Cursus Programmeren en Dataverwerking
Les #3

http://hay.github.io/codecourse

Vanmiddag (14.00 - 17.30)

- Opdracht van vorige week nakijken
- Bestanden openen in Python
- Functies
- Modules
- CSV parsen in Python
- Als er tijd over is: JSON en XML parsen
- Huiswerkopdracht
- 

Opdracht #6

- Herschrijf de vorige opdracht met een list in plaats van hoe het nu is, maar met **drie** namen ipv **twee**.
- Schrijf een programma dat om drie namen vraagt en een vrucht.
- Doe voor beide namen het volgende:
- Als de naam begint met een hoofdletter print je “Aangenaam $naam” en anders print je “Hoi $naam”
- Print of de lettercombinatie “te” voorkomt in de naam
- Print nu “$naam is een $vrucht”, waar $vrucht in hoofdletters is.

```
names = ["Bert", "Ernie", "Pino"]

for name in names:
    print "Aangenaam, " + name

print names[1]
print names[-1]

hellos = []

for name in names:
    hellos.append("Hallo " + name)

for hello in hellos:
    print hello

years = [1983, 1980, 1993]

for year in years:
    print year

print 1983 in years

eighties = range(1980, 1989)

for year in years:
    print "%s in de jaren tachtig? %s" % (year, year in eighties)

things = [42, True, "Hallo"]
```

Opdracht #7

- Maak een list met daarin twee dicts waar de titel en jaar van verschijnen instaan.
- Vraag vervolgens aan de gebruiker om zelf ook een film met titel en verschijningsjaar toe te voegen.
- Loop nu door de drie nieuwe films heen en print:
  - De naam van de film in KAPITALEN
  - Het jaar van verschijnen
  - Hoeveel jaar dit geleden is

```
person = {
    "name" : "Hay",
    "birthyear" : 1983
}

print "%s is geboren in %s" % (person["name"], person["birthyear"])

persons = [
    {
        "name" : "Hay",
        "birthyear" : 1983
    },
    {
        "name" : "Lotte",
        "birthyear" : 1981
    }
]

for person in persons:
    name = person["name"]
    birthyear = person["birthyear"]
    print "%s is geboren in %s" % (name, birthyear)
```

```
git pull
```

Bestanden openen en schrijven

```
people = open("people.txt")

for person in people:
    print person

people.close()
```

```
people = open("people.txt")

for person in people:
    print person

allpeople = people.readlines()
print allpeople

people.close()
```

```
with open("people.txt") as people:
    for person in people:
        print person
```

```
with open("people.txt") as people:
    allpeople = people.readlines()
    print allpeople
```

```
people = open("people.txt")
awesomepeople = open("awesomepeople.txt", "w")

for person in people:
    person = person.strip() # Verwijder newline
    aweseomeperson = "%s is echt super!\n" % person
    awesomepeople.write(aweseomeperson)

people.close()
awesomepeople.close()
```

```
people = open("people.txt")
awesomepeople = open("awesomepeople.txt", "w")

for person in people:
    person = person.strip() # Verwijder newline
    aweseomeperson = "%s is echt super!\n" % person
    awesomepeople.write(aweseomeperson)

people.close()
awesomepeople.close()
```

Opdracht #8

- Open zowel de bestanden ‘people.txt’ en ‘twitter.txt’.
- Combineer beide bestanden in een nieuw bestand waar dit in staat:

```
Hay Kranen zit op Twitter sinds mei 2007

Lotte Baltussen zit op Twitter sinds februari 2009
```

- Hint: voor de oplossing moet je technieken gebruiken die je nog kent van de les over *lists*.
  Je kan alle regels van een bestand in een *list* zetten, maar een bestand is zelf ook een *iterable.*

Functies

```
people = open("people.txt")

for person in people:
    person = person.strip() # Verwijder newline
    print "%s is echt super!" % person

people.close()
```

```
people = open("people.txt")

for person in people:
    person = person.strip() # Verwijder newline
    print "%s is echt super!" % person

people.close()

# Nog een keer!

people = open("people2.txt")

for person in people:
    person = person.strip() # Verwijder newline
    print "%s is werkelijk briljant!" % person

people.close()
```

```
people = open("people.txt")

for person in people:
    person = person.strip() # Verwijder newline
    print "%s is echt super" % person

people.close()

# Nog een keer!

people = open('people2.txt")

for person in people:
    person = person.strip() # Verwijder newline
    print "%s is werkelijk briljant!" % person

people.close()
```

```
def shownames(filename, remark):
    f = open(filename)

    for item in f: # Verwijder newline
        item = item.strip()
        print "%s is %s!" % (item, remark)

shownames("people.txt", "echt super")
print "---" * 10
shownames("people2.txt", "werkelijk briljant")
```

Opdracht #9

- Schrijf een functie die als argument een bestandsnaam accepteert. De functie opent het bestand, print de naam en print vervolgens alle regels uit het bestand met aan het begin het regelnummer

- Roep deze functie aan met de bestanden “people.txt” en “people2.txt”

- Output moet dus zo zijn:

```
Printing “people.txt”
0 Hay Kranen
1 Lotte Baltussen
…
```

```python
def shownames(filename, remark):
    f = open(filename)

    for item in f: # Verwijder newline
        item = item.strip()
        print "%s is %s!" % (item, remark)

shownames("people.txt", "echt super")
print "---" * 10
shownames("people2.txt", "werkelijk briljant")
```

```
def hoi():
    print "HOI!"
    hoi()

hoi()
```

```
counter = 0

def tel():
    counter = counter + 1
    print counter
    tel()

tel()
```

Modules

```
import functions
functions.shownames("people.txt", "echt super")
```

```python
def shownames(filename, remark):
    f = open(filename)

    for item in f: # Verwijder newline
        item = item.strip()
        print "%s is %s!" % (item, remark)

if __name__ == "__main__":
    shownames("people.txt", "echt super")
    print "---" * 10
    shownames("people2.txt", "werkelijk briljant")
```

```
import functions2
from functions2 import shownames

functions2.shownames("people.txt", "echt super")
shownames("people2.txt", "een supergoede muzikant")
```

string re struct difflib cStringIO textwrap codecs unicodedata stringprep
fpformat datetime calendar collections heapq bisect array sets sched mutex
weakref types new copy pprint repr numbers math cmath decimal fractions random
itertools functools operator ospath fileinput stat statvfs filecmp tempfile glob
fnmatch linecache shutil dircache macpath pickle cPickle copy_reg shelve marshal
anydbm whichdb dbm gdbm dbhash bsddb dumbdbm sqlite3 zlib gzip bz2 zipfile
tarfile csv robotparser netrc xdrlib plistlib hashlib hmac md5 sha os io time
argparse optparse getopt logging loggingconfig logginghandlers getpass curses
cursestextpad cursesascii cursespanel platform errno ctypes select threading
thread dummy_threading dummy_thread multiprocessing mmap readline rlcompleter
subprocess socket ssl signal popen2 asyncore asynchat email json mailcap mailbox
mhlib mimetools mimetypes mimify multifile rfc822 base64 binhex binascii quopri
uu sgmllib htmllib htmlentitydefs xmletreeElementTree xmldom xmldomminidom
xmldompulldom xmlsax xmlsaxhandler xmlsaxsaxutils xmlsaxxmlreader
xmlparsersexpat webbrowser cgi cgitb wsgiref urllib urllib2 httplib ftplib
poplib imaplib nntplib smtplib smtpd telnetlib uuid urlparse cookielib xmlrpclib
audioop imageop aifc sunau wave chunk colorsys imghdr sndhdr ossaudiodev gettext
locale cmd shlex ttk turtle pydoc doctest unittest 2to3 test testtest_support
bdb pdb hotshot timeit trace distutils ensurepip sys sysconfig __builtin__
future_builtins __main__ warnings contextlib abc atexit traceback __future__ gc
inspect site user fpectl code codeop rexec imp importlib imputil zipimport
pkgutil modulefinder runpy parser ast symtable symbol token keyword tokenize
tabnanny pyclbr py_compile compileall dis pickletools formatter msilib msvcrt
_winreg winsound posix pwd spwd grp crypt dl termios tty pty fcntl pipes
posixfile resource nis syslog commands ic macostools findertools autoGIL
gensuitemodule aetools aepack aetypes al cd fl flp fm gl imgfile jpeg
sunaudiodev

string **re** struct difflib cStringIO textwrap codecs unicodedata stringprep fpformat **datetime** calendar collections heapq bisect array sets sched mutex weakref types new copy pprint repr numbers **math** cmath decimal fractions **random** **itertools** functools operator ospath fileinput stat statvfs filecmp tempfile **glob** fnmatch linecache shutil dircache macpath pickle cPickle copy_reg shelve marshal anydbm whichdb dbm gdbm dbhash bsddb dumbdbm sqlite3 zlib gzip bz2 zipfile tarfile **csv** robotparser netrc xdrlib plistlib hashlib hmac md5 sha **os** io time **argparse** optparse getopt **logging** loggingconfig logginghandlers getpass curses cursestextpad cursesascii cursespanel platform errno ctypes select threading thread dummy_threading dummy_thread multiprocessing mmap readline rlcompleter subprocess socket ssl signal popen2 asyncore asynchat email **json** mailcap mailbox mhlib mimetools mimetypes mimify multifile rfc822 base64 binhex binascii quopri uu sgmllib htmllib htmlentitydefs xmletreeElementTree xmldom xmldomminidom xmldompulldom xmlsax xmlsaxhandler xmlsaxsaxutils xmlsaxxmlreader xmlparsersexpat webbrowser cgi cgitb wsgiref urllib urllib2 httplib ftplib poplib imaplib nntplib smtplib smtpd telnetlib uuid urlparse cookielib xmlrpclib audioop imageop aifc sunau wave chunk colorsys imghdr sndhdr ossaudiodev gettext locale cmd shlex ttk turtle pydoc doctest unittest 2to3 test testtest_support bdb **pdb** hotshot timeit trace distutils ensurepip sys sysconfig __builtin__ future_builtins __main__ warnings contextlib abc atexit traceback __future__ gc inspect site user fpectl code codeop rexec imp importlib imputil zipimport pkgutil modulefinder runpy parser ast symtable symbol token keyword tokenize tabnanny pyclbr py_compile compileall dis pickletools formatter msilib msvcrt _winreg winsound posix pwd spwd grp crypt dl termios tty pty fcntl pipes posixfile resource nis syslog commands ic macostools findertools autoGIL gensuitemodule aetools aepack aetypes al cd fl flp fm gl imgfile jpeg sunaudiodev

python™

search

» Package Index

PACKAGE INDEX »
- Browse packages
- Package submission
- List trove classifiers
- List packages
- RSS (latest 40 updates)
- RSS (newest 40 packages)
- Python 3 Packages
- PyPI Tutorial
- PyPI Security
- PyPI Support
- PyPI Bug Reports
- PyPI Discussion
- PyPI Developer Info

ABOUT »

NEWS »

DOCUMENTATION »

DOWNLOAD »

COMMUNITY »

FOUNDATION »

CORE DEVELOPMENT »

PyPI - the Python Package Index

The Python Package Index is a repository of software for the Python programming language. There are currently **62797** packages here.

To contact the PyPI admins, please use the Support or Bug reports links.

Not Logged In

Login
Register
Lost Login?
Use OpenID

Status

Nothing to report

Get Packages

To use a package from this index either "pip install *package*" (get pip) or download, unpack and "python setup.py install" it.

Package Authors

Submit packages with "python setup.py upload". The index hosts package docs. You may also use the web form. You must register. Testing? Use testpypi.

Infrastructure

To interoperate with the index use the JSON, OAuth, XML-RPC or HTTP interfaces. Use local mirroring or caching to make installation more robust.

Updated	Package	Description
2015-07-12	jsmin 2.1.2	JavaScript minifier. PLEASE UPDATE TO VERSION >= 2.0.6. Older versions have a serious bug related to comments.
2015-07-12	pyMorfologik 0.1.1	Binding for Morfologik tool
2015-07-12	nesterboro 1.2.0	A simple printer of nested lists

pypi-ranking.info

requests	HTTP requests
lxml	XML parsing
django	Web framework (zwaar)
flask	Web framework (licht)
xmltodict	Makkelijk XML parsen
pyquery	HTML scrapen
numpy	Processen van grote sets data
unicodecsv	CSV met UTF-8 support

	A	B	C
1	naam	startdatum	handle
2	Hay Kranen	mei 2007	@hayify
3	Lotte Baltussen	februari 2009	@lottebelice
4	Jesse de Vos	september 2009	@85jesse
5	Nienke Huitenga	juni 2010	@wzzzt
6	Maarten Brinkerink	maart 2009	@mbrinkerink
7	Erwin Verbruggen	juli 2010	@erwinverb

```
naam,startdatum,handle
Hay Kranen,mei 2007,@hayify
Lotte Baltussen,februari 2009,@lottebelice
Jesse de Vos,september 2009,@85jesse
Nienke Huitenga,juni 2010,@wzzzt
Maarten Brinkerink,maart 2009,@mbrinkerink
Erwin Verbruggen,juli 2010,@erwinverb
```

```
csvfile = open("twitter.csv")
keys = []

for index, line in enumerate(csvfile):
    fields = line.split(",")

    if index == 0:
        keys = fields
    else:
        for i, field in enumerate(fields):
            print "%s: %s" % (keys[i].strip(), field.strip())

    print "_____"
```

```
import csv

csvfile = open("twitter.csv")
reader = csv.reader(csvfile)

for row in reader:
    print row
```

```
import csv

csvfile = open("twitter.csv")
reader = csv.DictReader(csvfile)

for row in reader:
    for key, val in row.iteritems():
        print "%s: %s" % (key, val)

    print "____"
```

Opdracht #10

- Schrijf een programma dat het bestand “names.csv” inleest.
- Voor elke rij check je of er een rare naam in voorkomt (je mag zelf bedenken wat “raar” is). Schrijf de rare namen weg in een nieuw bestand.
- Voor bonuspunten: schrijf ze weg naar een nieuw csv-bestand met aantal keer dat de naam voorkomt + geslacht (je mag Googlen!)

```
import csv

csvfile = open("twitter.csv")
reader = csv.DictReader(csvfile)

for row in reader:
for key, val in row.iteritems():
    print "%s: %s" % (key, val)

print "___"
```

```
{
    "total_rows": 1146,
    "offset": 0,
    "rows": [
        {
            "id": "COLLECT.10002",
            "key": null,
            "value": {
                "_id": "COLLECT.10002",
                "_rev": "1-08f0c8e90217dddb5a56a96b4fa6a293",
                "formats": [
                    {
                        "type": "image/jpeg",
                        "url": "http://www.rijksmuseum.nl/media/assets/SK-A-3830"
                    },
                    {
                        "type": "text/html",
                        "url": "http://www.rijksmuseum.nl/collectie/SK-A-3830"
                    },
                    "hoogte 185 cm",
                    "breedte 150 cm",
                    "materiaal: doek"
                ],
                "identifier": "COLLECT.10002",
                "language": "Dutch",
                "publisher": "Rijksmuseum, Amsterdam",
                "rights": "License: CC-BY. http://creativecommons.org/licenses/by/3.0/nl/.
                    Bron: Rijksmuseum, Amsterdam",
                "date": "1757 -  1757",
                "description": "Portret van Theodorus Bisdom van Vliet (1698-1777),
                    burgemeester van Haastrecht en hoogheemraad van de Krimpenerwaard, met
                    zijn gezin op een terras in de tuin van zijn huis te Haastrecht voor een
                    beeld met Neptunus en Mercurius. Afgebeeld zijn verder: zijn vrouw Maria
                    van Harthals (1703-63) en zijn kinderen linksboven: Cornelis (1737-73),
                    Maria Theodora (1739-1828) en Adriana Elisabeth (1742-76), linksonder:
                    Agatha (1743-76) Johanna Margaretha (1735-64) en Johan de Wijs
                    (1740-62), met fluit, en rechts Elisabeth (1727-64) met waaier en gearmd
                    met Marcellus (1729-1806), rechtsachter te paard Adriaan Jacob (1732-90)
                    en Evert (1733-86).",
                "creator": "schilder: Stolker, Jan",
                "type": "schilderij",
                "title": "Theodorus Bisdom van Vliet (1698-1777). Burgemeester van
                    Haastrecht en hoogheemraad van de Krimpenerwaard, met zijn gezin in de
                    tuin van zijn huis te Haastrecht",
                "coverage": "derde kwart 18e eeuw",
                "height": "185",
                "width": "150"
            }
        }
    ]
}
```

```
{
    "total_rows": 1146,
    "offset": 0,
    "rows": [
        {
            "id": "COLLECT.10002",
            "key": null,
            "value": {
                "_id": "COLLECT.10002",
                "_rev": "1-08f0c8e90217dddb5a56a96b4fa6a293",
                "formats": [
                    {
                        "type": "image/jpeg",
                        "url": "http://www.rijksmuseum.nl/media/assets/SK-A-3830"
                    },
                    {
                        "type": "text/html",
                        "url": "http://www.rijksmuseum.nl/collectie/SK-A-3830"
                    },
                    "hoogte 185 cm",
                    "breedte 150 cm",
                    "materiaal: doek"
                ],
                "identifier": "COLLECT.10002",
                "language": "Dutch",
                "publisher": "Rijksmuseum, Amsterdam"
```

```python
import json

jsondata = open("rijksmuseum.json").read()
data = json.loads(jsondata)
paintings = []

for row in data["rows"]:
    value = row["value"]

    painting = {
        "title" : value["title"],
        "date" : value.get("date", None),
        "creator" : value["creator"],
    }

    paintings.append( painting )

paintingsfile = open("paintings.json", "w")
paintingsjson = json.dumps(paintings)
paintingsfile.write(paintingsjson)
```

Opdracht #11

- Schrijf een filmdatabase programma met JSON als opslagformaat.
- Bij binnenkomst kan de gebruiker kiezen om een lijst te zien van alle films (1), een film toe te voegen (2) of te stoppen (3)
- Bij 1: toon een lijst van de films met jaartal en titel en het totaal aantal films in de database
- Bij 2: vraag om een titel en jaartal, voeg het toe aan de lijst en save het JSON bestand.
- Bij 3: sluit het programma af
- Hint: je hebt functies nodig en hele simpele recursie

```
import json

jsondata = open("rijksmuseum.json").read()
data = json.loads(jsondata)
paintings = []

for row in data["rows"]:
    value = row["value"]

    painting = {
        "title" : value["title"],
        "date" : value.get("date", None),
        "creator" : value["creator"],
    }

    paintings.append( painting )

paintingsfile = open("paintings.json", "w")
paintingsjson = json.dumps(paintings)
paintingsfile.write(paintingsjson)
```