

Sortieren mit Python	Informatik
	Insertionsort

Der Sortieralgorithmus „Insertionsort“

Aufgabe 1

Formuliere das Prinzip (den Algorithmus) des Sortierens im abgebildeten Schema. Gehe dabei näher auf die Bezeichnung „Sortieren durch Einfügen“ ein.

Verfahren 3: Insertionsort	Algorithmische Idee
98 – 22 – 34 – 48 – 23 – 37	
22 – 98 – 34 – 48 – 23 – 37	
22 – 34 – 98 – 48 – 23 – 37	
22 – 34 – 48 – 98 – 23 – 37	
22 – 23 – 34 – 48 – 98 – 37	
22 – 23 – 34 – 37 – 48 – 98	

Aufgabe 2

Führe das Sortierverfahren Sortieren durch Einfügen am folgenden Beispiel weiter durch. Dokumentiere alle Schritte¹.

```
[25]  [17 32 56 25 19 8 66 29 6 20 29]
      ^
[17 25] [32 56 25 19 8 66 29 6 20 29]
      ^
[17 25 32] [56 25 19 8 66 29 6 20 29]
          ^
[17 25 32 56] [25 19 8 66 29 6 20 29]
              ^
[17 25 25 32 56] [19 8 66 29 6 20 29]
                ^
```

¹ Hier wirkt es zwar so, dass alle Elemente aus einer unsortierten Liste in eine neue sortierte wandern, dem ist aber nicht so. Der komplette Prozess erfolgt innerhalb ein und derselben Liste. Die vorliegende Darstellung dient lediglich der besseren Veranschaulichung, was konkret während des Sortiervorgangs passiert.

Sortieren mit Python	Informatik
	Insertionsort

Aufgabe 3

- a) Entwickeln Sie Ideen, wie Sie Insertionsort in Python implementieren würden und versuchen Sie Ihre Ideen umzusetzen. Versuchen Sie dabei darauf zu achten, dass Ihre Programmierschritte für das Plenum nachvollziehbar bleiben (Kommentierungen / Erläuterungen etc. anfügen). Falls Sie nicht weiterkommen, orientieren Sie sich an Aufgabe 3 b bzw. fahren Sie mit dieser Aufgabe fort.
- b) Implementiere mithilfe des vorliegenden Pseudocodes den Sortieralgorithmus „Insertionsort“.
1. Funktionskopf mit Namen der Funktion und einem Parameter (Liste)
 2. for-Schleife, die, beginnend an der 1. Indexposition, die gegebene Liste durchläuft
 3. der aktuelle Wert der Indexposition wird in eine Variable „wert“ geschrieben
 4. die aktuelle Indexposition wird um 1 vermindert und in die Variabel „i“ geschrieben
 5. while-Schleife mit der Bedingung „i > 0“
 6. es wird geprüft, ob der aktuelle Wert kleiner als der aktuelle Wert des Indexes ist
 - a. wenn ja:
 - der aktuelle Wert des Indexes „i“ (Vergleichswert) wird um eine Stelle nach rechts verschoben (i+1)
 - der aktuelle Wert wird an die Stelle des Indexes „i“ verschoben (nach links)
 - die aktuelle Indexposition „i“ wird um 1 verringert
 - b. wenn nicht:
 - die Schleife wird abgebrochen
 7. die sortierte Liste wird ausgegeben
- c) Testen Sie Ihr Programm mit den Zahlen auf Seite 1 des Arbeitsblattes und versehen Sie es mit Kommentaren, die das Programm nachvollziehbar machen.
- Arbeitshinweis: Gehen Sie das Programm händisch auf einem Blatt Papier Schritt für Schritt durch bzw. versehen Sie das Programm mit print-Ausgaben, um es für Sie verständlich zu machen.
- d) Bereiten Sie sich darauf vor, Ihre Ergebnisse dem Plenum zu präsentieren und in diesem Zuge auch zu erläutern / zu erklären.