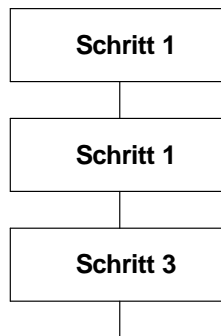


## I. Programmierung

---

Programmierung ist die Planung einer Abfolge von Schritten (Instruktionen), nach denen ein Computer handeln soll.



Der **Lebenszyklus** eines Programms

- Problemlösung

- **Analysieren** des Problems und Spezifizierung einer Lösung.
- Entwicklung einer generellen Lösung (**Algorithmus**) um das Problem zu lösen.

Ein Algorithmus ist eine Schritt-für-Schritt Anleitung, um ein Problem in endlicher Zeit zu lösen.

- **Überprüfen**, ob die Lösung das Problem wirklich löst.

- **Implementierung**

Die Übersetzung eines Algorithmus in eine Programmiersprache nennt man **Kodierung**

- **Wartung**

---

## II. Lineare Programme - Datentypen

---

---

### 2.1 Programmaufbau

---

**Beispiel :**

<pre><i>#Hallo Welt</i></pre> <pre><i>print "Hallo Welt !"</i></pre> <pre><i>print</i></pre> <pre><i>print "hallo world !"</i></pre>	<p>Kommentar</p> <p>Ausgabe eines Textes</p> <p>Leerzeile</p> <p>Ausgabe eines Textes</p>
--	---

Ein lineares Programm besteht aus einer Folge von **Kommentaren** und **Anweisungen**.

Kommentare dienen der Dokumentation eines Programms und werden vom **Doppelkreuz #** eingeleitet. Sie werden vom Programm nicht beachtet.

---

### 2.2 Variablen, Ein- und Ausgabe und Zuweisung

---

**Beispiel :**

<pre><i>#Eingabe von Zahlen und Zuweisung</i></pre> <pre><i>a = input("1.Summand : ")</i></pre> <pre><i>print</i></pre> <pre><i>b = input("2.Summand : ")</i></pre> <pre><i>s = a + b</i></pre> <pre><i>print "Summe : ", s</i></pre>	<p>Kommentar</p> <p>Eingabe und Wertzuweisung an die Variable a</p> <p>Leerzeile</p> <p>Eingabe und Wertzuweisung an die Variable b</p> <p>Addition der Speicherinhalte und Wertzuweisung an die Variable s</p> <p>Ausgabe eines Textes und des Wertes der Variablen s</p>
---	--

Die Namen für Speicherplätze heißen Variablen.

Das Einlesen von Zahlen, die Variablen zugewiesen werden sollen, erfolgt mit der "input"-Anweisung. Die Zuweisung erfolgt mit dem "="-Operator.

Die Addition von Zahlen erfolgt mit dem "+"-Operator.

Syntax :

```
variable = input("prompt")
```

Der prompt ist ein Text der die Eingabe erläutert.

**Beispiel :**

<i>#Eingabe von Zeichenketten</i>	Kommentar
<i>vorname = raw_input("Vorname : " )</i>	Eingabe einer Zeichenkette und Wertzuweisung an die Variable vorname
<i>name = raw_input("Name : ")</i>	Eingabe einer Zeichenkette und Wertzuweisung an die Variable name
<i>print "Hallo"+ " , vorname, name</i>	Ausgabe eines Textes und der Werte der Variablen Name und Vorname

Das Einlesen von Zeichen, die Variablen zugewiesen werden sollen, erfolgt mit der "raw\_input"- Anweisung.

Syntax :

```
variable = raw_input("prompt")
```

Ganze Zahlen und Zeichenketten sind Beispiele von **Datentypen (Objekttypen)**.

Man unterscheidet **numerische** (einfache) und **sequentielle** (zusammengesetzte) Datentypen. In Python wird der Datentyp an Hand der Eingabe identifiziert.

