

Slovníky

Poznámky k téme

Slovník (typ `dict`) uchováva dvojice **klúč** → **hodnota**. Funguje ako skutočný slovník alebo telefónny zoznam: podľa klúča (slova, mena) rýchlo nájdeš hodnotu (preklad, číslo). Kým v zozname hľadáme prvky podľa poradového čísla, v slovníku ich hľadáme podľa zmysluplného klúča – napríklad body podľa mena alebo počet výskytov podľa slova.

Čo sa naučíš

- vytvoriť slovník, čítať, pridávať, meniť a mazať hodnoty podľa klúča,
- bezpečne čítať hodnoty metódou `get`,
- prechádzať slovník (`keys`, `values`, `items`),
- počítat výskyty pomocou slovníka.

1 Klúče a hodnoty

Slovník zapíšeme do zložených zátvoriek ako dvojice `klúč: hodnota` oddelené čiarkou:

```
>>> ceny = {"chlieb": 1.89, "mlieko": 0.99, "maslo": 2.49}
>>> ceny["mlieko"]
0.99
>>> len(ceny)
3
>>> prazdny = {}
>>> len(prazdny)
0
```

K hodnote sa dostaneme zápisom `slovník[klúč]`. Ak klúč v slovníku nie je, nastane chyba `KeyError`:

```
>>> ceny = {"chlieb": 1.89, "mlieko": 0.99}
>>> ceny["syr"]
KeyError: 'syr'
```

1.1 Pridanie, zmena a odstránenie

Priradením do nového klúča dvojicu **pridáme**, do existujúceho klúča hodnotu **zmeníme**. Príkaz `del` (alebo metóda `pop`) dvojicu odstráni:

```
vek = {"Ema": 17, "Ján": 16}
vek["Sofia"] = 15          # pridanie
vek["Ján"] = 17           # zmena
vek["Ema"] += 1           # zmena podľa starej hodnoty
del vek["Sofia"]          # odstránenie
print(vek)
```

Výstup:

```
{'Ema': 18, 'Ján': 17}
```

1.2 Pravidlá pre kľúče

- Každý kľúč je v slovníku **najviac raz**; nové priradenie do toho istého kľúča prepíše hodnotu.
- Kľúčom musí byť **nemenná** hodnota: reťazec, číslo, n-tica. Zoznam kľúčom byť nemôže.
- Hodnoty môžu byť čokoľvek – aj zoznamy alebo ďalšie slovníky (téma Zložitejšie dátové štruktúry).
- Slovník si pamätá poradie, v akom boli kľúče **vložené**.

1.3 Je kľúč v slovníku?

Operátor `in` sa pri slovníku pýta na **kľúče** (nie na hodnoty):

```
>>> ceny = {"chlieb": 1.89, "mlieko": 0.99}
>>> "chlieb" in ceny
True
>>> "syr" in ceny
False
>>> 0.99 in ceny
False
>>> 0.99 in ceny.values()
True
```

1.4 Bezpečné čítanie: get

Metóda `get(kľúč, náhrada)` vráti hodnotu, a ak kľúč chýba, vráti **náhradnú hodnotu** – bez chyby. Ak náhradu nezadáš, vráti `None`:

```
>>> ceny = {"chlieb": 1.89, "mlieko": 0.99}
>>> ceny.get("chlieb", 0)
1.89
>>> ceny.get("syr", 0)
0
>>> print(ceny.get("syr"))
None
```

2 Prechádzanie slovníka

Zápis	Čo prechádza
<code>for k in d:</code>	kľúče
<code>for k in d.keys():</code>	kľúče (to isté, iba výslovne)
<code>for v in d.values():</code>	hodnoty
<code>for k, v in d.items():</code>	dvojice (kľúč, hodnota)

```
body = {"Ema": 47, "Ján": 38, "Sofia": 50}
for meno in body:
    print(meno, end=" ")
print()
print("Súčet bodov:", sum(body.values()))
for meno, b in body.items():
    print(f"{meno}: {b} bodov")
```

Výstup:

```
Ema Ján Sofia
Súčet bodov: 135
Ema: 47 bodov
Ján: 38 bodov
Sofia: 50 bodov
```

Zoznam kľúčov alebo hodnôt dostaneš ako `list(d)` alebo `list(d.values())`. Zoradené kľúče: `sorted(d)`.

