

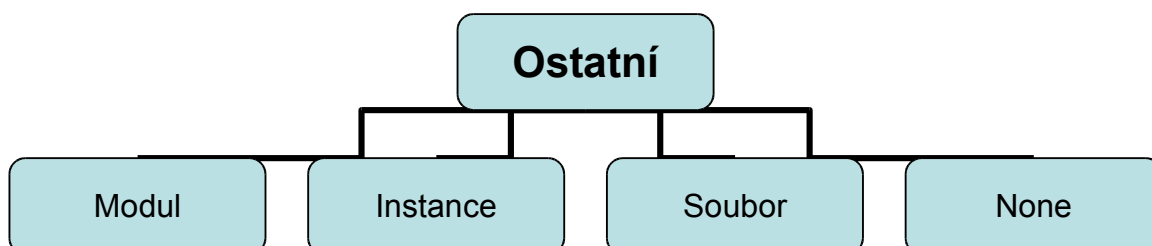
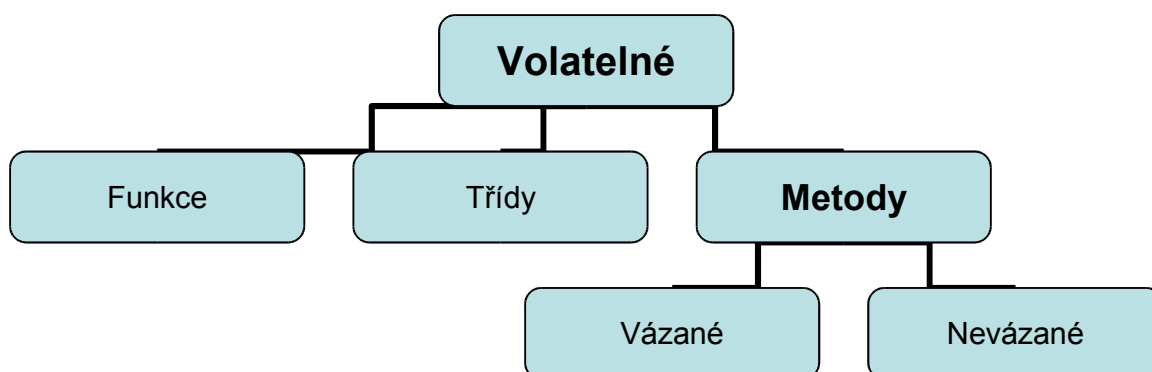
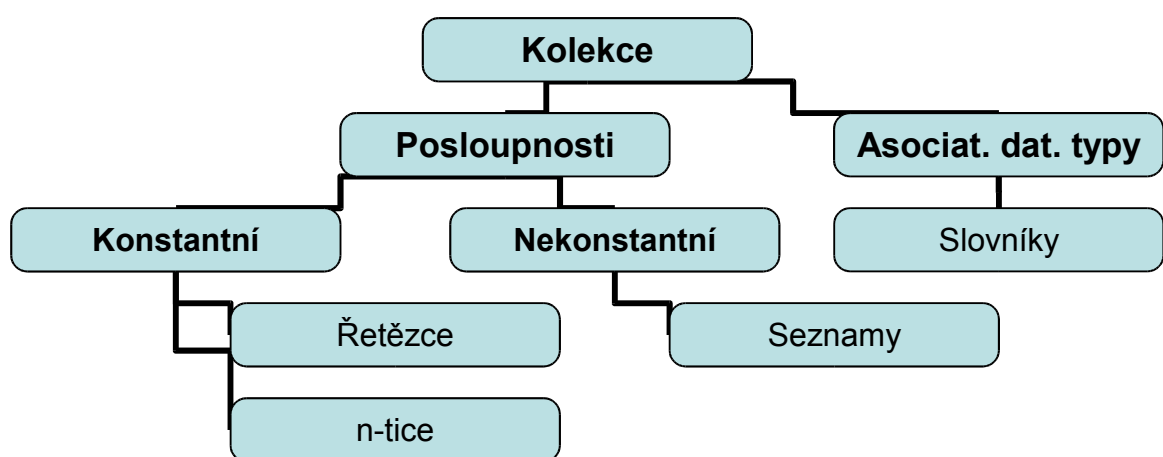
Přehled standardních typů, n-tice a typ None

