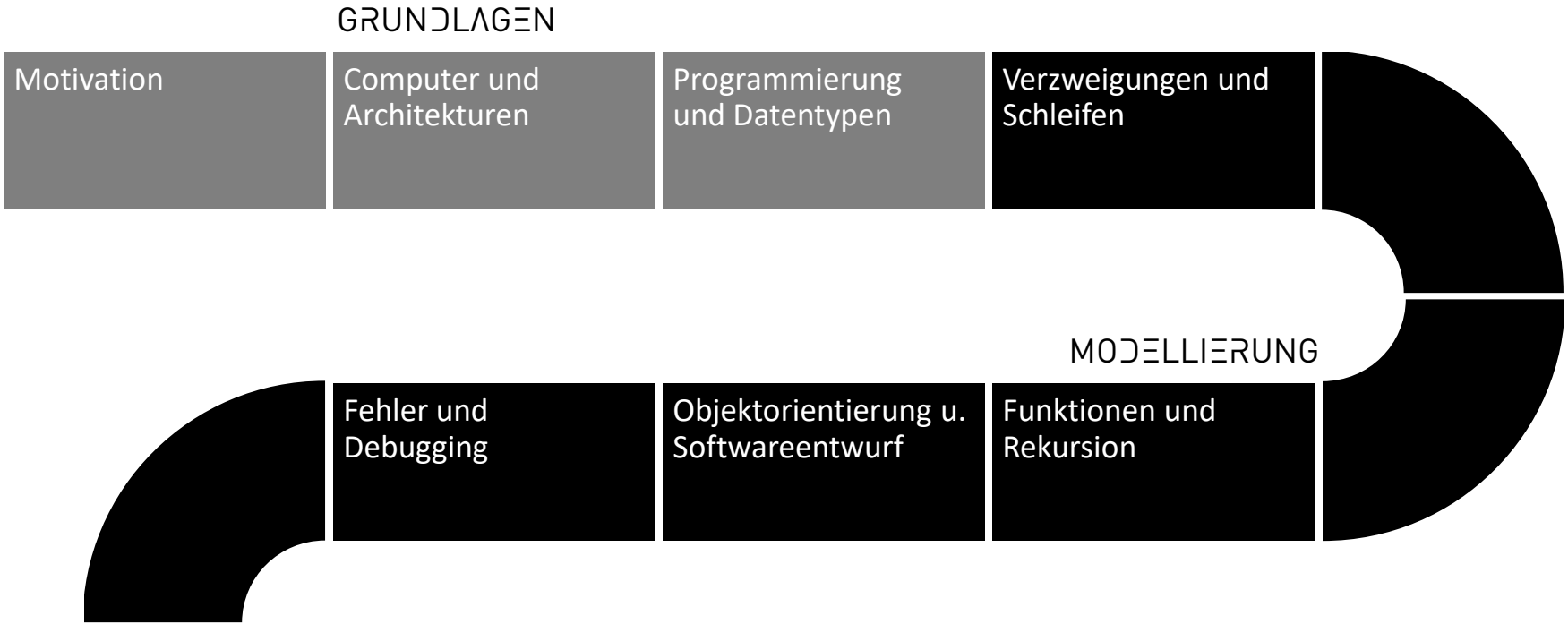
Einführung in Programmierung und Datenbanken

Joern Ploennigs

Programmierung und Datenbanken



Datentypen

DALL·E 2: Selfportrait

VARIABLEN

- Bei der Programmierung ist eine Variable ein Wert, der bei der Ausführung eines Computerprogramms auftritt und meist verändert werden kann. Eine Variable wird normalerweise im Quelltext durch einen Namen bezeichnet, hat einen Datentypen und eine Adresse im Speicher des Computers.

- Ein Wert der nicht verändert werden kann wird meist als Konstante bezeichnet.

Variablen in Python

Variablen in Python müssen nicht explizit deklariert werden (wie in vielen anderen Programmiersprachen), sondern werden durch Zuweisungsoperator (=) einem Wert zugeordnet.

- Beispiel:

```
a = 1   # a hat den Wert 1

a = 2  # a hat nun den Wert 2

a = "test"  # a hat nun den Wert "test"
```

Der Typ einer Variable kann mit der Funktion type(a) abgefragt werden.

Datentyp - Definition

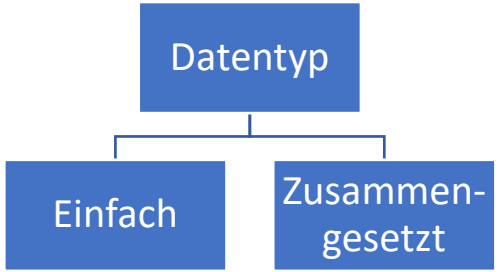
Der **Datentyp** gibt die *Art* der Daten sind, die mit ihm beschrieben werden und welche *Operationen* auf diesen ausgeführt werden können.

Datentyp

Datentypen - Struktur

Einfache Datentypen (Primitive Datentypen) können nur einen Wert des entsprechenden Wertebereichs aufnehmen.

Zusammengesetzte Datentypen (Komplexe Datentypen) sind ein Datenkonstrukt, das sich aus einfacheren Datentypen zusammensetzt. Da sie theoretisch beliebig komplex werden können, werden sie auch häufig schon zu den Datenstrukturen gezählt.



