

# Výrazy

## Priorita operátorů

## Rozšířené přiřazení

## Konverze typu

## Řazení porovnáváním

Výrazy jsou kombinace objektů a operátorů, které Python vyhodnotí jako objekt.

### Priorita operátorů

Následující tabulka je uvádí v přehledu operátory jazyka Python jejich prioritu:

Operátor	Popis
(...) [...] {...} `...`	n-tice, seznam, slovník, řetězec
x[i] x[i:j] x.y x(,,,)	výběr prvků a částí, bližší určení volání funkcí
-x +x	unární mínus a plus (aritmetická negace a její opak)
-x	bitový doplněk
x ** y	umocňování (asociativní zprava)
x * y x / y x % y	násobení (opakování), dělení, zbytek po celočíselném dělení (formátování řetězce)
x + y x - y	sčítání (skládání), odčítání
x << y x >> y	bitový posun
x & y	bitové and
x ^ y	bitové xor
x   y	bitové or
< <= > >= == <> !=	relační operátory
is is not	test identity
in not in	test příslušnosti k množině
not x	logická negace
x and y	logický součin
x or y	logický součet
lambda argumenty: výraz	definice anonymní funkce

Tab. – Priorita operátorů

### Použití závorek

Pomocí závorek lze prioritu měnit, postupuje se od vnitřních k vnějším, užívají se pouze kulaté.

### Rozšířené přiřazení

Jazyk Python poskytuje operátory pro rozšířené přiřazení jako např. x+=y což odpovídá x = x + y (podobně např. x\*=y nebo x^=y).

## Konverze typu

Funkce	Popis
<code>int(x[,base])</code>	Konvertuje x na celé číslo
<code>long(x[,base])</code>	Konvertuje x na dlouhé celé číslo
<code>float(x)</code>	Konvertuje x na číslo s pohyblivou řádovou čárkou
<code>complex(real[,imag])</code>	Vytváří komplexní číslo
<code>str(x)</code>	Konvertuje x na řetězcovou reprezentaci
<code>repr(x)</code>	Konvertuje x na výrazový řetězec
<code>eval(str)</code>	Vyhodnocuje řetězec a vrací objekt
<code>tuple(x)</code>	Konvertuje posloupnost x na n-tici
<code>list(x)</code>	Konvertuje posloupnost x na seznam
<code>chr(x)</code>	Konvertuje celé číslo na znak
<code>unichr(x)</code>	Konvertuje celé číslo na znak Unicode
<code>ord(x)</code>	Konvertuje jednotlivý znak na jeho celočíselnou hodnotu
<code>hex(x)</code>	Konvertuje celé číslo na hexadecimální řetězec
<code>oct(x)</code>	Konvertuje celé číslo na oktálový řetězec

Tab. 2 – Konverze typů

### Příklady:

```
int('FFFF',16)
65535
complex(12,-2)
(12-2j)
str(125)
'125'
eval('12+56')
68
repr(18*8-2)
'142'
tuple([1,])
(1,)
list((2,4,-3))
[2, 4, -3]
chr(65)
'A'
ord('c')
99
```

## Řazení porovnáním

V Pythonu lze pomocí standardních operátorů porovnávat libovolné dva objekty.

### Čísla

Čísla se porovnávají obvyklým způsobem. Pouze komplexní čísla nejsou porovnávána v souladu s matematikou, protože se nejdříve porovnává jejich reálná část a potom imaginární část. Takže výsledek porovnávání komplexních čísel prakticky nemá smysl.

### Řetězce

Řetězce se porovnávají lexikograficky. Velká písmena mají menší hodnotu než písmena malá.

- Posloupnosti** Libovolné posloupnosti stejného typu jsou porovnávány prvek po prvku. Je-li některý prvek jednoho seznamu menší než odpovídající prvek druhého seznamu, je menší i tento seznam. Dojde-li se při porovnávání až na konec některého seznamu a všechny prvky byly dosud stejné, je kratší seznam menší. Obdobně probíhá porovnávání i u dvou n-tic.
- Slovníky** Dva slovníky se porovnávají tak, že se jejich prvky setřídí podle klíče a pak se porovnávají prvek po prvku. V praxi se předpokládá, že se zjišťuje, zda dva slovníky mají pod stejnými klíči uloženy stejné objekty – tedy zda se rovnají.
- Objekty různých typů** Porovnávání dvou objektů rozdílného základního typu se provádí libovolně, ale důsledně. Znamená, že záleží na verzi Pythonu, ve které je vždy stanoveno, zda například seznamy jsou menší než řetězce (nebo naopak). V rámci jedné verze se pak na to dá spolehnout.
- Další typy** Porovnávání většiny dalších objektů (souborové objekty, funkce, třídy) se děje na základě porovnávání jejich adresy v paměti. Proto rozdílné objekty stejného typu i obsahu se sobě nemohou rovnat, protože mají rozdílné umístění v paměti. Chceme-li tuto nelogičnost odstranit, musíme ve třídě odpovídajícím způsobem definovat speciální metodu `_cmp_`, kterou potom Python užije jako porovnávací funkci, když bude zapotřebí porovnávat objekty této třídy.

### 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
- [6] Harms, D., McDonald K.: Začínáme programovat v jazyce Python, Computer Press, Brno 2003