

Spickzettel Swift

Blöcke	Blöcke werden in geschweiften Klammern notiert. <pre>{ // ... }</pre>
Kommentare	<pre>// einzeliger Kommentar /* mehrzeiliger Kommentar */</pre>
Primitive Datentypen	<pre>Bool: Wahrheitswert Int: Ganzzahl Float: Kommazahl (einfache Länge) Double: Kommazahl (doppelte Länge) Character: Zeichen, z. B. "C" Weitere: Int8, Int16, Int32, Int64</pre>
Zeichenketten	<pre>String: Zeichenkette, z. B. "Hallo"</pre>
Vereinbarung und Zuweisung bei primitiven Datentypen	<pre>var x: Int x = 6</pre>
Klasse mit Deklaration von Attributen und Konstruktor	<pre>class Haus { var strasse: String var hausnummer: Int public init(strasseNeu: String, hNrNeu: Int) { strasse = strasseNeu hausnummer = hNrNeu } }</pre>
Vererbung	<pre>class Reihenhaus: Haus Aufruf Oberklassenkonstruktor: super.init(...) Aufruf anderer Methoden: super.Methodenname(...)</pre>
Methode überschreiben	<pre>override func AktionAusführen(){ //...</pre>
Objekterzeugung	<pre>var h1: Haus = Haus(strasseNeu: "Schulstr. ", hNrNeu:8)</pre>
Referenz auf das ausführende Objekt	<pre>self</pre>
Leere Referenz	<pre>nil</pre>
Methode ohne Rückgabewert	<pre>func RückwärtsGehen(schritte: Int) { Gehen(schritte: -schritte) }</pre>
Methode mit Rückgabewert	<pre>func Quadrieren(zahl: Int) -> Int { return zahl*zahl }</pre>

Vergleichsoperatoren	<pre> == ist gleich != ungleich > größer >= größer gleich < kleiner <= kleiner gleich </pre>
Logische Operatoren	<pre> ! nicht && und oder </pre>
Bedingte Anweisung (bei einseitiger bedingter Anweisung entfällt der else-Teil)	<pre> if farbe == "rot" { anzahlRot = anzahlRot + 1 } else { //... } </pre>
Mehrfachauswahl	<pre> switch farbe { case "rot": anzahlRot = anzahlRot + 1 case "blau": anzahlBlau = anzahlBlau + 1 default: print("Farbe nicht bekannt") } </pre>
Zählwiederholung	<pre> for i in 0 ..< anzahlRechtecke { //zu wiederholende Sequenz } </pre>
Wiederholung mit Anfangsbedingung	<pre> while !Berührt() { Gehen(schritte: zehn) } </pre>
Konsolenausgabe	<pre> print(...) </pre>