

### Öffnen und schließen

Zeiger auf Datenstrom	FILE *Datei;
Datei öffnen	Datei = fopen("C:\\DATEI.txt", "rb");
Prüfen, ob Datei erfolgreich geöffnet wurde	if (Datei != NULL){ ... }
Datei schließen	fclose(Datei);

### Öffn Modi

Lesemodus (read)	"r"
Schreibmodus (write)	"w"
Anhängen (append)	"a"
Binärmodus (binary)	"b"
Textmodus (text)	"t"

### Kombination von Öffn Modi

binäres Lesen	"rb"
Anhängen an Textdatei	"at"
Lesen und Schreiben einer Textdatei	"rwt"

### Ausgabe in Dateien

Formatierte Ausgabe	int fprintf(FILE stream, char format[, argumnts]);
Ausgabe	stream: Datenstrom format: Zeichenkette arguments: Argumente

Beispiel

```
for (i = 0; i < 10; i++)
{
    fprintf(Datei, " %i \n", Zahl[i]);
}
```

unformatiertes Schreiben	int fputc(int c, FILE *stream);
zeichenspezifisches Schreiben	c: zu schreibendes Zeichen stream: der Datenstrom

Beispiel

```
fputc('z', Datei);
```

Wenn das Funktionsergebnis von fgetc == EOF, dann Fehler!

Wenn das Funktionsergebnis von fscanf == EOF, dann Fehler oder Dateiende

### Einlesen von Dateien

Zeilenweises Einlesen	int fscanf(FILE stream, char format[, argum]);
-----------------------	--

Beispiel

```
FILE *Quelle;
int i, Zahl[10];
char* fname = "U: \\TE ST.T XT ";
```

```
Quelle = fopen(fname, "r");
if(Quelle != NULL){
    for(i = 0; i < 10; i++){
        fscanf(Quelle, "%i", &zahl[i]);
    }
    fclose(Quelle);
```

Zeichenspezifisches Einlesen	int fgetc(FILE *stream);
Rückgabewert: eingelesenes Zeichen	

Wenn Funktionsergebnis == EOF, dann Fehler oder Dateiende!