

Python – jak na to?

Jan Švec
honza@py.cz

11. 12. 2006

Co je Python?

- Programovací jazyk
 - Interpretovaný
 - Objektivě orientovaný
 - Široce podporovaný
 - Otevřený
 - Perspektivní
 - „Bohatý“
 - Multiplatformní



Programovací jazyk

- Čitelný kód
 - Použití odsazení k označení bloků kódu
 - Klíčová slova (*for, if, def, class, import, ...*)
 - Volná typová kontrola
 - Modulární (programové moduly a balíčky)
- Datové typy
 - Řetězce (i UNICODE)
 - Seznamy a vektory
 - Asociativní pole
 - Množiny
 - Čísla (reálná, celá, long)

Použití Pythonu

- Skriptovací jazyk
 - automatizace činností
 - nadstavba nízkoúrovňového API
- Webové aplikace
 - seznam.cz, google.com
- Vědecké výpočty
- Programování GUI aplikací

- Cokoli jiného ...

Příklad: Výpočet faktoriálu I.

mod_fact.py

```
1 def fact(x):  
2     'Výpočet faktoriálu `x`'  
3     ret = 1  
4     for i in range(2, x+1):  
5         ret = ret * i  
6     return ret  
7  
8 print fact(10)  
3628800
```

Příklad:

Výpočet faktoriál II.

„Kolik má faktoriál 10 000 na konci nul?“

```
1 from mod_fact import fact
2
3 f10000 = fact(10000)
4
5 s_nulami = str(f10000)
6 bez_nul = s_nulami.rstrip('0')
7 print len(s_nulami) - len(bez_nul)
2499
```

Příklad:

Objektová kalkulačka I.

calc.py

```
1 from mod_fact import fact
2 class Calculator(object):
3     def __init__(self):
4         self.ans = 0
5     def add(self, x):
6         self.ans += x
7         return self.ans
8     def fact(self, x):
9         self.ans = fact(x)
10    return self.ans
```

Příklad:

Objektová kalkulačka II.

```
1 from calc import Calculator
2
3 obj = Calculator()
4 print obj.fact(10)
  3628800
5 obj.add(13)
6 print obj.ans
  3628813
```


Standardní knihovna

- Debugger, profiler
- Vlákna
- Serializace objektů
- Internetové protokoly a standardy
 - HTTP, FTP, email, CGI, XML-RPC
- Značkové jazyky
 - XML, HTML
- GUI

Další moduly a programy

- `scipy`
 - vědecké výpočty
- `epydoc`, `docutils`
 - generování dokumentace
- `IPython`
 - interaktivní shell
- `PyChecker`
 - kontrola zdrojového kódu

Odkazy a dokumentace I.

- python.org – domácí stránka
 - dokumentace, download, cheeseshop
- [Python Sidebar](#)
 - sidebar pro Firefox s dokumentací
- QuickRef.org
 - dokumentační portál nejen k Pythonu
- [Google Code Search](#)
 - vyhledávání ve zdrojových kódech
 - `lang:python`

Odkazy a dokumentace II.

- py.cz – Python v Čechách
 - konference python@py.cz
- py.cz/Honza
 - „Učebnice jazyka Python“



UNICODE a Python

Jan Švec
honza@py.cz

11. 12. 2006

Řetězce a Python

- Neměnný datový typ
- Pole znaků
- 2 typy
 - 8bitové řetězce
 - nutno určit kódování (iso-8859-2, cp1250, utf-8)
 - UNICODE řetězce
 - při komunikaci s vnějším světem konverze na 8bit
 - mnoho modulů pracuje přímo s UNICODE
 - PyXML
 - GUI moduly

Zápis řetězců

- Specifikace kódování zdrojového souboru

```
1 # -*- coding: iso-8859-2 -*-
```

- Zápis

```
2 retezec8 = "toto je osmibitový řetězec"
```

```
3 retezecUni = u"toto je unicode řetězec"
```

- Používání

```
4 print retezec8[10]
```

```
m
```

```
5 print retezecUni[0:4:2]
```

```
tt
```

Konverze řetězců

- 8bit → UNICODE
 - 1 `unicode(retezec8, "iso-8859-2")`
 - 2 `retezec8.decode("iso-8859-2")`
- UNICODE → 8bit
 - 3 `retezecUni.encode("utf-8")`
- možnost vzniku chyb
 - řízení dalším parametrem
 - 'strict', 'ignore', 'replace'

Čtení/zápis řetězců

- Výchozí kódování
 - nastavení v modulu site

```
sys.setdefaultencoding(kodovani)
```
 - zjištění

```
sys.setdefaultencoding()
```
- Čtení ze souboru → 8bit řetězec
- Zápis do souboru → převod do výchozího kódování

Čtení/zápis UNICODE řetězců

```
1 import codecs
2 import sys
3 fr = codecs.open(fn, "r", "utf-8")
4 for line in fr:
5     print line
6     # použití výchozího
7     # kódování
```