# JSONTalk User Study

The aim of this user study is to investigate the effectiveness of JSONTalk. JSONTalk is a command-line tool that has been developed to generate natural language descriptions of JSON files. The primary use case of this tool will be to assist visually impaired programmers to quicker determine the structure and contents of a JSON file.

This evaluation will last roughly 15 minutes, and will consist of the following:

1. You will read through a recap on JSON syntax and semantics.

2. I will demonstrate how to use the tool.

3. You will be given several screen reader transcripts of JSON file descriptions and asked a few questions about them. You will then use the tool to assist you in answering the questions. Please remember that it is the system, not you, that is being evaluated.

You should fill in this form with your answers, which will be stored securely and anonymously on my laptop. Feel free to ask me any questions throughout the experiment. You are free to withdraw at any point during the experiment.

If you have any questions following the experiment, contact me at 2463548h@student.gla.ac.uk

* Required

1. Do you consent to take part in this user study and have your answers stored anonymously? *

*Mark only one oval.*

○ Yes

JSON Briefing / Recap

JSON (JavaScript Object Notation) is a text-based, lightweight data interchange format used to store and exchange data. It is based on a subset of the JavaScript programming language and is easy for humans to read and write and for machines to parse and generate.

Syntax:

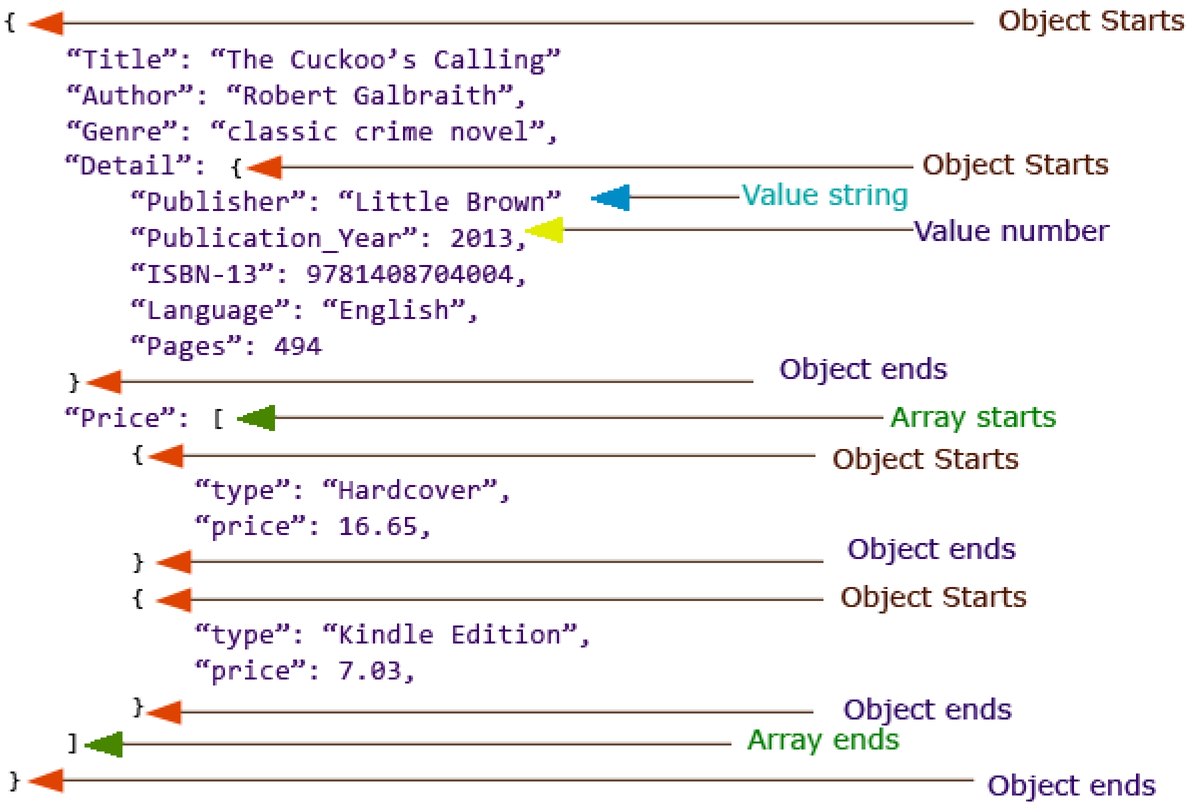
- Data is represented in key-value pairs, separated by a comma and enclosed within curly braces {}.
- Keys are strings, enclosed in double quotes "".
- Values can be a string, number, object, array, Boolean, or null. Strings are enclosed in double quotes "", while numbers don't have quotes.

