

## Funktionen

Funktion	Selbstständiger Codeabschnitt
<p>Wichtige Befehle:</p> <p><b>Definition:</b>    <code>int my_fun (bool arg1, float arg2) {...}</code> <b>Rückgabe:</b>     <code>return my_val;</code> <b>Aufruf:</b>       <code>my_fun(true, 3.75f)</code></p> <p>Der <b>Rückgabewert</b> wird immer zum <b>Rückgabety</b> konvertiert.</p> <p>Jede Funktion, die nicht den <b>Rückgabety</b> <code>void</code> hat, muss ein <b>return</b> haben.</p>	
<pre>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 &gt; 0);     return count; }  int main () {     std::cout &lt;&lt; bin_digits(3) &lt;&lt; "\n"; // Output: 2     std::cout &lt;&lt; bin_digits(8) &lt;&lt; "\n"; // Output: 4     return 0; }</pre>	

// PRE: ... // POST: ...	Funktionsbeschreibung
<p><b>PRE-/POST-Conditions</b> gehören vor <b>jede</b> Funktionsdefinition ausser der <b>main</b>-Funktion. (In diesen Programmier-Befehlszusammenfassungen werden sie aber manchmal aus Platzgründen weggelassen.)</p> <p>Man kann beispielsweise <b>assert</b> verwenden, um das Programm abubrechen, falls die Funktion doch mal mit Argumenten aufgerufen wird, welche die <b>PRE-Condition</b> verletzen.</p>	

( ... )

# Programmier-Befehle - Woche 5

( ... )

```
// POST: return value is a^4
int power_4 (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 (double width, double height) {
    assert(width >= 0 && height >= 0);
    return width * height;
}
```

## Datentypen

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