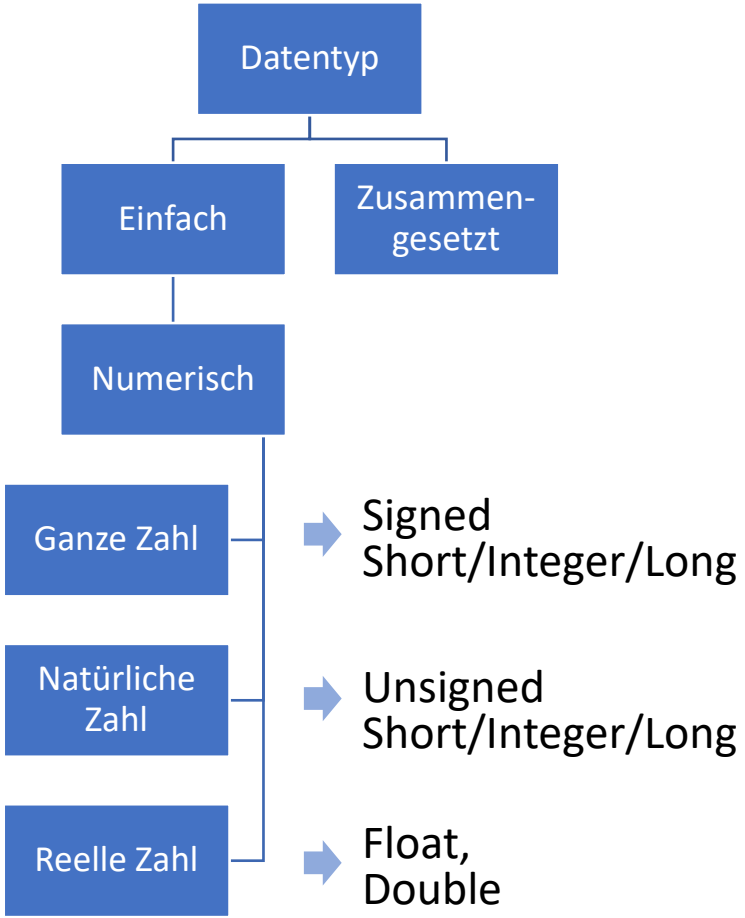
Einfache Datentypen - Numerisch

Numerische Datentypen repräsentieren Zahlen.

Ganze und Natürliche Zahlen werden als signed und unsigned Integer abgebildet. Nach Speicherkapazität unterscheidet man Short Integer (8 Bit), Integer (32 Bit) und Long (64 Bit).

Reelle Zahlen werden als Gleitkommazahl Float (32 Bit) oder Double (64 Bit) abgebildet.

Die unterschiedlichen Varianten werden in verschiedenen Programmiersprachen unterschiedlich bezeichnet.



Einfache Datentypen – Numerisch Gleitkommazahl

- Reelle Zahlen sind im Speicher eines Computers schwer zu repräsentieren
- Können extrem lang oder sogar unendlich lang sein

- Daher: angenäherte Darstellung durch **Gleitkommazahlen**
- Haben ein *Vorzeichen*, eine *Mantisse* m und einen *Exponenten* e

$$\pm m * 10^{e}$$

- Mantisse liegt immer zwischen 1 und 10
- D. h.: $257{,}6432 = +2{,}576432 * 10^{2}$

(Quelle: T. H. Kolbe, TU München, Informatik I)

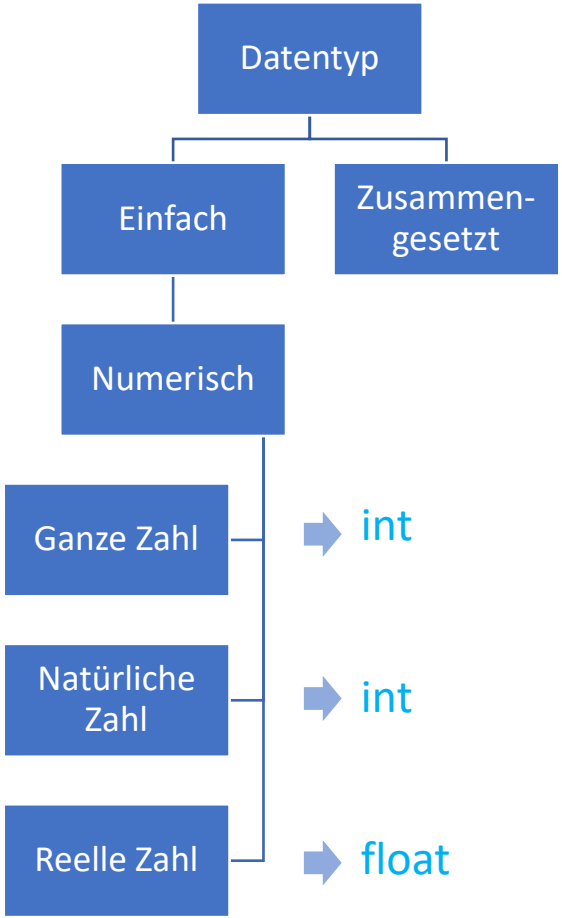
Einfache Datentypen – Numerisch in Python

Numerische Datentypen repräsentieren Zahlen.

