

Zoznamy

Poznámky k téme

Zatiaľ sme mali v každej premennej jednu hodnotu. Čo ak potrebujeme uložiť známky celej triedy, výsledky sto meraní alebo nákupný zoznam? Na to slúži **zoznam** (typ `list`) – usporiadaná postupnosť hodnôt uložená pod jedným menom. Zoznamy sú v Pythone asi najpoužívanejšia dátová štruktúra a bez nich sa nezaobíde takmer žiadna maturitná úloha.

Čo sa naučíš

- vytvoriť zoznam, prístupovať k prvkom cez indexy a meniť ich,
- používať metódy zoznamov (`append`, `insert`, `pop`, `remove`, `sort`, ...),
- prechádzať zoznam cyklom a používať funkcie `len`, `sum`, `min`, `max`, `sorted`,
- načítavať zoznam zo vstupu a vytvoriť náhodný zoznam,
- zapisovať zoznamy stručne pomocou generátorového zápisu.

1 Vytvorenie zoznamu a indexy

Zoznam zapíšeme do hranatých zátvoriek, prvky oddelíme čiarkou:

```
>>> znamky = [1, 3, 2, 1, 4]
>>> mena = ["Ema", "Ján", "Sofia"]
>>> prazdny = []
>>> zmiesany = [7, "text", 3.5, True]
>>> len(znamky)
5
>>> len(prazdny)
0
```

Prvky zoznamu sú očíslované **indexmi** presne ako znaky reťazca – od nuly, záporné od konca:

index	0	1	2	3	4
prvok	10	20	30	40	50
záporný index	-5	-4	-3	-2	-1

```
>>> z = [10, 20, 30, 40, 50]
>>> z[0]
10
>>> z[-1]
50
>>> z[1:3]
[20, 30]
>>> z[::-1]
[50, 40, 30, 20, 10]
>>> z[5]
IndexError: list index out of range
```

Rezy fungujú rovnako ako pri reťazcoch a vytvárajú **nový** zoznam.

1.1 Zoznamy sa dajú meniť

Na rozdiel od reťazcov je zoznam **meniteľný**: prvok môžeme prepísať priradením.

```
>>> z = [10, 20, 30]
>>> z[1] = 99
>>> z
[10, 99, 30]
>>> z[-1] += 5
>>> z
[10, 99, 35]
```

1.2 Zoznam z iných hodnôt

```
>>> list(range(1, 6))
[1, 2, 3, 4, 5]
>>> list("ahoj")
['a', 'h', 'o', 'j']
>>> "a,b,c".split(",")
['a', 'b', 'c']
>>> [0] * 5
[0, 0, 0, 0, 0]
>>> [1, 2] + [3, 4]
[1, 2, 3, 4]
>>> 3 in [1, 2, 3]
True
```

`[0] * 5` vytvorí zoznam piatich núl – hodí sa ako počiatočný stav napríklad pre počítadlá. `+` spojí dva zoznamy do nového, `in` zistí, či je hodnota v zozname.

2 Metódy zoznamov

Metóda	Čo urobí
<code>z.append(x)</code>	pridá <code>x</code> na koniec
<code>z.insert(i, x)</code>	vloží <code>x</code> na index <code>i</code> (ostatné sa posunú)
<code>z.pop()</code>	odoberie a vráti posledný prvok
<code>z.pop(i)</code>	odoberie a vráti prvok na indexe <code>i</code>
<code>z.remove(x)</code>	odstráni prvý výskyt hodnoty <code>x</code> (ak nie je, chyba)
<code>z.index(x)</code>	vráti index prvého výskytu <code>x</code> (ak nie je, chyba)
<code>z.count(x)</code>	koľkokrát je <code>x</code> v zozname
<code>z.sort()</code>	zoradí zoznam (vzostupne; <code>reverse=True</code> zostupne)
<code>z.reverse()</code>	obráti poradie prvkov
<code>z.extend(iny)</code>	pridá na koniec všetky prvky iného zoznamu
<code>z.copy()</code>	vráti kópiu zoznamu
<code>z.clear()</code>	vymaže všetky prvky

```
nakup = ["chlieb", "mlieko"]
nakup.append("vajcia")
nakup.insert(0, "jablká")
print(nakup)
odobrane = nakup.pop()
print("odobrané:", odobrane, "→", nakup)
nakup.remove("chlieb")
print(nakup)
print("index mlieka:", nakup.index("mlieko"))
```

