

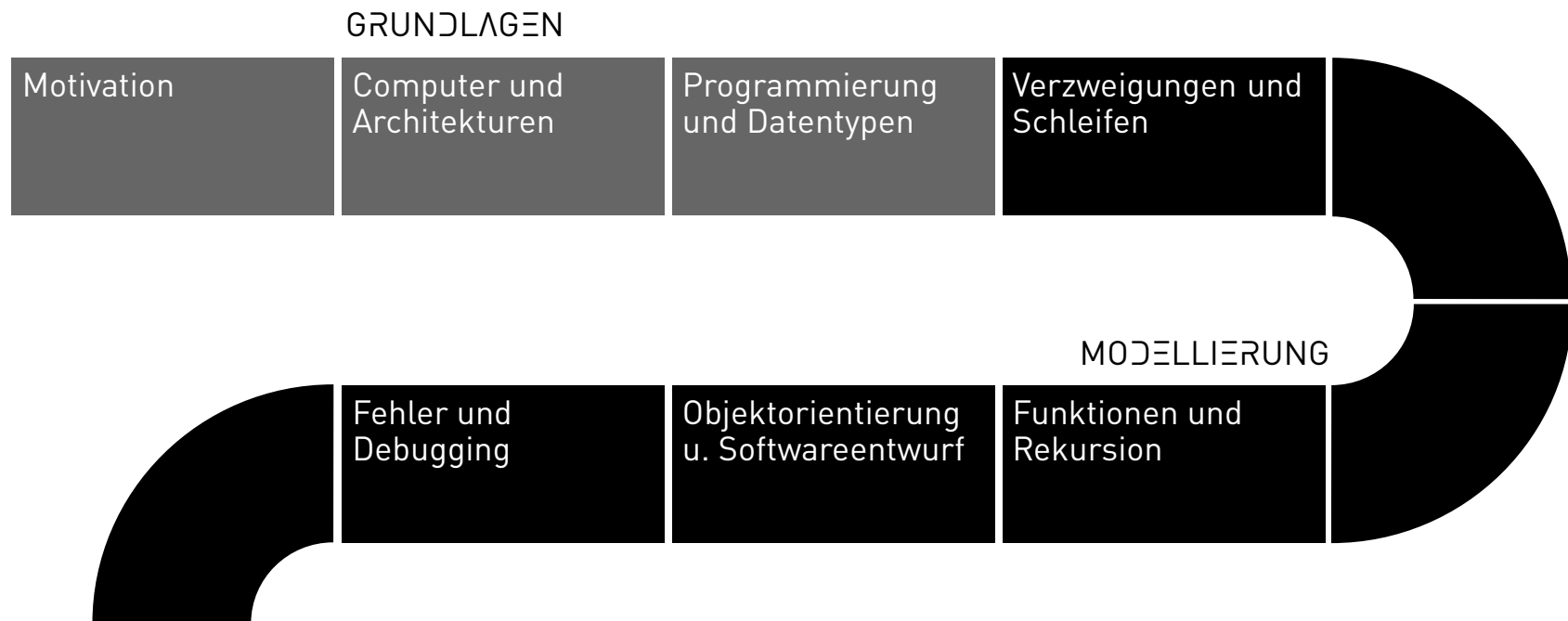
programmierung und datenbanken

Joern Ploennigs

Datentypen

MIDJOURNEY: DATATYPE JUNGLE, REF. HENRI ROUSSEAU

ABLAUF



VARIABLEN



Definition: 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.



Definition: Konstante

Eine Konstante ist ein Wert, der nicht verändert werden kann, nachdem er einmal zugewiesen wurde.

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.

VARIABLENNAMEN

- Verwenden Sie Namen, die die Bedeutung und den Inhalt der Variablen klar vermitteln.
- Vermeiden Sie übermäßig generische Namen wie "Daten" oder "v".
- Halten Sie die Namenskonventionen in Ihrem Code konsistent.
- Verwenden Sie Kleinbuchstaben mit Unterstrichen (_) zur Trennung von Wörtern (Snake Case), z.B. `meine_variable`.
- Vermeiden Sie Sonderzeichen wie ä, ö, ü, ß da diese in der Zeichenkodierung Probleme verursachen können.
- Verwenden Sie keine reservierten Schlüsselwörter der Programmiersprache als Variablennamen (z.B. `if`, `for`, `while`).
- Vermeiden Sie Abkürzungen oder Akronyme, es sei denn, sie sind allgemein verständlich (z.B. "BIP").
- Streben Sie nach einem Gleichgewicht zwischen Klarheit und Kürze.