Semantics:

- An object is an unordered set of key-value pairs.
- An array is an ordered collection of values, enclosed in square brackets [].
- A value can be a string, number, object, array, Boolean, or null.
- Strings are sequences of Unicode characters.
- Numbers can be integers or floating-point values.
- Booleans have only two values: true and false.
- The null value represents a deliberate non-value.

Here's an example JSON object:

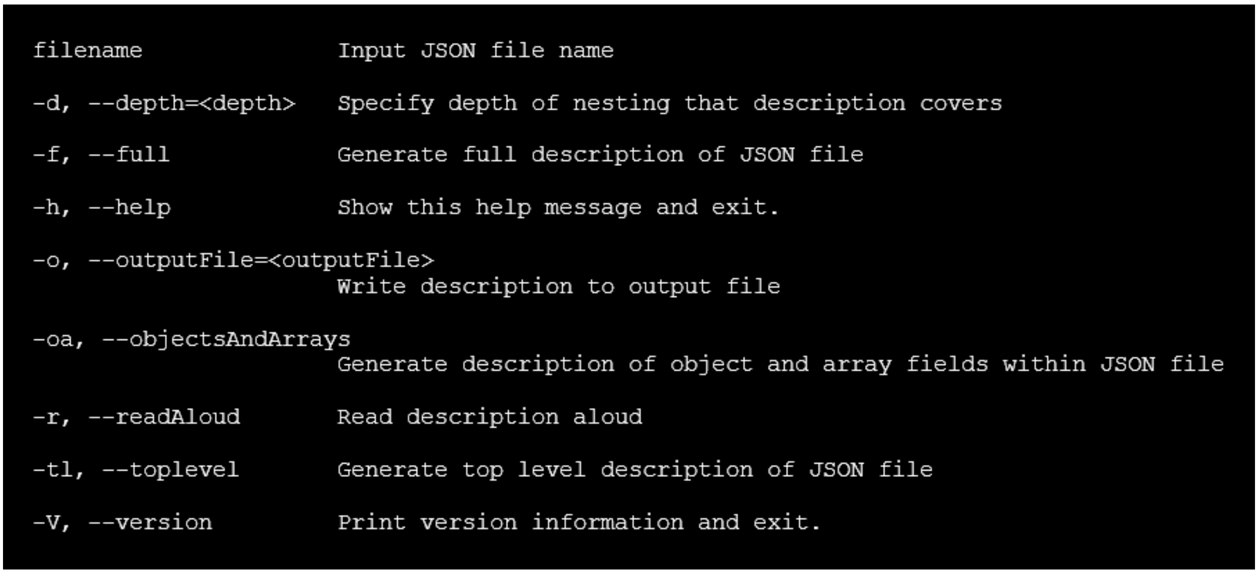
```
{
    "name": "John Doe",
    "age": 32,
    "isStudent": false,
    "courses": [
        "math",
        "history",
        "english"
    ],
    "address": {
        "street": "123 Main St",
        "city": "Anytown",
        "state": "XX"
    }
}
```



```
{
    “Title”: “The Cuckoo’s Calling”
    “Author”: “Robert Galbraith”,
    “Genre”: “classic crime novel”,
    “Detail”: {
        “Publisher”: “Little Brown”
        “Publication_Year”: 2013,
        “ISBN-13”: 9781408704004,
        “Language”: “English”,
        “Pages”: 494
    }
    “Price”: [
        {
            “type”: “Hardcover”,
            “price”: 16.65,
        }
        {
            “type”: “Kindle Edition”,
            “price”: 7.03,
        }
    ]
}
```

| Tool Briefing | JSONTalk is a command line tool that takes an input JSON file and generate various different natural language descriptions of the file.<br>Usage: jsonCLI [-fhlrV] [-oa] [-tl] [-d=<depth>] [-o=<outputFile>] filename |
|---|---|