Ganze und Natürliche Zahlen werden in Python als Integer int abgebildet.

Reelle Zahlen werden als Gleitkommazahl float abgebildet.

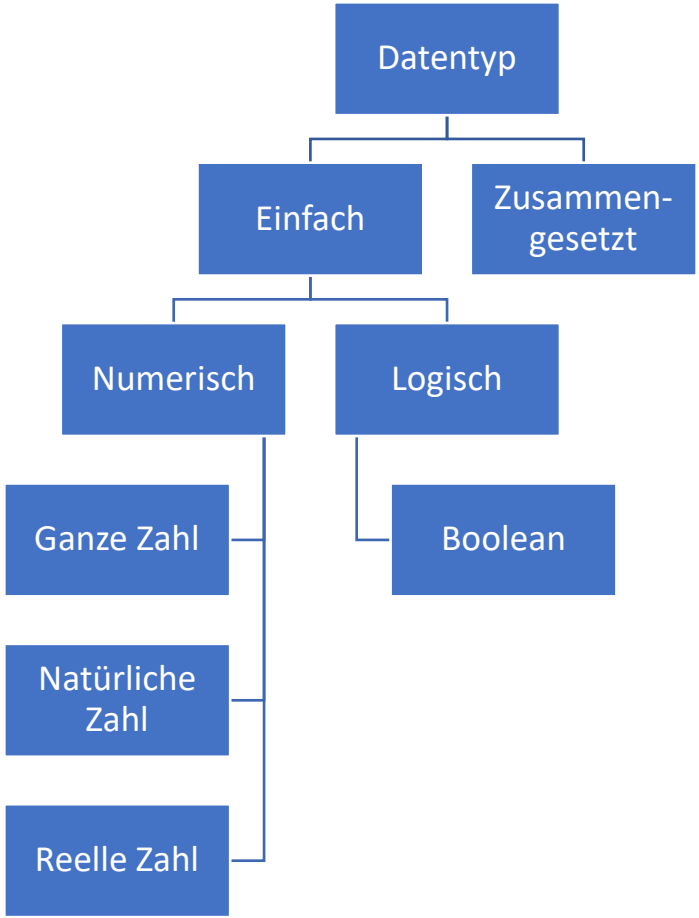
Anmerkung: Cython, eine Variante von Python die in C compiliert wird (also nicht interpretiert) unterscheidet hingegen zwischen signed short/int/long Typen und float/double um schneller zu laufen.



Einfache Datentypen - Logische

Logische Datentypen repräsentieren binäre Wahr/Falsch Werte.

Wahrheitswerte werden als boolesche Werte bezeichnet.



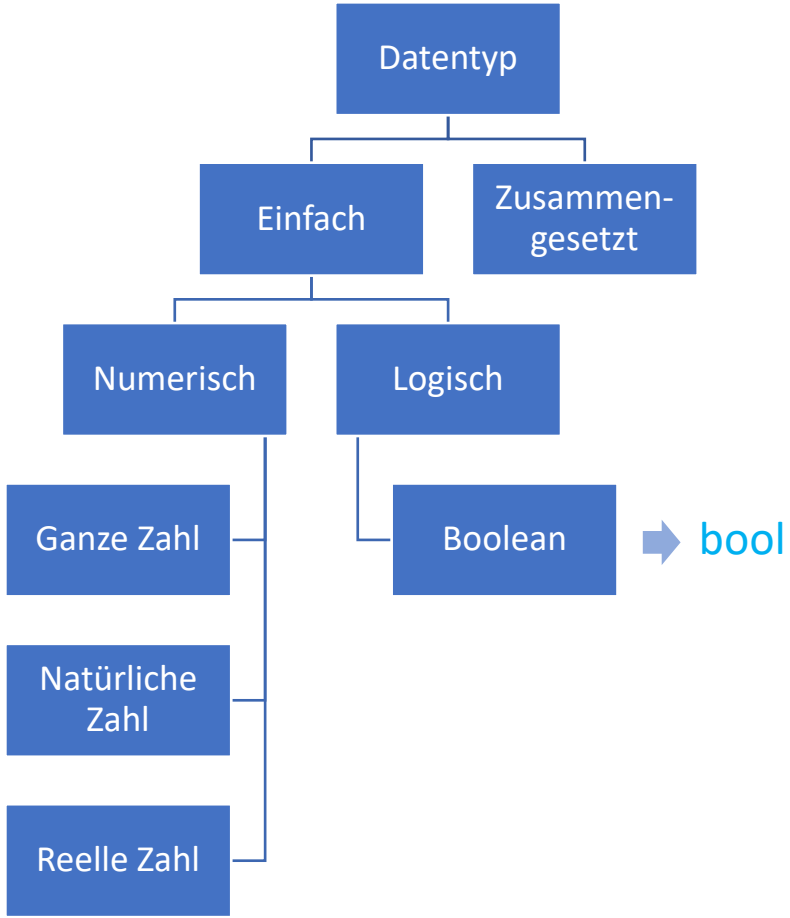
Einfache Datentypen – Logische in Python

Logische Datentypen repräsentieren binäre Wahr/Falsch Werte.