BEISPIELE FÜR GUT UND SCHLECHTE VARIABLENNAMEN

Gut	Schlecht	Beschreibung
age	a	Steht eindeutig für das Alter einer Person.
first_name	fn	Beschreibt eindeutig den Vornamen.
birth_year	by	Eindeutig und verständlich für das Geburtsjahr.
email_address	email	Verdeutlicht, dass es sich um eine E-Mail-Adresse handelt.
is_on	on	is_ zeigt, dass es sich um einen Wahrheitswert handelt.
product_list	products	Macht deutlich, dass es sich um eine Liste handelt.
score_total	score	Zeigt, dass es sich um die Gesamtsumme handelt.
user_count	count	Gibt an, dass es sich um die Anzahl der Nutzer handelt.

DATENTYP

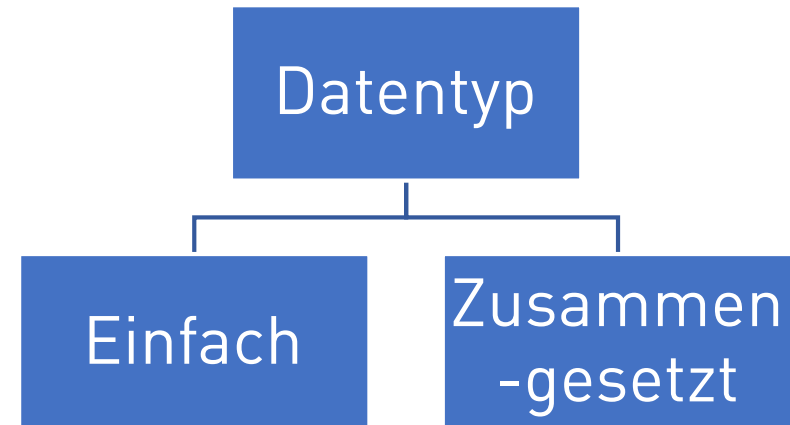


Definition: Datentyp

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

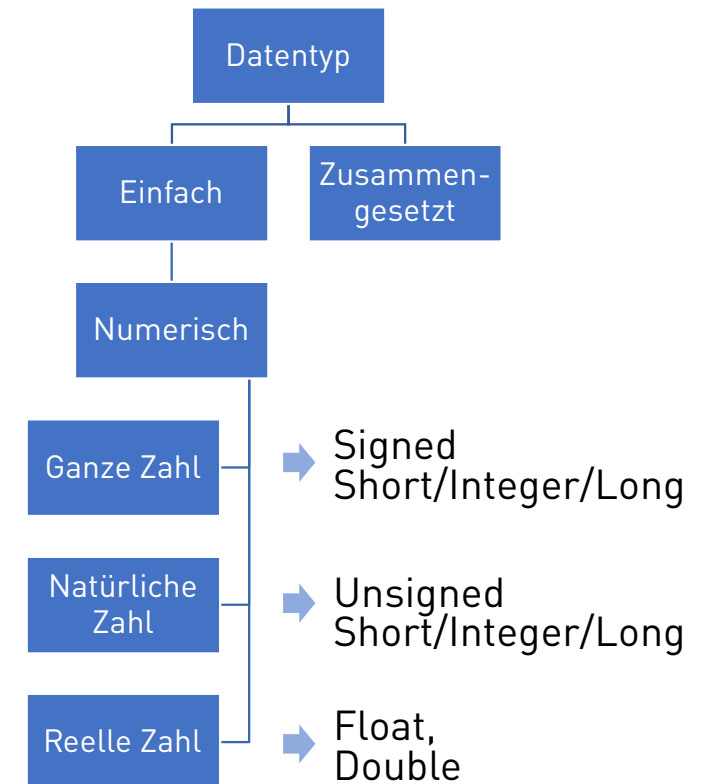
DATENTYPEN - STRUKTUR

- 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.