Parameters and options



```
filename                Input JSON file name

-d, --depth=<depth>     Specify depth of nesting that description covers

-f, --full              Generate full description of JSON file

-h, --help              Show this help message and exit.

-o, --outputFile=<outputFile>
                        Write description to output file

-oa, --objectsAndArrays
                        Generate description of object and array fields within JSON file

-r, --readAloud         Read description aloud

-tl, --toplevel         Generate top level description of JSON file

-V, --version           Print version information and exit.
```

Tool demo

I will now show you how to use the tool and you have the opportunity to ask any questions.

JSON Transcript 1

Transcript:

"Object, name: 'store', brace open. Object, name: 'store', brace open. Property: 'book', value: Array, bracket open. Object, brace open. Property: 'category', value: 'reference', comma. Property: 'author', value: 'Nigel Rees', comma. Property: 'title', value: 'Sayings of the Century', comma. Property: 'price', value: 8.95, brace close, comma. Object, brace open. Property: 'category', value: 'fiction', comma. Property: 'author', value: 'Evelyn Waugh', comma. Property: 'title', value: 'Sword of Honour', comma. Property: 'price', value: 12.99, brace close, comma. Object, brace open. Property: 'category', value: 'fiction', comma. Property: 'author', value: 'Herman Melville', comma. Property: 'title', value: 'Moby Dick', comma. Property: 'isbn', value: '0-553-21311-3', comma. Property: 'price', value: 8.99, brace close, comma. Object, brace open. Property: 'category', value: 'fiction', comma. Property: 'author', value: 'J. R. R. Tolkien', comma. Property: 'title', value: 'The Lord of the Rings', comma. Property: 'isbn', value: '0-395-19395-8', comma. Property: 'price', value: 22.99, brace close, bracket close, comma. Property: 'bicycle', value: Object, brace open. Property: 'color', value: 'red', comma. Property: 'price', value: 19.95, brace close, brace close, comma. Property: 'expensive', value: 10, brace close."

2. With the 'store' object being at depth 1 and the depth increasing by 1 with each level of nesting, what depth is the 'Bicycle' object at? *

________________________________

3. Did you use the JSONTalk tool to help you answer the above question? *

*Mark only one oval.*

○ Yes

○ No

4. Are there any objects with the same structure within the described JSON file? *

(same structure meaning they have identical keys)