2.1 Zoradenie podľa hodnoty a najväčšia hodnota

Funkciám `max`, `min` a `sorted` sa dá povedať, podľa čoho majú porovnávať (parameter `key`). `max(d, key=d.get)` vráti kľúč s najväčšou hodnotou:

```
body = {"Ema": 47, "Ján": 38, "Sofia": 50, "Tomáš": 44}
najlepsi = max(body, key=body.get)
print("Najviac bodov:", najlepsi, body[najlepsi])
for meno in sorted(body, key=body.get, reverse=True):
    print(meno, body[meno])
```

Výstup:

```
Najviac bodov: Sofia 50
Sofia 50
Ema 47
Tomáš 44
Ján 38
```

3 Počítanie výskytov

Najčastejšie použitie slovníka: zistiť, koľkokrát sa ktorá hodnota vyskytuje. Kľúčom je hodnota, ktorú počítame, hodnotou je počet. Pri prvom výskyte kľúč v slovníku ešte nie je – preto `get(kľúč, 0)`:

```
text = "abrakadabra"
pocety = {}
for znak in text:
    pocety[znak] = pocety.get(znak, 0) + 1
print(pocety)
```

Výstup:

```
{ 'a': 5, 'b': 2, 'r': 2, 'k': 1, 'd': 1 }
```

Ten istý postup s podmienkou (dlhší, ale možno názornejší):

```
slova = "pes mačka pes vták pes mačka".split()
pocety = {}
for slovo in slova:
    if slovo in pocety:
        pocety[slovo] += 1
    else:
        pocety[slovo] = 1
for slovo, pocet in pocety.items():
    print(slovo, pocet)
```

Výstup:

```
pes 3
mačka 2
vták 1
```

Tip: Python má na počítanie aj hotový nástroj `Counter` z modulu `collections`:

`Counter("abrakadabra").most_common(2)` vráti dva najčastejšie znaky. Na písomke či maturite je však dobré vedieť to naprogramovať aj bez neho.

4 Riešené úlohy

4.1 Telefónny zoznam

Zadanie: Vytvor jednoduchý telefónny zoznam. Program načíta meno a vypíše číslo, alebo oznámi, že meno v zozname nie je.

```
kontakty = {"Ema": "0901 111 222", "Ján": "0902 333 444", "Sofia": "0903 555 666"}
meno = input("Koho hľadáš? ")
if meno in kontakty:
    print(meno, "má číslo", kontakty[meno])
else:
    print("Toto meno v zozname nie je.")
```

Výstup:

```
Koho hľadáš? Ján
Ján má číslo 0902 333 444
```

4.2 Najčastejšie slovo

Zadanie: V texte nájdí slovo, ktoré sa vyskytuje najčastejšie (veľké a malé písmená ani interpunkciu nerozlišuj).

Riešenie: Text prevedieme na malé písmená, odstránime čiarky a bodky, rozdelíme na slová a spočítame ich výskyty. Najčastejšie slovo je kľúč s najväčšou hodnotou.

```
text = "Mačka sedí na okne. Pes sedí pod oknom, mačka spí a pes tiež spí."
cisty = text.lower().replace(",", "").replace(".", "")
pocety = {}
for slovo in cisty.split():
    pocety[slovo] = pocety.get(slovo, 0) + 1
najcastejsie = max(pocety, key=pocety.get)
print(najcastejsie, pocety[najcastejsie])
```

Výstup:

```
mačka 2
```

Ak má viac slov rovnaký najvyšší počet, `max` vráti to, ktoré sa v slovníku objavilo skôr.