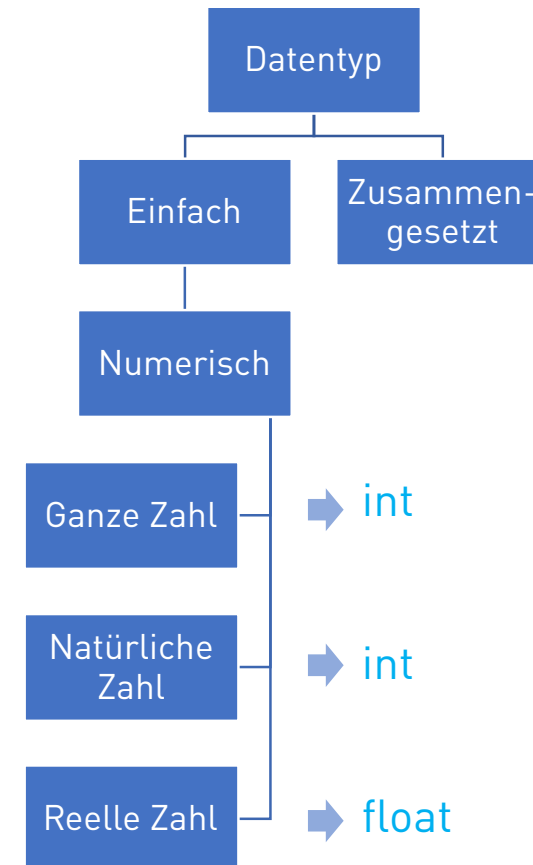
EINFACHE DATENTYPEN - NUMERISCH

- 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.
- Diese Varianten werden in verschiedenen Programmiersprachen unterschiedlich bezeichnet.



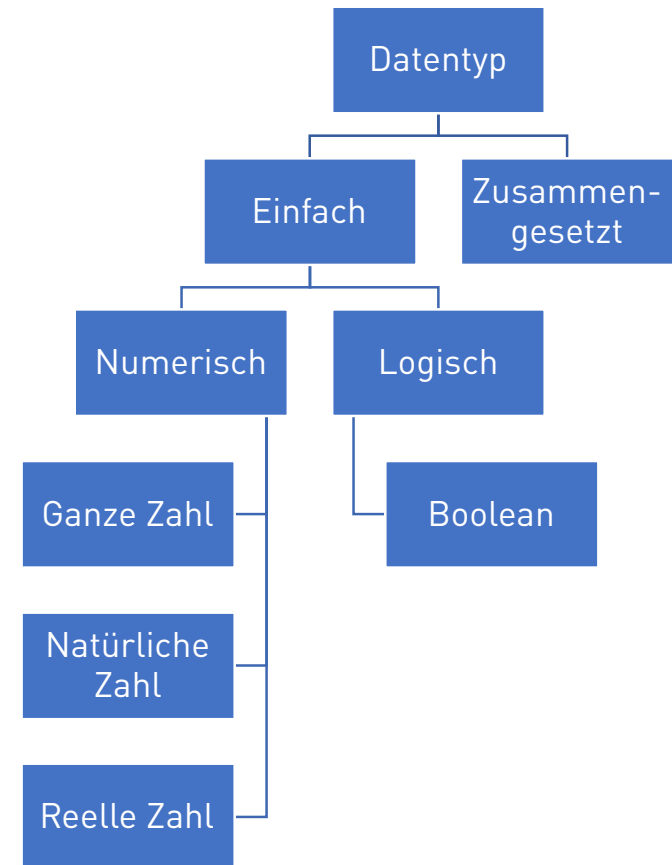
EINFACHE DATENTYPEN – NUMERISCH IN PYTHON

- Ganze und Natürliche Zahlen werden in Python als Integer `int` abgebildet und nicht unterschieden.
- Reelle Zahlen werden als Gleitkommazahl `float` abgebildet.
- **Info:** Cython, eine typisierte Variante von Python die nach C compiliert wird (also nicht interpretiert) unterscheidet zwischen `(signed) short/int/long` Typen und `float/double` um schneller zu laufen.



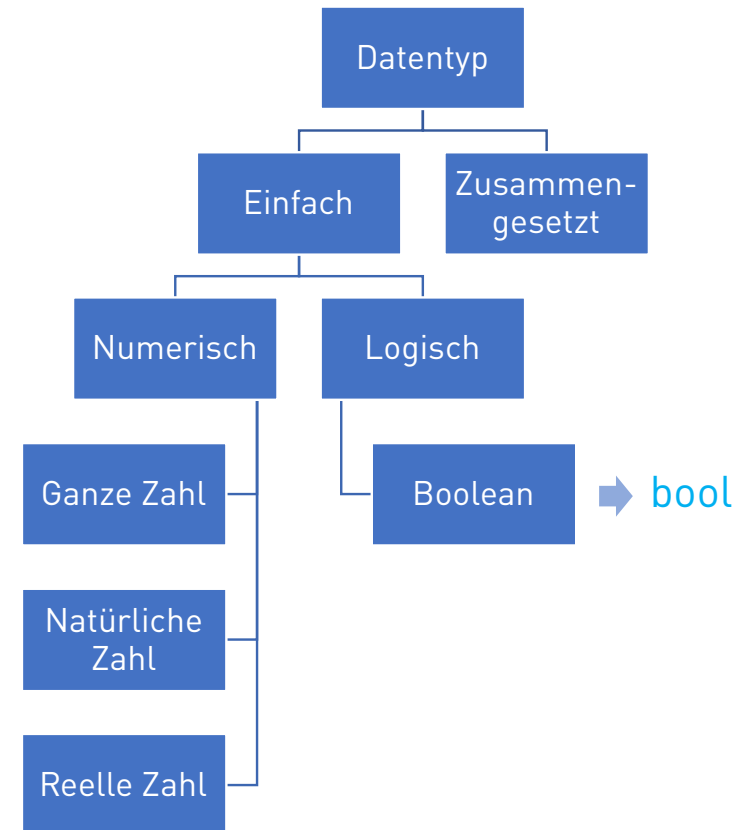
EINFACHE DATENTYPEN - LOGISCHE

- Logische Datentypen repräsentieren binäre Werte wie Wahr oder Falsch.
- Wahrheitswerte werden als boolesche Werte bezeichnet.