Die wichtigsten Datentypen sind

### Numerische Datentypen

Datentyp	Beispiele
Ganze Zahlen	1, 1000
Lange ganze Zahlen	1L, 12345678987654321L
Gleitkommazahlen	1.0, 123.04, 1.02E3
Komplexe Zahlen	2+3j, 1.5+0.5j

## Sequentielle Datentypen

<b>Datentyp</b>	<b>Beispiele</b>
Zeichenketten(strings)	"Hans"
Listen	[1,2,3], ["rot", "schwarz"], ['a', 'b', 'c']
Tupel	(1,2,3), ("Adam", "Eva"), ('1', '2')
Dictionaries	[1:"sehr gut", 2:"gut"]

### III. Programmstrukturen

---

---

#### 2.1 Die einseitige Auswahl

---

Bei einer einseitigen Auswahl werden Anweisungen nur ausgeführt, wenn eine Bedingung erfüllt ist. Die Programmstruktur einer einseitigen Auswahl lautet

<pre><i>#Einseitige Auswahl</i> <i>a = input("Zahl : ")</i>  <i>print</i>  <i>if a &gt;= 0 :</i>      <i>r = a**0.5</i>  <i>print "Wurzel : ",r</i></pre>	<p>Kommentar</p> <p>Eingabe eines Zahlenwerts</p> <p>Leerzeile</p> <p>Falls Bedingung erfüllt ist</p> <p>Berechnung der Wurzel</p> <p>Ausgabe der Wurzel</p>
---	--

Struktur einer einseitigen Auswahl

```
if bedingung :  
    anweisung 1  
    anweisung 2  
    .....
```

Die Anweisungen im eingerückten Block werden nur dann ausgeführt, wenn die Bedingung erfüllt ist.

Bedingungen erstellt man oft mit **Vergleichsoperatoren**. Mit ihnen werden Werte von Variablen miteinander oder mit Zahlen verglichen.

Operator	Bedutung
>	<b>größer</b>
<	<b>kleiner</b>
>=	<b>größer gleich</b>
<=	<b>kleiner gleich</b>
==	<b>gleich</b>
!=	<b>ungleich</b>

Bemerkung :

Ein Vergleich liefert immer einen Wert vom Typ **Boolean**. Dieser hat entweder den Wert **True** oder **False**.

---

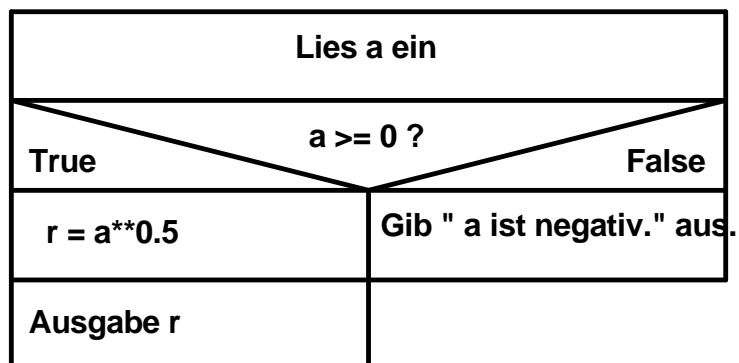
## 2.2. Die zweiseitige Auswahl

---

**Beispiel :**

<i>#Zweiseitige Auswahl</i>	Kommentar
<i>a = input("Zahl : ")</i>	Eingabe eines Zahlenwerts
<i>print</i>	Leerzeile
<i>if a &gt;= 0 :</i>	Falls Bedingung erfüllt ist
<i>r = a**0.5</i>	Berechnung der Wurzel
<i>print "Wurzel : ",r</i>	Ausgabe der Wurzel
<i>else :</i>	Falls Bedingung nicht erfüllt ist
<i>print "Eingegebene Zahl ist negativ."</i>	Ausgabe eines Textes

**Struktogramm :**



Struktur einerzweiseitigen Auswahl

<b>if</b> <i>bedingung</i> : <i>anweisung 1</i> <i>anweisung 2</i> ..... <b>else</b> <i>anweisung 1</i> <i>anweisung 2</i> .....
---

