

N-tice

Poznámky k téme

N-tica (typ `tuple`) je usporiadaná skupina hodnôt – podobne ako zoznam, ale **nedá sa meniť**. Používa sa na údaje, ktoré patria pevne k sebe: súradnice bodu `(x, y)`, dátum `(deň, mesiac, rok)`, meno a body `("Ema", 47)`. S n-ticami sa stretávaš častejšie, než si myslíš – napríklad keď funkcia vráti viac hodnôt naraz alebo keď vymeniš dve premenné zápisom `a, b = b, a`.

Čo sa naučíš

- vytvoriť n-ticu a pracovať s jej prvkami,
- rozbaľovať n-tice do premenných (aj v cykle `for`),
- vracať z funkcie viac hodnôt,
- rozhodnúť, kedy použiť n-ticu a kedy zoznam.

1 Vytvorenie n-tice

N-ticu zapíšeme do okrúhlych zátvoriek, prvky oddelíme čiarkou. Zátvorky sa dajú aj vynechať – n-ticu v skutočnosti vytvára **čiarka**:

```
>>> bod = (3, 7)
>>> datum = (1, 10, 2026)
>>> ziak = ("Ema", 17, 1.68)
>>> farby = "červená", "zelená", "modrá"
>>> farby
('červená', 'zelená', 'modrá')
>>> type(farby)
<class 'tuple'>
```

Dva špeciálne prípady:

```
>>> prazdna = ()
>>> len(prazdna)
0
>>> jedna = (5,)
>>> type(jedna)
<class 'tuple'>
>>> type((5))
<class 'int'>
```

N-tica s **jedným** prvkom potrebuje čiarku: `(5,)`. Zápis `(5)` je iba číslo 5 v zátvorkách.

1.1 Práca s prvkami

Indexy, rezy, `len`, `in`, `count`, `index` a cyklus `for` fungujú rovnako ako pri zoznamoch:

```
>>> t = (10, 20, 30, 40)
>>> t[0]
10
>>> t[-1]
40
>>> t[1:3]
(20, 30)
>>> len(t)
4
>>> 20 in t
True
>>> max(t)
40
```

1.2 N-tica sa nedá meniť

N-tica je **nemenná** – prvky nemožno prepisovať, pridávať ani odoberať. Nemá preto metódy `append`, `remove`, `sort`:

```
>>> t = (10, 20, 30)
>>> t[0] = 99
TypeError: 'tuple' object does not support item assignment
>>> t.append(40)
AttributeError: 'tuple' object has no attribute 'append'
```

Ak potrebuješ „zmenenú“ n-ticu, vytvor novú (napríklad `t + (40,)`), alebo n-ticu preved' na zoznam: `list(t)`. Opačne `tuple(z)` vytvorí zo zoznamu n-ticu.

2 Rozbaľovanie

Hodnoty n-tice môžeme naraz priradiť do viacerých premenných – **rozbaľiť** ju. Počet premenných sa musí zhodovať s počtom prvkov:

```
bod = (3, 7)
x, y = bod
print("x =", x, "y =", y)

meno, vek, vyska = ("Ema", 17, 1.68)
print(meno, vek, vyska)
```

Výstup:

```
x = 3 y = 7
Ema 17 1.68
```

```
x, y = (1, 2, 3)
```

Výstup (program skončí chybou):

```
Traceback (most recent call last):
  File "program.py", line 1, in <module>
    x, y = (1, 2, 3)
    ^^^^
ValueError: too many values to unpack (expected 2)
```

2.1 Výmena hodnôt

Vďaka n-ticiam sa dajú dve premenné vymeniť jedným riadkom. Pravá strana `b, a` najprv vytvorí n-ticu zo **starých** hodnôt a až potom sa rozbalí do `a` a `b`:

```
a = 1
b = 2
a, b = b, a
print(a, b)
```

Výstup:

```
2 1
```

Rovnaký trik sa používa pri posúvaní hodnôt, napríklad vo Fibonacciho postupnosti: `a, b = b, a + b`.

2.2 Rozbaľovanie v cykle

Ak je v zozname viac n-tíc, cyklus `for` ich môže rozbaľovať priamo:

```
vysledky = [("Ema", 47), ("Ján", 38), ("Sofia", 50)]
for meno, body in vysledky:
    print(f"{meno:<6} {body:>3} bodov")
```

Výstup:

```
Ema      47 bodov
Ján      38 bodov
Sofia    50 bodov
```

Funkcia `enumerate` (téma Zoznamy) vracia práve takéto dvojice (`index`, `prvok`) a funkcia `zip` spojí dva zoznamy do dvojíc:

```
mena = ["Ema", "Ján", "Sofia"]
body = [47, 38, 50]
for meno, b in zip(mena, body):
    print(meno, b)
```

