

Einleitend

- Installiert Python 3.8
- <https://gogs.naeefe.eu/>

Nicht-Ziele

- Python-Grundkurs
- Vollständiger Überblick über Python
- Python 3.8 Review

Roadmap

- Slots
- Dataclasses
- Decorators
- Pause

- Python3.8
 - f-strings
 - Walrus Operator
 - Position-only Arguments
- Context Managers
- Collections
- Pause

- Dictionary Implementation
- Debugging
- Python Language Services

Collections

- Spezialisierte Daten Container
- Collections

Dict Implementation

- hash tables
- Index des Elements wird berechnet auf Basis des Hashes des Keys

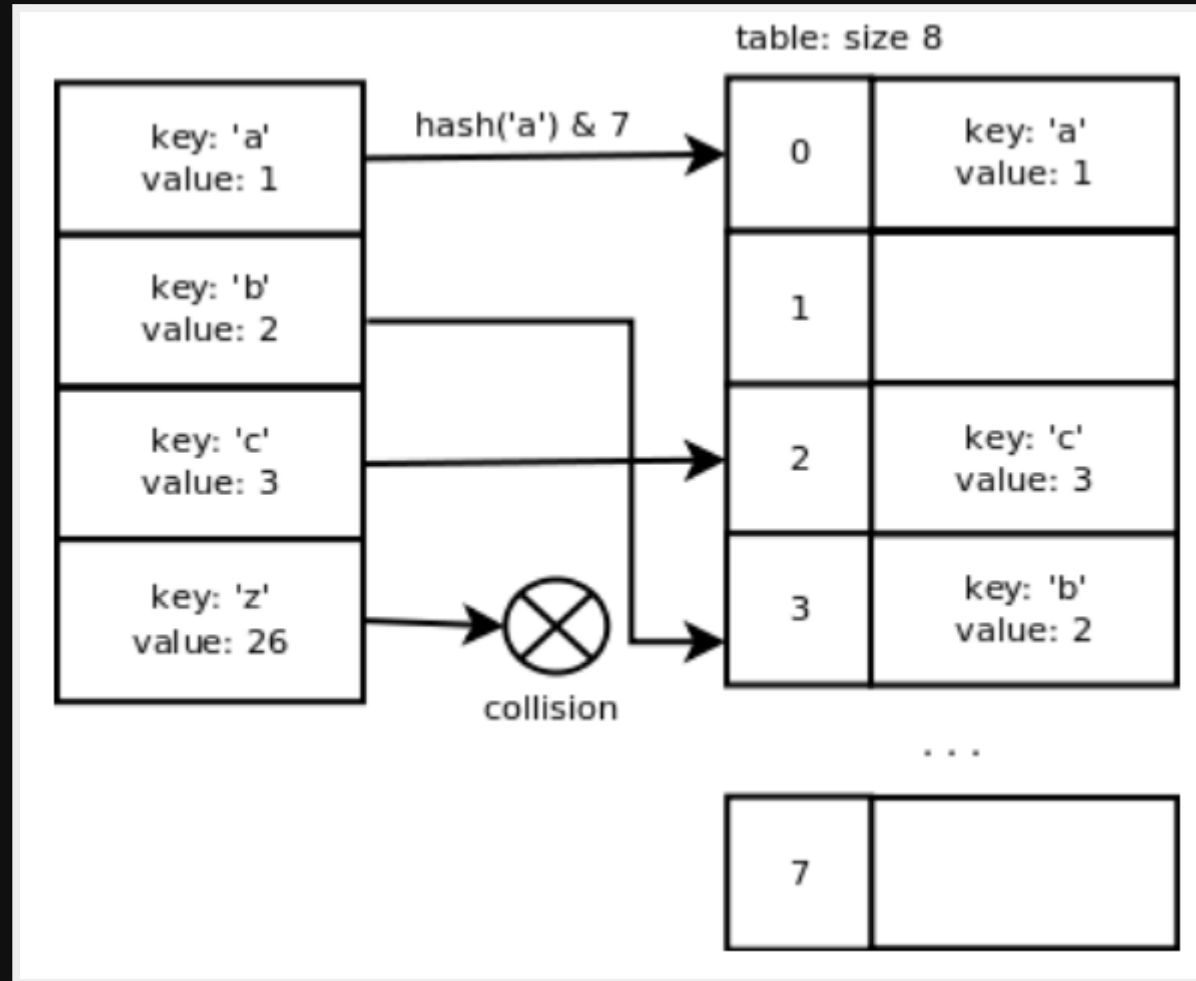