Výstup:

```
['jablká', 'chlieb', 'mlieko', 'vajcia']
odobrané: vajcia → ['jablká', 'chlieb', 'mlieko']
['jablká', 'mlieko']
index mlieka: 1
```

```
cisla = [5, 3, 8, 1, 9, 2]
cisla.sort()
print(cisla)
cisla.sort(reverse=True)
print(cisla)
cisla.reverse()
print(cisla)
print("počet trojok:", cisla.count(3))
```

Výstup:

```
[1, 2, 3, 5, 8, 9]
[9, 8, 5, 3, 2, 1]
[1, 2, 3, 5, 8, 9]
počet trojok: 1
```

Pozor: Metódy, ktoré zoznam menia (append, sort, reverse, insert, remove...), vracajú None. Zápis `cisla = cisla.sort()` preto zoznam zoradí – a vzápätí ho stratíš, lebo do `cisla` uložíš None. Správne je iba `cisla.sort()`.

3 Prechádzanie zoznamu

Najčastejšie prechádzame priamo **prvky**:

```
teploty = [12.5, 14.0, 9.5, 17.0]
for t in teploty:
    print(t, "°C")
```

Výstup:

```
12.5 °C
14.0 °C
9.5 °C
17.0 °C
```

Ak potrebujeme aj **pozíciu** prvku (alebo chceme prvky meniť), prechádzame **indexy**:

```
body = [45, 38, 50]
for i in range(len(body)):
    body[i] = body[i] + 2      # každému pridáme 2 body
print(body)
```

Výstup:

```
[47, 40, 52]
```

Pohodlne získaš index aj prvok naraz funkciou `enumerate`:

```
mena = ["Ema", "Ján", "Sofia"]
for poradie, meno in enumerate(mena, start=1):
    print(f"{poradie}. {meno}")
```

Výstup:

```
1. Ema
2. Ján
3. Sofia
```

4 Funkcie pre zoznamy

```
>>> z = [4, 9, 1, 7]
>>> len(z)
4
>>> sum(z)
21
>>> min(z)
1
>>> max(z)
9
>>> sum(z) / len(z)
5.25
>>> sorted(z)
[1, 4, 7, 9]
>>> sorted(z, reverse=True)
[9, 7, 4, 1]
>>> z
[4, 9, 1, 7]
```

`sorted(z)` vráti **nový** zoradený zoznam a pôvodný nechá tak; `z.sort()` zoradí priamo pôvodný zoznam. Funkcie `min`, `max` a `sorted` fungujú aj so zoznamom reťazcov (abecedne podľa kódov znakov).

5 Kopírovanie: pozor na dve mená pre jeden zoznam

Priradenie `b = a` **nevytvorí kópiu** zoznamu – `a` aj `b` sú dve mená pre **ten istý** zoznam:

```
a = [1, 2, 3]
b = a
b.append(4)
print("a =", a)
print("b =", b)
```

Výstup:

```
a = [1, 2, 3, 4]
b = [1, 2, 3, 4]
```

Ak chceš samostatnú kópiu, použi `a.copy()` alebo rez `a[:]`:

```
a = [1, 2, 3]
b = a.copy()
b.append(4)
print("a =", a)
print("b =", b)
```

Výstup:

```
a = [1, 2, 3]
b = [1, 2, 3, 4]
```

Rovnako to platí pri funkciách: ak funkcii odovzdáš zoznam a ona ho zmení (napr. `append`), zmena je viditeľná aj mimo funkcie.