Výstup:

```
Ema 47
Ján 38
Sofia 50
```

3 Funkcia vracajúca viac hodnôt

Funkcia môže vrátiť viac hodnôt oddelených čiarkou – vráti jednu n-ticu, ktorú pri volaní rozbalíme:

```
def statistika(cisla):
    return min(cisla), max(cisla), sum(cisla) / len(cisla)

najmensie, najvacsie, priemer = statistika([4, 9, 2, 7])
print(najmensie, najvacsie, priemer)
print(statistika([1, 1]))
```

Výstup:

```
2 9 5.5
(1, 1, 1.0)
```

Zabudovaná funkcia `divmod` robí to isté: vráti podiel aj zvyšok naraz.

```
hodiny, minuty = divmod(135, 60)
print(f"135 minút = {hodiny} h {minuty} min")
```

Výstup:

```
135 minút = 2 h 15 min
```

4 N-tica alebo zoznam?

	N-tica ()	Zoznam []
dá sa meniť	nie	áno
typické použitie	pevná skupina rôznych údajov o jednej veci	veľa údajov rovnakého druhu, ktorých počet sa mení
príklad	(x, y), ("Ema", 17)	známky triedy, nákupný zoznam
kľúč v slovníku	áno	nie

N-tica chráni údaje pred nechcenou zmenou a dá sa použiť ako kľúč v slovníku (téma Slovníky) alebo prvok množiny – zoznam nie.

5 Riešené úlohy

5.1 Vzdialenosť dvoch bodov

Zadanie: Napíš funkciu, ktorá vráti vzdialenosť dvoch bodov v rovine zadanych ako n-tice (x, y).

Riešenie: Podľa Pytagorovej vety je vzdialenosť $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Body vo funkcii rozbalíme na súradnice.

```
def vzdialenost(a, b):
    x1, y1 = a
    x2, y2 = b
    return ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 0.5

print(vzdialenost((0, 0), (3, 4)))
print(round(vzdialenost((1, 2), (4, -2)), 3))
```

Výstup:

```
5.0
5.0
```

5.2 Najlepší výsledok

Zadanie: V zozname dvojíc (meno, body) nájdí meno s najviac bodmi.

```
vysledky = [("Ema", 47), ("Ján", 38), ("Sofia", 50), ("Tomáš", 44)]
najlepsi_meno, najlepsie_body = vysledky[0]
for meno, body in vysledky:
    if body > najlepsie_body:
        najlepsi_meno, najlepsie_body = meno, body
print(najlepsi_meno, najlepsie_body)
```

Výstup:

```
Sofia 50
```

Funkcia `max` to vie aj sama, ak jej povieme, podľa čoho porovnávať: `max(vysledky, key=lambda d: d[1])`. Zápis `lambda d: d[1]` je krátka funkcia, ktorá z dvojice vráti body.

5.3 Zoradenie dvojíc

Zadanie: Zoraď výsledky od najvyššieho počtu bodov.

```
vysledky = [("Ema", 47), ("Ján", 38), ("Sofia", 50), ("Tomáš", 44)]
for meno, body in sorted(vysledky, key=lambda d: d[1], reverse=True):
    print(meno, body)
```

Výstup:

```
Sofia 50
Ema 47
Tomáš 44
Ján 38
```

Bez `key` by sa n-tice zoradili podľa **prvého** prvku (mena) – n-tice sa porovnávajú postupne prvok po prvku.

5.4 Čas zo sekúnd

Zadanie: Napíš funkciu `rozloz_cas(sekundy)`, ktorá vráti n-ticu (hodiny, minúty, sekundy).

```
def rozloz_cas(sekundy):
    minuty, s = divmod(sekundy, 60)
    h, m = divmod(minuty, 60)
    return h, m, s

h, m, s = rozloz_cas(7384)
print(f"{h}:{m:02}:{s:02}")
```

Výstup:

```
2:03:04
```

Zhrnutie

- N-tica `(a, b, c)` je usporiadaná a **nemenná**; jednoprvková sa píše `(5,)`.
- Indexy, rezy, `len`, `in`, `for` fungujú ako pri zoznamoch; metódy na zmenu nemá.
- Rozbalenie: `x, y = bod` – počet premenných sa musí zhodovať s počtom prvkov.
- `a, b = b, a` vymení hodnoty; `for meno, body in zoznam_dvojic:` rozbaľuje v cykle.

- `return a, b` vráti n-ticu; pri volaní ju rozbalíme do premenných.
- N-tica pre pevnú skupinu údajov o jednej veci, zoznam pre premenlivý počet údajov rovnakého druhu.

Úlohy na precvičenie

1. Napíš funkciu, ktorá pre dva body vráti stred úsečky ako n-ticu.
2. Načítaj dátum v tvare `1.10.2026` a ulož ho ako n-ticu troch celých čísel.
3. Zo zoznamu dvojíc (mesto, teplota) vypíš najteplejšie a najchladnejšie mesto.
4. Napíš funkciu, ktorá pre zoznam čísel vráti súčet párných a súčet nepárnych naraz.
5. Bez spustenia urči, čo vypíše: `a, b, c = 1, 2, 3, a, b, c = c, a, b, print(a, b, c)`.

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