

Jython: Numerické funkcie (pokračovanie)

Na ilustráciu vstavaných matematických funkcií možno uviesť tie, ktoré sa používajú najčastejšie:

```
abs(číslo)

Vráti absolútnu hodnotu argumentu.

coerce(číslo1, číslo2)

Vráti tuple ... (výsledok1, výsledok2). Tuple je dátový typ, o ktorom bude reč neskôr, a to v sekcii pojednávajúcej o sekvenčných dátových typoch. Výsledky reprezentujú čísla (číslo1 a číslo2) prevedené na rovnaký číselný typ. Výsledným dátovým typom je presnejší dátový typ.

>>> cislo1 = 123
>>> cislo2 = 2L
>>> coerce(cislo1, cislo2)
(123L, 2L)
>>>
>>> cislo3 = 0.5
>>> coerce(cislo1, cislo3)
(123.0, 0.5)
>>>
>>> coerce(cislo2, cislo3)
(2.0, 0.5)
```

```
divmod(x, y)

Vráti tuple celých čísel. Prvá položka je celočíselný podiel x/y, druhá položka je zostatok po delení. Čísla odovzdané funkciou nemusia mať celočíselný dátový typ. (Tuple, vid. coerce ...).
```

```
pow(základ, exponent [, mod])

Vráti hodnotu základ ** exponent. Pokiaľ je zadaná i nepovinná hodnota mod, výsledok je rovný (základ ** exponent) % mod.
```

```
round(číslo, počet_číslic=0)

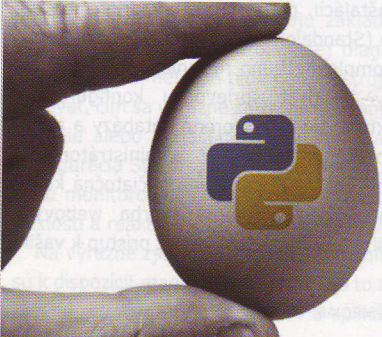
Vráti číslo zaokrúhlené na zadaný počet číslic.
```

Na typovú konverziu možno použiť funkcie int(), long(), float() a complex(), ako je zjavné z uvedených príkladov:

```
>>> int(3.5)
3
>>>
>>> long(22)
22L
>>>
>>> float(3)
3.0
```

Reťazce

Pretože v Jythone je všetko objektom, i reťazce sú objekty. To znamená, že s reťazcom sa pracuje ako s objektom, a nie ako s počtom znakov. Treba dať pozor na skutočnosť, že reťazce sú konštantné, čo znamená, že raz vytvorený reťazec už nemožno meniť. Jython neobsahuje vstavanú triedu premenných reťazcov, ktorá by bola ekvivalentom triedy StringBuffer v Jave. Obdobne ako v Jave i v Jythone sú reťazce vnútorne kódované v kódovaní Unicode.



Existuje niekoľko rôznych spôsobov, ako možno vytvárať reťazcové konštanty. Reťazce môžu byť uzatvorené v dvojítych i v jednoduchých úvodzovkách:

```
>>> "Retazec v dvojitych uvodzovkach"
'Retazec v dvojitych uvodzovkach'
>>>
>>> 'Retazec v jednoduchych uvodzovkach'
'Retazec v jednoduchych uvodzovkach'
>>>
>>> msg = "Warovanie!!!" je asi zle napisane!
>>> msg
'Warovanie!!!' je asi zle napisane'
```

Vnútri úvodzoviek možno uviesť špeciálne znaky pomocou escape sekvencií. Escape sekvencie použiteľné v Jythone sú uvedené v tabuľke 2-1.

| Escape sekvencie | Význam |
|------------------|---|
| \\ | Spätná lomka |
| ' | Apoštrof |
| " | Úvodzovky |
| \r | Návrat vozika |
| \t | Tabulátor |
| \xxxx | 16-bitový znak Unicode, xxxx (hex) |
| \b | Cúvnutie (znak backspace) |
| \n | Nová stránka |
| \N | Nový riadok |
| \N{meno} | Znak Unicode identifikovaný pomocou meno |
| \Uxxxxxxx | 32-bitový znak Unicode, xxxxxxx (hex) |
| \xhh | 8-bitový znak Unicode v hexadecimálnom zápise |

