1. Du weisst, dass Python bei Dictionaries die Elemente aus Effizienzgründen in einer scheinbar zufälligen Reihenfolge speichert. (→ Hashtabellen)
2. Du kannst ein Dictionary mit Werten initialisieren:

   `D={<key1>: <value1>, <key2>: <value2>, ...}`
3. Du kannst ein leeres Dictionary mit `D=dict()` anlegen.
4. Du kannst Schlüssel-Wert-Paare mit `D[<key>]=<value>` zu einem Dictionary hinzufügen.
5. Du kannst mit dem Schlüssel `D[<key>]` oder mit `D.get(<key>)` auf den zugehörigen Wert im Dictionary zugreifen.
6. Du kannst den zu einem Schlüssel gehörenden Wert mit `D[<key>]=<newvalue>` ändern.
7. Du kannst mit `len(D)` die Anzahl der Elemente eines Dictionaries bestimmen.
8. Du kannst mit `if <key> in D:` oder `if <key> not in D:` testen, ob ein Schlüssel in einem Dictionary (nicht) vorhanden ist.
9. Du kannst ein Schlüssel-Wert-Paar mit `D.pop(<key>)` oder `del D[<key>]` aus einem Dictionary entfernen.
10. Du kannst mit `D.update(E)` ein Dictionary `D` mit einem Dictioary `E` aktualisieren.
11. Du kannst ein Dictionary mit `for <var> in D:` oder `for <var> in D.keys():` schlüsselweise durchlaufen.
12. Du kannst ein Dictionary mit `for <var> in sortetd(D.keys()):` nach sortierten Schlüsseln durchlaufen.
13. Du kannst ein Dictionary mit `for <var> in D.values():` werteweise durchlaufen.
14. Du kannst ein Dictionary mit `for (<keyvar>,<valuevar>) in D.items():` paarweise durchlaufen.
15. Du kannst mit `D.copy()` eine vom Original unabhängige Kopie des Dictionaries `D` herstellen und weisst, dass die Zuweisung `E = D` nur eine Referenz auf `D` in `E` speichert.
16. Du kannst Python-Programme schreiben, die einfache Zählaufgaben (Zeichenketten, Listenelemente) mittels eines Dictionaries erledigen.