Wahrheitswerte werden als boolesche Werte bezeichnet.

In Python werden boolesche (binäre) Werte als bool bezeichnet.

Der Wert „Wahr“ wird in Python als True geschrieben und „Falsch“ als False.

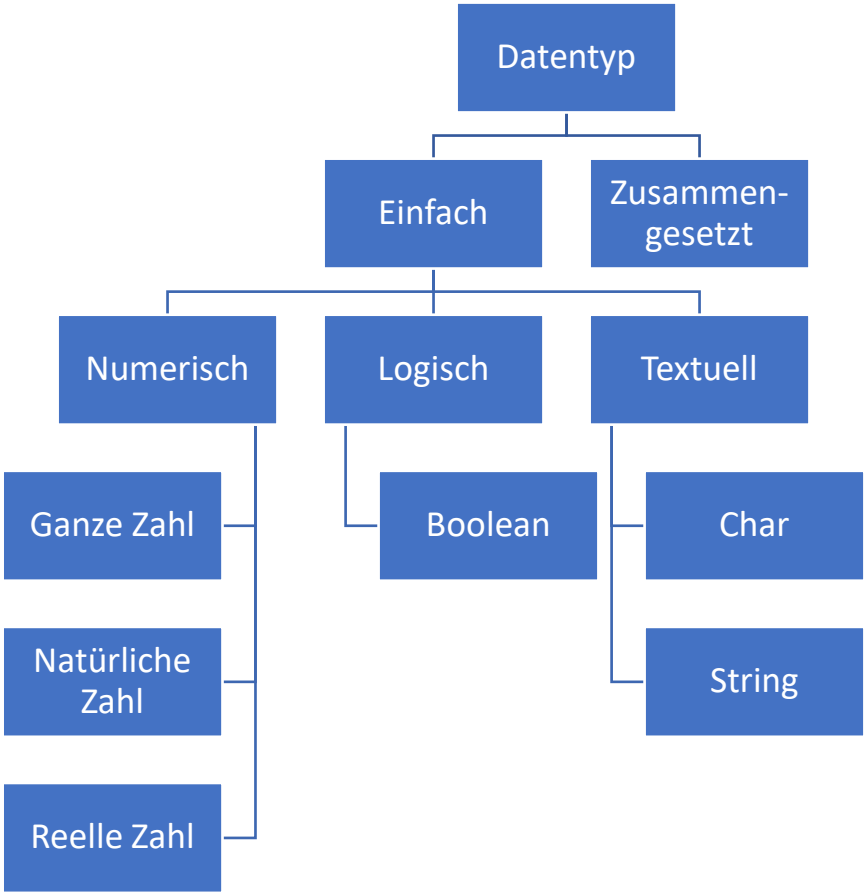


Einfache Datentypen - Textuell

Textuelle Datentypen repräsentieren Buchstaben.

Einzelne textuelle Zeichen werden als Char bezeichnet.

Mehrere textuelle Zeichen werden als String bezeichnet (Sie werden z. T. auch zu den zusammengesetzten Datentypen gezählt).



Einfache Datentypen – Textuell in Python

Textuelle Datentypen repräsentieren Buchstaben.

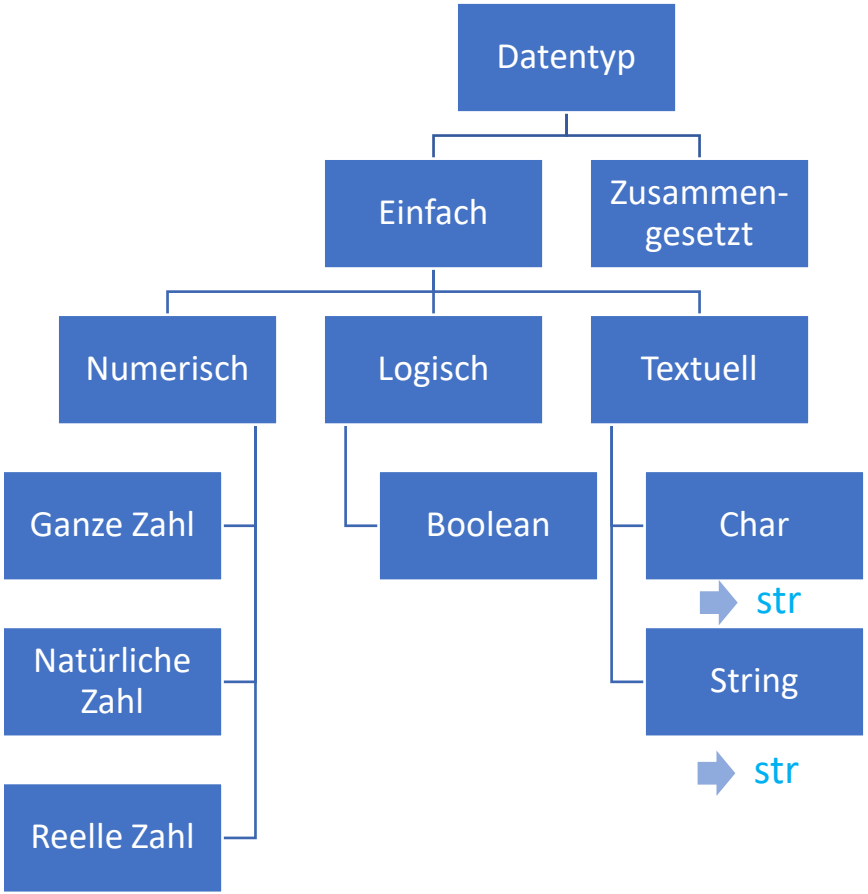
Einzelne textuelle Zeichen werden in Python als str mit der Länge 1 abgebildet.