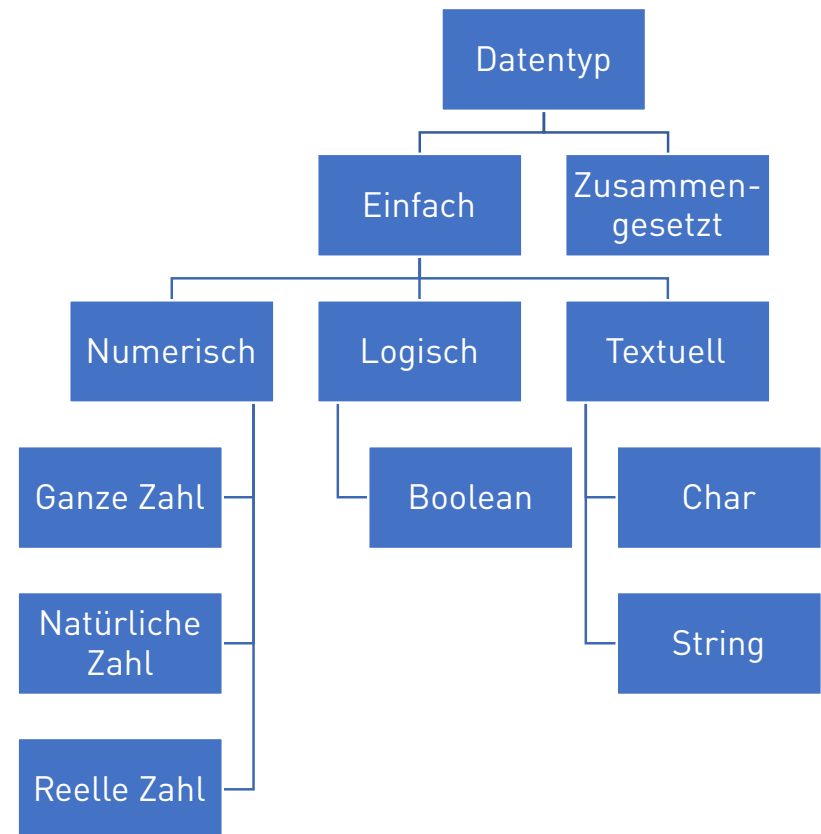
EINFACHE DATENTYPEN – LOGISCHE IN PYTHON

- 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

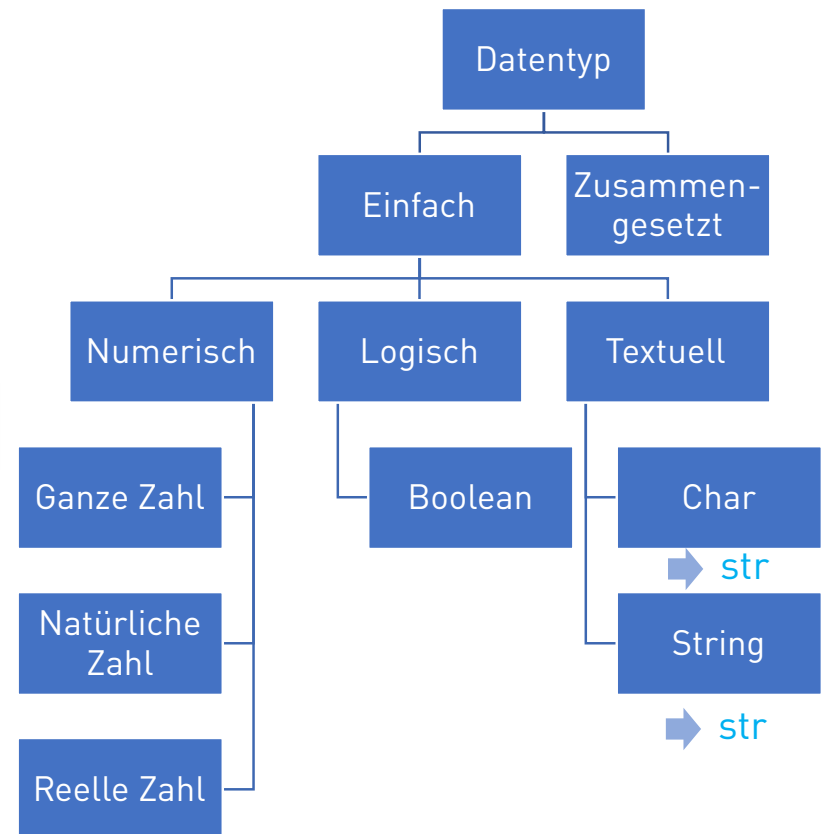
- Textuelle Datentypen repräsentieren Buchstaben.
- Einzelne textuelle Zeichen werden als Char bezeichnet.
- Mehrere textuelle Zeichen werden als String bezeichnet (Sie werden z. T. auch zu den zusammengesetzten Datentypen gezählt).