Tab. 2-1 Najdôležitejšie sekvencie Escape použiteľné v Jythone

Teraz sa pozrieme na základné operácie s reťazcami. Reťazce možno spájať. Reťazce uvedené vedľa seba sú spojené automaticky. Dá sa použiť i operátor + (krížik):

```
>>> 'prvy' + 'druhy reťazec'
'prvy druhy reťazec'
>>>
>>> slovo = 'prvy reťazec' + ' a druhy reťazec'
>>>
>>> slovo
'prvy reťazec a druhy reťazec'
```

Operátor * (hviezdica) slúži na opakovanie reťazca:

```
>>> lava_zatvorka = '<'
>>> prava_zatvorka = '>'
>>> lava_zatvorka * 3 + "AHOJ" + prava_zatvorka * 3
'<<<AHOJ>>>'
```

Reťazce môžu byť takisto indexované. To znamená, že možno pristupovať k ľubovoľnej časti reťazca prostredníctvom indexov. Prvý znak v reťazci má index 0. V Jythone neexistuje žiadny dátový typ analogický s dátovým typom char v Jave. Znak je jednoducho reťazec s dĺžkou 1.

```
>>> slovo = "Popokatepetl"
>>>
>>> slovo
'Popokatepetl'
>>> slovo[0]
'P'
>>> slovo[4]
'k'
```

Reťazec možno indexovať aj so zápornými indexmi. Posledný znak v reťazci sa dá získať napríklad takto:

```
>>> slovo
'Popokatepetl'
>>> slovo[-1]
'l'
```



Ako sme už uviedli, reťazce sú konštantné a nemožno ich zmeniť. Pokus o zmenu na indexovanej pozícii sa skončí chybou:

```
>>> slovo
'Popokatepetl'
>>> slovo[0] = 'K'
Traceback (innermost last):
  File "<console>", line 1, in ?
TypeError: can't assign to immutable object
```

Medzi pokročilejšie prvky v práci s reťazcami – a ako neskôr uvidíme, nielen s reťazcami – patria takzvané výrezy (anglicky slices, slice annotation). Ide o konštrukciu, keď z určitého reťazca vyberiete iba niektoré znaky. Výrezy možno zadávať uvedením rozmedzia od : az_do. Horné rozmedzie je od dolného oddelené dvojbodkou.

```
>>> slovo = 'Popokatepetl'
>>>
>>> slovo[0:4]
'Popo'
>>>
>>> slovo[4:6]
'ka'
```

Výrezy majú definované užitočné default hodnoty. Prvý index, pokiaľ mu nezadáme, má default hodnotu 0, druhý index je defaultne nastavený na dĺžku reťazca.

```
>>> slovo = 'Popokatepetl'
>>> slovo[:2] #Prve dva znaky
'Po'
>>> slovo[:4] #Prve štyri znaky
'Popo'
>>> slovo[:4:] #Všetky znaky okrem prvých 4
'katepetl'
>>> slovo[-2:] #Posledné dva znaky
'tl'
>>> slovo[:-2] #Všetky znaky okrem posledných 2
'Popokatepe'
```

Pozn.: Výrez vždy vracia nový objekt definovaný rozmedzím od : az_do.

Spomedzi základných funkcií na prácu s reťazcami možno uviesť nasledujúce:

```
len(reťazec)

Vráti počet znakov v reťazci.
>>> s = 'supercalifragilisticexpialidocious'
>>> len(s)
34

ord(znak)

Vráti celé číslo zodpovedajúce kódovaniu znaku.

chr(číslo)

Vráti reťazec dĺžky 1, obsahujúci znak zodpovedajúci danému kódovaniu. Argument číslo môže naberať hodnotu 1 – 255.

unichr(číslo)

Vráti Unicode reťazec dĺžky 1, obsahujúci znak zodpovedajúci danému kódovaniu. Argument číslo môže naberať hodnotu 1 – 65535. V Jythone nie je rozdiel medzi funkciami chr() a unichr().
```

ŠTEFAN HAVLIČEK, Sales Engineer, InterSystems B.V.