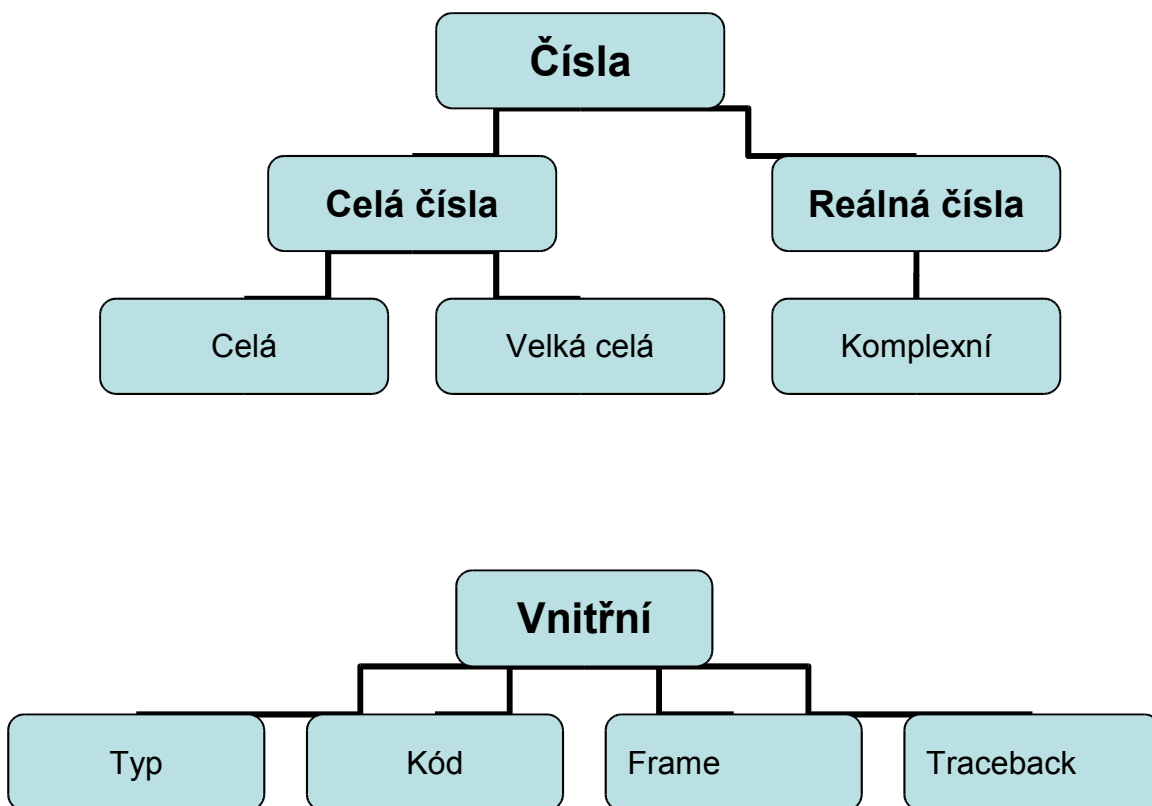
Přehled standardních typů

N-tice (tuple)

Typ None

Přehled standardních typů





N-tice (tuple)

N-tice (tuple) jsou typem kolekce. N-tice jsou:

- konstantními posloupnostmi a umožňují přístup přes pořadí
- mají pevnou délku, mohou být heterogenní a vnořené
- jsou poli referencí na objekty.

