

Blatt 12

Vincent Kümmerle und Elvis Gnaglo

18. Januar 2026

1 Datentypen

```
1 #include <iostream>
2 #include <string>
3 #include <cmath>
4 #include <typeinfo>
5
6 int main() {
7     auto v1 = 3 + 5; // Ganzzahl-Addition
8     std::cout << "3 + 5 = " << v1 << " | Typ: " << typeid(v1).name()
9         << std::endl;
10
11     auto v2 = 3 + 5.0; // Misch-Addition
12     std::cout << "3 + 5.0 = " << v2 << " | Typ: " << typeid(v2).name
13         () << std::endl;
14
15     // "3" + "5" würde einen Compilerfehler verursachen (Zeiger-
16     // Addition)
17
18     auto v4 = std::string("3") + "5"; // String-Zusammenfügung
19     std::cout << "std::string(\"3\") + \"5\" = " << v4 << " | Typ: "
20         << typeid(v4).name() << std::endl;
21
22     auto v5 = 3 / 2; // Ganzzahl-Division
23     std::cout << "3 / 2 = " << v5 << " | Typ: " << typeid(v5).name()
24         << std::endl;
25
26     auto v6 = 3.0 / 2; // Gleitkomma-Division
27     std::cout << "3.0 / 2 = " << v6 << " | Typ: " << typeid(v6).name
28         () << std::endl;
29
30     auto v7 = int(2.71828); // Explizite Typumwandlung von Kommazahl
31     // zu Ganzzahl
```

```

25     std::cout << "int(2.71828) = " << v7 << " | Typ: " << typeid(v7).
        name() << std::endl;
26
27     auto v8 = std::round(2.71828); // Mathematisches Runden
28     std::cout << "std::round(2.71828) = " << v8 << " | Typ: " <<
        typeid(v8).name() << std::endl;
29
30     return 0;
31 }
32 }

```

Output: int = Ganzzahl, double = Kommazahl

```

1 3 + 5 = 8 | Typ: int
2 3 + 5.0 = 8 | Typ: double
3 std::string("3") + "5" = 35 | Typ: class std::basic_string<char,
    struct std::char_traits<char>, class std::allocator<char> >
4 // C++ String-Objekt
5 3 / 2 = 1 | Typ: int
6 3.0 / 2 = 1.5 | Typ: double
7 int(2.71828) = 2 | Typ: int
8 std::round(2.71828) = 3 | Typ: double

```

2 Zahlen Raten mit C++

```

1 #include <iostream>
2 #include <cstdlib> // für rand() und srand()
3 #include <ctime>   // für time()
4
5 int main(int argc, char* argv[])
6 {
7     // Zufallsgenerators mit der aktuellen Zeit
8     std::srand(std::time(NULL));
9
10    int maximum;
11    int guess;
12    int guess_count = 1;
13    int random_number;
14
15    std::cout << "Gib ein Maximum fuer die Zufallszahl ein: ";
16    std::cin >> maximum;
17
18    // Generiert eine Zahl zwischen 1 und dem Maximum
19    random_number = (std::rand() % maximum) + 1;
20

```

```

21     std::cout << "Rate die Zahl (1 bis " << maximum << "): ";
22     std::cin >> guess;
23
24     while (guess != random_number) {
25         // Ueberprüfung der Einggegebenen Zahl
26         if (guess > random_number) {
27             std::cout << "Die Zahl ist kleiner als " << guess << ".
                Versuche es nochmal: ";
28         }
29         else {
30             std::cout << "Die Zahl ist groesser als " << guess << ".
                Versuche es nochmal: ";
31         }
32
33         // Neue Eingabe innerhalb der Schleife
34         std::cin >> guess;
35
36         // Zähler erhöhen
37         guess_count++;
38     }
39
40     // Erfolgsmeldung nach der Schleife
41     std::cout << "Glueckwunsch! Du hast die Zahl " << random_number
        << " erraten." << std::endl;
42     std::cout << "Du hast dafuer " << guess_count << " Versuche
        gebraucht." << std::endl;
43
44     return 0;
45 }

```

3 Funktionen in C++

```

1 #include <iostream>
2
3 // Definition der Prüffunktion
4 bool pruefeTipp(int tipp, int loesung) {
5     if (tipp == loesung) {
6         std::cout << "Treffer! Gut gemacht." << std::endl;
7         return true; // Rückgabe für True
8     }
9     else if (tipp > loesung) { // Tipp größer als die Lösung
10         std::cout << "Zu hoch!" << std::endl;
11     }
12     else {

```

```

13         std::cout << "Zu niedrig!" << std::endl;
14     }
15     return false; // Rückgabe für False
16 }
17
18 int main() {
19     int guess;
20     bool gefunden = false;
21
22     std::cout << "Willkommen zum Zahlen-Raten!" << std::endl;
23
24     while (!gefunden) {
25         std::cout << "Dein Tipp: ";
26         std::cin >> guess;
27
28         gefunden = pruefeTipp(guess, secret);
29     }
30
31     return 0;
32 }

```