Mehrere textuelle Zeichen werden in Python als str definiert.

Ein str in Python kann mit einem oder zwei Anführungsstrichen angefangen und beendet werden.

name = “Joern“

name = ‘Joern‘

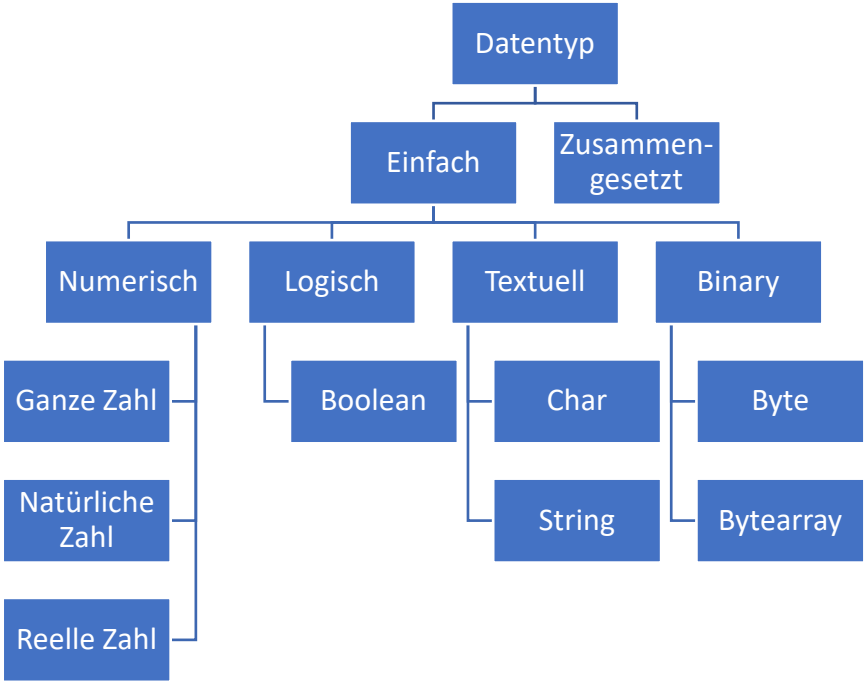


Einfache Datentypen - Binär

Binäre Datentypen können beliebige Zeichen repräsentieren.

Einzelne binäre Zeichen werden als Byte bezeichnet.

Mehrere Zeichen werden als Bytearray oder Bytestring bezeichnet (Sie werden z.T. auch zu den zusammengesetzten Datentypen gezählt).



Einfache Datentypen – Binär in Python

Binäre Datentypen können beliebige Zeichen repräsentieren.

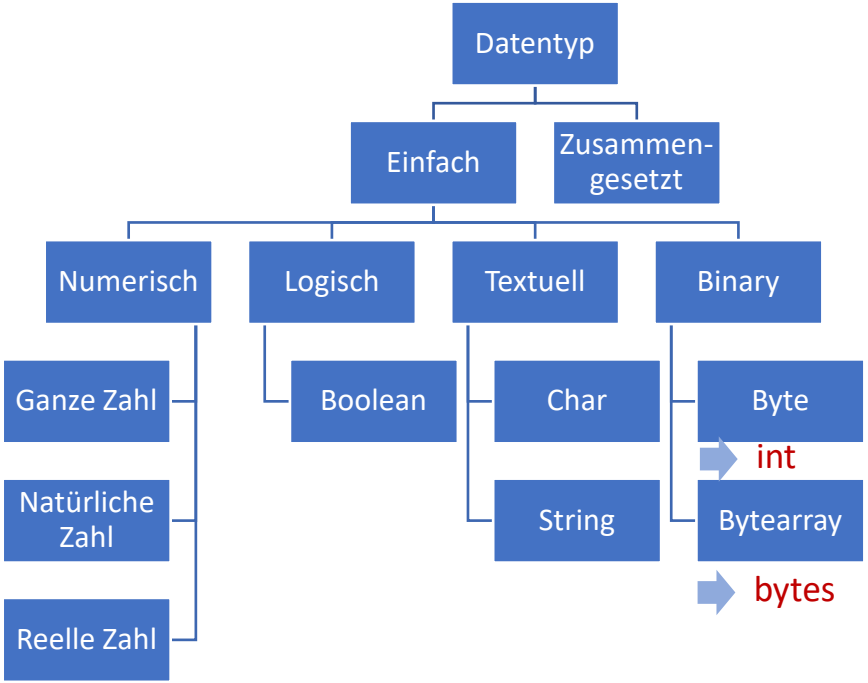
Einzelne binäre Zeichen werden in Python als int abgebildet.

