data-encoding-cheatsheet.md 1/31/2022

Data Encoding cheatsheet

CSV

Python CSV documentation [https://docs.python.org/3/library/csv.html]

CSV Reading

reader() returns an object which will iterate over the lines in the given file. DictReader() returns an object which will iterate over the lines as a Dict.

```
import csv

# open people.csv and read as string
with open("people.csv", 'r') as file:
    reader = csv.reader(file, delimiter=',')
    for row in reader:
        print(row)

# open people.csv and read as dict
with open("people.csv", 'r') as file:
    csv_file = csv.DictReader(file)
    for row in csv_file:
        print(row)
```

CSV Writing

writer() creates an object that allows for writing to CSV using a list of values DictWriter() creates an object that allows for writing to CSV using a dictwriterow() adds a row to the file writeheader() writes a row with the fieldnames supplied in the constructor

```
import csv

# open people.csv and write row
with open('people.csv', mode='w') as file:
    writer = csv.writer(file, delimiter=',')
    # instruct the write to write a row
    writer.writerow(['Joe', 'Bloggs', 40])
    writer.writerow(['Jane', 'Smith', 50])

# open the people.csv and write row from dict
with open('people.csv', mode='w') as file:
    # set the headers for the CSV
    fieldnames = ['first_name', 'last_name', 'age']
    writer = csv.DictWriter(file, fieldnames=fieldnames)
    # instruct the writer to know to write the headers
    writer.writeheader()
```

data-encoding-cheatsheet.md 1/31/2022

```
# instruct the writer to write the row
writer.writerow({
    'first_name': 'Jan',
    'last_name': 'Smith',
    'age': 60
})
```

JSON

Python JSON documentation: [https://docs.python.org/3/library/json.html]

JSON to Python

loads () converts JSON String to a python object

```
import json

# some JSON:
x = '{ "first_name": "Joe", "age": 30, "last_name": "Bloggs"}'

# parse x:
y = json.loads(x)

# the result is a Python dictionary:
print(y["age"])
```

Python to JSON

dumps () converts a python object to JSON String

```
import json

# a Python object (dict):
x = {
    "first_name": "Joe",
    "age": 30,
    "last_name": "Bloggs"
}

# convert into JSON:
y = json.dumps(x)

# the result is a JSON string:
print(y)
```

XML

Python XML documentation: [https://docs.python.org/3/library/xml.etree.elementtree.html]

Reading XML

ET.parse() loads a XML file into a Tree object ET.getroot() returns the root Element of the XML Tree ET.fromstring() parses XML from a string into Element object Element.findall() query the XML using XPath Element.text returns the text value of the XML Element Element.attrib returns a dict of the attributes on the XML Element Element.tag returns the name of the Element

```
import xml.etree.ElementTree as ET

tree = ET.parse('country_data.xml')
root = tree.getroot()

root = ET.fromstring(country_data_as_string)
```

Writing XML

```
import xml.etree.ElementTree as ET

person = ET.Element('person')

last_name = ET.SubElement(person, 'last_name')
first_name = ET.SubElement(person, 'first_name')
age = ET.SubElement(person, 'age')

person_xml_string = ET.dump(a)
print(person_xml_string)
```