---

### 2.3. Die mehrseitige Auswahl

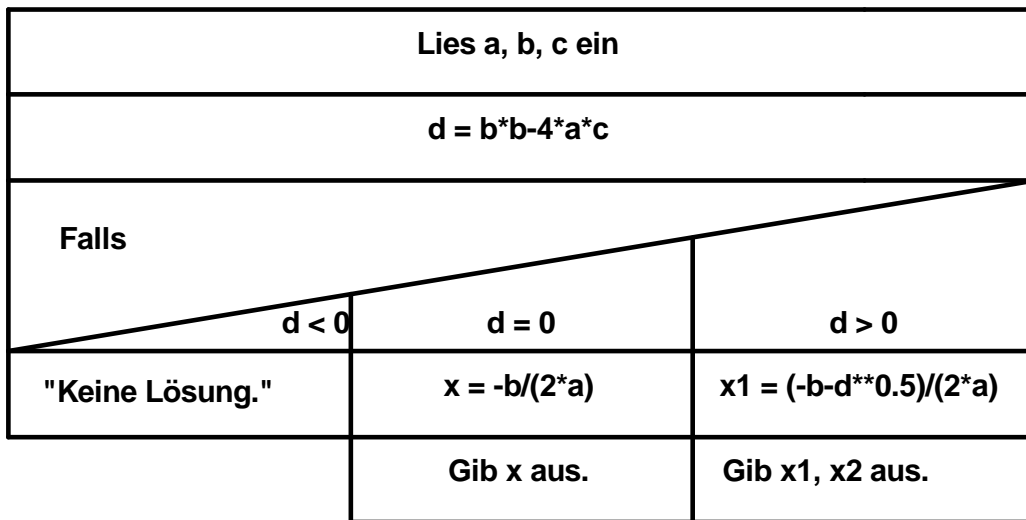
---

#### Beispiel :

Bestimmung der Lösung einer quadratischen Gleichung

<pre><i>#Mehrseitige Auswahl</i>  a = input("Zahl : ")  print  b = input("Zahl : ")  print  c = input("Zahl : ")  print  d = b**2-4*a*c  if d &gt; 0 :      x_1 = (-b-d**0.5)/(2*a)      x_2 = (-b+d**0.5)/(2*a)      print "Zwei Loesungen : ", x_1, x_2  elif d == 0 :      x = -b/(2*a)      print "Eine Loesung : ", x  else :      print "Keine Loesung !"</pre>	<p>Kommentar</p> <p>Eingabe und Wertzuweisung an die Variable a</p> <p>Leerzeile</p> <p>Eingabe und Wertzuweisung an die Variable b</p> <p>Leerzeile</p> <p>Eingabe und Wertzuweisung an die Variable b</p> <p>Leerzeile</p> <p>Berechnung der Diskriminante und Wertzuweisung an die Variable</p> <p>Bedingung für Zweig 1</p> <p>Berechnung der Lösungen und Wertzuweisung an die Variablen x_1 und x_2</p> <p>Ausgabe eines Textes und der Werte der Variablen x_1 und x_2</p> <p>Bedingung für Zweig 2</p> <p>Berechnung der Lösung und Wertzuweisung an die Variablen x</p> <p>Ausgabe eines Textes und des Wertes der Variablen x</p> <p>Fall keine Bedingung erfüllt</p> <p>Ausgabe eines Textes</p>
---	---

**Struktogramm :**



Struktur einer zweiseitigen Auswahl

```

if bedingung :
    anweisung 1
    anweisung 2
    .....
elif bedingung :
    anweisung 1
    anweisung 2
    .....
else :
    anweisung 1
    anweisung 2
    .....
    
```



## 2.4 Die for-Schleife

---

Bei einer Iteration wird eine Sequenz (Liste, Tupel, Zeichenkette) jeweils ein Anweisungsblock durchlaufen.

