

JSON

Read and Write

JSON Editor

Practice Problems

Discussion 10: JSON

SI 206: Data-Oriented Programming

Instructor: Dr. Barbara (Barb) Ericson

GSI: Kexuan (Michael) Huang

IA: Cristina & Jade

School of Information
University of Michigan

Fall 2023



JSON

Read and Write

JSON Editor

Practice Problems

Reminders

- Final project plans due on 11/20 - time to find a group!
- 2-3 students
- Instructions are up on Canvas

Deadlines

- Project 2 due this Friday (11/10)

JavaScript Object Notation

JSON

Read and Write

JSON Editor

Practice Problems

- Store and exchange data (as a **string**)
- Similar to nested dictionaries and lists in Python (use **null** instead of None)

JSON Example

```
1 {"employees": [
2     {"firstName": "John", "lastName": "Doe"},
3     {"firstName": "Anna", "lastName": "Smith"},
4     {"firstName": "Peter", "lastName": "Jones"}
5 ]}
```



Read and Write JSON

[JSON](#)[Read and Write](#)[JSON Editor](#)[Practice Problems](#)

- `json.loads(string)`: Takes a JSON string as input and returns a Python object (Dictionary or List)
- `json.dumps(object)`: Takes a Python object as input and returns a JSON string

Read and Write JSON

JSON

Read and Write

JSON Editor

Practice Problems

JSON Example

```
1 # content = '{"employees": [{"firstName": "John", "lastName":  
    ↪  "Doe"}, {"firstName": "Anna", "lastName": "Smith"}, {"firstName":  
    ↪  "Peter", "lastName": "Jones"}]}'  
2 import json  
3  
4 source_dir = os.path.dirname(__file__)  
5 full_path = os.path.join(source_dir, 'com.json')  
6 with open(full_path, 'r') as file:  
7     contents = file.read()  
8  
9 data = json.loads(content)  
10 data['employees'][2]['lastName'] # returns "Jones"
```



JSON Editor

- JSON Editor Online: <https://jsoneditoronline.org>
- It's super helpful to see the structure in a readable format
- We can also use pretty print library: `from pprint import pprint`

JSON Editor Online

New Open ▾ Save ▾ Settings ▾ Help

powered by ace

The screenshot shows the JSON Editor Online interface. On the left, there is a code editor window containing a JSON object. The object has a response_code of 0 and results with two items. The first result has a category of "Science & Nature", type of "boolean", difficulty of "medium", and a question about the word "Hippopotomonstrosesquippedaliophobia". The second result has a category of "Science & Nature", type of "boolean", difficulty of "hard", and a question about scientists growing teeth from urine. Both results have a correct_answer of "True" and incorrect_answers of ["False"].

object ► results ► 1 ►

- object {2}
 - response_code : 0
- results [2]
 - 0 {6}
 - category : Science & Nature
 - type : boolean
 - difficulty : medium
 - question : Hippopotomonstros esquippedaliophob ia is the irrational fear of long words.
 - correct_answer : True
- incorrect_answers [1]