*Mark only one oval.*

○ Yes

○ No

○ Don't know

5. Did you use the JSONTalk tool to help you answer the above question? *

*Mark only one oval.*

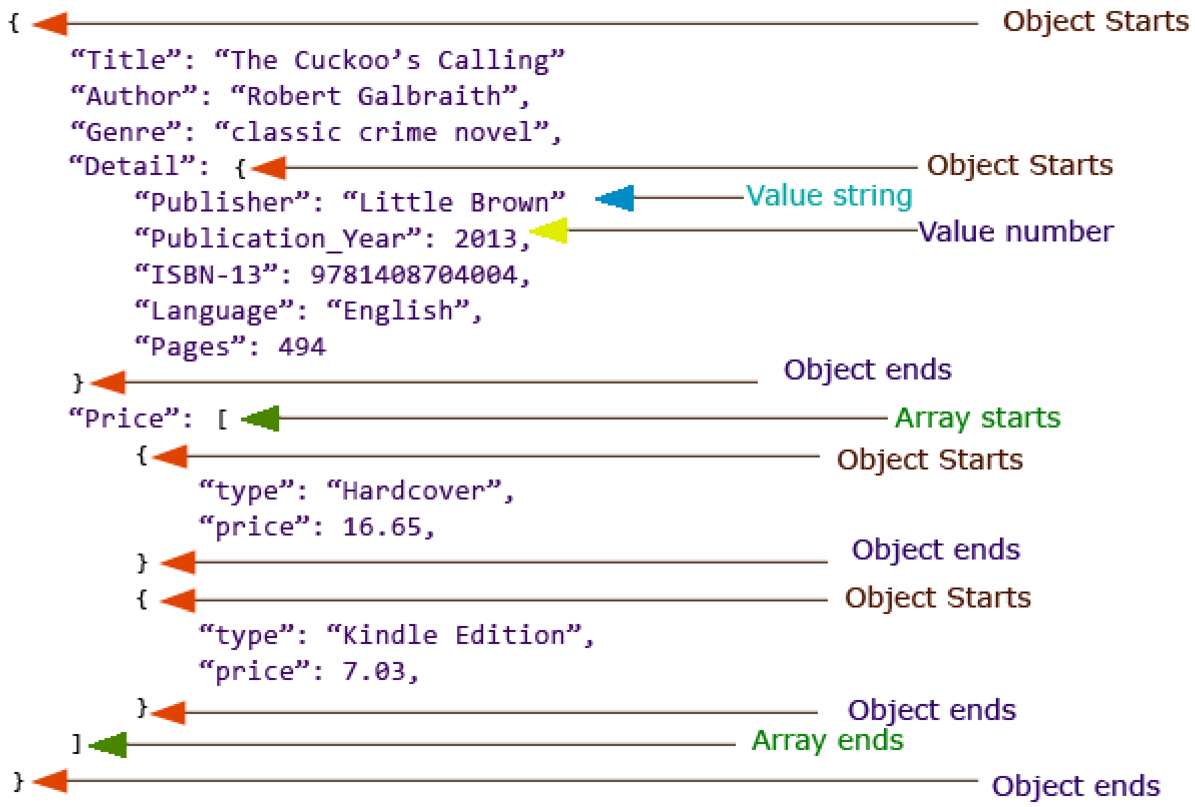
○ Yes

○ No

6. Re-write the JSON file transcript into the standard JSON syntax. *

Transcript (to avoid having to scroll to the top):
"Object, name: 'store', brace open. Object, name: 'store', brace open. Property: 'book', value: Array, bracket open. Object, brace open. Property: 'category', value: 'reference', comma. Property: 'author', value: 'Nigel Rees', comma. Property: 'title', value: 'Sayings of the Century', comma. Property: 'price', value: 8.95, brace close, comma. Object, brace open. Property: 'category', value: 'fiction', comma. Property: 'author', value: 'Evelyn Waugh', comma. Property: 'title', value: 'Sword of Honour', comma. Property: 'price', value: 12.99, brace close, comma. Object, brace open. Property: 'category', value: 'fiction', comma. Property: 'author', value: 'Herman Melville', comma. Property: 'title', value: 'Moby Dick', comma. Property: 'isbn', value: '0-553-21311-3', comma. Property: 'price', value: 8.99, brace close, comma. Object, brace open. Property: 'category', value: 'fiction', comma. Property: 'author', value: 'J. R. R. Tolkien', comma. Property: 'title', value: 'The Lord of the Rings', comma. Property: 'isbn', value: '0-395-19395-8', comma. Property: 'price', value: 22.99, brace close, bracket close, comma. Property: 'bicycle', value: Object, brace open. Property: 'color', value: 'red', comma. Property: 'price', value: 19.95, brace close, brace close, comma. Property: 'expensive', value: 10, brace close."