**Beispiele :**

<pre><i># Iteration ueber eine Liste</i> for i in [-2, -1, - 0.5, 0, 0.5, 1, 2] :     q = i*i     print "Zahl :", i, "Quadrat :", q</pre>	<p>Kommentar</p> <p>Suzessive Wahl der Elemente der Liste</p> <p>Quadratur der Elemente und Wertzuweisung</p> <p>Ausgabe</p>
---	--

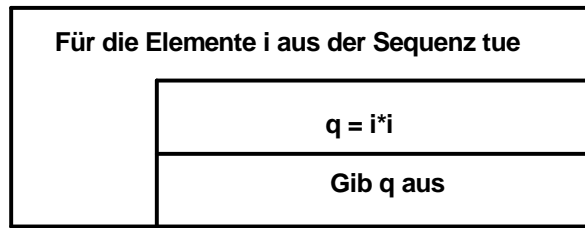
<pre><i># Iteration ueber eine Zeichenkette</i> for i in "Mathematik" :     print i</pre>	<p>Kommentar</p> <p>Suzessive Wahl der Buchstaben der Zeichenkette</p> <p>Ausgabe des jeweiligen Buchstaben</p>
---	---

Mit **range (anfangswert, endwert, schrittweite)** lässt eine regelmäßige Liste ganzer Zahlen erzeugen.

**range(anfangswert, endwert)** erzeugt einer Liste aller ganzen Zahlen von anfangswert bis endwert-1.

**range(endwert)** erzeugt die Liste aller ganzen Zahlen von 0 bis endwert-1.

<pre><i># Iteration ueber eine Liste</i> for i in range(-3, 3, 2) :     q = i*i     print "Zahl :", i, "Quadrat :", q</pre>	<p>Kommentar</p> <p>Suzessive Wahl der ganzen Zahlen von -3 bis ausschließlich 3 in Abständen von 2</p> <p>Quadratur der Elemente und Wertzuweisung</p> <p>Ausgabe</p>
---	--



Struktur einer for-Schleife :

```
for element in sequenz :  
    anweisung1  
    anweisung2;  
    .....;
```



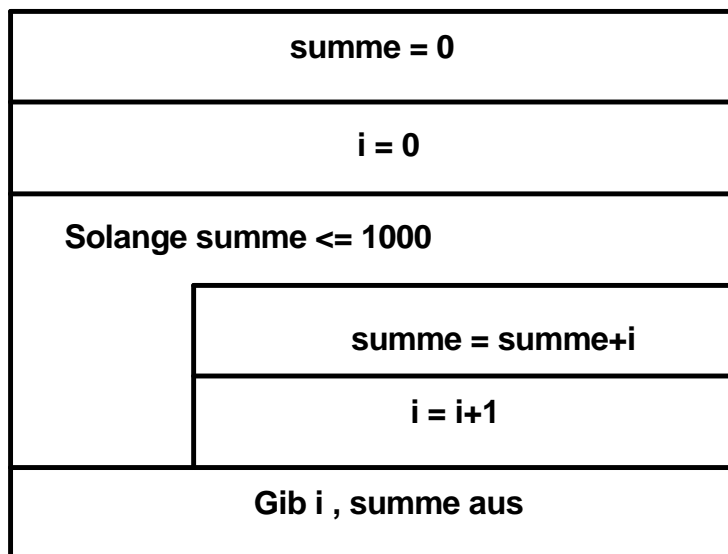
## 2.5 Die while Schleife

Bei einer while-Schleife wird ein Anweisungsblock solange durchlaufen, wie eine Bedingung erfüllt ist.

Das folgende Programm bestimmt, bis zu welcher Zahl man alle natürlichen Zahlen addieren kann, ohne dass die Summe größer als 100 wird.

<i># while Schleife</i>	Kommentar
<i>summe = 0</i>	Wertzuweisung
<i>i = 0</i>	Wertzuweisung
<i>while summe &lt;= 1000 :</i>	Bedingung für das Durchlaufen der while Schleife
<i>summe = summe+i</i>	Wertzuweisung
<i>i = i+1</i>	Werterhöhung
<i>print i, summe</i>	

Struktogramm :



Struktur einer while-Schleife :

```
while bedingung :
    anweisung1
    anweisung2;
    .....
```

## Aufgaben

---

---

### I. Lineare Programme

---

1. Die Botschaft "Mein erstes Python-Programm" soll ausgegeben werden.
2. Der Tarif für den elektrischen Strom setzt sich aus einem Grundpreis, der monatlich 18 € beträgt und unabhängig vom Stromverbrauch ist, sowie dem Preis von 14 Cent je Kilowattstunde zusammen.

