Cheatography

C- Funktionen Cheat Sheet

by TimSch via cheatography.com/29480/cs/8667/

Allgemeine Begriffe

Begriff	Beschreibung
Funktionsdeklaration	Bekanntmachung des Namens der Funktion. `Rückgabetyp Funktionsname(Datantyp [Parametername], ...);`

Funktionsdefinition	Belegung eines Speicherbereichs
	`Rückgabetyp Funktionsname (Datentyp Parametername, ...)`
	`{`
	`Anweisungen;`
	`}`

Rückgabewert	*Funktionsergebnis* Wird innerhalb der Funktion mit `return Wert;` angegeben. Die Funktion wird damit beendet.
Funktionsparameter	Daten (Variablen), die zur Verarbeitung an die Funktion übergeben werden können. *Eine Funktion kann beliebig viele Parameter haben.*
Übergabe per Wert	Wert wird in den Stack kopiert. Wird er verändert, bleibt das Original unberührt.
Übergabe per Zeiger	Der Zeiger wird im Stack abgelegt. Über diesen Zeiger wird auf das eigentliche Objekt zugegriffen und damit auch verändert.
Rekursiver Funktionsaufruf	Der Aufruf einer Funktion durch sich selbst.
Zeiger auf Funktionen	`double (*Fkt) (double);` Zeiger auf Funktion, die ein double als Parameter erhält und ein double zurückgibt.

Beispiel	`#include <math.h>`
	`double (*TrigFkt) (double); //Zeiger auf Funktion, die double als Parameter erhält und double zurückgibt`
	`TrigFkt = sin;`

Beispiele

Parameter

```
int i = 5;
void funktion (int i) {
printf("%i", i);
}

//Ausgabe: 5
```

Beispiele (cont)

Rückgabewert

```
int i;
int funktion(int);

int main(){
i = funktion(5);
printf("%i", i);
}

int funktion (int i) {
return i += 5;
}

//Ausgabe: 10
```

Rekursiver Funktionsaufruf

```
int funktion(int);

void main(){
i = funktion(5);
printf("%i", i);
}

int funktion (int i) {
int p = i;
p += 5;
if (p < 15)
   p = funktion(p);
return p;
}

//Ausgabe: 15
```

Die Funktion main()

einfache Form

```
`int main()
  return 0; //Exit Code
}
```

komplexere Form

```
int main (int argc, char* argv[])
{
return 0; //Exit Code
}
```

Beispielaufruf komplexere Form

```
./prog 1 Test

argv[0] = "./prog"
argv[1] = "1"
argv[2] = "Test"
```