

# PYTHON

SEZNÁMENÍ S PROGRAMOVACÍM JAZYKEM

**Textové soubory**  
**Práce s moduly**  
**Moduly třetích stran**

Lukáš Kotek

# TEXTOVÉ SOUBORY

## ▪ Použití základních nástrojů pro práci se soubory

- **open** a **close** – zpřístupnění a uzavření souboru + vrácení OS
- **read** vs. **readline** vs. **readlines** – jakým způsobem načítat data
- **write** – zápis řetězce → analogicky jako u funkce **print**
- direktiva **with** – syntaxe pro blok kódu → nepoužije se **close**

## ▪ Vybrané režimy práce s textovými soubory

- "r" – režim pouze pro čtení
- "w" – režim pro zápis
- "a" – připojování na konec souboru

```
soubor = open("data.txt", "r")
print(soubor.readlines())      # Vrati obsah jako seznam
print(soubor.readline())      # Vrati 1 radek ...
print(soubor.read())          # Vrati soubor jako string
soubor.close()
```

# POUŽITÍ FUNKCE OPEN

## ▪ Použití funkce

- Možnost práce s různými kódováními
- Výchozí chování v Pythonu 3, dříve vhodné použít modul **codecs**

```
soubor = open("data.txt", "r", encoding="utf8")
radky = soubor.readlines()
for radek in radky:
    x = radek.split(",") # Metoda pro rozdeleni retezce
    print(type(x))
    print(x)
```

# Alternativne s direktivou **with**:

```
with open("data.txt", "r", encoding="utf8") as soubor:
    # Zbytek zapisu je analogicky
```

# ZPŮSOBY IMPORTU MODULŮ

## ▪ Použití příkazu **import**

- Zpřístupnění funkcí / proměnných / objektů u modulu
- Přistupujeme vždy pře název modulu

```
import math  
math.pow(4, 2)
```

## ▪ Použití příkazu **from**

- Určujeme, které prvky z modulu se mají zpřístupnit přímo

```
from math import pow # Zprístupni pouze funkci pow  
pow(4, 2)
```

```
from math import * # Pozor, zprístupni vse z modulu
```

## ▪ Použití příkazu **as**

- Pro zpřehlednění a zjednodušení zápisu

```
import math as m  
m.pow(4, 2)
```

# MODULY TŘETÍCH STRAN

## ▪ Proč použít modul třetí strany?

- Velké množství modulů v základu... nicméně obrovské celkově
  - Patrně největší síla Pythonu
- Zajímavé možnosti v kombinaci s tvořenými virtualenvs
- Předkompilované moduly s příponou \*.whl („wheel“)
  - Praktická možnost na MS Windows
    - <http://www.lfd.uci.edu/~gohlke/pythonlibs/>

## ▪ Nástroj pip

- Základní použití z příkazového řádku MS Windows / unixového shellu

```
pip install nazev_baliku
```

```
pip freeze      # nainstalovane baliky
```

# MODUL LOGGING

## ▪ Jak vytvářet logovací záznamy?

- Pomocí běžné práce s textovými soubory... ale proč vymýšlet kolo?
- Pomocí specializovaného modulu, např. logging

```
import logging
import datetime
import getpass # Pro zjistiení uzivatele

logger = logging.getLogger(__name__)
FORMAT = "%(levelname)-8s [%(asctime)-15s] [%(user)-8s]
[%(process)d] [%(funcName)-14s] - %(message)s"

logging.basicConfig(filename='log_{0}.log'.format(
    datetime.date.today()), format=FORMAT)
logger.setLevel("INFO") # Co reálně logovat
d = {'user': getpass.getuser()}

logger.debug("Podrobne info", extra=d) # Nezaologuje se
logger.info("Zakladni info", extra=d) # Zalohuje se
```

# JAK PRACOVAT S DATY V XML

## ▪ Použití modulu xml

- Načíst ze souboru pomocí metody **parse** nebo ... transformovat řetězec pomocí metody **fromstring**
- Když je „vše ztraceno“... je tu **BeautifulSoup!**

```
import xml.etree.ElementTree as XML

strom = XML.parse("zdrojovy_soubor.xml")
print(type(strom))
print(strom)

struktura = strom.getroot()

for zaznam in struktura:
    print(str(zaznam.tag))          # Pristup k datum
    print(str(zaznam.attrib))
```

<https://docs.python.org/2/library/xml.etree.elementtree.html>

<https://www.crummy.com/software/BeautifulSoup/bs4/doc>

# JAK PRACOVAT S DATY V JSON

- **Použití modulu json**

- Reprezentace dat je mnohem **přirozenější pro Python** než XML
- Lze použít zápis odpovídající běžným datovým (dict a list)

```
import json
```

```
# Nactení bezneho textoveho souboru obsahujiciho JSON
```

```
zdroj = open("data.json", "r", encoding="utf-8")  
data = json.load(zdroj)    # Prevedeni na JSON  
zdroj.close()
```

```
print(type(data))  
print(data)
```

*<https://docs.python.org/2/library/json.html>*



# JAK VYTVOŘIT HTTP REQUEST

- **Použití modulu urllib**

- K dispozici sada předpřipravených funkcí
- Lze definovat proxy a port, typ odesílaných dat  
→ plnohodnotné využití pro cokoliv
- Alternativa? Modul **requests**!

```
import urllib.request

web = urllib.request.urlopen("http://www.root.cz")
dokument = web.read()
data = dokument.decode("utf-8")    # Predpokladame utf-8

print(type(data))
print(data)
print(web.getcode())    # Zjistí výsledný http kód
```

*<https://docs.python.org/2/library/urllib.html>*

Otázky?