Mehrere binäre Zeichen werden als bytes bezeichnet.

Ein bytes in Python wird als String deklariert mit einem führendem b.

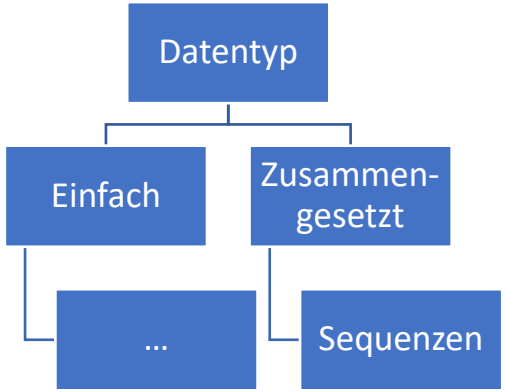
name = b“Joern“

name = b‘Joern‘



Zusammengesetzte Datentypen – Sequenzen

Sequenzen sind eine geordnete Abfolge an Werten, meist vom selben Datentyp.



Sequenzen werden in Programmiersprachen meist als Array bezeichnet. Arrays haben oft eine feste, unveränderliche Länge, die bei der Erzeugung definiert wird. Die Werte sind veränderlich.

Beispiel in Cython:

- Variablendeklaration :

  cdef int a = 5

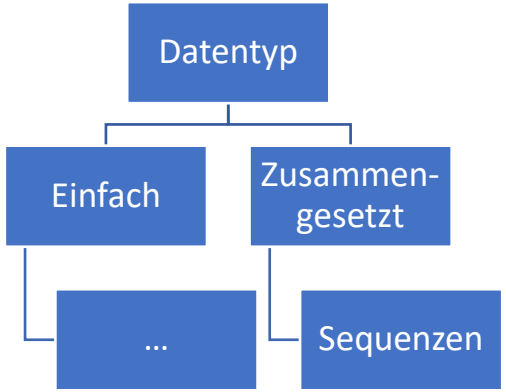
- Array-Deklaration :

  cdef int a[5] = [0, 1, 2, 3, 4]

Listen sind ein weiterer typischer Datentyp für Sequenzen. Listen haben oft keine feste Länge und können beliebig erweitert werden.

Zusammengesetzte Datentypen – Sequenzen in Python

Python unterstützt keine Arrays sondern nutzt list und tuple. Sie werden durch eckige oder runde Klammern deklariert.



Liste: x = [0, 1, 2, 3, 1]

Tuple: x = (0, 1, 2, 3, 1)

tuple haben eine feste Länge in Python. Sie sind unveränderlich (immutable).

Zusätzlich gibt es den besonderen Datentyp range um Sequenz von ganzen Zahlen zu erzeugen. Sie werden durch eckige Klammern mit drei Punkten deklariert.

Range: x = range(0,10)

Zusammengesetzte Datentypen – Sequenzen Zugriff auf Elemente

Um auf ein Element zuzugreifen wird die Elementzahl in eine eckige Klammer geschrieben.

In Python wird der Index in einer Liste ab 0 gezählt (ab 1 in einigen Programmiersprachen).

```
x[0] # greift also auf das erste Element zu
x[1] # greift auf das zweite Element zu
```

Eine Besonderheit in Python ist dass auch negative Indizes erlaubt sind um das Ende von Listen zuzugreifen (sowas nennt man unter Programmierern Syntax-Suggar)

```
x[-1] # greift auf das letzte Element zu
x[ len(x) – 1 ] # wäre die übliche umständliche Variante über die Länge len(x)
```

In Python kann durch Slicing auch auf Listenteile zugegriffen werden

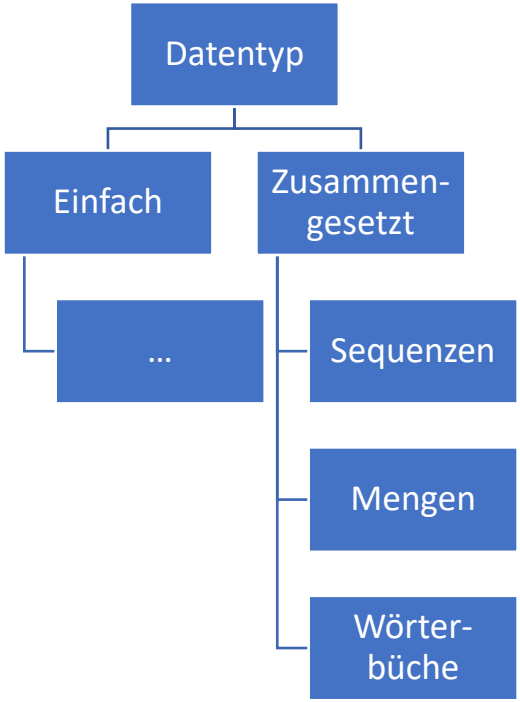
```
x[0:10] # greift auf die ersten zehn Elemente zu (ohne 10)
x[:10]   # greift auch auf die ersten zehn Elemente zu (ohne 10)
x[-10:] # greift auf die letzten zehn Elemente zu
```

Zusammengesetzte Datentypen – Mengen

Mengen stellen eine Menge an Werten ohne Wiederholungen dar, so wie in der Mathematik.

