

Kurzübersicht

Deklaration und Definition	<code>int *zeiger;</code> <i>Deklaration eines Zeigers zeiger auf einen int</i>
Speicheradresse einer Variable in Zeiger speichern	<code>int i;</code> <code>int *ip;</code> <code>ip = &i;</code>
Auf den Wert (nicht die Speicheradresse) eines Zeigers zugreifen	<code>int i;</code> <code>int *ip;</code> <code>ip = *i;</code>
Zeigerwert NULL	In <i>stdio.h</i> definiert. Wird genutzt, um zu vermeiden, dass ein Zeiger irgendwohin zeigt. <i>NULL ist ein Zeiger!</i>
unveränderbare Zeiger	Werden wie Variablen mit dem Schlüsselwort <code>const</code> definiert.
Zeiger auf Zeiger	Zeiger können auch auf andere Zeiger zeigen. <code>int Wert = 1234;</code> <code>int *p1 = &Wert;</code> <code>int **p2 = &p1;</code>

Einen Zeiger auf einen Zeiger erstellt man mit dem doppelten Variablenoperator `**`

Zeiger Operatoren

Operator	Funktion
<code>&</code>	Adressoperator Greift auf die Speicheradresse der Variablen, auf die der Zeiger zeigt zu.
<code>*</code>	Variablenoperator Greift auf den Wert der Variablen, auf die der Zeiger zeigt, zu.

Zeiger und Arrays

Array	Zeiger	Adresse	Inhalt
<code>IntArray[0]</code>	<code>*IntZeiger</code>	00004711h	5
<code>IntArray[1]</code>	<code>*(IntZeiger + 1)</code>	00004715h	4

```
int *IntZeiger = NULL;
int IntArray[5];
```

```
IntZeiger = IntArray;
IntArray[0] = 5;
IntArray[1] = 4;
*IntZeiger = 5;
```

Da Zeichenketten ein Spezialfall von Arrays sind, gelten diese Regeln dort ebenfalls.

Zeiger-Arithmetik Regeln

Zeiger können mit ganzen Zahlen addiert und subtrahiert werden.

Zwei Zeiger **gleichen Typs** lassen sich subtrahieren.

`ip1 = 1 - ip2` wäre verboten, da 1 vom Typ `int` und `ip2` vom Typ `int *` ist.

Alle anderen Operationen, wie Multiplikation oder Division usw. sind nicht erlaubt.

Es wird nicht in Bytes, sondern in Anzahl von Elementen des Datentyps, auf den der Zeiger zeigt, gerechnet.

Wird 1 zu einem Zeiger auf `int` addiert, zeigt er nicht auf die nächste Speicheradresse, sondern auf den nächsten `int`.

Zeiger-Arithmetik Beispiele

```
int i1 = 5, i2 = 7;
int ip1 = &i1, ip2 = &i2;
```

Operation erlaubt?

`ip2 = ip1 + 1;` ja - Zeiger und ganze Zahl

`ip1 = ip2 - 1;` ja - Zeiger und ganze Zahl

`ip2 = 1 + ip1;` ja - Zeiger und ganze Zahl

`ip1 = 1 - ip2;` nein - es können nur Zeiger gleichen Typs subtrahiert werden

`i1 = ip1 + ip2;` nein - keine ganze Zahl, sondern zwei Zeiger

`i2 = ip2 - ip1;` ja - zwei Zeiger vom gleichen Typ

Übersicht unveränderbare Zeiger

```
Definition int i1 = 5
           int const i2 = 3;
           int* ip1 = &i1; //veränderbarer Zeiger auf
           int const * ip2 = &i2; //veränderbarer Zeiger
           int* const ip3 = &i1; //unveränderbarer Zeiger
           int const * const ip4 = &i2;
           *(ip1++); //erlaubt
           (*ip1)++ //erlaubt
           *(ip2++); //erlaubt
           (*ip2)++; //Fehler - unveränderbare Variable
           *(ip3++); //Fehler - unveränderbarer Zeiger
           (*ip3)++ //erlaubt
           *(ip4++); //Fehler - unveränderbarer Zeiger
           (*ip4)++; //Fehler - unveränderbare Variable
```



By **TimSch**
cheatography.com/timsch/

Published 24th July, 2016.
Last updated 17th March, 2018.
Page 1 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>