

<p>Variable</p>	<p>Ein <i>Speicherplatz</i>, der Werte aufnehmen kann, die sich ändern.</p> <p>Beispiel in Python:</p> <pre>name = "Anna"</pre> <p>In diesem Beispiel ist <code>name</code> die Variable, und ihr wird der Wert "Anna" zugewiesen.</p>
<p>Programmierschleife</p>	<p>Eine Wiederholung eines Codeblocks, solange eine Bedingung erfüllt ist.</p> <pre>i = 0 while i < 5: print(i) i += 1</pre> <p>Die Schleife wiederholt sich also, bis <code>i</code> den Wert 5 erreicht - dann wird sie beendet.</p> <p>Schleifen tragen zur Effizienz von Programm bei: Wiederholende Aufgaben müssen nicht mehrfach geschrieben werden.</p>
<p>Algorithmus</p>	<p>Eine eindeutige Folge von Schritten zur Lösung eines Problems.</p> <p>Beispiel: Finde die größte Zahl von drei Zahlen.</p> <p>Algorithmus (in Schritten):</p> <p>Vergleiche die erste Zahl mit der zweiten. Merke dir die größere von beiden. Vergleiche diese gemerkte Zahl mit der dritten. Die größere Zahl ist das Ergebnis.</p>
<p>Programmbeschreibung</p>	<p>Eine Programmbeschreibung ist eine schriftliche Erklärung oder Dokumentation, die beschreibt, was ein Programm tut, wie es funktioniert und welche Ziele es verfolgt. Sie kann Informationen enthalten wie:</p> <p>Ziel des Programms: Was soll das Programm erreichen? Funktionalitäten: Welche Aufgaben oder Funktionen bietet das Programm? Eingaben und Ausgaben: Welche Daten benötigt das Programm und welche Ergebnisse gibt es aus? Technische Details: Welche Programmiersprachen, Bibliotheken etc. wurden verwendet?</p> <p>Beispiel:</p> <p><i>Ziel des Programms:</i> Dieses Programm berechnet die Summe aller Zahlen von 1 bis zu einer angegebenen Zahl <code>n</code>.</p> <p>Funktionen: Der Benutzer gibt eine Zahl <code>n</code> ein.</p>

	Das Programm berechnet und gibt die Summe der Zahlen von 1 bis n aus, usw.
Programmspezifikation	Die Programmspezifikation sorgt dafür, dass alle Anforderungen klar und präzise sind, bevor mit der Programmierung begonnen wird. Sie hilft auch, Missverständnisse zu vermeiden und stellt sicher, dass das entwickelte Programm alle Anforderungen erfüllt.
Problemzerlegung	Das Problem wird in kleinere, besser handhabbare Teilprobleme zerlegt.
Quellcode	<p>Quellcode ist der Code, den ein Programmierer in einer Programmiersprache wie Python, Java oder C schreibt. Dieser Code ist für Menschen lesbar.</p> <p>Python: <code>print("Hallo, Welt!")</code> # Dieser Code gibt "Hallo, Welt!" auf dem Bildschirm aus</p>
Maschinencode	<p>Maschinencode ist der Code, den der Computer tatsächlich ausführt. Er besteht aus binären Zahlen (0 und 1) und wird durch einen Compiler oder Interpreter aus dem Quellcode erzeugt.</p> <p>Beispiel für Maschinencode: 10110000 01100001</p>
Sequenz	<p>Sequenz ist ein grundlegendes Konzept, da es beschreibt, wie die Befehle eines Programms nacheinander abgearbeitet werden.</p> <p>Beispiel:</p> <ul style="list-style-type: none"> • Schritt 1 - Der Benutzer gibt eine Zahl ein (Eingabe). • Schritt 2 - Diese Zahl wird verdoppelt (Berechnung). • Schritt 3 - Das Ergebnis wird ausgegeben (Ausgabe) <p>Die Sequenz stellt sicher, dass alle Schritte in der richtigen Reihenfolge durchgeführt werden.</p> <p>Wenn die Reihenfolge der Anweisungen nicht beachtet wird, können unerwartete Fehler auftreten. Zum Beispiel würde ein Versuch, eine Zahl zu verdoppeln, bevor sie eingegeben wird, zu einem Fehler führen.</p>
Schritte bei der Erstellung eines Programms	Analyse → Entwurf → Programmierung → Testen → Erweiterung

<p>Flussdiagramm</p>	<p>Ein Flussdiagramm ist eine visuelle Darstellung eines Prozesses oder Algorithmus. Es verwendet standardisierte Symbole und Pfeile, um den Ablauf von Aktivitäten zu zeigen und wie verschiedene Schritte miteinander verbunden sind. Flussdiagramme helfen dabei, komplexe Prozesse einfach und verständlich darzustellen.</p> <p>Beispiel Flussdiagramm:</p> <pre> [Start] [Frage nach Zahl 1] → [Frage nach Zahl 2] → [Frage nach Zahl 3] [Vergleiche Zahlen] [Zeige größte Zahl] [Ende]</pre>
<p>Text (Strings)</p>	<p>Strings sind einfach Texte oder Zeichenketten, die in Python durch Anführungszeichen (" oder ') definiert werden.</p> <p>Beispiel in Python:</p> <pre>text = "Hallo, Welt!"</pre>
<p>Zahlen (Integer und Float)</p>	<ul style="list-style-type: none"> • Integer (int) sind ganze Zahlen, die keine Dezimalstellen haben. • Float sind Gleitkommazahlen, die Dezimalstellen enthalten.
<p>Boolean</p>	<p>Ein Boolean ist ein Datentyp, der nur zwei mögliche Werte haben kann: True (wahr) oder False (falsch).</p>
<p>for</p>	<p>Beispiel in Python:</p> <pre>namen = ["Anna", "Max", "Lena", "Tom"] for name in namen: print(name)</pre> <p>Erklärung:</p> <ul style="list-style-type: none"> • Wir haben eine Liste <code>namen</code> mit den Namen "Anna", "Max", "Lena" und "Tom". • Die <code>for</code>-Schleife geht jedes Element der Liste durch und gibt es mit <code>print()</code> aus.
<p>If ... then ... else ...</p>	<p>Ein einfaches Beispiel für eine if-else-Bedingung in Python: Überprüfe, ob eine Zahl positiv oder negativ ist.</p> <p>Python:</p> <pre>zahl = int(input("Gib eine Zahl ein: "))</pre>

```
if zahl > 0:
    print("Die Zahl ist positiv.")
else:
    print("Die Zahl ist negativ oder null.")
```

Erklärung:

- Der Benutzer wird nach einer Zahl gefragt, die in der Variablen `zahl` gespeichert wird.
- Mit der **if-Bedingung** wird überprüft, ob die Zahl **größer als 0** ist.

Wenn ja, wird "Die Zahl ist positiv." ausgegeben.
 Falls nicht (also wenn die Zahl null oder negativ ist), wird der **else-Teil** ausgeführt, der "Die Zahl ist negativ oder null." anzeigt.