Re-call the proper syntax below:



```
{
    “Title”: “The Cuckoo’s Calling”
    “Author”: “Robert Galbraith”,
    “Genre”: “classic crime novel”,
    “Detail”: {
        “Publisher”: “Little Brown”
        “Publication_Year”: 2013,
        “ISBN-13”: 9781408704004,
        “Language”: “English”,
        “Pages”: 494
    }
    “Price”: [
        {
            “type”: “Hardcover”,
            “price”: 16.65,
        }
        {
            “type”: “Kindle Edition”,
            “price”: 7.03,
        }
    ]
}
```

| | |
|---|---|
| Transcript 2 | Transcript:<br>"Object, name: 'name', brace open. Property: 'name', value: 'Jane Doe', comma. Property: 'email', value: 'jane.doe@example.com', comma. Property: 'address', value: Object, brace open. Property: 'street', value: '123 Main St', comma. Property: 'city', value: 'Anytown', comma. Property: 'state', value: 'CA', comma. Property: 'zip', value: '12345', brace close, comma. Property: 'phoneNumbers', value: Array, bracket open. Object, brace open. Property: 'type', value: 'home', comma. Property: 'number', value: '555-555-1234', brace close, comma. Object, brace open. Property: 'type', value: 'work', comma. Property: 'number', value: '555-555-5678', brace close, bracket close, comma. Property: 'age', value: 35, comma. Property: 'isMarried', value: true, comma. Property: 'hobbies', value: Array, bracket open. Value: 'reading', comma. Value: 'traveling', comma. Value: 'cooking', bracket close, brace close." |

7. With the 'name' object being at depth 1 and the depth increasing by 1 with each level of nesting, what depth is the 'zip' property at? *

_______________________________

8. Did you use the JSONTalk tool to help you answer the above question? *

*Mark only one oval.*

○ Yes

○ No

9. Are there any objects with the same structure within the described JSON file? *

(same structure meaning they have identical keys)