6 Zoznam zo vstupu

Čísla na jednom riadku – rozdelíme `split()` a každú časť prevedieme na číslo:

```
cisla = []
for cast in input("Čísla: ").split():
    cisla.append(int(cast))
print(cisla, "súčet:", sum(cisla))
```

Výstup:

```
Čísla: 4 8 15 16 23 42
[4, 8, 15, 16, 23, 42] súčet: 108
```

Čísla na samostatných riadkoch – najprv počet, potom čísla:

```
n = int(input("Počet: "))
cisla = []
for _ in range(n):
    cisla.append(int(input()))
print(cisla)
```

Výstup:

```
Počet: 3
12
7
30
[12, 7, 30]
```

6.1 Náhodný zoznam

Pomocou modulu `random` (téma Modulárny program) vytvoríme zoznam náhodných čísel. V ukážke je nastavené `random.seed(1)`, aby bol výsledok vždy rovnaký:

```
import random

cisla = []
for _ in range(10):
    cisla.append(random.randint(1, 100))
print(cisla)
print("najväčšie:", max(cisla), "na indexe", cisla.index(max(cisla)))
```

Výstup:

```
[18, 73, 98, 9, 33, 16, 64, 98, 58, 61]
najväčšie: 98 na indexe 2
```

7 Generátorový zápis

Zoznam, ktorý vzniká z iného zoznamu, sa dá zapísať jedným riadkom – **generátorovým zápisom** (*list comprehension*):

[výraz for prvok in postupnosť] alebo s podmienkou [výraz for prvok in postupnosť if podmienka]

```
>>> [x * x for x in range(1, 6)]
[1, 4, 9, 16, 25]
>>> [x for x in range(20) if x % 3 == 0]
[0, 3, 6, 9, 12, 15, 18]
>>> [slovo.upper() for slovo in ["ahoj", "svet"]]
['AHOJ', 'SVET']
>>> [int(c) for c in "2026"]
[2, 0, 2, 6]
```

Zápis je skratkou za tento cyklus:

```
stvorce = []
for x in range(1, 6):
    stvorce.append(x * x)
print(stvorce)
```

Výstup:

```
[1, 4, 9, 16, 25]
```

Načítanie zoznamu čísel zo vstupu sa tak skrúti na jeden riadok: `cisla = [int(x) for x in input().split()]`. Náhodný zoznam: `[random.randint(1, 100) for _ in range(10)]`.

8 Riešené úlohy

8.1 Priemer a nadpriemerné výsledky

Zadanie: Z bodov z písomky vypočítaj priemer a vypíš, koľko výsledkov je nad priemerom.

Riešenie: Priemer je `sum / len`. Potom zoznam prejdeme ešte raz a spočítame hodnoty väčšie ako priemer – na to potrebujeme mať všetky čísla uložené, preto zoznam.

```
body = [38, 45, 22, 50, 31, 47, 28]
priemer = sum(body) / len(body)
nad = 0
for b in body:
    if b > priemer:
        nad += 1
print(f"Priemer: {priemer:.2f}")
print("Nad priemerom:", nad)
```

Výstup:

```
Priemer: 37.29
Nad priemerom: 4
```

8.2 Prvky, ktoré sú v zozname iba raz

Zadanie: Vypíš prvky, ktoré sa v zozname vyskytujú práve raz, v pôvodnom poradí.

```
z = [3, 7, 3, 1, 9, 7, 4]
jedinecne = []
for x in z:
    if z.count(x) == 1:
        jedinecne.append(x)
print(jedinecne)
```

Výstup:

```
[1, 9, 4]
```

8.3 Najčastejší prvok

Zadanie: Nájdi hodnotu, ktorá sa v zozname vyskytuje najčastejšie (ak je ich viac, stačí prvá).

Riešenie: Pre každý prvok zistíme jeho počet výskytov a pamätáme si doteraz najlepšie. Pre veľké zoznamy je rýchlejšie počítanie cez slovník (téma Slovníky).

```
z = [2, 5, 2, 8, 5, 5, 1, 2, 5]
najcastejsi = z[0]
for x in z:
    if z.count(x) > z.count(najcastejsi):
        najcastejsi = x
print(najcastejsi, "-", z.count(najcastejsi), "krát")
```

Výstup:

```
5 - 4 krát
```

8.4 Druhé najväčšie číslo

Zadanie: Nájdi druhú najväčšiu **rôznu** hodnotu v zozname (zoznam má aspoň dve rôzne hodnoty).

