

<AusdruckL> , <AusdruckR>

- Zusammenfassung von zwei Ausdrücken zu einem einzigen Ausdruck, so dass die enthaltenen Ausdrücke von links nach rechts ausgewertet werden
 - **AusdruckL** wird ausgewertet und das **Ergebnis vergessen**
 - **AusdruckR** wird ausgewertet zum **Ergebnis des Komma-Ausdrucks**

```
int a = 0, b = 0, c = 0;
a = 1;
b = 2;
c = a + b;
( a = 1, b = 2, c = a + b );
c = ( a = 1, b = 2, a + b );
```

Von links nach rechts ausgew.

↳ Zuweisungen besitzen Rückwert

```
char *s1 = ..., *s2 = ...;
char *l = NULL, *r = NULL;
int equals = 1;
for( l = s1, r = s2 ; equals && *l && *r ; l++, r++ ) {
    equals &= (*l == *r);
}
equals &= (*l == *r);
```

Rückgabe d. Kommaausdrucks

	Operand AusdruckL	Operand AusdruckR	Ergebnis
Typen	Beliebiger Typ TL	Beliebiger Typ TR	TR, ist kein lvalue
Anmerkungen	Auswertung von links nach rechts (AusdruckL vor AusdruckR), Ergebnis von AusdruckL wird verworfen		Wert von AusdruckR