4.3 Nákupný košík

Zadanie: Cenník je slovník *tovar* → *cena*. Košík je zoznam nakúpených položiek (môžu sa opakovať). Vypočítaj cenu nákupu a vypíš položky, ktoré v cenníku nie sú.

```
cennik = {"chlieb": 1.89, "mlieko": 0.99, "maslo": 2.49, "jablko": 0.35}
kosik = ["mlieko", "jablko", "jablko", "čokoláda", "chlieb", "jablko"]
spolu = 0
for polozka in kosik:
    if polozka in cennik:
        spolu += cennik[polozka]
    else:
        print("Nie je v cenníku:", polozka)
print(f"Spolu: {spolu:.2f} €")
```

Výstup:

```
Nie je v cenníku: čokoláda
Spolu: 3.93 €
```

4.4 Prekladový slovník

Zadanie: Prelož vetu slovo po slove pomocou slovníka. Slová, ktoré v slovníku nie sú, nechaj nepreložené.

```
sk_en = {"ja": "I", "mám": "have", "psa": "a dog", "a": "and", "mačku": "a cat"}
veta = "ja mám psa a mačku doma"
preklad = []
for slovo in veta.split():
    preklad.append(sk_en.get(slovo, slovo))
print(" ".join(preklad))
```

Výstup:

```
I have a dog and a cat doma
```

Metóda `get(slovo, slovo)` vráti preklad, a ak preklad chýba, vráti samotné slovo.

4.5 Známky v triede

Zadanie: Ku každému menu patrí zoznam známok. Vypíš priemer pre každé meno na dve desatinné miesta.

```
znamky = {  
    "Ema": [1, 2, 1, 1],  
    "Ján": [3, 2, 4],  
    "Sofia": [2, 2, 1, 3, 1],  
}  
  
for meno, z in znamky.items():  
    print(f"{meno:<6} {sum(z) / len(z):.2f}")
```

Výstup:

```
Ema      1.25  
Ján      3.00  
Sofia    1.80
```

5 Časté chyby

- **Čítanie neexistujúceho kľúča** (`d["x"]`) skončí `KeyError` – použi `in` alebo `get`.
- `in` **hľadá kľúče, nie hodnoty** – na hodnoty použi `in d.values()`.
- **Prázdny slovník** je `{}`, prázdna množina je `set()`.
- **Zoznam ako kľúč** nie je možný (`TypeError: unhashable type: 'list'`) – použi `n`-ticu.
- **Zmena slovníka počas prechádzania** (pridávanie alebo mazanie kľúčov v cykle `for k in d`) skončí chybou; prechádzaj radšej kópiu `list(d)`.

Zhrnutie

- Slovník `{kľúč: hodnota, ...}`; prístup `d[kľúč]`, pridanie či zmena `d[kľúč] = hodnota`, odstránenie `del d[kľúč]`.
- Kľúče sú jedinečné a nemenné (reťazce, čísla, `n`-tice); hodnoty môžu byť čokoľvek.
- `kľúč in d` testuje kľúče; `d.get(kľúč, náhrada)` číta bez chyby.
- Prechádzanie: `for k in d`, `d.values()`, `for k, v in d.items()`.
- Počítanie výskytov: `pocety[x] = pocety.get(x, 0) + 1`.
- Kľúč s najväčšou hodnotou: `max(d, key=d.get)`.

Úlohy na precvičenie

1. Načítaj vetu a vypíš, koľkokrát sa v nej vyskytuje každé písmeno (bez medzier), abecedne.
2. Vytvor slovník *mesto* → *teplota* pre päť miest a vypíš najteplejšie a najchladnejšie mesto.
3. Načítavaj dvojice „meno body“, kým sa nezadá prázdny riadok; ak sa meno zopakuje, body sa sčítajú. Na konci vypíš poradie.
4. Spočítaj, koľko čísel v náhodnom zozname 100 hodov kockou padlo na každú stranu kocky.
5. Vytvor „obráteneý“ slovník – z `{"pes": "dog"}` urob `{"dog": "pes"}`.

Úlohy k tejto téme nájdeš na learnwithmente.com v časti Informatika → Python.