Structs I

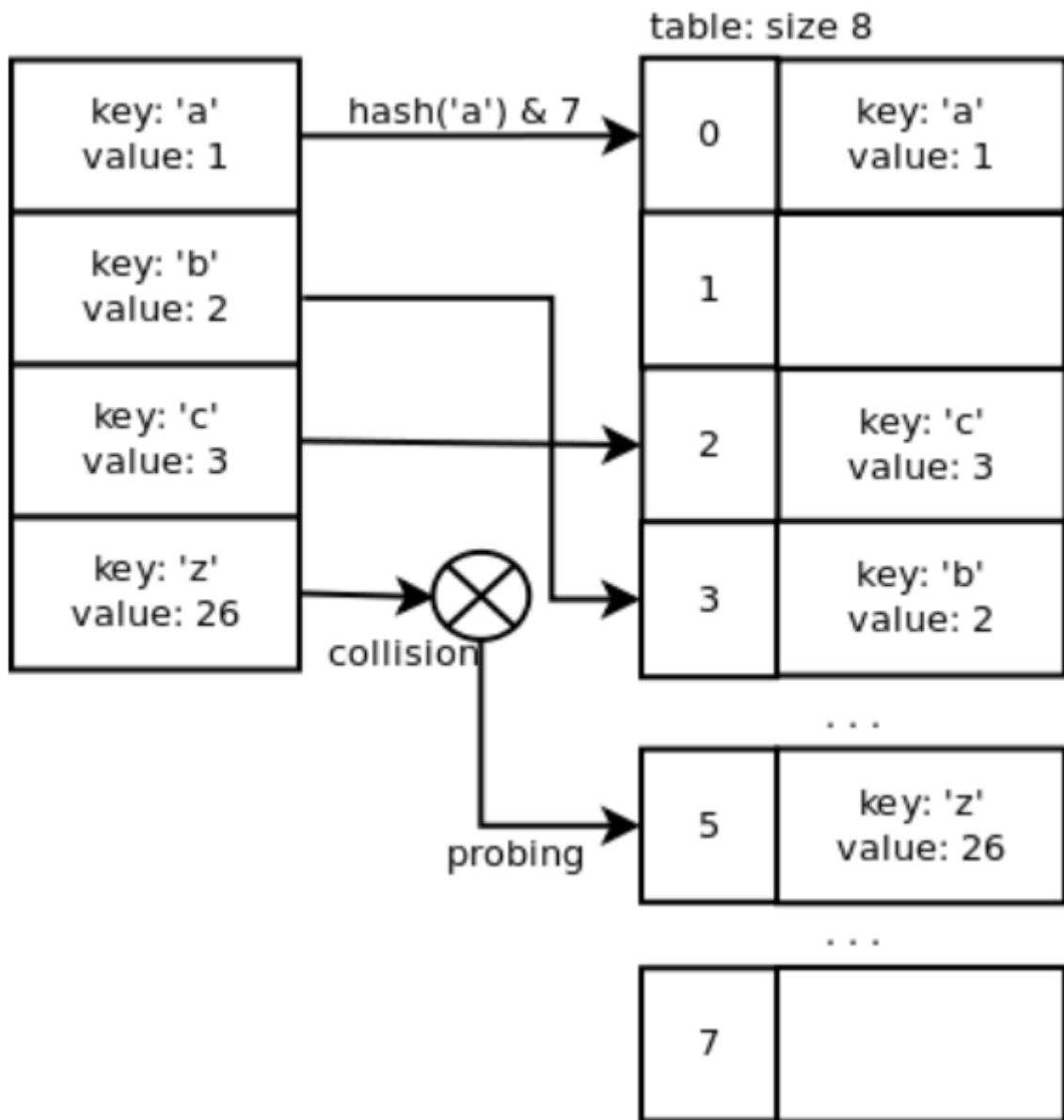
```
typedef struct _dictobject PyDictObject;
struct _dictobject {
    PyObject_HEAD
    Py_ssize_t ma_fill; // Number of used slots + dummy slots
    Py_ssize_t ma_used; // Used slots
    Py_ssize_t ma_mask; // Array size minus 1
    PyDictEntry *ma_table;
    PyDictEntry *(*ma_lookup) (PyDictObject *mp, PyObject *key, 1
    PyDictEntry ma_smalltable[PyDict_MINSIZE];
}
```


Structs II

```
typedef struct {  
    Py_ssize_t me_hash;  
    PyObject *me_key;  
    PyObject *me_value;  
} PyDictEntry;
```

