[5];
char *ptr;
for (ptr = arrayname; ptr <= arrayname+4; ptr++)
{
    *ptr = 0;
}
```

Arrayüberschreitung	gibt keinen Fehler aus. Überschreibt einfach falsche Zelle
----------------------------	------------------------------------------------------------

Ein/Ausgabe

5.0 = Gleitpunktkonstante

Post-Inkrement	<code>int a = i++;</code>	<code>//a = i;</code>
Pre-Inkrement	<code>int a = ++i;</code>	<code>//a = 1+i;</code>

Bedingungen & Schleifen

If (else)	<code>if (something_is_true)</code> { <code>/* do one thing */;</code> } <code>else /* optional part */</code> { <code>/* do something else */;</code> }	<code>? (something_is_true) :</code> <code>/* do one thing */;</code> <code>:</code> <code>/* do something else */;</code>	<u>Wenn Bedingung wahr ist</u>
(do) while	<code>while (condition)</code> { <code>/* do something */;</code> }	<code>do</code> { <code>/* do something */;</code> } <code>while (condition)</code>	<u>Durchlauf sofern wahr</u> do = mindestens 1 Durchlauf
switch	<code>switch (integer value)</code> { <code>case 1: do something */;</code> <code>break; /* optional line */</code> <code>case 2: do something */;</code> <code>break; /* optional line */</code> <code>default: default statement;</code> <code>break; /* optional line */</code> }		<u>Unterscheidung eines int-Wertes</u> <code>break</code> beendet case <code>default</code> falls kein obiges getroffen hat
for	<code>for (statement1; condition; statement2)</code> { <code>/* do something */</code> }		<u>Zählschleife</u> <code>statement1: int i=1</code> <code>statement2: i=i+1</code>

<http://www.iu.hio.no/~mark/CTutorial/CTutorial.html>