

Standardní číselné typy a operace s čísly

Standardní číselné typy

Operace s čísly

Standardní číselné typy

Mezi standardní číselné typy patří:

- Celé číslo
- Celé číslo s libovolnou přesností
- Číslo s pohyblivou řádovou čárkou
- Komplexní číslo

Celé číslo (IntType)

Celé číslo (*IntType* – *integer*) zapisujeme jako číslo v desítkové soustavě s případným znaménkem a bez desetinné tečky. Celé číslo je zpravidla implementováno jako slovo, má tedy délku 32 bitů.

Celá čísla lze zapisovat i osmičkové a šestnáctkové soustavě:

- čísla v **osmičkové** soustavě začínají **nulou**. Příklad:

```
>>> 023
```

```
19
```

- čísla v **šestnáctkové** soustavě začínají **0x** nebo **0X**. Příklad:

```
>>> 0x2a
```

```
42
```

Celé číslo s libovolnou přesností (LongType)

Připojíme-li k celému číslu **L** nebo **l**, získáme z něj **velké celé číslo** (*LongType* - *long integer*). Můžeme tak pracovat s téměř neomezenou přesností čísel, protože u tohoto typu jsme omezeni jen velikostí paměti. Nevýhodou je menší početní rychlost při výpočtu.

Příklad:

```
>>> 123456789L*3
```

```
370370367L
```

Operátory posuvu a bitové logické operátory popsané v části věnované logickým operacím a operacím s bity je možno aplikovat pouze na celá čísla a dlouhá celá čísla.

Číslo s pohyblivou řádovou čárkou (FloatType)

Racionální čísla – čísla s pohyblivou řádovou čárkou (*FloatType*), mohou obsahovat číslice, desetinnou tečku, znaménko, exponent uvozený písmenem **e** nebo **E** a znaménko exponentu. V paměti počítače jsou většinou reprezentována 64 bity. Maximální přesnost je většinou 17 platných číslic a exponent je v rozsahu -308 až 307. Příklad:

```
>>> -2.33E-2*-860
```

```
20.038
```

Pokud chceme získat podíl dvou čísel jako číslo racionální, musí alespoň jedno být ve tvaru racionálních čísla. Příklad:

```
>>> 5/2.0
```

```
2.5
```

Komplexní číslo (ComplexType)

Komplexní čísla (*ComplexType*) jsou v Pythonu zapisována ve tvaru *<reálná část>* *<imaginární část>*; imaginární jednotka se zapisuje jako **j** nebo **J**. Vnitřně jsou komplexní

Čísla reprezentována dvojicí čísel v pohyblivé řádové čárce. Operace s nimi se řídí pravidly pro komplexní čísla.

Příklady:

```
>>> 1+2j
(1+2j)
>>> 1j**2
(-1+0j)
>>> (2+3j)*2
(4+6j)
```

Operace s čísly

Pro všechny numerické typy jsou definovány tyto operátory:

Operátor	Význam	Příklad
$-x$	unární mínus (aritmetická negace)	>>> -5 -5
$x ** y$	Umocňování	>>> 2**3 8
$x * y$	Násobení	>>> 7*5 35
x / y	celočíselné dělení	>>> 7/3 2
$x \% y$	modulo (zbytek po celočíselném dělení)	>>> 7%3 1
$x + y$	Sčítání	>>> 12+8 20
$x - y$	Odčítání	>>> 31-6 25

Pro jednoduchost jsou všechny příklady pro celočíselný typ.

Operátor modulo obecně vrací zbytek po dělení x / y . Ačkoliv tato operace má praktický smysl pouze pro zbytek po celočíselném dělení, lze ji užít i pro výpočet zbytku při dělení čísel s pohyblivou čárkou, kdy vrací výsledek $x - \text{int}(x / y) * y$. Operace modulo pro komplexní čísla vrací výsledek $x - \text{int}(x / y1.\text{real}) * y$.

Vestavěné funkce pro všechny číselné typy

Funkce	Význam
<code>abs(x)</code>	Absolutní hodnota
<code>divmod(x, y)</code>	Vrací $(\text{int}(x / y), x \% y)$
<code>pow(x, y [, modulo])</code>	Vrací $(x ** y) \% \text{modulo}$
<code>round(x, [n])</code>	Zaokrouhlení na nejbližší násobek 10^{-n} (pouze pro čísla s pohyblivou řádovou čárkou)

Funkce `pow()` může nahradit operaci umocňování, s argumentem modulo pak v sobě kombinuje operace umocňování a modulo, což se často užívá v šifrovacích algoritmech.

Funkce `round()` zaokrouhluje číslo s pohyblivou řádovou čárkou na nejbližší násobek 10^{-n} . Není-li argument `n` uveden, je standardně nastaven na číslo 0, takže se zaokrouhluje na celky. Pokud je hodnota mezi dvěma násobky přesně uprostřed, zaokrouhluje se na číslo s větší absolutní hodnotou (tedy 0,5 na 1, -0,5 na -1).

Operace s čísly jsou platné jen tehdy, jsou-li operandy téhož typu. Pokud se operandy liší, provádí se konverzní operace, která konvertuje jeden typ na druhý:

Informatika a výpočetní technika

1. Je-li kterýkoliv operand komplexním číslem, konvertuje se druhý operand na komplexní číslo.
2. Je-li kterýkoliv operand číslem s pohyblivou řádovou čárkou, konvertuje se druhý operand na číslo s pohyblivou řádovou čárkou.
3. Je-li kterýkoliv operand dlouhým celým číslem, konvertuje se druhý operand na dlouhé celé číslo.
4. Jinak obě čísla musí být čísla celými a žádná konverze se neprovádí.

Konverze typů

Funkce	Význam
<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 v pohyblivé řádové čárce
<code>Complex(real [,imag])</code>	Vytváří komplexní číslo

Literatura:

- [1] Rubeš, J.: Nebojte se programovat, Computer Media, 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 circus, Python tutoriál, 2003