EINFACHE DATENTYPEN – TEXTUELL IN PYTHON

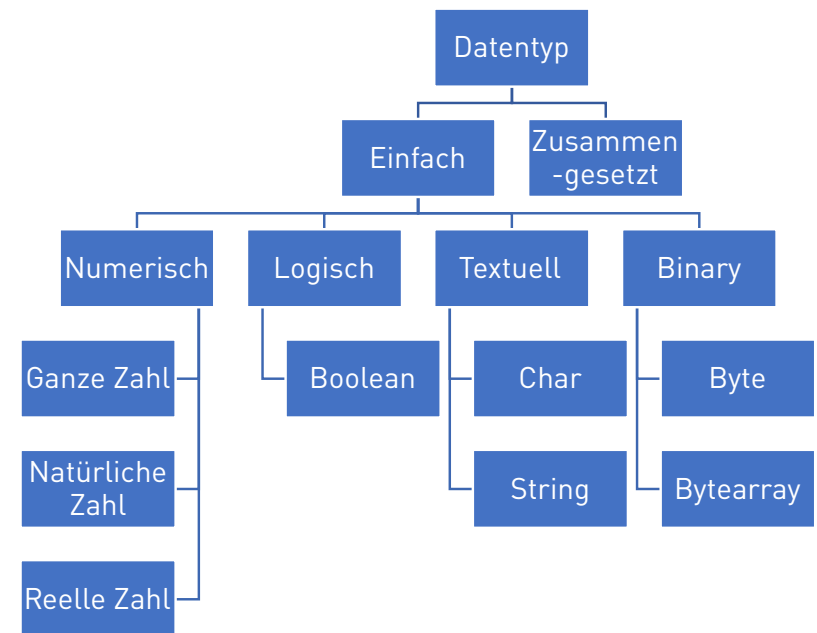
- Einzelne textuelle Zeichen werden in Python als `str` mit der Länge 1 abgebildet.
- Mehrere textuelle Zeichen werden in Python als `str` definiert.
- Ein `str` in Python kann mit einem `'` oder zwei `"` Anführungsstrichen angefangen und beendet werden.

```
name = "Joern"  
name = 'Joern'
```



EINFACHE DATENTYPEN - BINÄR

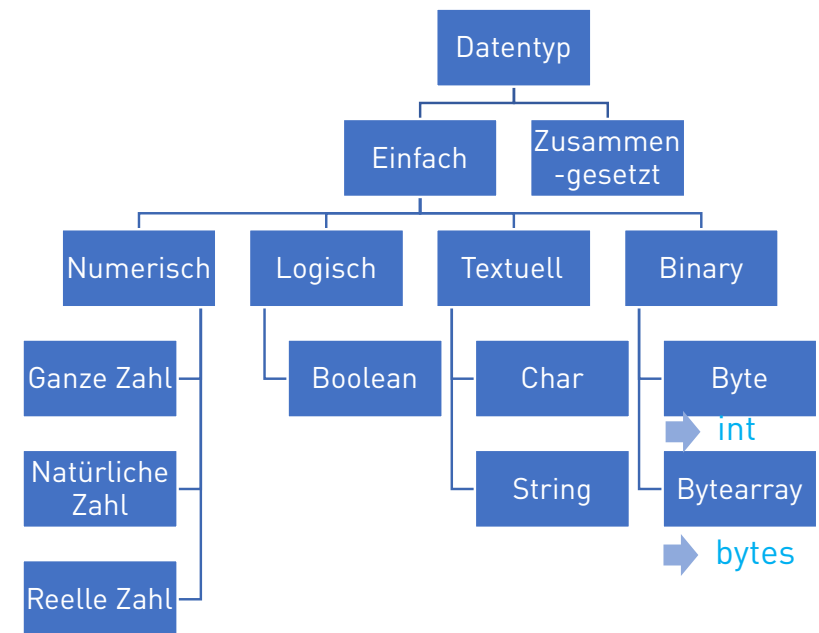
- Binäre Datentypen können beliebige Zeichen repräsentieren.
- Einzelne binäre Zeichen werden als Byte bezeichnet.
- Mehrere Zeichen werden als Bytearray oder Bytestring bezeichnet (Sie werden z.T. auch zu den zusammengesetzten Datentypen gezählt).