Insert





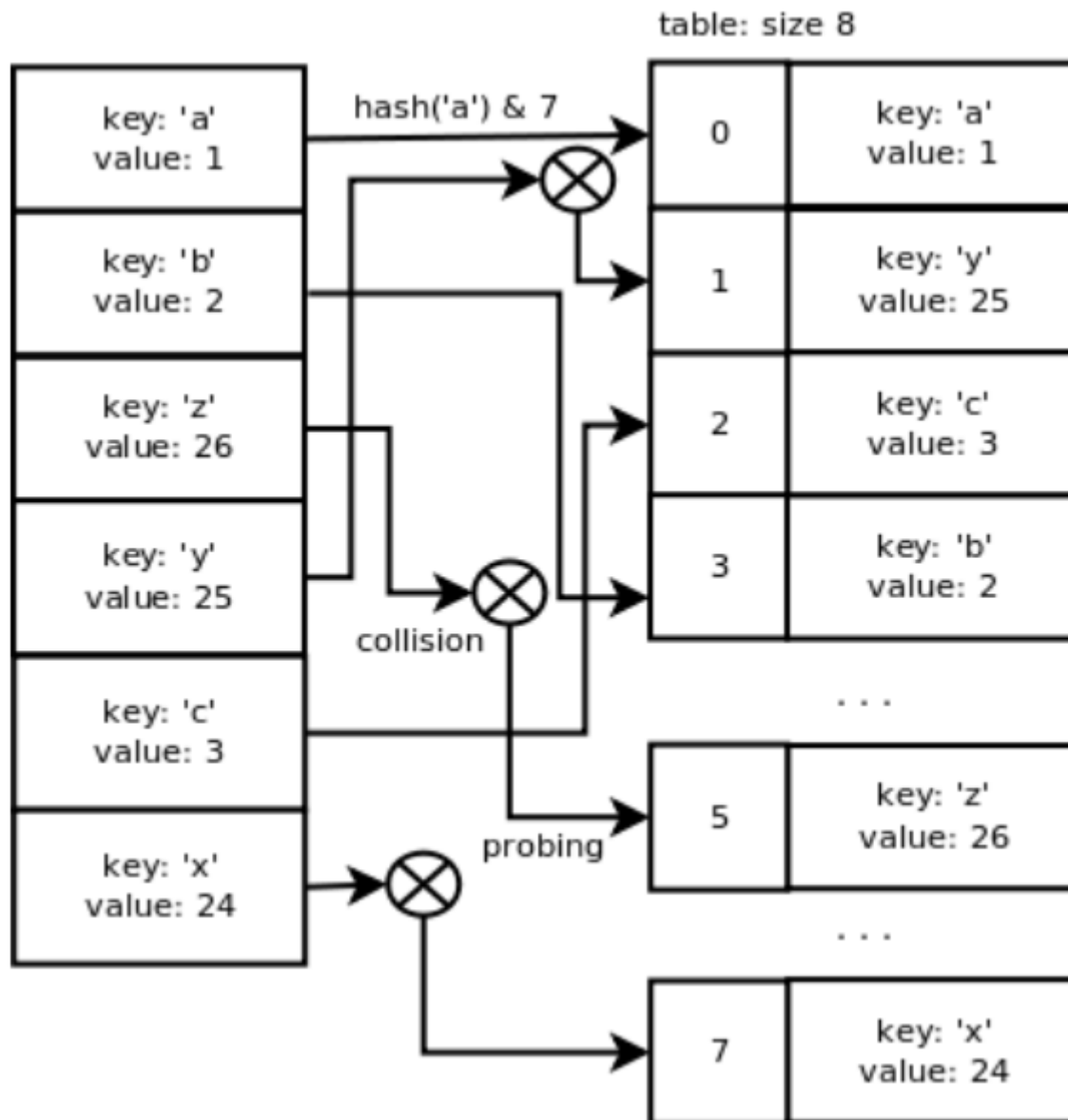
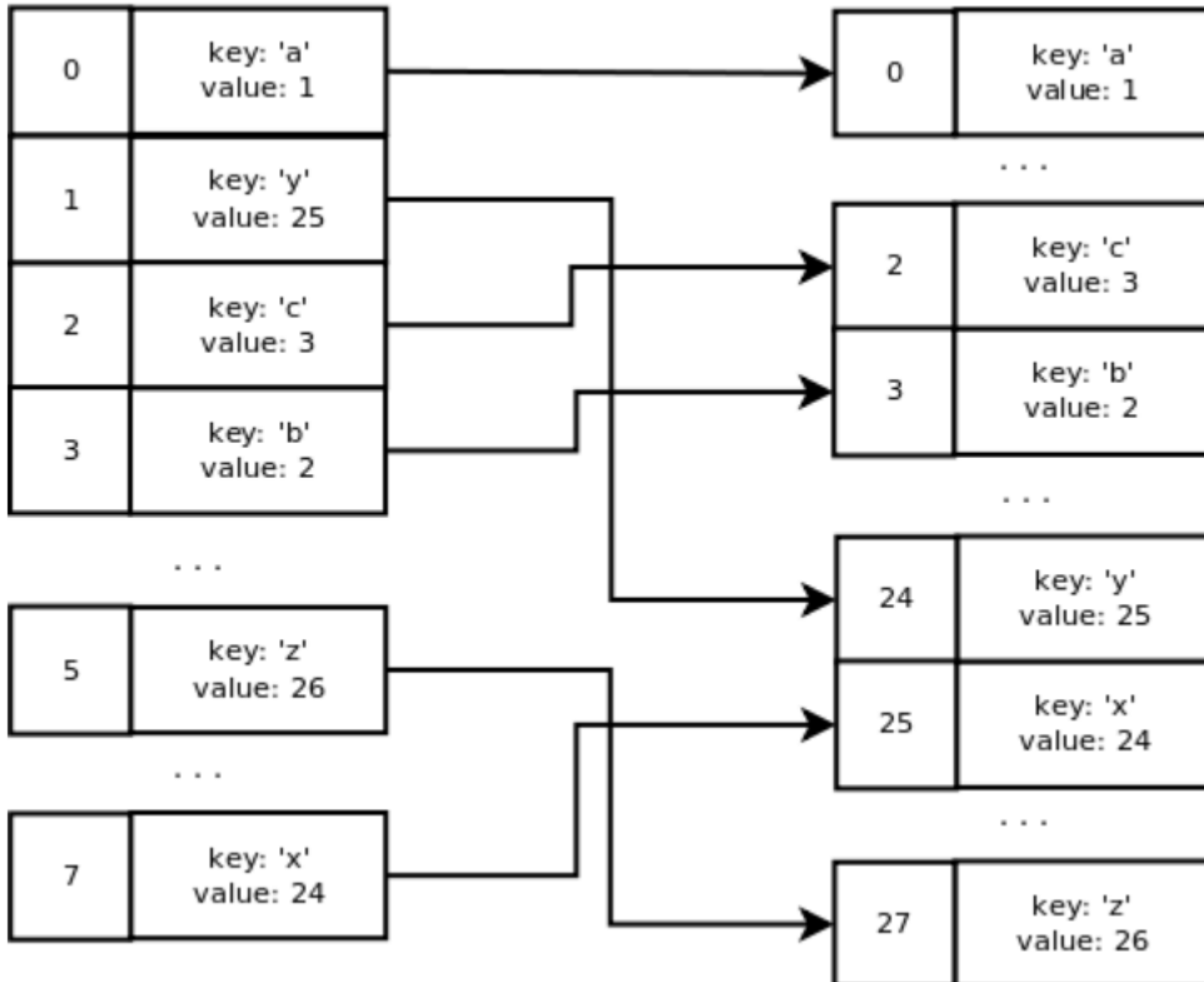