Schreibe ein Programm, das nach Eingabe des Verbrauchs in kWh die Stromrechnung ausgibt.

3. Nach Eingabe von Kapital, Zinssatz und Zinsdauer sollen die Zinsen ausgegeben werden.
4. Schreibe ein Programm, das nach Eingabe der Koordinaten zweier Punkte deren Abstand berechnet.

5. Wie lang ist der Anhalteweg eines Autos nach dem Erkennen einer Gefahr ?

Der Anhalteweg setzt sich zusammen aus dem Reaktionsweg und dem Bremsweg.

Der Reaktionsweg ist die Strecke, die vom Erkennen einer Gefahr bis zum Beginn des Bremsens zurückgelegt wird. In dieser Zeit, die bei routinierten Fahrern etwa 0,8 Sekunden beträgt fährt das Auto mit unverminderter Geschwindigkeit weiter

Nach einer Faustformel ist der Reaktionsweg in Metern gleich

#### **"Tachoanzeige durch 10 mal 3"**

Der Bremsweg ist die Strecke, die das Fahrzeug noch bis zum Stillstand zurücklegt, nachdem der Fahrer die Bremse betätigt hat. Ihre Länge hängt von verschiedenen Faktoren (Straßenbeschaffenheit, Beladung usw.) ab. Die Faustformel lautet hier

#### **"Tachoanzeige durch 10 zum Quadrat".**

Berechnen Sie Reaktions-, Brems- und Anhalteweg für Geschwindigkeiten von 50, 100, 110, 120, 130 und 150  $\frac{\text{km}}{\text{h}}$ . Schreiben dazu ein Pythonprogramm.

6. Schreibe ein Programm, das die Werte zweier Variablen vertauscht.

#### **Hinweis :**

Ein klassisches Problem

---

## II. Auswahl

---

1. Nach Eingabe zweier verschiedener Zahlen, soll die die größere der beiden ausgegeben werden. Erstelle eine Programmablaufplan und und schreibe das Programm.

---

2. Für die Versicherung eines bestimmten Kraftwagentyps gilt folgendes Tarifsysteem :

Grundprämie : 640 €

Tarif I (0 - 2 Jahre unfallfrei)	200% der Grundprämie
Tarif II (3 - 5 Jahre unfallfrei)	Grundprämie
Tarif III : (6 - 10 Jahre unfallfrei)	80% der Grundprämie
Tarif IV (11 Jahre unfallfrei und mehr)	50% der Grundprämie

Schreibe ein Programm das nach Eingabe der Anzahl der unfallfreien Jahre die Höhe der Versicherungsprämie ausgibt.

### Hinweise :

a) Python wertet auch Doppelungleichungen aus d.h.  $3 \leq x \leq 5$  macht Sinn (!).

b) Zwei Ungleichungen lassen sich mit dem logischen Operator **and** verknüpfen d.h.,

$3 \leq x \leq 5$  ist gleichwertig mit  $x \geq 3$  and  $x \leq 5$

---

3. Bei einer Schadensversicherung hat der Versicherte bei Schäden über 100 Euro 20% der Schadensumme, mindestens jedoch 100 Euro selbst zu tragen. Erstelle ein Programm, das nach Eingabe der Schadensumme den Anteil des Versicherten und die Zahlung der Versicherung ausgibt.

Erstelle einen Programmablaufplan (Struktogramm) und und schreibe das Programm.

---

4. Schreibe ein Programm, dass feststellt, ob eine eingegebene natürliche Zahl gerade oder ungerade ist.

### Hinweis :

Für zwei natürliche Zahlen m und n ergibt  $m \% n$  den Rest, den m beim Teilen durch n lässt.