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'
```



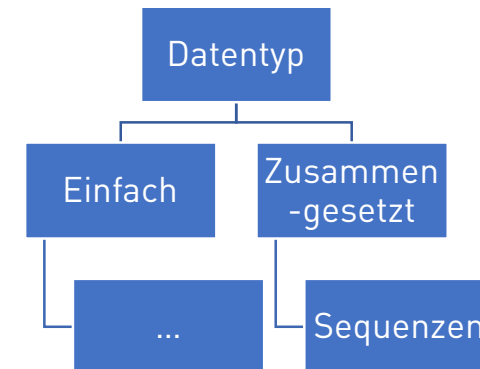
ZUSAMMENGESETZTE DATENTYPEN – SEQUENZEN

- 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:

```
cdef int a = 5 #  
Variablendeklaration  
cdef int a[5] = [0, 1, 2, 3, 4]  
#Array-Deklaration
```

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



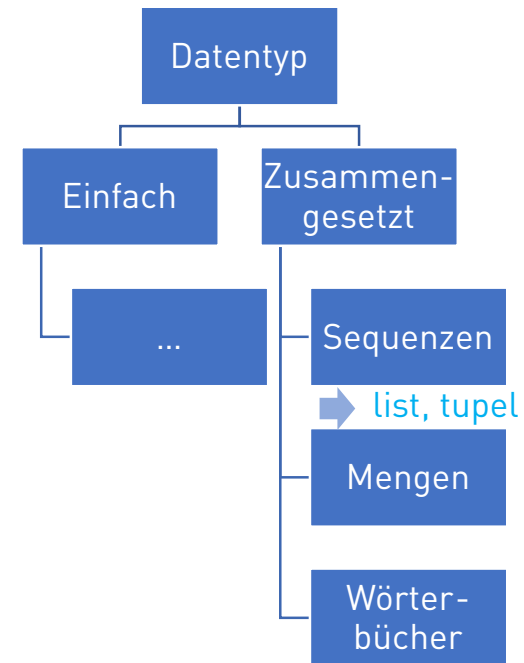
ZUSAMMENGESETZTE DATENTYPEN – SEQUENZEN IN PYTHON

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

```
x = [0, 1, 2, 3, 1] # Liste  
x = (0, 1, 2, 3, 1) # Tuple
```

- `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.

```
x = range(0,10) # Range Funktion
```



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 Sprachen).

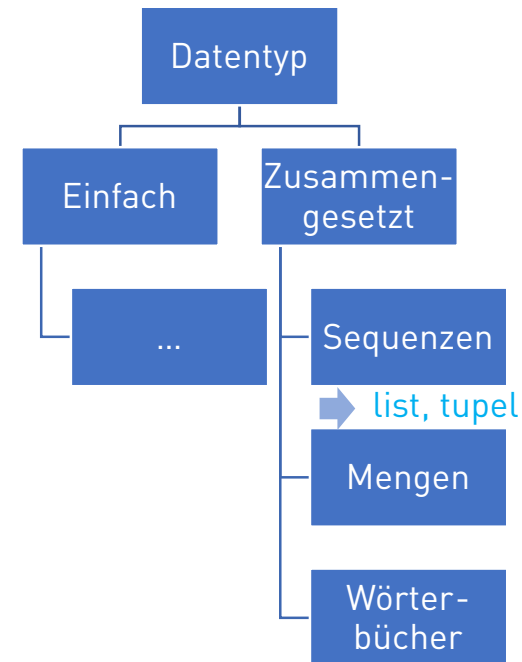
```
x[0] # 1. Element  
x[1] # 2. Element
```

- Eine Besonderheit in Python ist dass auch negative Indizes erlaubt sind um das Ende von Listen zuzugreifen (a.k.a. Syntax-Suggar)

```
x[-1] # letzte Element  
x[len(x)-1] # alternatives  
Vorgehen
```

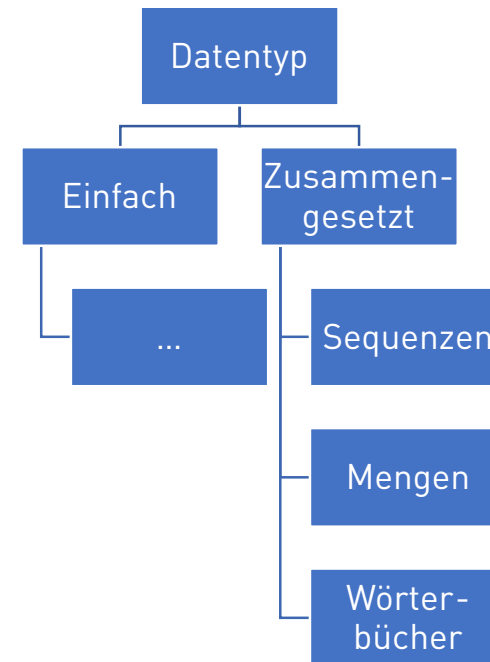
- In Python kann durch Slicing auch auf Listenteile zugegriffen werden

