I523 Big data Applications
Group Assignment: Data Formats

Project members:
Jiaan Wang
Dhawal Chaturvedi
Matthew Millard
Weixuan Wang
Qiaoyi Liu
Shiqi Shen
Geng Niu

Details:
yaml, json, xml
1. Prepare a collaborative answer about what these data formats are and how they help. Contrast them and identify where you want to use the one or the other

Answer:
YAML:

- YAML is a data serialization language. Data serialization is the concept of converting structured data into a format that allows it to be shared or stored in such a way that its original structure can be recovered.
- It uses indentation format to define the structure just like python.
- YAML was initially called **Y**et **A**nother **M**arkup **L**anguage, referencing its purpose as a markup language, but it was then repurposed as **Y**AML **A**in't **M**arkup **L**anguage , to distinguish its purpose as data-oriented, rather than document markup.
- It essentially allows you to provide powerful configuration settings, without having to learn a more complex code type like CSS, JavaScript, and PHP.
- YAML doesn't support the use of tabs. Instead of tabs, it uses spaces which are not supported universally.
- YAML is case sensitive.
- YAML supports some basic data types which can be used with programming languages such as : **Scalars, Sequences** and **Mappings.**

Helpful links:

- https://en.wikipedia.org/wiki/YAML
- https://learn.getgrav.org/advanced/yaml
- https://www.tutorialspoint.com/grav/grav_yaml_syntax.htm
- http://www.yamllint.com/

YAML example:

```
# Employee records
- Dhawal:
    name: Dhawal Chaturvedi
    job: Developer
    skills:
      - Python
      - C
      - Java
- Weixuan:
    name: Weixuan Wang
    job: Developer
    skills:
      - lisp
      - fortran
      - erlang
```

JSON:

JSON stands for JavaScript Object Notation.
It is easy for humans to read and write. It is easy for machines to parse and generate.
JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others.

JSON's syntax matches JavaScript, but typically a parse function is used to convert JSON text to a JavaScript object. This adds a level of protection from malicious code since JSON data is often sent over the Internet on an unsecure channel. It also addresses bad data issues.

Helpful links:

https://www.w3schools.com/js/js_json_intro.asp
http://www.json.org/
http://www.electronicdesign.com/dev-tools/whats-difference-between-json-xml-and-yaml
http://bob.ippoli.to/archives/2005/12/05/remote-json-jsonp/

Json Example:

```
{
   "glossary": {
      "title": "example glossary",
                  "GlossDiv": {
         "title": "S",
                           "GlossList": {
            "GlossEntry": {
               "ID": "SGML",
                                          "SortAs": "SGML",
                                          "GlossTerm": "Standard Generalized Markup Language",
                                          "Acronym": "SGML",
                                          "Abbrev": "ISO 8879:1986",
                                          "GlossDef": {
                  "para": "A meta-markup language, used to create markup languages such as
DocBook.",
                                                   "GlossSeeAlso": ["GML", "XML"]
               },
                                          "GlossSee": "markup"
            }
         }
      }
   }
}
```

XML:

XML, or Extensible Markup Language, is a markup language that you can use to create your own tags. It was created by the World Wide Web Consortium (W3C) to overcome the limitations of HTML, the Hypertext Markup Language that is the basis for all Web pages. Like HTML, XML is based on SGML -- Standard Generalized Markup Language. Although SGML has been used in the publishing industry for decades, its perceived complexity intimidated many people that otherwise might have used it. The XML standard is a flexible way to create information formats and electronically share structured data via the public Internet, as well as via corporate networks.

- XML stands for EXtensible Markup Language
- XML is a markup language much like HTML.
- XML was designed to describe data.
- XML tags are not predefined in XML. You must define your own tags.
- XML is self describing.
- XML uses a DTD (Document Type Definition) to formally describe the data.

Helpful links:

- http://searchmicroservices.techtarget.com/definition/XML-Extensible-Markup-Language
- https://www.w3schools.com/xml/xml_whatis.asp

XML example:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Comparison of yaml, json and xml:

Summary:

Comparison of data formats:

- YAML stands for "*YAML Ain't Markup Language*".
- JSON stands for "*JavaScript Object Notation*".
- XML is "*eXtensible Markup Language"* .
- XML uses a tag to define the structure just like HTML.
- YAML uses indentation to define the structured data. So the each block in the YAML is differentiated by the number of white spaces.
- All three mentioned serialization language has same extension as their name. (.yaml for YAML, .json for JSON, .xml for XML).

Comparison of their usage:

- When you talk about Javascript, JSON is most prominent serialization language.
- For Java programming, you must have seen XML is widely used.
- Python has same indentation technique same as YAML. So Pythonista finds YAML more friendly than other serialization languages.

Helpful links:

- http://www.electronicdesign.com/dev-tools/whats-difference-between-json-xml-and-yaml
- http://www.csestack.org/yaml-vs-json-vs-xml-difference/

2. As we do lots of python, how does python use configurations. Is there a better alternative to use configurations in python?

Answer:
The simplest way to write configuration files is to simply write a separate file that contains Python code. It could be called something like databaseconfig.py. Then the line *config.py could be added to your .gitignore file to avoid uploading it accidentally.
A configuration file could look like this:

```
#!/usr/bin/env python
import preprocessing
mysql = {'host': 'localhost',
         'user': 'root',
         'passwd': 'my secret password',
         'db': 'write-math'}
preprocessing_queue = [preprocessing.scale_and_center,
                       preprocessing.dot_reduction,
                       preprocessing.connect_lines]
use_anonymous = True
```

Within the actual code, it can be used like this:

```
#!/usr/bin/env python
import databaseconfig as cfg
connect(cfg.mysql['host'], cfg.mysql['user'], cfg.mysql['password'])
```

4 Ways to manage the configuration in Python

- Using built-in data structure
  It uses the built-in data structure for managing the configuration.

- Using external configuration file such as **INI, JSON, XML or YAML**
  It loads the configuration values defined in the external file, not the built-in data structures.

- Using environment variables
  It uses system environment variables as configuration values.

- Using dynamic loading
  This is a more advanced way of using **built-in data structures** . In this approach, the config file does not have to located on importable path and can even be located on other repository.

Helpful links:

- https://martin-thoma.com/configuration-files-in-python/
- https://www.red-dove.com/config-doc/
- https://hackernoon.com/4-ways-to-manage-the-configuration-in-python-4623049e841b
- https://wiki.python.org/moin/ConfigParserShootout