

Python – Základní operace

Vestavěné funkce a standardní moduly Pythonu

Michal Čihař

mcihar@suse.cz



Vestavěné funkce

Vestavěné funkce

- Převody mezi typy
- Základní matematické operace
- Práce s objektovým modelem
 - Přístup k prvkům či jejich vypsání
 - Dekorátory pro atributy či třídní metody
- V Pythonu 3 mnoho nahrazeno metodami odpovídajících objektů

Převody mezi typy

- Některé typy nelze vytvořit přímo
 - `bytearray('abc')`
 - `set((1, 3, 5))`
- Vynucené převody
 - `int()`, `str()`, `float()`, `tuple()`, `bool()`, ...

Manipulace s objekty

- Dědičnost
 - `super()`, `isinstance()`, `issubclass()`
- Práce s atributy
 - `setattr()`, `getattr()`, `delattr()`, `hasattr()`
 - `dir()`
- Dekorátory
 - Podrobněji probereme příště
 - `property()`, `classmethod()`, `staticmethod()`

Matematické operace

- Výpočty
 - `abs()`, `divmod()`, `pow()`
- Práce s poli
 - `max()`, `min()`, `sum()`
- Převody mezi soustavami
 - `oct()`, `hex()`, `bin()`
 - `int(x, base)`

Vstup a výstup

- Jednoduchý vstup
 - `input()`
 - Pokud je natažen modul `readline`, tak se použije
- Výstup
 - `print()`
 - V Pythonu 2.x se používalo `print "x"`

Nápověda

- Zobrazuje dokumentaci v interaktivním režimu
 - help(objekt)
 - help('jmeno')

Spouštění kódu

- Spuštění libovolného kódu
 - eval()
- Kompilace do byte-kódu
 - compile()
- Spuštění zkompilevaného kódu
 - exec()

Práce se soubory

Souborové objekty

- Standardní rozhraní pro souborové objekty
- Nemusí mít společného předka
- Téměř všude se dá použít vlastní objekt, pokud implementuje toto rozhraní

Otevření souboru

- Vestavěná funkce open
 - `open(soubor, režim, ...)`
- Režim otevírání
 - Přístup (`r/w/x/a`)
 - Zpracování textu (`t/b`)

Ošetření chyb

- Je možné použít try...except...finally
- Od Pythonu 2.6 je vhodnější použít kontext, který se automaticky postará o uzavření při odchodu z bloku:

```
with open('/etc/passwd') as passwd:  
    passwd.read()
```

Standardní knihovna

Standardní knihovna

- Obsahuje všechno důležité
- Práce s textem
 - string, re, difflib, unicodedata, ...
- Práce s čísly
 - math, cmath, decimal, fractions, random, ...
- Soubory a adresáře
 - os, stat, tempfile, glob, fnmatch, shutil, ...
- Formáty souborů
 - csv, configparser, xdrlib, ...

Přístup k systému a informace o něm

Modul sys

- Obsahuje základní informace o systému
- Konfigurace Pythonu
- Informace o příkazové řádce

Parametry z příkazové řádky

- Parametry uloženy v `sys.argv` – odpovídá `argv` v C

```
#!/usr/bin/python
```

```
import sys
```

```
for param in sys.argv:  
    print(param)
```

Informace o systému

- Informace o platformě
 - `sys.byteorder`
 - `sys.float_info`
 - `sys.getsizeof()`
 - `sys.platform`
- Informace o Pythonu
 - `sys.version`
 - `sys.api_version`
 - `sys.flags`

Standardní vstup a výstup

- Aktuální hodnoty, dají se měnit
 - `sys.stdin`
 - `sys.stdout`
 - `sys.stderr`
- Hodnoty ze spuštění
 - `sys.__stdin__`
 - `sys.__stdout__`
 - `sys.__stderr__`
- Některé mohou být `None`

Vyhledávací cesta

- Určuje adresáře, ve kterých Python hledá moduly
- Dá se nastavit proměnnou prostředí PYTHONPATH
- Základní cesty nastavené při kompilaci
- Standardně se vyhledává i v aktuálním adresáři
- Přístupné jako `sys.path`
- Je možné měnit za běhu:

```
#!/usr/bin/python
```

```
import sys
```

```
sys.path.append('/path/to/modules')
```

Rozhraní operačního systému

Modul os

- Základní rozhraní operačního systému
- Práce se soubory, procesy a terminály
 - Většina toho co je nadefinovaná v unistd.h
- Mnoho funkcí není dostupných na Windows