---

5. a) Schreibe ein Programm, das von 3 eingegebenen Zahlen die kleinste ausgibt.

b) Schreibe ein Programm, das 3 eingegebene Zahlen in einer fallenden Ungleichungskette ausgibt.

**Hinweis :**Man kann Auswahlen verschachteln.

---

6. Schreibe ein Programm, das die Lösung der Gleichung  $ax + b = c$  bestimmt.

Fallunterscheidung "

---

### III. Schleifen

---

1. Schreibe ein Programm, das den Mittelwert der Daten einer Liste berechnet:

---

2. Schreibe ein Programm, das die Summe der ersten 100 natürlichen Zahlen berechnet.

---

3. Schreibe ein Programm, das nach Eingabe einer natürlichen Zahl  $n$ , die Summe der Quadrate der natürlichen Zahlen von 1 bis  $n$  berechnet und ausgibt.

---

4. Schreibe ein Programm, das 6 Zahlen einliest und deren Summe berechnet.

---

5. Schreibe ein Programm, das die Funktionswerte der Funktion  $f: x \rightarrow y = 2x^2 - 3x + 5$  für  $x=1,3,5,\dots,21$  ausgibt.

---

6. Schreibe ein Programm, das die kleinste natürliche Zahl  $n$  bestimmt, so dass

$$3^n > 1000000 \text{ ist. Verallgemeinere dein Programm.}$$

---

7. Schreibe ein Programm, das mit Hilfe einer Schleife feststellt, ab welcher natürlichen Zahl  $n$  die Summe der Kehrwerte der natürlichen Zahlen von 1 bis  $n$  größer als 5 wird.

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

Verallgemeinere dein Programm.

---

8. Schreibe ein Programm, das ermittelt nach wieviel Jahren verdoppelt sich ein Kapital von 1 € bei einem Zinssatz von 10% verdoppelt?

Verallgemeinere dein Programm.

---

9. Das Heronverfahren

Für jede positive Zahl  $x_0$  nähern sich die Zahlenwerte

$$x_1 = \frac{1}{2} \left( x_0 + \frac{2}{x_0} \right)$$

$$x_2 = \frac{1}{2} \left( x_1 + \frac{2}{x_1} \right) \text{ usw.}$$

immer mehr der Quadratwurzel von 2.

a) Schreibe ein Programm, das die Quadratwurzel einer eingegeben Zahl mit Hilfe des Heronsverfahrens näherungsweise bestimmt.

b) Schreibe ein Programm, das die kleinste natürliche Zahl bestimmt so, dass sich  $x_n$  um weniger als 0,001 von  $\sqrt{2}$  unterscheidet.

13. Schreibe ein Programm, das eine Liste aller Teiler einer Zahl ausgibt.

---

14. Eine Zahl heißt vollkommen, wenn sie gleich der Summe ihrer Teiler ohne sie selbst ist. Schreibe ein Programm, das Summe alle vollkommenen Zahlen zwischen 1 und 10000 bestimmt.

---

15. Schreibe ein Programm, das feststellt ob ein eingegebene Zahl  $p$  prim ist.

---

16. Bestimme alle ganzen Zahlen  $n$  und  $m$  mit  $1 \leq n, m \leq 100$ , die Lösungen der Gleichung  $3n - 5m = 1$  sind.

---

17. Bestimme alle natürlichen Zahlen  $a$  und  $b$  mit  $1 \leq a, b \leq 100$  für die  $a^2 + b^2$  ein Quadrat ist.

---

18. Schreibe ein Programm, das die Werte von  $1^3, 1^3 + 2^3, \dots, 1^2 + 2^3 + \dots + 9^3$  ausgibt.

Was vermutest du ?

---

19. Die sog. Fibonacci-Zahlen sind 0, 1, 1, 2, 3, 5, 8, ..., wobei jede Zahl die Summe der beiden vorhergehenden ist.

Schreibe ein Programm, das nach Eingabe einer Zahl  $n > 2$  die ersten  $n$  Fibonacci-Zahlen ausgibt.

---