

Datentypen

<code>void</code>	Datentyp für Funktion ohne Rückgabe .
void-Funktionen haben keinen Rückgabewert, aber sinnvollerweise einen Effekt (z.B. Textausgabe im Beispiel unten).	
<pre>void print_account (const int a, const double b) { std::cout << "Your assets: " << a << "\n" << "Your interest: " << b << "\n"; }</pre>	

Funktionen

Funktion	Selbstständiger Codeabschnitt
Wichtige Befehle: Definition: <code>int my_fun (bool arg1, float arg2) {...}</code> Rückgabe: <code>return my_val;</code> Aufruf: <code>my_fun(true, 3.75f)</code>	
Der Rückgabewert wird immer zum Rückgabotyp konvertiert.	
Jede Funktion, die nicht den Rückgabotyp <code>void</code> hat, muss ein return haben.	

(...)

Programmier-Befehle - Woche 5

(...)

```
unsigned int bin_digits (unsigned int n) {
    if (n == 0)
        return 1; // stops function and returns 1
    unsigned int count = 0;
    do {
        n /= 2;
        ++count;
    } while (n > 0);
    return count;
}

int main () {
    std::cout << bin_digits(3) << "\n"; // Output: 2
    std::cout << bin_digits(8) << "\n"; // Output: 4
    return 0;
}
```

// PRE: ... // POST: ...	Funktionsbeschreibung
PRE-/POST-Conditions gehören vor jede Funktionsdefinition ausser der main-Funktion. (In diesen Programmier-Befehlszusammenfassungen werden sie aber manchmal aus Platzgründen weggelassen.)	
<pre>// POST: return value is a^4 int power_4 (const unsigned int a) { return a*a*a*a; } // PRE: width >= 0 and height >= 0 // POST: returns the rectangle area given by width and height double area (const double width, const double height) { assert(width >= 0 && height >= 0); return width * height; }</pre>	