

values found by name instead of by position, one key pointing to exactly one value

A dictionary maps keys to values

- a dictionary holds key colon value pairs, separated by commas, and written inside curly braces
- student = the name marks paired with 90, the name grade paired with A, is one such dictionary
- a value is looked up by its key, never by a position, a dictionary keeps no index numbers at all
- **mutable**: pairs can be added, changed, or removed after the dictionary is created

Reading and writing a value *

- student['marks'] returns the value stored under the key marks, and raises a KeyError if that key does not exist
- student.get('marks') does the same lookup safely, returning None instead of raising an error when the key is missing
- student.get('rank', 0) also takes a second argument, a default value handed back only when the key is missing
- student['rank'] = 3 both adds a brand new pair when rank is not already a key, and overwrites the value when it is

One dictionary, key by key

key	value
name	Priya
marks	90
grade	A

every key in a dictionary is unique, writing the same key a second time simply overwrites its old value

Common dictionary methods

- d.keys() returns a view of every key, d.values() a view of every value, d.items() a view of every key value pair together
- d.update(other) merges another dictionary in, adding new keys and overwriting the values of any key already shared
- d.pop(key) removes that key and returns its value, and raises a KeyError if the key is missing
- del d[key] also removes a pair, but returns nothing at all
- len(d) reports the number of pairs currently stored

Dictionary views



one root branching into the three view methods a dictionary offers over its own contents

Iterating a dictionary *

Q. d = the key a paired with 1, the key b paired with 2, the key c paired with 3, trace: for k, v in d.items(): print(k, v)

- 1 d.items() gives one pair per turn, as a two-value tuple
- 2 first turn: k is a, v is 1, prints a 1
- 3 second turn: k is b, v is 2, prints b 2
- 4 third turn: k is c, v is 3, prints c 3
- 5 the order printed matches the order the pairs were inserted in, dictionaries remember insertion order

Answer

a 1, then b 2, then c 3, one line each