

# Zeiger Cheat Sheet

## 01 Anlegen und Zuweisen

**Datentyp (Integer)**  
**Variablenname**  
`int i = 5;` → Wert  
**Datentyp (Integer-Zeiger)**  
**Variablenname**  
`int* pi = NULL;` → Wert

- Beim Anlegen einer Variablen entscheidet der Datentyp, ob es sich um einen Zeiger handelt.
- Dazu wird dem Datentyp (bspw. `int`) ein `*` angehängt.
- Ein leerer Zeiger wird explizit mit dem Wert `NULL` definiert.

**Variablenname**  
`i = 5;` → Wert  
`pi = &i;` → Wert  
**Adressoperator**  
**Variablenname**

- Bei der Wertzuweisung an einen Zeiger, wird lediglich der Variablenname benötigt.
- Einem Zeiger kann nur eine Adresse zugewiesen werden, bspw. mit Hilfe des `&` Operators.

## 02 Zugriff auf Zeiger

`int i = 5;`  
`int* pi = &i;`  
`*pi = 7;`  
**Dereferenzierungsoperator**

- Ein gültiger Zeiger zeigt stets auf einen Bereich im Speicher.
- Für den Zugriff auf den Wert auf die die Zeigervariable zeigt, wird der `*` Operator vorangestellt.

`int j = *pi + 8;`  
**Dereferenzierungsoperator**  
`int* pj = pi;`

- Durch den `*` Operator wird der Wert auf den `pi` zeigt verwendet.
- Eine Zuweisung ist immer möglich, wenn beide Seiten des `=` den identischen Typ haben.

Zugriff	Beispielwert
<code>i</code>	5
<code>*pj</code>	15

Zugriff	Beispielwert
<code>pi</code>	0x7FFEE76B1B70
<code>pj</code>	0x7FFEE76B1B70

## 03 Arrays und Zeiger

**Integer-Array**  
`int iValues[4];`  
`iValues[0] = 5;`  
`iValues[1] = 7;`  
**Index-Operator**

- Die Deklaration eines Arrays erfolgt durch die Typ- und Größenangabe (hier: 4 Integer).
- Mit Hilfe des Index-Operators kann auf die Array-Elemente zugegriffen werden. Das erste Element befindet sich an Index 0.

`*iValues = 5;`  
**Dereferenzierungsoperator**  
`*(iValues + 1) = 7;`  
**Dereferenzierungsoperator**

- Die Array-Elemente liegen im Speicher hintereinander.
- Der Variablenname ist immer ein Zeiger auf das erste Element.
- Mit `+1` wird der Zeiger um 4 Byte (Größe `int`) verschoben.

Array	Zeiger	Beispieladresse
<code>iValues[0]</code>	<code>*iValues</code>	0x7FFEE76B1B70
<code>iValues[0]</code>	<code>*(iValues + 0)</code>	0x7FFEE76B1B70
<code>iValues[1]</code>	<code>*(iValues + 1)</code>	0x7FFEE76B1B74

## 04 Zeiger auf Zeiger

`int i = 5;`  
`int* pi = &i;`  
**Adressoperator**  
`int** ppi = &pi;`

- Ein Zeiger kann auf einen weiteren Zeiger im Speicher verweisen.
- Die Adresse eines Zeigers kann ebenfalls mit dem `&` Operator ermittelt werden.
- Zeiger auf Zeiger Deklaration: `**`.

`int iMatrix[4][4];`  
`iMatrix[0][0] = 5;`  
`**iMatrix = 0;`

- Für mehrdimensionale Arrays sind Zeiger auf Zeiger erforderlich.
- Index- oder Dereferenzierung für den Elementzugriff.