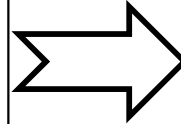


- Implizit **Zeigerarithmetik** für den Zugriff auf Array-Elemente
  - `int arr[4]`: `arr[0]` liefert **lvalue** zum ersten Element, `arr[1]` zum zweiten usw.
  - Index-Operator `[]` aber nur für Zeiger-Werte definiert – wie funktioniert das?
  - Im Rahmen eines Ausdrucks wird ein Array-Bezeichner wie `arr` (Typ `int[4]`) **implizit** in einen Zeiger auf das erste Array-Element konvertiert (hier: Typ `int*`)

```
int arr[4] = { 0 };  
for(i = 0 ; i < 4 ; i++) {  
    fprintf(stdout, "%d ", arr[i]);  
}
```



```
int arr[4] = { 0 };  
for(i = 0 ; i < 4 ; i++) {  
    fprintf(stdout, "%d ", *(arr'+i));  
    /* arr': int*, Zeiger auf erstes  
       Element von arr */  
}
```

```
int arr[4] = { 0 };  
mit sizeof(int) == 4
```