table: size 8

0	key: 'a' value: 1
1	key: 'y' value: 25
2	key: 'c' value: 3
3	key: 'b' value: 2
...	
5	key: 'z' value: 26
...	
7	key: 'x' value: 24

table: size 32

0	key: 'a' value: 1
...	
2	key: 'c' value: 3
3	key: 'b' value: 2
...	
24	key: 'y' value: 25
25	key: 'x' value: 24
...	
27	key: 'z' value: 26



Usage

**Raw
Capacity**

**Resize
Threshold**

**Empty
space**

8

6

2

32

22

10

128

86

42

...

...

...

2048

1366

682

...

...

...

32768

21846

10922

Import system

- Modul = Source Code File
- Package = Sammlung von Modulen und Sub-Packages
- Unterscheidung zwischen
 - Regular Package
 - Namespace Package

Importieren

- Normalerweise import Statement
 1. Suche nach dem Modul (Cache)
 2. Finder sucht nach dem Modul
 3. Loader
 4. Binding
- Import Hooks

Debugging

- Beobachtung: Viel Print-Debugging bei Python Anfängern
- Python debugger framework bdb
 - Basis-Funktionen eines Debuggers
 - Breakpoints, Steps, Stack-Trace

--

Faulthandler

- Dumps Python Tracebacks
- Eingeschränkte verwendung von bdb
- Aktiviert durch
 - expliziten Aufruf
 - automatisch in „catastrophic“ Situationen
 - reaktion auf registriertes User Signal

pdb

- Frontend zu bdb
- `python3 -m pdb bugged.py`
- Commands
 - `break`
 - `list`
 - `step`
 - `next`
 - `quit`

Python Language Services

- Arbeiten mit der Python Language
- Module für
 - Tokenizing
 - Parser
 - Syntax analysis
 - Bytecode
 - ...