Riešenie: Zoradíme hodnoty bez opakovania (`set` odstráni duplikáty – téma Množiny) a vezmeme predposlednú. Bez množiny by sme najprv našli maximum a potom najväčší prvok, ktorý je od neho menší.

```
z = [4, 9, 2, 9, 7, 1]
rozne = sorted(set(z))
print("Druhé najväčšie:", rozne[-2])

maximum = max(z)
druhe = None
for x in z:
    if x < maximum and (druhe is None or x > druhe):
        druhe = x
print("Druhé najväčšie:", druhe)
```

Výstup:

```
Druhé najväčšie: 7
Druhé najväčšie: 7
```

8.5 Spoločné prvky dvoch zoznamov

Zadanie: Vypíš prvky, ktoré sa nachádzajú v oboch zoznamoch (bez opakovania).

```
a = [1, 4, 6, 8, 4, 10]
b = [4, 5, 6, 7, 10, 12]
spolocne = []
for x in a:
    if x in b and x not in spolocne:
        spolocne.append(x)
print(spolocne)
```

Výstup:

```
[4, 6, 10]
```

8.6 Párne a nepárne, zoradené

Zadanie: Rozdeľ zoznam na párne a nepárne čísla a obidva zoznamy vypíš zoradené.

```
cisla = [15, 8, 3, 22, 7, 10, 1]
parne = sorted([x for x in cisla if x % 2 == 0])
neparne = sorted([x for x in cisla if x % 2 == 1])
print("párne:", parne)
print("nepárne:", neparne)
```

Výstup:

```
párne: [8, 10, 22]
nepárne: [1, 3, 7, 15]
```

9 Časté chyby

- **Index mimo rozsahu** (`IndexError`): posledný prvok je `z[-1]` alebo `z[len(z) - 1]`.
- `z = z.sort()`: výsledok je `None`; správne je `z.sort()` alebo `z = sorted(z)`.
- `b = a` **namiesto kópie**: obe mená menia ten istý zoznam; kópia je `a.copy()`.
- **Odstraňovanie počas prechádzania**: ak v cykle `for x in z` voláš `z.remove(...)`, cyklus preskakuje prvky. Radšej vytvor nový zoznam s prvkami, ktoré chceš ponechať:

```
z = [2, 4, 4, 5, 6]
for x in z:
    if x % 2 == 0:
        z.remove(x)
print("zle:", z)

z = [2, 4, 4, 5, 6]
z = [x for x in z if x % 2 != 0]
print("dobre:", z)
```

Výstup:

```
zle: [4, 5]
dobre: [5]
```

- `remove` a `index` **hodnoty, ktorá v zozname nie je**, skončia chybou `ValueError` – najprv over `if x in z:`.

Zhrnutie

- Zoznam `[a, b, c]` je usporiadaný a meniteľný; indexy a rezy fungujú ako pri reťazcoch.
- Pridávanie: `append`, `insert`, `extend`; odoberanie: `pop`, `remove`; ďalej `sort`, `reverse`, `count`, `index`.
- Metódy meniace zoznam vracajú `None` – ich výsledok neukladaj.
- `len`, `sum`, `min`, `max`, `sorted`; priemer je `sum(z) / len(z)`.
- `for x in z` prechádza prvky, `for i in range(len(z))` indexy, `enumerate` oboje.
- `b = a` nie je kópia – použi `a.copy()`.
- Generátorový zápis: `[výraz for x in z if podmienka]`.

Úlohy na precvičenie

1. Načítaj čísla oddelené medzerou a vypíš súčet kladných a počet záporných.
2. Vytvor zoznam 20 náhodných čísel od 1 do 6 (hody kockou) a vypíš, koľkokrát padla šestka.
3. Vypíš zoznam odzadu bez použitia `reverse` a `[::-1]` (pomocou cyklu).
4. Načítaj mená hostí, kým sa nezadá prázdny riadok, a vypíš ich abecedne zoradené.
5. Posuň prvky zoznamu o jedno miesto doprava: `[1, 2, 3, 4] → [4, 1, 2, 3]`.
6. Napíš funkciu, ktorá vráti index najmenšieho prvku zoznamu (bez `min` a `index`).

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