

Default-Argumente

Diese funktionieren in C++ wie in Python: Eine Funktion kann (**nur bei ihrer erstmaligen Deklaration oder Definition**) Default-Werte für einige oder alle Parameter definieren. Diese **dürfen** dann beim Aufruf fehlen und werden in diesem Fall durch die Default-Werte ersetzt:

```
double f(double a, double b = 1.0, double c = 3.0)
{
    return a + b + c;
}
...
f(-1.0, 2.0, 7.0); // korrekt: Rückgabewert ist 8.0 (-1.0 + 2.0 + 7.0)
f(-1.0, 2.0);      // korrekt: Rückgabewert ist 5.0 (-1.0 + 2.0 + 3.0)
f(-1.0);           // korrekt: Rückgabewert ist 3.0 (-1.0 + 1.0 + 3.0)
f();               // Fehler: Argument a fehlt
```

Selbstverständlich können auch alle Argumente einen Default-Wert bekommen. Defaultwerte können in C++ (wie in Python) nur für die **letzten** Argumente vergeben werden, d.h. die folgende Definition wäre **nicht erlaubt** (hat ein Argument einen Default-Wert, müssen **auch alle danach folgenden** einen Default-Wert besitzen)!

```
double add(double a = -1.0, double b) // Fehler!!
{
    return a + b;
}
```

Prinzipiell darf man auch Funktionen mit Default-Argumenten überladen, es darf nur kein Aufruf dieser Funktionen mehrdeutig sein (siehe Text über Überladen)