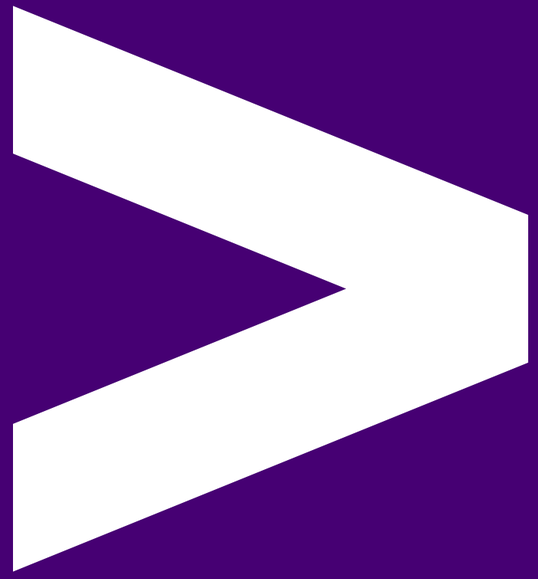
School of Tech
part of accenture

Data Encoding



School of Tech
part of accenture

Overview

- What is data encoding and why do we do it?
- CSV
- XML
- JSON

School of Tech
part of accenture

Learning Objectives

- To be able to explain what data encoding is and why we do it
- To gain an understanding of what CSV, XML & JSON are

What is data encoding?

Encoding is the process of converting data into a specified format.

Decoding is the reverse process - to extract information from the converted format.

Common formats:

- JPG, MP4, AVI
- Morse code, Braille
- JSON, XML, CSV
- Analog, Digital

Why do we encode data?

- Easier to store for computers (humans work with text, computers work with bytes)
- Removes redundancies from data (such as whitespace) so data size decreases
- Smaller data means it's more efficient to store and retrieve data

CSV (Comma Separated Values)

- A plain text file that uses specific structuring to arrange tabular data
- Only contains text data
- Each line of the file is a data record
- Is separated by a delimiter (comma, colon, tab etc.)

```
first_name, last_name, age
John,       Smith,     20
Sally,      Bloggs,    30
```

CSV - Dealing with commas

What about data that contains a comma?

Well, in that scenario, we quote the data

```
first_name, last_name, age, test_scores
John,       Smith,     20,  "80, 76, 92"
Sally,      Bloggs,    30,  "72, 84, 90"
```

CSV - Dealing with double quotes

How about data that contains a double quote?

We 'escape' the quote by using two of them together

```
tv,         size
"Samsung", "24"" TV"
"LG",      "41"" TV"
```

CSV in Python

Luckily for us, Python has its own csv library to read and write to/from CSV files.

CSV - Reading a File

Looking back to the previous module, opening a CSV can be done in a few lines like so:

```
import csv

with open('people.csv', 'r') as file:
    reader = csv.reader(file, delimiter=',')
    for row in reader:
        print(row)
```

`reader` is a function in the CSV library which returns an object which will iterate over the lines in the given file.

Reading to a Dictionary

We can read our CSV directly into a dictionary using `DictReader`:

```
import csv
with open('people.csv', 'r') as file:
    csv_file = csv.DictReader(file)
    for row in csv_file:
        print(row)
```

Output:

```
{'first_name': 'John', 'last_name': 'Smith', 'age:' 20}
{'first_name': 'Sally', 'last_name': 'Bloggs', 'age:' 30}
```

CSV - Writing to a File

```
import csv

with open('people.csv', mode='w') as file:
    writer = csv.writer(file, delimiter=',')

    writer.writerow(['Joe', 'Bloggs', 40])
    writer.writerow(['Jane', 'Smith', 50])
```

Writing from a Dictionary to CSV

```
with open('people.csv', mode='w') as file:
    fieldnames = ['first_name', 'last_name', 'age']
    writer = csv.DictWriter(file, fieldnames=fieldnames)

    writer.writeheader()
    writer.writerow({
        'first_name': 'Jan',
        'last_name': 'Smith',
        'age': 60
    })
```

`fieldnames` is required when writing from dictionary to csv.

School of Tech
part of accenture

Emoji Check:

Do you feel you understand CSV encoding format and how to utilise it with Python? Say so if not!

1. 😢 Haven't a clue, please help!
2. 😕 I'm starting to get it but need to go over some of it please
3. 😐 Ok. With a bit of help and practice, yes
4. 🙂 Yes, with team collaboration could try it
5. 😀 Yes, enough to start working on it collaboratively

School of Tech
part of accenture

Quiz Time! 🤓

When data is encoded in CSV format, what do we call the character (such as a comma or tab), which is used to separate different fields within a record?

1. `separator`
2. `limiter`
3. `fielder`
4. `delimiter`

Answer: `4`

You want to import the contents of a CSV file, and view the imported contents in dictionary format. Which of the following lines is likely to appear in your code?

1. `reader = csv.reader(file, delimiter=',')`
2. `csv_file = csv.DictReader(file)`
3. `writer = csv.writer(file, delimiter=',')`
4. `writer = csv.DictWriter(file, fieldnames=fieldnames)`

Answer: 2

Exercise - 15 mins

> Distribute exercise file: `exercises/data-encoding-exercises.md`.
>
> Utilise the `exercises/ford_escort.csv` file for the exercise.
>
> Let's all do "Part 1 - Reading and writing CSV", taking a look at working with CSV files.

Emoji Check:

How did you find exercises on using CSV file encoding with Python?