N-tice se uzavírají do závorek, může existovat prázdná n-tice, v případě jednoprvkové n-tice musíme za prvek napsat čárku, aby bylo zřejmé, že se nejedná o výraz, ale o n-tici. Opakováním a skládáním získáme novou n-tici. N-tice jsou konstantní, nemůžeme prvky přidávat či odebrat, aniž bychom vytvořili novou n-tici. N-tice mohou obsahovat libovolné objekty včetně seznamů, slovníků a dalších n-tic.

Příklady:

```

>>> t1 = () # prázdná n-tice
>>> t2 = (1,) # jednoprvková n-tice
>>> t3 = (1, 2, 3, 4) # čtyřprvková n-tice
>>> t4 = (5, (6, 4)) # vnořená n-tice
>>> t3 + t4 # složená n-tice
(1, 2, 3, 4, 5, (6, 4))
>>> len(t4) # délka n-tice - je 2, protože je složená
2
>>> t2 * 5 # opakování prvků n-tice
(1, 1, 1, 1, 1)
    
```

Informatika a výpočetní technika

```
>>> 3 in t3          # test na výskyt v n-tici, výsledek: ano
1
>>> for x in t3: x   # výpis prvků n-tice, výstup čísel
1
2
3
4
>>> for x in t3: x,   # výpis prvků - jednoprvkových n-tic
(1,)
(2,)
(3,)
(4,)
>>> t3[1]           # výběr prvku n-tice
2
>>> t3[0:2]        # výběr části n-tice
(1, 2)
>>> t5 = t4 + t3    # nová n-tice
>>> t5
(5, (6, 4), 1, 2, 3, 4)
>>> t6 = (t4, t3)
>>> t6
((5, (6, 4)), (1, 2, 3, 4))
```

Operací nazývanou **skládání n-tice** se rozumí přiřazení více prvků jedné proměnné. Tato proměnná pak bude tvořena n-ticí obsahující tyto prvky.

Opačnou operací je **rozklad sekvencí**. V tomto případě stojí na levé straně proměnné oddělené čárkou a straně pravé pak n-tice obsahující prvky, které budou přiřazeny proměnné:

```
>>> x, y, z = t
```

Rozklad sekvencí vyžaduje, aby délka n-tice na pravé straně byla stejná jako délka výčtu proměnných na straně levé. Je zde určitá asymetrie, zatímco skládání vždy vytvoří n-tici, rozklad funguje na libovolnou sekvenci (tj. jak na n-tici, tak i na řetězce a seznamy)

Pro konverzi posloupnosti `s` na n-tici se užívá funkce `tuple(s)`.

Seznamy nebo n-tice

N-tice jsou konstantní, takže poskytují na rozdíl od seznamů jistotu, že se nemohou změnit. Proto některé funkce vyžadují argumenty právě n-tice a nikoli seznamy.

Základní pravidlo: pokud by se uspořádaná kolekce měla měnit, použijeme seznamy, pokud nikoli, dáme přednost n-tici.

Typ None

Typ `None` označuje prázdný objekt. V Pythonu je právě jeden prázdný objekt, který je v programu zapsán jako `None`. Tento objekt je vrácen funkcemi, které nevracejí žádnou hodnotu. `None` se používá jako standardně předdefinovaná hodnota volitelných argumentů, aby funkce mohla zjistit, že při volání byla ve skutečnosti použita nějaká hodnota pro daný argument.

`None` nemá žádné atributy a v booleovských výrazech se vyhodnocuje jako nepravda (`false`).

Literatura:

- [1] Rubeš, J.: Nebojte se programovat, ComputerMedia, Bedihošť 2001
- [2] Lutz, M., Ascher, D.: Naučte se Python, Grada, Praha 2003
- [3] Beazley, D. M.: Python, Neocortex, Praha 2002
- [4] Python Reference Manual
- [5] Švec, J.: Létající cirkus, Python tutoriál, 2003