```
x[0:10] # ersten 10 Elemente  
(ohne 10)  
x[:10] # ersten 10 Elemente  
(ohne 10)  
x[-10:] # letzten 10 Elemente
```



ZUSAMMENGESETZTE DATENTYPEN – MENGEN

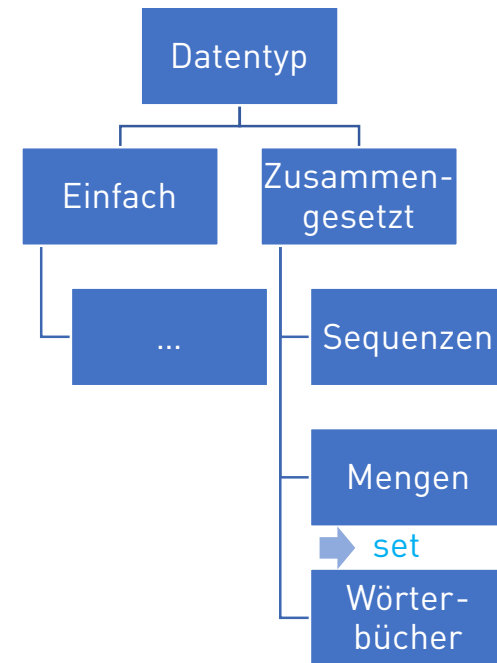
- Mengen stellen eine Menge an Werten ohne Wiederholungen dar, so wie in der Mathematik.
- In den meisten Programmiersprachen werden Mengen als Set bezeichnet.
- Es wird häufig auch zwischen Datentypen für sortierte und unsortierte Sets unterschieden.



ZUSAMMENGESETZTE DATENTYPEN – MENGEN IN PYTHON

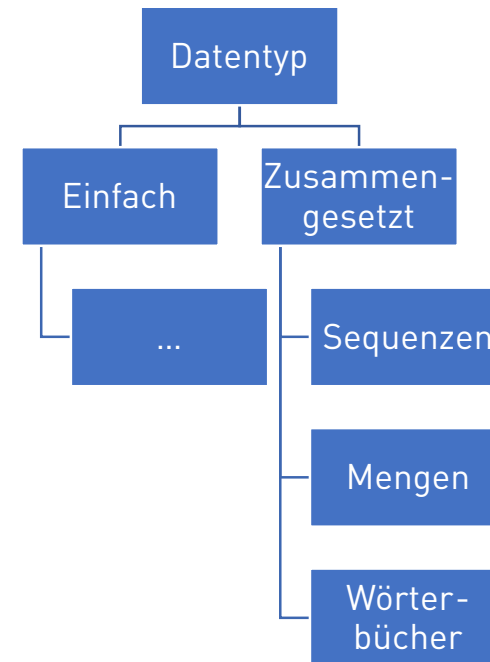
- Der Datentyp für Mengen in Python heißt `set`. Sie werden durch geschweifte Klammern deklariert.

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



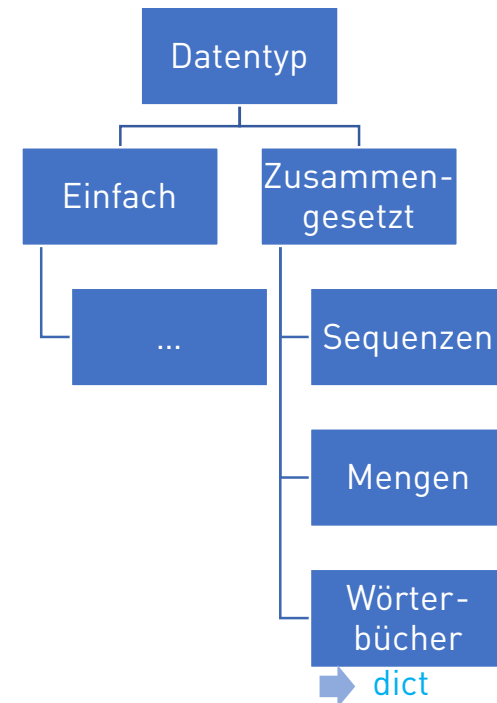
ZUSAMMENGESETZTE DATENTYPEN – WÖRTERBÜCHER IN PYTHON

- 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.

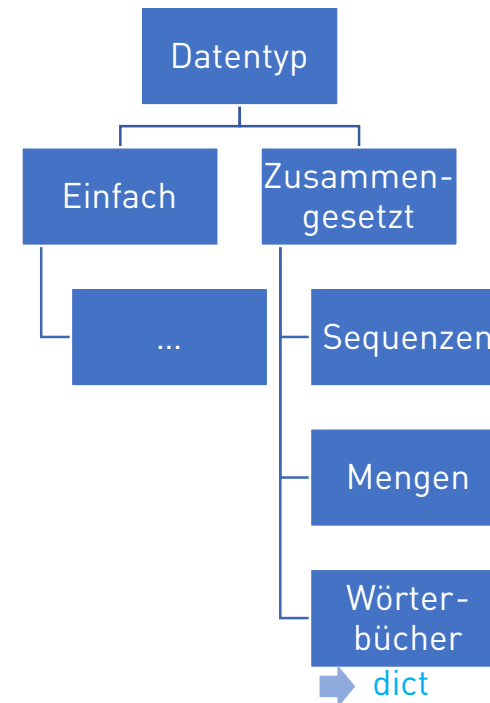
```
x["Baujahr"]
```

- 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