*Mark only one oval.*

○ Yes

○ No

○ Don't know

10. Did you use the JSONTalk tool to help you answer the above question? *

*Mark only one oval.*

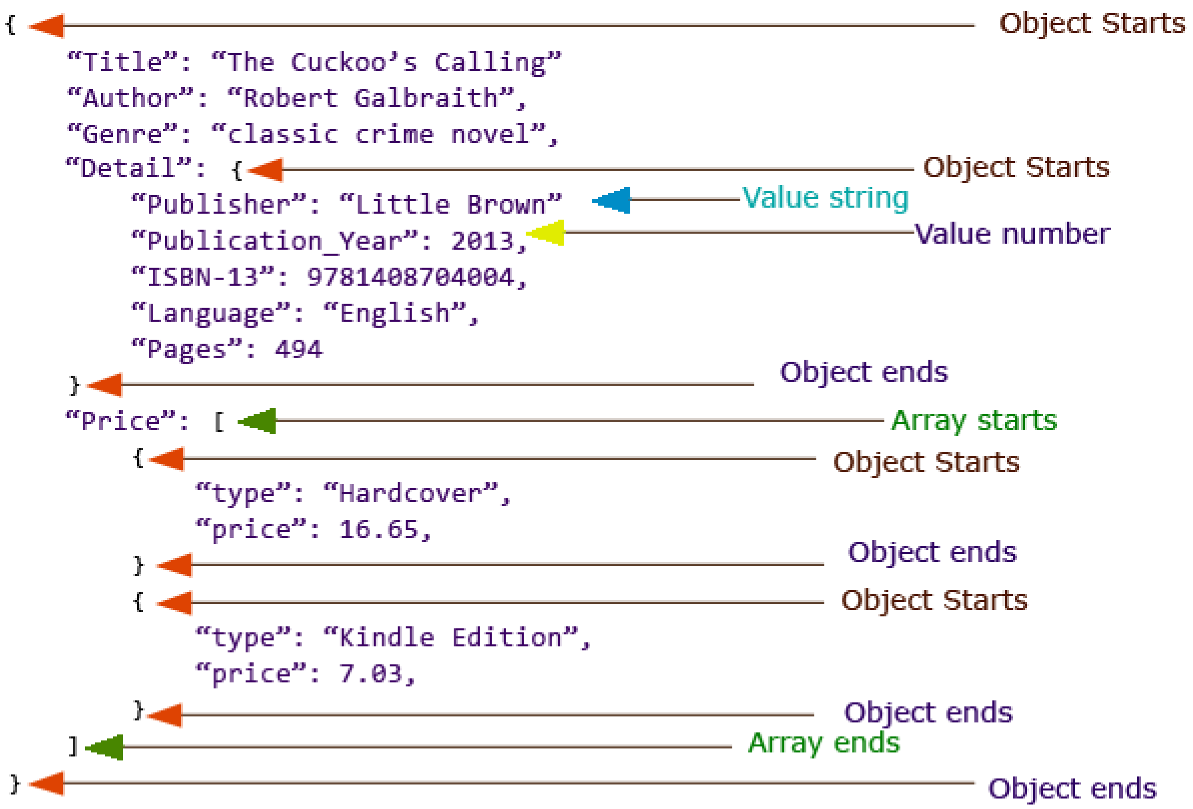
○ Yes

○ No

11. Re-write the JSON file transcript into the standard JSON syntax. *

Transcript (to avoid having to scroll to the top):
"Object, name: 'name', brace open. Property: 'name', value: 'Jane Doe', comma. Property: 'email', value: 'jane.doe@example.com', comma. Property: 'address', value: Object, brace open. Property: 'street', value: '123 Main St', comma. Property: 'city', value: 'Anytown', comma. Property: 'state', value: 'CA', comma. Property: 'zip', value: '12345', brace close, comma. Property: 'phoneNumbers', value: Array, bracket open. Object, brace open. Property: 'type', value: 'home', comma. Property: 'number', value: '555-555-1234', brace close, comma. Object, brace open. Property: 'type', value: 'work', comma. Property: 'number', value: '555-555-5678', brace close, bracket close, comma. Property: 'age', value: 35, comma. Property: 'isMarried', value: true, comma. Property: 'hobbies', value: Array, bracket open. Value: 'reading', comma. Value: 'traveling', comma. Value: 'cooking', bracket close, brace close."

Re-call the proper syntax below:



```
{
    “Title”: “The Cuckoo’s Calling”
    “Author”: “Robert Galbraith”,
    “Genre”: “classic crime novel”,
    “Detail”: {
        “Publisher”: “Little Brown”
        “Publication_Year”: 2013,
        “ISBN-13”: 9781408704004,
        “Language”: “English”,
        “Pages”: 494
    }
    “Price”: [
        {
            “type”: “Hardcover”,
            “price”: 16.65,
        }
        {
            “type”: “Kindle Edition”,
            “price”: 7.03,
        }
    ]
}
```

12. Did you use the JSONTalk tool to help you answer the above question? *

*Mark only one oval.*

○ Yes

○ No

| | |
|---|---|
| Transcript 3 | Transcript:<br>"Object, property: 'person', value: Object, brace open. Property: 'name', value: Object, brace open. Property: 'first', value: 'John', comma. Property: 'last', value: 'Doe', brace close, comma. Property: 'age', value: 35, comma. Property: 'email', value: 'john.doe@example.com', comma. Property: 'address', value: Object, brace open. Property: 'street', value: '456 Elm St', comma. Property: 'city', value: 'Smallville', comma. Property: 'state', value: 'KS', comma. Property: 'zip', value: '67890', comma. Property: 'location', value: Object, brace open. Property: 'latitude', value: 37.7749, comma. Property: 'longitude', value: -122.4194, comma. Property: 'weather', value: Object, brace open. Property: 'current', value: Object, brace open. Property: 'temperature', value: 72, comma. Property: 'humidity', value: 0.65, brace close, comma. Property: 'forecast', value: Object, brace open. Property: 'high', value: 75, comma. Property: 'low', value: 68, comma. Property: 'chanceOfRain', value: 0.10, brace close, brace close, brace close, brace close, comma. Property: 'phoneNumbers', value: Array, bracket open. Object, brace open. Property: 'type', value: 'home', comma. Property: 'number', value: '555-555-1234', comma. Property: 'extension', value: 123, brace close, comma. Object, brace open. Property: 'type', value: 'work', comma. Property: 'number', value: '555-555-5678', comma. Property: 'extension', value: 456, brace close, bracket close, comma. Property: 'hobbies', value: Array, bracket open. 'reading', comma. Object, brace open. Property: 'outdoor', value: Array, bracket open. 'hiking', comma. 'camping', comma. Object, brace open. Property: 'equipment', value: Array, bracket open. 'tent', comma. 'sleeping bag', comma. 'backpack', bracket close, brace close, bracket close, comma. 'cooking', bracket close, comma. Property: 'isMarried', value: true, brace close." |