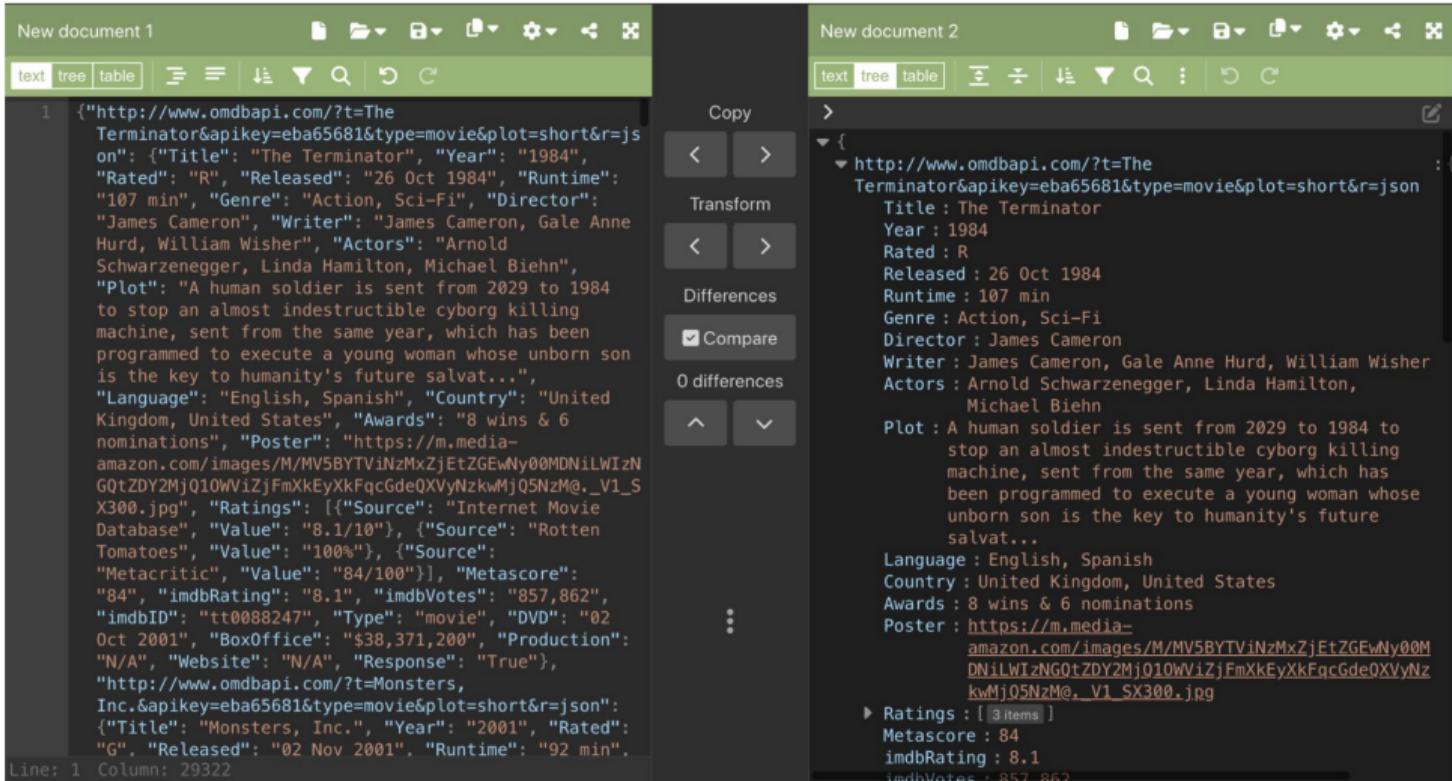
JSON Editor

JSON

Read and Write

JSON Editor

Practice Problems



The screenshot shows the JSON Editor application interface with two documents open.

New document 1:

```
1 {"http://www.omdbapi.com/?t=The Terminator&apikey=eba65681&type=movie&plot=short&r=json": {"Title": "The Terminator", "Year": "1984", "Rated": "R", "Released": "26 Oct 1984", "Runtime": "107 min", "Genre": "Action, Sci-Fi", "Director": "James Cameron", "Writer": "James Cameron, Gale Anne Hurd, William Wisher", "Actors": "Arnold Schwarzenegger, Linda Hamilton, Michael Biehn", "Plot": "A human soldier is sent from 2029 to 1984 to stop an almost indestructible cyborg killing machine, sent from the same year, which has been programmed to execute a young woman whose unborn son is the key to humanity's future salvation...", "Language": "English, Spanish", "Country": "United Kingdom, United States", "Awards": "8 wins & 6 nominations", "Poster": "https://m.media-amazon.com/images/M/MV5BYTViNzMXZjEtZGEwNy00MDNiLWIzNGQtZDY2MjQ10WViZjFmXkEyXkFqcGdeQXVyNzkwMjQ5NzM@._V1_SX300.jpg", "Ratings": [{"Source": "Internet Movie Database", "Value": "8.1/10"}, {"Source": "Rotten Tomatoes", "Value": "100%"}, {"Source": "Metacritic", "Value": "84/100"}], "Metascore": "84", "imdbRating": "8.1", "imdbVotes": "857,862", "imdbID": "tt0088247", "Type": "movie", "DVD": "02 Oct 2001", "BoxOffice": "$38,371,200", "Production": "N/A", "Website": "N/A", "Response": "True"}, {"http://www.omdbapi.com/?t=Monsters, Inc.&apikey=eba65681&type=movie&plot=short&r=json": {"Title": "Monsters, Inc.", "Year": "2001", "Rated": "G", "Released": "02 Nov 2001", "Runtime": "92 min"}}
```

Line: 1 Column: 29322

New document 2:

```
>
{
  "http://www.omdbapi.com/?t=The Terminator&apikey=eba65681&type=movie&plot=short&r=json": {
    "Title": "The Terminator",
    "Year": "1984",
    "Rated": "R",
    "Released": "26 Oct 1984",
    "Runtime": "107 min",
    "Genre": "Action, Sci-Fi",
    "Director": "James Cameron",
    "Writer": "James Cameron, Gale Anne Hurd, William Wisher",
    "Actors": "Arnold Schwarzenegger, Linda Hamilton, Michael Biehn",
    "Plot": "A human soldier is sent from 2029 to 1984 to stop an almost indestructible cyborg killing machine, sent from the same year, which has been programmed to execute a young woman whose unborn son is the key to humanity's future salvation...",
    "Language": "English, Spanish",
    "Country": "United Kingdom, United States",
    "Awards": "8 wins & 6 nominations",
    "Poster": "https://m.media-amazon.com/images/M/MV5BYTViNzMXZjEtZGEwNy00MDNiLWIzNGQtZDY2MjQ10WViZjFmXkEyXkFqcGdeQXVyNzkwMjQ5NzM@._V1_SX300.jpg",
    "Ratings": [
      {"Source": "Internet Movie Database", "Value": "8.1/10"},
      {"Source": "Rotten Tomatoes", "Value": "100%"},
      {"Source": "Metacritic", "Value": "84/100"}
    ],
    "Metascore": "84",
    "imdbRating": "8.1",
    "imdbVotes": "857,862",
    "imdbID": "tt0088247",
    "Type": "movie",
    "DVD": "02 Oct 2001",
    "BoxOffice": "$38,371,200",
    "Production": "N/A",
    "Website": "N/A",
    "Response": "True"
  }
}
```

Copy
< >
Transform
< >
Differences
 Compare
0 differences
^ v



Practice Problems

JSON

Read and Write

JSON Editor

Practice Problems

Go to Canvas → Assignment → Discussion 10 and clone the GitHub Repo.

Your tasks

- Implement `read_json(file)`: reads a JSON document, decodes the file content, and returns a dictionary.
- Implement `top_movies(rated, data)`: returns the top three movies based on the movie ratings category (`rated`) specified.
- Implement `get_longest_movie(data)`: returns the title of the longest movie.
- Implement `get_rotten_tomatoes(data)`: return a list of movies with a 95% - 100% Rotten Tomatoes rating.