

# Jython: Komentáre

Ako každý programovací jazyk aj Jython má možnosť *komentárov*. Prax odporúča používať ich. Komentár by sa mal vždy objaviť v mieste, kde ste použili nejaký nezvyčajný programátorský obrat, a potom i v mieste, kde ste niečo skúšali. Ak to neurobite, dá sa očakávať, že v budúcnosti čas, ktorý strávite premýšľaním nad týmito neokomentovanými časťami, bude podstatne dlhší než čas, ktorý potrebujete na napísanie komentára.

Jython umožňuje iba *jednoriadkové komentáre*. Komentár sa začína znakom # (hash) a všetko, čo je za ním do konca riadka, sa chápe ako komentár.

```
>>> #Toto je komentar
>>>
>>> x = 1      #Toto je take komentar
```

## Premenné a identifikátory

Základným kameňom všetkých programov je premenná. Je to pomenovaná oblasť pamäte, v ktorej je niečo uložené. Čo to je, to závisí už od konkrétneho programovacieho jazyka. V Jythone premenná takmer vždy obsahuje odkaz (*referenciu*) na nejaký objekt. Objektom môže byť objekt základných vstavaných dátových typov, ako číslo, zoznam, tuple (nie, to nie je preklep, čo je *tuple*, to sa dozviete neskôr), asociatívne pole či iná inštancia z bohatého výberu tried Jythonu či Javy.

Názvy premenných, *identifikátory*, môžu byť ľubovoľné. Je však lepšie dodržiavať už osvedčené

pravidlo: použiť ľubovoľne dlhú postupnosť písmen anglickej abecedy, číslíc a podčiarkovníkov, ktorá sa nezačína číslicom.

Pomocou identifikátorov sa neoznačujú len premenné, ale i rad ďalších prvkov jazyka. Môžu to byť *funkcie, moduly, triedy*. Aby bol výsledný kód čitateľný i pre iných programátorov, a keďže je Jython úzko spätý s prostredím jazyka Java, je vhodné dodržiavať určité konvencie. Tie sú totiž dodržiavané tak v Jythone, ako aj v celom Java Core API a ich znalosť vám umožní (okrem iného) vyznať sa v ňom.

Jython rovnako ako Java rozlišuje malé a veľké písmená a pomerne často sa stretnete s tým, že dva identifikátory sú rozlíšené iba veľkosťou písmen. Množstvo znakov použitých v identifikátore nehrá rolu. Medzery nie sú povolené.

Identifikátory nemôžu mať rovnaké mená ako *klúčové slová*, ktoré sú v Jythone rezervované, a ich zoznam je uvedený v tabuľke.

and	def	finally	in	print
as	del	for	is	raise
assert	elif	from	lambda	return
break	else	global	not	try
class	except	if	or	while
continue	exec	import	pass	

## Vznik premennej

Jython je skriptovací jazyk, v ktorom premenné nie sú explicitne deklarované. Vznikajú automaticky pri ich prvom použití. To však neznamená, že tu neexistuje typová kontrola. Naopak, typová kontrola tu existuje, ale je pred používateľom

skrytá. Robí sa interne. V porovnaní s inými programovacími jazykmi môžeme tvrdiť, že *Jython patrí medzi jazyky s dynamickým priradovaním dátových typov so silnou typovou kontrolou*.

Premenná teda vznikne tak, že sa pomocou *priradovacieho príkazu vytvorí väzba medzi identifikátorom a objektom nejakého dátového typu*. Premenná potom obsahuje odkaz na tento objekt. Teda nie vlastnú hodnotu, ako napr. číslo 4, ale odkaz na objekt, ktorý toto číslo nejakým spôsobom reprezentuje.

## Základné dátové typy

### Numerické typy

#### Integer

Implementácia tohto typu zodpovedá dátovému typu `int` jazyka Java, t. j. 32-bitová celočíselná hodnota so znamienkom. **Jython nepodporuje kratšie celočíselné dátové typy korešpondujúce s javovými typmi `byte`, `short`.**

Výsledkom akejkoľvek operácie medzi celými číslami je vždy typ `integer`. To môže spôsobiť občas „čudné“ chyby pri delení:

```
>>> 1 / 2
0
>>> 7 / 3
2
>>> 7 / -3
-3
```

#### Long

Celočíselný dátový **typ s ľubovoľnou presnosťou**. Nesmie sa pliesť so základným dátovým typom `long`, ktorý poznáte z Javy. Jython reprezentuje

dátový typ `long` ako inštanciu triedy `java.math.BigInteger`. Literály typu `long` sa riadia rovnakými pravidlami ako typ `integer`, s jedinou výnimkou: sú ukončené písmenom `L` alebo `L`:

```
>>> # dlhe cislo
>>> d1 = 12345678123456789L
>>> d2 = 123456123654321L
>>>
>>> d1
12345678123456789L
>>>
>>> d2
123456123654321L
>>>
>>> d1/d2
100L
```

## Float

Reprezentuje dátový typ reálnych čísel. Jython implementuje reálne čísla, tak ako Java implementuje dátový typ `double`, t. j. so 64-bitovou presnosťou. Obdobne ako v prípade typu `integer` kratšie reálne dátové typy Jython neimplementuje. Reálne čísla sa zapisujú rovnako ako celé čísla, ale obsahujú desatinnú bodku.

Aritmetické operátory majú implementovanú plnú podporu reálnych čísel. V prípade, že aritmetický operátor obsahuje celočíselné aj reálne hodnoty, sú celočíselné hodnoty automaticky prevedené na reálne:

```
>>> 10 / 3.
3.3333333333333335
>>>
>>> 1. / 2.
```

```
0.5
>>>
>>> 0.025 / 3.
0.008333333333333333
```

## Complex

Jython obsahuje aj podporu komplexných čísel. Komplexné čísla sú reprezentované ako usporiadaná dvojica reálnych čísel. Zápis komplexných čísel je nasledujúci:

```
>>> 1 + 2j
(1+2j)
>>>
>>> 3.0j
3j
>>> 1.3 + 0.5j
(1.3+0.5j)
>>> complex(1,5)
(1+5j)
>>>
>>> (1 + 2j) / (3 - 1j)
(0.1+0.7j)
>>> a = 1 + 05.j
>>> a.real
1.0
>>> a.imag
5.0
```

## Numerické funkcie

Jython obsahuje niekoľko vstavaných funkcií, ktoré umožňujú pracovať s numerickými hodnotami. Okrem týchto funkcií možno použiť štandardný modul `math`, v ktorom sú definované ďalšie užitočné matematické funkcie.

■ ŠTEFAN HAVLIČEK,  
Sales Engineer, InterSystems B.V.