13. With the 'person' object being at depth 1 and the depth increasing by 1 with each level of nesting, what depth is the 'chanceOfRain' object at? *

14. Did you use the JSONTalk tool to help you answer the above question? *

*Mark only one oval.*

○ Yes

○ No

15. Are there any objects with the same structure within the described JSON file? *

(same structure meaning they have identical keys)

*Mark only one oval.*

○ Yes

○ No

○ Don't know

16. Did you use the JSONTalk tool to help you answer the above question? *

*Mark only one oval.*

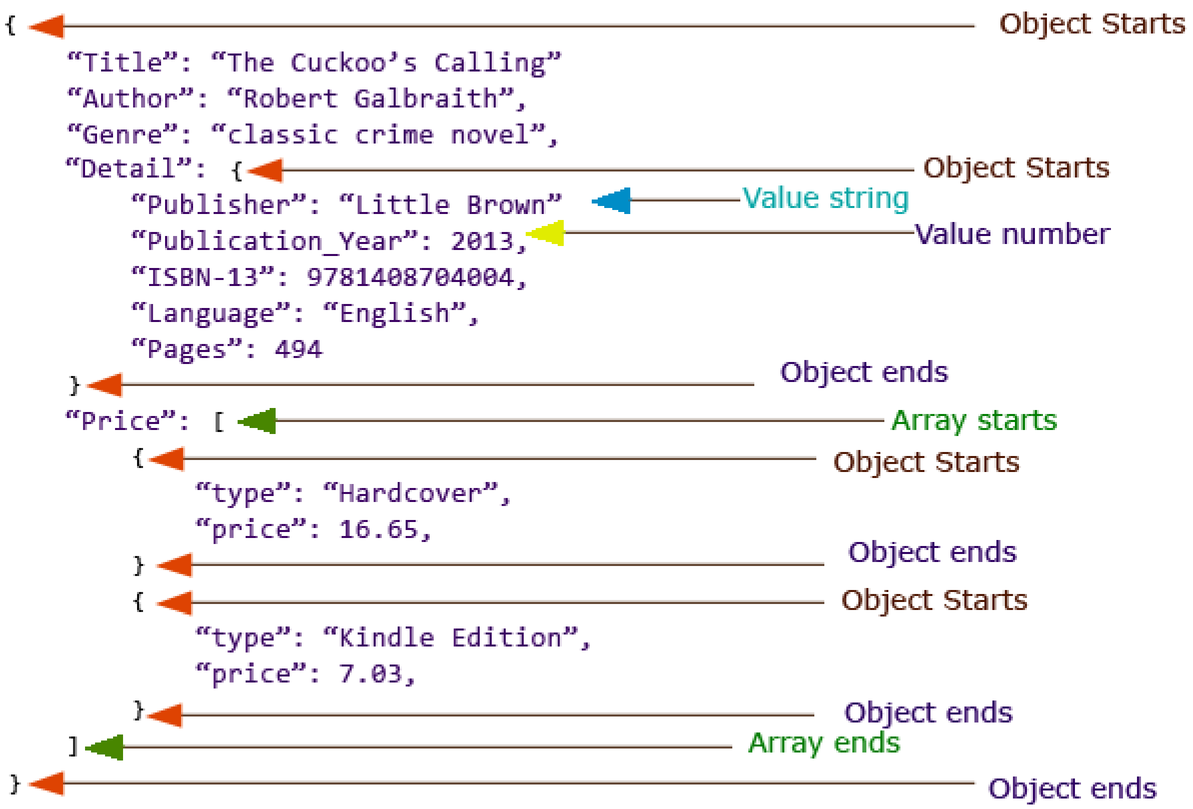
○ Yes

○ No

17. Re-write the JSON file transcript into the standard JSON syntax. *

Transcript (to avoid having to scroll to the top):
"Object, property: 'person', value: Object, brace open. Property: 'name', value: Object, brace open. Property: 'first', value: 'John', comma. Property: 'last', value: 'Doe', brace close, comma. Property: 'age', value: 35, comma. Property: 'email', value: 'john.doe@example.com', comma. Property: 'address', value: Object, brace open. Property: 'street', value: '456 Elm St', comma. Property: 'city', value: 'Smallville', comma. Property: 'state', value: 'KS', comma. Property: 'zip', value: '67890', comma. Property: 'location', value: Object, brace open. Property: 'latitude', value: 37.7749, comma. Property: 'longitude', value: -122.4194, comma. Property: 'weather', value: Object, brace open. Property: 'current', value: Object, brace open. Property: 'temperature', value: 72, comma. Property: 'humidity', value: 0.65, brace close, comma. Property: 'forecast', value: Object, brace open. Property: 'high', value: 75, comma. Property: 'low', value: 68, comma. Property: 'chanceOfRain', value: 0.10, brace close, brace close, brace close, brace close, comma. Property: 'phoneNumbers', value: Array, bracket open. Object, brace open. Property: 'type', value: 'home', comma. Property: 'number', value: '555-555-1234', comma. Property: 'extension', value: 123, brace close, comma. Object, brace open. Property: 'type', value: 'work', comma. Property: 'number', value: '555-555-5678', comma. Property: 'extension', value: 456, brace close, bracket close, comma. Property: 'hobbies', value: Array, bracket open. 