In den meisten Programmiersprachen werden Mengen als Set bezeichnet.

Es wird häufig auch zwischen Datentypen für sortierte und unsortierte Sets unterschieden.

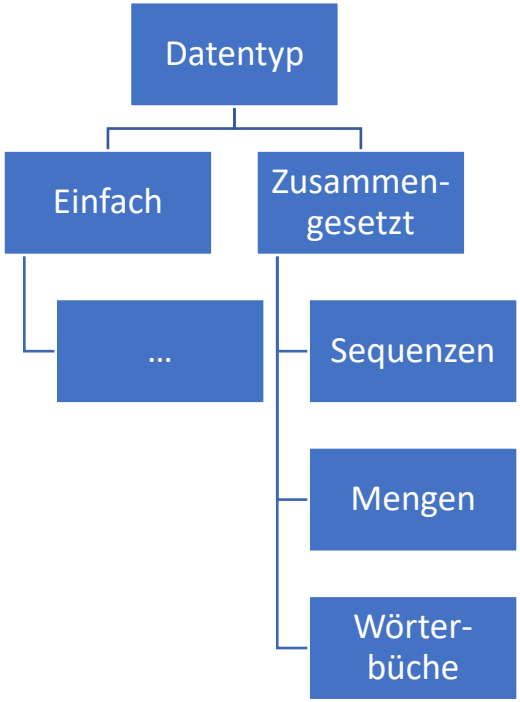


Zusammengesetzte Datentypen – Mengen in Python

Mengen stellen eine Menge an Werten ohne Wiederholungen dar, so wie in der Mathematik.

Der Datentyp für Mengen in Python heißt set. Sie werden durch geschweifte Klammern deklariert.

```
x = {0, 1, 2, 3}
```

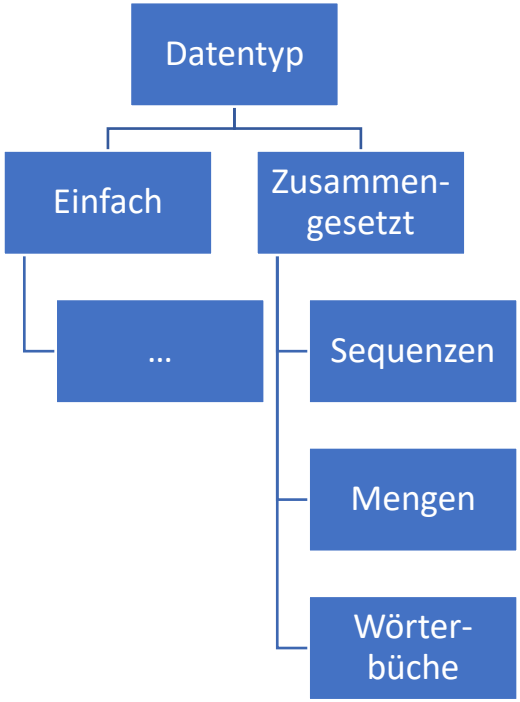


Zusammengesetzte Datentypen – Wörterbücher

- Wörterbücher bilden eine Abbildung von einer Menge an Schlüsseln auf eine Menge an Werten (Key-Value). Die Menge der Schlüssel darf keine Wiederholung aufweisen, die Menge der Werte schon.

- Wörterbücher werden in den meisten Programmiersprachen als Map (von engl. Mapping = Abbildung) bezeichnet.

- Anmerkung: Sets werden häufig intern als Map ohne Werte gespeichert, weil Schlüssel keine Dopplung aufweisen dürfen.



Zusammengesetzte Datentypen – Wörterbücher in Python

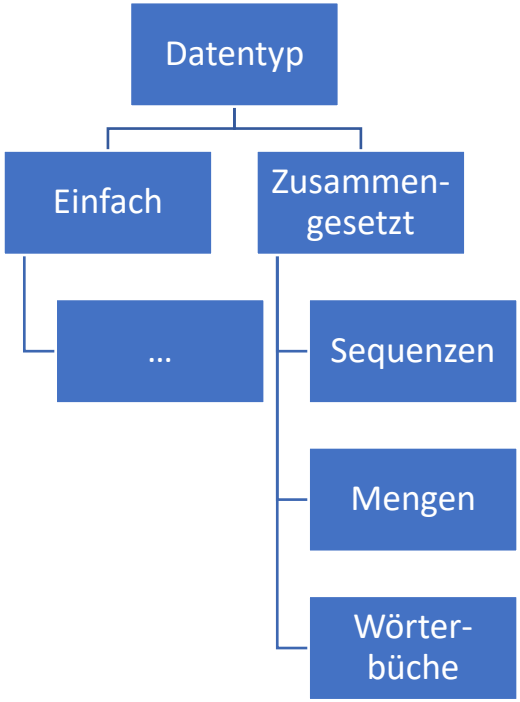
Wörterbücher bilden eine Abbildung einer Menge an Schlüsseln auf eine Menge an Werten (Key-Value).

