Modern Database Concepts - SS 2022
Prof. Dr. Florian Heinz
florian.heinz@oth-regensburg.de

OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Exercise Sheet 4

Exercise 1: BSON

Using a programming language and modules/libraries of your choice:

Create or find two files with semi-structured data (xml, json or yaml). These should contain at least one 32-bit integer

Import them to native data structures

Export them to a file using BSON (just append the second one to the first)

Dump them with "bsondump"

Change one of the integer values with a hexeditor and dump them again.

Exercise 2: YAML, JSON, XML

Using a programming language and modules/libraries of your choice:

Choose a JSON Api, for example from https://github.com/public-apis/public-apis

Request some data sets from the API

Convert the JSON responses to internal data structures

Merge these structures in a suitable way

Export it to a YAML document to have a nice looking output

Export it to an XML document for further processing with BaseX or similar tools

Exercise 3: JSONSchema

Using a programming language and modules/libraries of your choice:

Fetch the document http://mdc.sysv.de/ex4b.json

Build a suitable JSON-Schema for this file

Validate the document (for example with the python module "jsonschema") and find the 2 errors

Modern Database Concepts - SS 2022
Prof. Dr. Florian Heinz
florian.heinz@oth-regensburg.de
Exercise Sheet 4

OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Cheatsheet for python3

To install the required modules on the CIP-Pool hosts:

```
wget https://bootstrap.pypa.io/get-pip.py
python3 get-pip.py
python3 -m pip install bson jxmlease jsonschema pymongo
```

Reading a file:

```
open("filename", "r").read()
```

Writing to a file:

```
open("filename", "w").write("string")
```

Requesting an URL with **urllib.request**:

```
import urllib.request
urllib.request.urlopen('https://www.oth-regensburg.de/').read()
```

JSON encoding/decoding:

```
import json
data = json.loads('[1,2,3]')
json.dumps(data)
```

BSON encoding/decoding:

```
import bson
open("file.bson", "wb").write( bson.encode({"foo":"bar"}) )
bson.decode(open("file.bson", "rb").read())
```

YAML encoding/decoding:

```
import yaml
data = yaml.safe_load('[1,2,3]')
yaml.dump(data)
```

XML encoding/decoding:

```
import jxmlease
data = jxmlease.parse('bar')
jxmlease.XMLDictNode(data).emit_xml()
```

Hex-Editing without hexeditor:

```
xxd -g 1 file.bin > file.hex
edit file.hex
xxd -r file.hex > file2.bin
```