

Verzweigungen / Fallunterscheidung	
Verzweigungen	Syntax
if	<pre>if (Bedingung) Anweisung1; else Anweisung2;</pre>
Kurzschreibweise	<code>(Bedingung) ? Anweisung1 : Anweisung2;</code>
Fallunterscheidungen	Syntax
Switch	<pre>switch (Ausdruck) { case Wert1: Anweisungen1; break; case Wert2: Anweisungen2; break; ... default: Ersatz anweisungen;</pre>
Der Datentyp von Ausdruck muss int sein! <code>break;</code> beendet die Anweisung, wenn es fehlt werden die nachfolgenden Anweisungen auch ausgeführt. Alternativ kann eine Anweisung auch mit <code>continue</code>, <code>goto</code> und <code>return</code> beendet werden.	
Schleifen	
Schleife	Syntax
while-Schleife (1 Anweisung)	<pre>while (Bedingung) Anweisung;</pre>
while-Schleife (länger)	<pre>while (Bedingung) { Anweisung1; Anweisung2; }</pre>
do-while-Schleife (1 Anweisung)	<pre>do Anweisung; while (Bedingung);</pre>
do-while-Schleife (länger)	<pre>do { Anweisung1; Anweisung2; } while (Bedingung);</pre>
Schleifen (cont)	
for-Schleife (1 Anweisung)	<pre>for ([Variablen-Initialisierung]; [Bedingung]; [Anweisung]) Anweisung;</pre>
for-Schleife (länger)	<pre>for ([Variablen-Initialisierung]; [Bedingung]; [Anweisung]) { Anweisung1; Anweisung2; }</pre>
Wichtiges zur for-Schleife: Werden Teile der Parameter weggelassen (z.B. keine Initialisierung), muss das Semikolon trotzdem gesetzt werden. <i>Beispiel:</i> <pre>int i = 5; for (; i >= 0; i--) Anweisung;</pre> anderes: <code>break</code> beendet eine Schleife manuell, unabhängig von der Bedingung. <code>continue</code> überspringt den restlichen Schleifenkörper und springt direkt zum Anfang (Prüfung der Bedingung).	