

Wie wird in C der Ausdruck "1 + 2 << 3" ausgewertet und wie in Pascal der entsprechende Ausdruck "1 + 2 shl 3"



In beiden Sprachen ist entscheidend, **welche Priorität** ( operator precedence ) die Operatoren haben.

**C:** 1 + 2 << 3

In C hat + **höhere Priorität** als << .

Der Ausdruck wird also so geklammert:

```
C
(1 + 2) << 3
```

Zuerst:

```
C
1 + 2 = 3
```

Dann Linksverschiebung um 3 Bit:

```
C
3 << 3
```

Das entspricht:

```
3 * 2^3 = 24
```

Ergebnis in C:

```
24
```

## Pascal: $1 + 2 \text{ shl } 3$

In Pascal (klassisches Pascal / Delphi / FreePascal) hat `shl` die **gleiche Priorität wie** `*`, `div`, `mod` und damit **höhere Priorität als** `+`.

Deshalb wird ausgewertet als:

**pascal**

$$1 + (2 \text{ shl } 3)$$

Zuerst:

**pascal**

$$2 \text{ shl } 3 = 16$$

Dann:

**pascal**

$$1 + 16 = 17$$

Ergebnis in Pascal:

17

## Zusammenfassung

| Sprache | Auswertung               | Ergebnis |
|---------|--------------------------|----------|
| C       | $(1 + 2) \ll 3$          | 24       |
| Pascal  | $1 + (2 \text{ shl } 3)$ | 17       |

Das ist ein klassisches Beispiel dafür, dass sich Operatorprioritäten zwischen Sprachen unterscheiden können.



