Python

Zdeněk Philipp - Seznam.cz

Něco málo omáčky

- http://www.python.org
- objektově orientovaný, interpretovaný, dynamický a silně typovaný
- multiplatformní (Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, Nokia mobile phones)
- jednoduchost, čitelná syntaxe
- snadno pochopitelný
- bohatství knihoven

Něco málo omáčky

- vysoce produktivní
- lehce rozšiřitelný (C, C++)
- lehce integrovatelný s ostatními programovacími jazyky
  - Java (Jython)
  - .NET (IronPython?)
  - C, C++
- nepotřebujeme IDE

K čemu to bylo navrženo a k čemu je to dobré

- původně byl zamýšlen jako druhý jazyk k nějakému systémovému jazyku
- postupem času se z něj stal výkonný nástroj, hlavní jazyk (Bruce Eckel)
- vhodný pro výuku programování
- prototypování aplikací

Co se v tom dá vytvořit

- webové aplikace (klient / server)
- síťové aplikace (TCP / UDP)
- skripty, utility
- systémové programy
- GUI aplikace (QT)
- práce s grafikou
- práce s textovými dokumenty, XML
- lze napsat i server (multithreadové, multiprocesové aplikace)

Na co se používá v seznam.cz

- webové aplikace (klient / server)
- aplikační servery
- doplňkové skripty

Výkon

- Pomalejší než C++, Java
- Zhruba na stejno jako Perl, PHP
- Kompiler Psyco 2-100x rychlejší
- Rozumná úvaha rychlost x efektivita
  - 5x méně ukecaný než Java a C++

Ještě než začneme (nic v tom nehledejte!)

```
a = 5
if a > 0:
    print "a je vetsi nez 0: %d" % a
    for i in a:
        print i
else:
    print "a neni vetsi nez 0"
```

středník není nutný

(Nejen) Objektově orientovaný

- paradigma programování
  - strukturované (procedurálním)
  - objektově orientované
  - funkcionální
- návrh jazyka (C, C++, Java, PHP)
- v Pythonu je všechno objekt
- podporuje objektově orientované
- podporuje strukturované programování
- umožňuje používat i některé užitečné techniky známé z funkcionálního programování

Python je Interpretovaný

- využívá byte code (Java, .NET)
- vyžaduje interpretr
- automaticky překládaný

Nenutí, nepřikazuje

- není nutné používat jednu ze zvolených metod programování
- pejsek s kočičkou vařili dort, Python ale uvařili velmi dobří kuchaři, vlastnosti zkombinovali velmi umně a chutně
- umožnuje přetěžování operátorů
- umožňuje skriptování v konzoli

Konzole

- help, dir
- Python jako kalkulačka

Základní datové typy

- číselné – bool, int, float, long
- stringové – str, unicode

Python dynamicky a silně typovaný

```
// C / C++ example
#include <stdio.h>

int main(int *argc, char *argv[]){
  printf("Vysledek scitani int a char = %d\n", (1 +
  "1"));
  return(0);
}
```

Python dynamicky a silně typovaný

```
// Java example
class Main {
    public static void main(String args[]) {
        System.out.println(
            "Vysledek scitani int a char = " + 1 + "1");
    }
}
```

Python dynamicky a silně typovaný

```
# python example
print 'Vysledek scitani int a string =', 1 + "1"
```

Zakladní datové struktury

- tuple
  - `t = “postel”, “lednicka”, “pocitac”`
- list
  - `l = [“postel”, “lednicka”, “pocitac”]`
- dictionary
  - `d = {“postel” : 5000, “lednicka” : 10000, “pocitac” : 15000}`
- object – necháme na později

Zakladní konstrukce

- **if-elif-else** - podmíněné vykonávání kódu
- **for-else** - opakované vykonávání kódu s konečným počtem cyklů (break)
- **while-else** - opakované vykonávání kódu s podmíněným počtem cyklů
- **try-except-finally-else** - zachycování výjimek

Hello world

```
#!/usr/bin/python

print “Hello world”
```

```
#!/usr/bin/python

def hello():
    print “Hello world”

if __name__ == “__main__”:
    hello()
```

Definice funkce

```
def funkce():
    print "Ja jsem funkce"


def funkce(a):
    print "Ja jsem funkce s parametrem %s" % a


def funkce(a = “1”):
    print "Ja jsem funkce s nepovinnym parametrem %s" % a
```

funkce nelze přetěžovat

Reference

- základní datové typy se kopírují
- datové struktury se předávají jako reference

Importy

- sdílení kódu (metody, objekty)
- používání celých knihoven
- vytváření knihoven a modulů
- import a from
  - from modul import knihovna
  - from modul.knihovna import funkce

Objekty v pythonu

- co to je objekt ?
- třída x objekt
- jak vytvořit objekt ?
- nepodporuje řízení přístupu ke členům objektů (na rozdíl od kompilovaných OOP jazyků, všechny členy objektů jsou public)
- umožňuje jejich skrytí (přístup celým jménem, mangling)
- Python je dynamický jazyk, umožňuje vytváření a modifikaci tříd přímo za běhu programu

Objekty v pythonu

```
# vytvoreni objektu
class Volant:
    pass

a = Volant()

# konstruktor a destruktor
class Volant:

    def __init__(self):
        print “Ahoj ja jsem konstruktor tridy Volant”

    def __del__(self):
        print “Ahoj ja jsem destruktor tridy Volant”

    def barva(self):
        print “Cerna”
```

Nakonec pár příkladů

- načtení souborů
- zmenšení křečka
- možná ještě něco ???

Načtení dat ze souboru

```
f = open("krecek.jpg", "r")
i = f.read()
f.close()
```

Příklad zmenšení křečka

```
import Image
i = Image.open("krecek.jpg")
i = i.resize((800, 600), 1)
i.save("krecek1.jpg")
```

KONEC

sejdeme se na přijímacím pohovoru :-)