1. 😢 Haven't a clue, please help!
2. 😕 I'm starting to get it but need to go over some of it please
3. 😐 Ok. With a bit of help and practice, yes
4. 🙂 Yes, with team collaboration could try it
5. 😄 Yes, enough to start working on it collaboratively

XML

- Stands for 'Extensible Markup Language'
- Like HTML but for storing data rather than displaying data
- Multidimensional
- A form of semi-structured data

School of Tech
part of accenture

XML Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<people>
  <person>
    <first_name>John</first_name>
    <last_name>Cole</last_name>
    <age>20</age>
  </person>
  <person>
    <first_name>Sally</first_name>
    <last_name>Bloggs</last_name>
    <age>30</age>
  </person>
</people>
```

XML Advantages

- Can store highly structured data
- Human readable
- Well understood and used

XML Disadvantages

- 'Wordy' - metadata takes up a lot of space
- Becomes progressively inefficient the more complicated the structure of the data

School of Tech
part of accenture

Emoji Check:

Do you feel you understand XML encoding format? Say so if not!

1. 😢 Haven't a clue, please help!
2. 😕 I'm starting to get it but need to go over some of it please
3. 😐 Ok. With a bit of help and practice, yes
4. 🙂 Yes, with team collaboration could try it
5. 😄 Yes, enough to start working on it collaboratively

JSON

- JavaScript Object Notation
- A file format that uses human-readable text to store and transmit data objects
- Can also store semi-structured data like XML
- Also maps to multidimensional data
- Can hold any combination of objects using `{}` and lists using `[]`

JSON Example - Object

JSON objects start with { and end with }. JSON files often contain a root object, like so:

```
{
  "person": {
    "first_name": "John"
  }
}
```

School of Tech
part of accenture

JSON Example - List

JSON lists start with `[` and end with `]`. JSON files can contain lists, like so:

```
[
    {
        "first_name": "John",
        "last_name": "Smith"
    },
    {
        "first_name": "Sally",
        "last_name": "Matthews"
    }
]
```

This would be analogous to the CSV data we saw before. The above would be a valid json file.

School of Tech
part of accenture

JSON Example

JSON files can be arbitrarily complex, depending on your needs; This would be analogous to the XML example we saw before:

```
{
  "people": [
    {
      "person": {
        "first_name": "John"
      }
    },
    {
      "person": {
        "first_name": "Sally"
      }
    }
  ]
}
```

JSON Example - Code

```
import json

person = {
    "name": "Rob",
    "phone": 123456
}
# encode & write to a file
with open('example.json', 'w') as f:
    json.dump(person, f)
```

```
# read file and decode its content
with open('example.json', 'r') as f:
    my_data = json.load(f)
```

Python objects and their equivalent conversion to JSON

Python	JSON Equivalent
dict	object
str	string
list, tuple	array
int, float	number
True	true
False	false
None	null

JSON Advantages

- Human readable and much less wordy than XML
- Has become a data transfer and storage "standard"

JSON Disadvantages

- JSON isn't as robust a data structure as XML is.
- Can't use comments

School of Tech
part of accenture

Emoji Check:

Do you feel you understand JSON encoding format and how to utilise it with Python? Say so if not!

1. 😢 Haven't a clue, please help!
2. 😕 I'm starting to get it but need to go over some of it please
3. 😐 Ok. With a bit of help and practice, yes
4. 🙂 Yes, with team collaboration could try it
5. 😀 Yes, enough to start working on it collaboratively

School of Tech
part of accenture

Quiz Time! 🤓

School of Tech
part of accenture

Which of these is valid JSON?

```
// 1
{
  person: {
    first_name : "John"
  }
}

// 2
{
  "person": {
    "first_name" = "John"
  }
}
```

```
// 3
{
  "person": {
    "first_name": "John"
  }
}

// 4
{
  "person": [
    "first_name": ["John"]
  ]
}
```

Answer: 3

Exercise - 15 mins

> Continue working with: `exercises/data-encoding-exercises.md`.
>
> Utilise the `exercises/menu_items.json` file for the exercise.
>
> Lets all do "Part 2 - Reading and writing JSON", taking a look at working with JSON files.

Emoji Check:

How did you find exercises on using JSON file encoding with Python?

1. 😢 Haven't a clue, please help!
2. 😕 I'm starting to get it but need to go over some of it please
3. 😐 Ok. With a bit of help and practice, yes
4. 🙂 Yes, with team collaboration could try it
5. 😀 Yes, enough to start working on it collaboratively

Terms and Definitions - recap

CSV: A delimited text file that uses commas to separate values.

XML: a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

JSON: An open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and array data types (or any other serializable value).

Terms and Definitions - recap

Encoding: A system of rules to convert information into another form for communication through a communication channel or storage in a storage medium.

Parse: The process of analysing a string of symbols, either in natural language, computer languages or data structures, conforming to the rules of a formal grammar.

School of Tech
part of accenture

Overview - recap

- What is data encoding and why do we do it?
- CSV
- XML
- JSON

School of Tech
part of accenture

Learning Objectives - recap

- To be able to explain what data encoding is and why we do it
- To gain an understanding of what CSV, XML & JSON are

Further Reading

Reading and Writing CSV Files in Python

Reading and Writing XML Files in Python

Working with JSON Data in Python

School of Tech
part of accenture

Emoji Check:

On a high level, do you think you understand the main concepts of this session? Say so if not!

1. 😢 Haven't a clue, please help!
2. 😕 I'm starting to get it but need to go over some of it please
3. 😐 Ok. With a bit of help and practice, yes
4. 🙂 Yes, with team collaboration could try it
5. 😀 Yes, enough to start working on it collaboratively