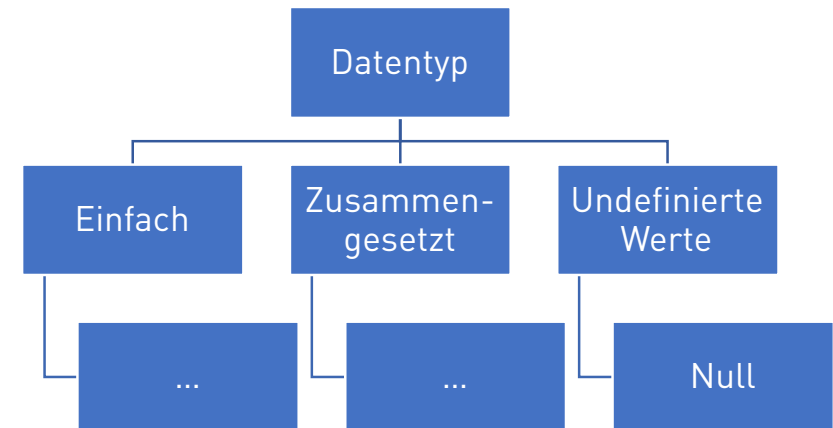
```
del x["Baujahr"]
```



DATENTYPEN – UNDEFINIERTER WERT

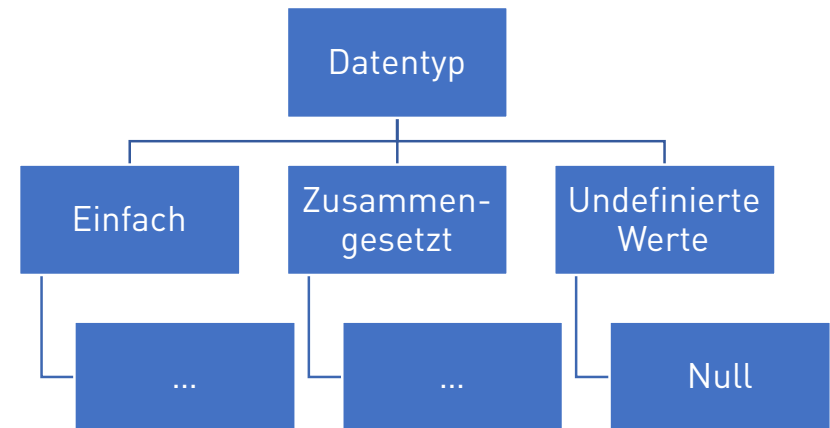
- In vielen Programmiersprachen gibt es auch einen Wert um einen undefinierten Wert darzustellen, z.B. wenn etwas nicht da ist.
- Dieser undefinierte Wert wird häufig als **Null**-Wert bezeichnet.

Die Notwendigkeit für einen Null-Wert ist heutzutage sehr umstritten, da Null-Werte schnell Fehler erzeugen. Deshalb haben einige moderne Sprachen keinen Null-Wert.



DATENTYPEN – UNDEFINIERTER WERT IN PYTHON

- 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)

Definition: Mutability

Modifizierbarkeit 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.

Mutabel (änderbar) Immutabel (nicht änderbar)

<code>list</code>	<code>tuple</code>
<code>set</code>	<code>frozenset</code>
<code>dict</code>	<code>frozendict</code>
<code>bytearray</code>	<code>bytes</code>

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

fragen?

