

- Typische Nutzung von Zeigern:
Nachbildung von Referenzparametern an Funktionen

```
int getDb1(char *str, double *fp) {  
    char dummy = '\0';  
    double lfp = 0.0; Parse-Ergebnis  
    if( sscanf(str, "%lf%c", &lfp, &dummy) == 1 ) {  
        *fp = lfp; Nötig da d. Zeiger beschrieben wird, selbst  
        return 1; bei erfolglosem Aufruf (weiter chars nach  
    } else { Fließkommazahl)  
        return 0;  
    }  
}
```

```
void foo(char *numstr, ...) {  
    ...  
    double num = 0.0;  
    if( getDb1(numstr, &num) ) {  
        ...  
    }  
    ...  
}
```

- Parameter `double *fp`:**
Aufrufer kann einen Zeiger auf eine lokale Variable übergeben (hier: `num` in `foo()`)
- Im Rumpf von `getDb1()` kann der Wert der Variablen des Aufrufers geändert werden (per `*fp`)
 - Zuweisungen auf `fp` (z.B. `fp = NULL`) haben lediglich lokale Wirkung in `getDb1()`
 - Zuweisungen auf `*fp` (z.B. `*fp = 42.0`) hingegen wirken auf den Zustand im aufrufenden Programmcode (hier auf die Variable `num` in `foo()`)