Wörterbücher werden in Python als dict bezeichnet. Sie werden durch geschweifte Klammern und Schlüssel/Wert-Paaren definiert.

```
x = {
        "Gebäudetyp":"Wohnhaus",
        "Baujahr":2022
}
```

neue Werte können auch dynamisch zugewiesen werden:

```
x["Bauweise"] = "Holzbauweise"
```



Zusammengesetzte Datentypen – Wörterbücher Zugriff auf Elemente

Um auf ein Element in einem dict zuzugreifen wird der Schlüssel in einer eckige Klammer geschrieben.

```
x["Baujahr"]
```

Anmerkung: Dies funktioniert nicht bei Mengen (set), da ja kein Wert dahinter steht, sondern nur interessant ist ob der Schlüssel in der Menge enthalten ist. Was man so abfragt

```
"Baujahr" in x
```

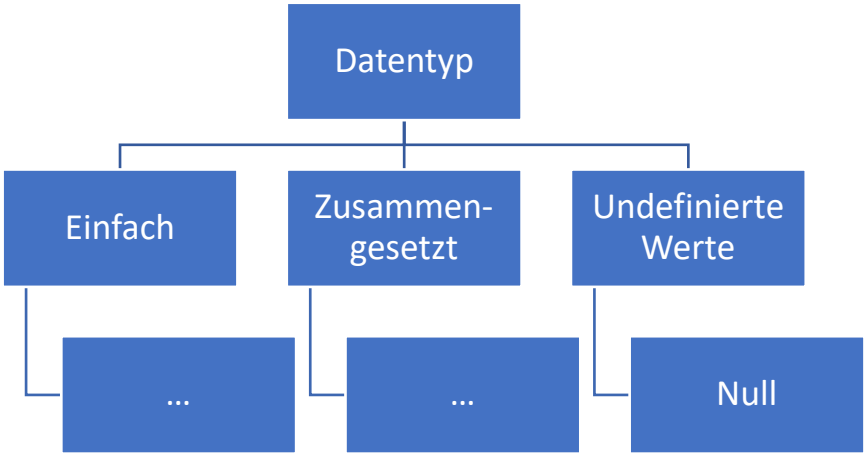
Um Werte aus einem dict oder set zu entfernen, nutzt man

```
del x["Baujahr"]
```

Datentypen – Undefinierte Werte

In vielen Programmiersprachen gibt es auch einen Wert um einen undefinierten Wert darzustellen, z. B. wenn etwas nicht da ist.

Dieser undefinierte Wert wird häufig als Null-Wert bezeichnet.



Anmerkung: Die Notwendigkeit für einen Null-Wert ist heutzutage sehr umstritten, da Null-Werte in Programmiersprachen sehr viele Fehler erzeugen. Deshalb haben einige moderne Sprachen keinen Null-Wert.

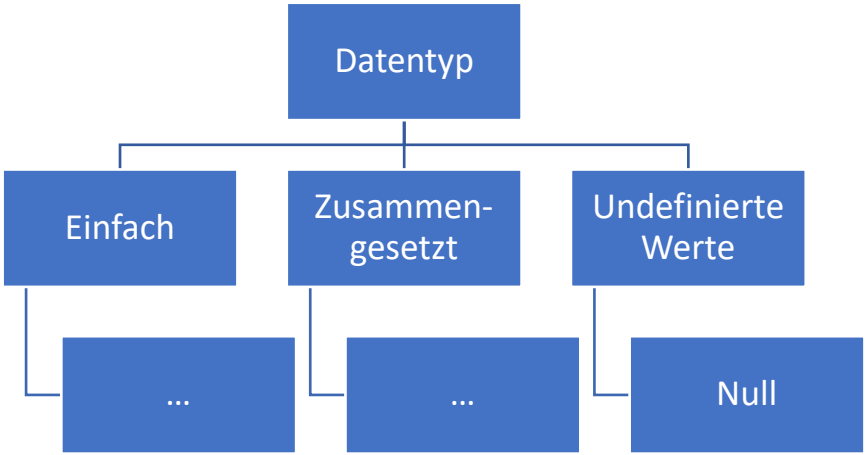
Datentypen – Undefinierte Werte in Python

In vielen Programmiersprachen gibt es auch einen Wert um einen undefinierten Wert darzustellen, z. B. wenn etwas nicht da ist.

Der Null-Wert in Python heißt `None`.

Bedeutet dass eine Variable keinen Wert zugewiesen hat, oder eine Operation keinen Wert zurück gibt.

Der Datentyp einer Variable vom Wert `None` ist NoneType.



Datentypen – Modifizierbarkeit (Mutability)

- Besonders für Listen ein wichtiges Konzept – beschreibt die Veränderbarkeit von Datenstrukturen
- Ist ein Datentyp *mutable* können Variablen von diesem Typ direkt verändert werden.
- Ist er *immutable* kann man diese nur durch eine komplette Neubelegung verändern.

Mutable	Immutable
bytearray	bytes
set	frozenset
list	tupel

Um Programmierfehler zu verhindern und Zugriffssicherheit zu gewährleisten, unterscheiden einige Programmiersprachen sehr strikt zwischen mutable und immutable Datentypen.

HÖRSAALFRAGE

FRAGEN?

DALL·E 2: A psychedelic DJ with a question mark for a head