

💡 Ďalšie vstavané dátové typy

NÁZOV	UKLADÁ	PRÍKLADY
list (zoznam)	usporiadanú kolekciu (akýchkoľvek) hodnôt	[3, 7, 42]
tuple (n-tica)	usporiadanú kolekciu pevného počtu hodnôt	(159, 222, 200)
range (rozsah)	kolekciu diskretných hodnôt medzi začiatkom a koncom	range(2, 10, 2)
dict (slovník)	kolekciu jedinečných kľúčov a ich príslušných hodnôt	{name: "John", age: 13}
set (množina)	neusporiadanú kolekciu jedinečných hodnôt	{apple, pear, banana}

Zisti viac o iných dátových typoch [tu](#) 🔗.

💡 Operácie na dátových typoch

ZOZNAMY (LISTS)

OPERÁCIA	SYNTAX
prístup k prvku	nazovlistu[1]; nazovlistu[-2]; nazovlistu[2:5]; nazovlistu[2:]; nazovlistu[:4]
zmena prvku	nazovlistu[idx] = x; nazovlistu.insert(pozicia, nova_hodnota)
rozšírenie	nazovlistu.append(nova_hodnota); nazovlistu.extend(iny_zoznam)
zoradenie	nazovlistu.sort()

Zisti viac o operáciách so zoznamami [tu](#) 🔗.

N-TICE (TUPLES)

OPERÁCIA	SYNTAX
prístup k prvku	nazovtuple[idx]
rozbalenie	x, y, z = nazovtuple

Zisti viac o operáciách s n-ticami [tu](#) 🔗.

SLOVNÍKY (DICTIONARIES)

OPERÁCIA	SYNTAX
prístup k prvku	nazovdict[kluc]; nazovdict.keys(); nazovdict.values()
pridanie prvkov	nazovdict[novy_kluc] = nova_hodnota
odstránenie prvkov	nazovdict.pop(kluc)

Zisti viac o operáciách so slovníkmi [tu](#) 🔗.

MNOŽINY (SETS)

OPERÁCIA	SYNTAX
pridanie prvkov	nazovsetu.add(novy_prvok)
odstránenie prvkov	nazovsetu.remove(prvok)

Zisti viac o teórii množín na lepšie pochopenie operácií [tu](#) 🔗. Zisti viac o operáciách s množinami v Pythone [tu](#) 🔗.

💡 Základná logika: Riadiace štruktúry

Všeobecná štruktúra riadiacich štruktúr, vykomentované časti (označené ##) nie sú povinné:

🔗 if-else

```
if [podmienka]:
    následok
##elif [podmienka]:
    ##následok
##else:
    ##následok
```

🔗 match-case

```
match [premenná]:
    case [hodnota_1]:
        následok_1
    ##case [hodnota_2]:
        ##následok_2
    ##case [hodnota_3]:
        následok_3
    ...
```

🔗 while loops

```
while [podmienka]:
    akcia
```

🔗 for loops

```
for [prvok] in [sekvencia]:
    akcia
```

💡 Riadiace štruktúry: Všeobecné poznámky

- 1 Všetky riadiace štruktúry musia byť správne odsadené ako je ukázané vyššie (inak ich Python nebude vedieť zparsovať). Na odsadenie používaj klávesu 'tab'.
- 2 Podmienky typicky obsahujú logické, porovnávacie, identitné alebo členské operátory. (Spomeň si na rôzne operátory [tu](#) 🔗.)
- 3 Uisti sa, že cykly majú zadaný *nejaký koniec*. Inak sú celkom dobré v crashovaní počítačov. Vždy zahrň akciu, ktorá pri iterácii pozmení podmienku a/alebo pridaj break príkaz.
- 4 Pre if-else a match-case príkazy: následky nemôžu byť prázdne—zväž použitie pass príkazu, ktorý plní v podstate tú istú funkciu (= nerobí nič).
- 5 Loops sú iteratívne príkazy, ktoré môžu prechádzať cez akýkoľvek *iterovateľný* dátový typ. Iterovateľné dátové typy zahŕňajú lists, sets, tuples, dictionaries, strings alebo ranges.