

Task 14:

Schreibe eine Funktion, die beliebig viele Argumente bekommt und ihre Summe zurückgibt.

Task 15:

Schreibe eine Funktion, die zu den gegebenen Argumenten x , a , b und c (wobei a , b und c optional sind) die Lösung des Polynoms $ax + bx^2 + cx^3$ berechnet.

Task 16:

Erweitere das Programm aus Task 11, so, dass der Nutzer die auszuführende Operation mit einer Eingabe (via `input()`) bestimmen kann. Der Variablenname der Funktion, die zur Berechnung benutzt wird, soll davon unabhängig aber gleich bleiben.

Task 17:

Erweitere das Programm aus Task 16 so, dass der Nutzer zusätzlich die Operanden eingeben kann und diese als Integer interpretiert werden.

Was passiert, wenn man dennoch eine Kommazahl (in englischer Schreibweise, z.B. "3.141") eingibt?

Task 18:

Schreibe jeweils für die Umwandlung von `int` → `bool`, `float` → `bool`, `str` → `bool` eine alternative Funktion, die nicht auf `bool()` zurückgreift, aber die gleichen Ergebnisse liefert.

Task 19:

Schreibe eine Funktion, die einen String als Argument bekommt und eine Liste zurückgibt, deren Elemente die Zeilen des Strings in Kleinbuchstaben sind.

Task 20:

Schreibe eine Funktion, die einen String s und einen Integer n als Argument bekommt und einen String zurückgibt, in dem jedes Zeichen in s um n auf dem Alphabet verschoben ist.

Task 21:

Benutze die Funktionen aus Task 19 und 20, um die Funktionen `caesar_encrypt(text, key)` und `caesar_decrypt(text, key)` zu implementieren.
(`text = caesar_decrypt(caesar_encrypt(text, key), key)`)

Task 22:

Schreibe eine Funktion, die einen String `s` und einen Buchstaben `c` (String der Länge 1) als Argument bekommt und daraus den Caesar-Schlüssel ableitet, mit dem ein Klartext zu `s` verschlüsselt wurde, wenn `c` das am häufigsten im Klartext verwendete Zeichen war.