'reading', comma. Object, brace open. Property: 'outdoor', value: Array, bracket open. 'hiking', comma. 'camping', comma. Object, brace open. Property: 'equipment', value: Array, bracket open. 'tent', comma. 'sleeping bag', comma. 'backpack', bracket close, brace close, bracket close, comma. 'cooking', bracket close, comma. Property: 'isMarried', value: true, brace close."

Re-call the proper syntax below:



```
{
    “Title”: “The Cuckoo’s Calling”
    “Author”: “Robert Galbraith”,
    “Genre”: “classic crime novel”,
    “Detail”: {
        “Publisher”: “Little Brown”
        “Publication_Year”: 2013,
        “ISBN-13”: 9781408704004,
        “Language”: “English”,
        “Pages”: 494
    }
    “Price”: [
        {
            “type”: “Hardcover”,
            “price”: 16.65,
        }
        {
            “type”: “Kindle Edition”,
            “price”: 7.03,
        }
    ]
}
```

18. Did you use the JSONTalk tool to help you answer the above question? *

*Mark only one oval.*

○ Yes

○ No

## Tool feedback

The final stage of this evaluation involves you answering a short series of questions about the JSONTalk tool you have just used.

19. I think that I would like to use this system frequently *

*Mark only one oval.*

Strongly disagree

1 ○

2 ○

3 ○

4 ○

5 ○

Strongly agree

20. I found the system unnecessarily complex *

*Mark only one oval.*

Strongly disagree

1 ○

2 ○

3 ○

4 ○

5 ○

Strongly agree

21. I thought the system was easy to use *

*Mark only one oval.*

Strongly disagree

1 ○

2 ○

3 ○

4 ○

5 ○

Strongly agree

22. I think that I would need the support of a technical person to be able to use this system *

*Mark only one oval.*

| | |
|---|---|
| | Strongly disagree |
| 1 | ○ |
| 2 | ○ |
| 3 | ○ |
| 4 | ○ |
| 5 | ○ |
| | Strongly agree |

23. I found the various functions in this system were well integrated *

*Mark only one oval.*

| | |
|---|---|
| | Strongly disagree |
| 1 | ○ |
| 2 | ○ |
| 3 | ○ |
| 4 | ○ |
| 5 | ○ |
| | Strongly agree |

24. I thought there was too much inconsistency in this system *

*Mark only one oval.*

| | |
|---|---|
| | