Python (3/3)

PU Deep Learning

Nils Reiter, nils.reiter@uni-koeln.de

May 4, 2021 (Summer term 2021)

Recap

- List Comprehension
  - Define lists by specifying a pattern
    - `[x*2 for x in l1 if x < 10]`
- Functions
  - Weakly typed arguments
  - Named arguments, default values
  - Return values, `None` and `NoneType`

Today

Input/Output

Exception Handling

Python Packages

Exercise

Section 1

Input/Output

Introduction

- IO: Input/Output
- Standard model: Streams of bytes/characters: Most programming languages

Introduction

- IO: Input/Output
- Standard model: Streams of bytes/characters: Most programming languages
- Regular workflow
  1. Open a file
     - Configure: Mode, encoding, file type, ...
  2. Read from file or write to file
  3. Close file

IO in Python

Listing: File Reading

```
1  # open file and create a file
2  # object
3  fh = open(FILENAME)
4
5  # call a method on file object
6  for line in fh.readlines():
7    # process each line
8
9  # close object
10 # (important, because open files
11 # consume resources)
12 fh.close()
```

Listing: File Writing

```
1  # open file and create a file
2  # object
3  fh = open(FILENAME, mode="w")
4
5  # write list content into file
6  for line in listOfResults:
7    fh.write(line)
8
9  # close object
10 # (important, because open files
11 # consume resources)
12 fh.close()
```

IO in Python

open()

```
open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None, closefd=True, opener=None)
```

- https://docs.python.org/3/library/functions.html#open
- mode: Reading or writing
- buffering: Controls when and in which chunks data is actually written to the disk
- encoding: Specify encoding (hint: Make sure that UTF-8 is used)
- errors: How to handle encoding errors
- newline: Which characters are used to indicate a new line?
- ...

IO in Python

Methods of the File Object

- `https://docs.python.org/3/library/io.html#module-io`
- `close()`: Flush and close the stream
  - In Java, `close()` only closes without flushing
- `flush()`: Flush the stream
- `readline(size=-1)`: Read and return one line from the stream
- `readlines(hint=-1)`: Read and return a list of lines from the stream.
- `write(s)`: Write the string s to the stream and return the number of characters written.
- ...

Section 2

Exception Handling

Introduction

- Program errors
  - Variables have the wrong type (i.e., we try to multiply two strings)
  - IO issues: Disk full, file/directory doesn't exist
  - Parse errors: Input isn't how we expect it to be
  - Users errors: A user enters something unexpected
  - ...

Introduction

- Program errors
  - Variables have the wrong type (i.e., we try to multiply two strings)
  - IO issues: Disk full, file/directory doesn't exist
  - Parse errors: Input isn't how we expect it to be
  - Users errors: A user enters something unexpected
  - ...
- Error prevention: Verify before doing
  - Some errors cannot be anticipated
- Exception handling: Do, and specify behavior if error occurs

Exception Handling

- Anytime, code can ›throw an exception‹ (Python: `raise`)
- The exception can be caught by other code
- Documentation: `https://docs.python.org/3/tutorial/errors.html`

Exception Handling

- Anytime, code can ›throw an exception‹ (Python: `raise`)
- The exception can be caught by other code
- Documentation: `https://docs.python.org/3/tutorial/errors.html`

Example

```
1  l1 = [1,2,3,4,5,6]
2
3  try:
4    userInput = input()
5    intInput = int(userInput) # may raise a TypeError
6    print(l1[intInput])       # may raise an IndexError
7  except TypeError:
8    print("Please enter an integer number!")
9  except IndexError:
10   print("Doesn't exist")
```

Example

```
1  def myFunc(a1):
2    if type(a1) == str:
3      print(a1)
4    else:
5      raise TypeError
6
7  try:
8    myFunc("bla")
9  except TypeError:
10   print("An error occurred")
11
12 try:
13   myFunc(15)
14 except TypeError:
15   print("An error occurred")
```

Built-In Errors

- A number of error types are built into the standard library
- They are also used by library functions
- `https://docs.python.org/3/library/exceptions.html#Exception`

Built-In Errors

- A number of error types are built into the standard library
- They are also used by library functions
- `https://docs.python.org/3/library/exceptions.html#Exception`
- Feel free to use them!

Built-In Errors

- A number of error types are built into the standard library
- They are also used by library functions
- `https://docs.python.org/3/library/exceptions.html#Exception`
- Feel free to use them!

Error type misuse

- Nothing prevents you to use a `TypeError` for something unrelated to types
- But: Code becomes more difficult to maintain
- Don't do it

Define Your Own Exceptions

- Exceptions are classes that inherit from `BaseException`
- Own exceptions should be derived from `Exception`

```
1 class MyException(Exception):
2   pass
3
4 if a > len(l):
5   raise MyException
```

Exception Payload

It's possible to pass information through exceptions (e.g., the file name that doesn't exist)

Exception Payload

It's possible to pass information through exceptions (e.g., the file name that doesn't exist)

```
1  def openAndRead(filename):
2    fo = open(filename)
3    content = ""
4    for l in fo.readlines():
5      content.append(l)
6    fo.close()
7    return content
8
9  try:
10   content = openAndRead("filename.txt")
11 except OSError as err:
12   print("File " + err.filename + " does not exist.")
```

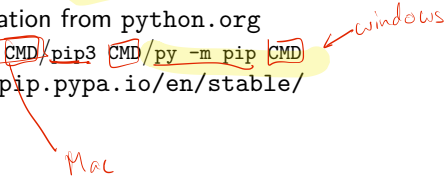
Section 3

Python Packages

Java: Maven, Gradle

Python Packages

- Central repository for python libraries: `https://pypi.org`
- Installation and local management: `pip`
  - Contained in python installation from `python.org`
  - Run on command line: `pip CMD` / `pip3 CMD` / `py -m pip CMD`
  - Documentation: `https://pip.pypa.io/en/stable/`



Python Packages

- Central repository for python libraries: `https://pypi.org`
- Installation and local management: `pip`
  - Contained in python installation from `python.org`
  - Run on command line: `pip CMD`/`pip3 CMD`/`py -m pip CMD`
  - Documentation: `https://pip.pypa.io/en/stable/`
- CMD: Various commands
  - Most important: `install LIBRARY`

Python Packages

- Central repository for python libraries: `https://pypi.org`
- Installation and local management: `pip`
  - Contained in python installation from `python.org`
  - Run on command line: `pip CMD`/`pip3 CMD`/`py -m pip CMD`
  - Documentation: `https://pip.pypa.io/en/stable/`
- CMD: Various commands
  - Most important: `install LIBRARY`

Example

```
1 pip install numpy
```

Section 4

Exercise

Exercise 04

https://github.com/idh-cologne-deep-learning/exercise-04