Prostředí

- Kompletní prostředí přístupné přes `os.environ`
 - Chová se jako standardní slovník
 - Změny se propagují do prostředí (pokud to platforma podporuje)
 - Načte se při prvním importu modulu `os` (typicky při startu)
- Funkce
 - Změny se neprojeví v `os.environ`
 - `os.getenv()`
 - `os.putenv()`
 - `os.unsetenv()`

Aktuální adresář

- Podobně jako v C
 - `os.chdir()`
 - `os.fchdir()`
 - `os.getcwd()`

Uživatelé a skupiny

- Podobně jako v C
 - `os.getuid()`
 - `os.getgid()`
 - `os.seteuid()`
 - `os.getgroups()`
 - ...

Práce s adresáři

- Výměna adresáře
 - `os.listdir()`
- Vytváření
 - `os.mkdir()`
 - `os.makedirs()`
- Procházení
 - `os.walk()`
- Vymazání
 - `os.rmdir()`

Práce se soubory

- Klasické operace
 - `os.chmod()`
 - `os.chown()`
 - `os.link()`
 - `os.mknod()`
 - `os.remove()`
 - `os.stat()`

File descriptor

- Nízkoúrovňová práce se soubory
- Vytvoření souborového objektu
 - `os.fdopen()`
- Běžné operace
 - `os.open()`, `os.dup()`
 - `os.read()`, `os.write()`
 - `os.close()`
 - ...

Práce s cestami

Modul os.path

- Manipulace s cestami souborů
- Testování vlastností cest

Převody cest

- Absolutní a relativní cesty
 - `os.path.abspath(path)`
 - `os.path.relpath(path[, start])`
- Uživatelské cesty (`~/Documents`)
 - `os.path.expanduser(path)`
- Kanonická cesta
 - `os.path.realpath(path)`

Rozdělení a spojení názvu souboru

- Rozdělování

- `os.path.basename()`

- Chová se jinak než `basename` v případě, že cesta končí na `/`

- `os.path.dirname()`

- `os.path.splitdrive()`

- `os.path.splitext()`

- Spojování

- `os.path.join()`

Testování cest

- Existence
 - `os.path.exists()`
- Různé typy
 - `os.path.isdir()`
 - `os.path.isfile()`
 - `os.path.islink()`
 - `os.path.ismount()`

Manipulace se soubory

Modul shutil

- Obsahuje funkce pro práci se soubory
- Operace podobné jako v shellu
 - `shutil.copy()`
 - `shutil.copytree()`
 - `shutil.rmtree()`
 - `shutil.move()`
- Mnoho rozšíření v Pythonu 3.2 a 3.3
 - 3.2: `shutil.make_archive()`, `shutil.unpack_archive()`
 - 3.3: `shutil.chown()`, `shutil.which()`, `shutil.disk_usage()`

Práce s textem

Formátování

- Python 2

```
>>> name = 'John'  
>>> print 'Hello %s' % name  
Hello John  
>>> print 'Hello %(name)s' % {'name': name}  
Hello John
```

- Python 3

```
>>> print('Hello {1}'.format(name))  
Hello John  
>>> print('Hello {name}'.format(name=name))  
Hello John
```


Regulární výrazy

- Modul re
- Syntaxe podobná jako v Perlu
- Různá rozšíření v podobě (?...)

-

```
>>> import re
>>> match = re.match(r'id=(?P<id>\d*)', 'id=123456')
>>> print(match.group('id'))
'123456'
```

Ukecané regulární výrazy

- Umožní se vyznat v regulárních výrazech
 - Ignoruje mezery komentáře

```
PRINTF_RE = r'''%(      # initial %
(?P<fullvar>
    [+#-]*           # flags
   (?:\d+)?         # width
   (?:\.\d+)?       # precision
    (hh\|h\|l\|ll)? # length formatting
    (?P<type>[\w%])) # type (%s, %d, etc.)
)'''
```

```
REGEXP = re.compile(PRINTF_RE, re.VERBOSE)
```


Další zdroje informací

Dokumentace

- Oficiální dokumentace
 - Popis jazyka, tutorial, popis standardní knihovny, API
 - <http://docs.python.org/2/>
 - <http://docs.python.org/3/>
- Knihy
 - Dive into Python 3 / Ponořme se do Pythonu 3
 - Expert Python Programming
- Vestavěná nápověda v příkazové řádce
- Prezentaci naleznete na
 - <http://cs.cihar.com/talks/2013/python/>

Nějaké dotazy?

Thank you.





Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org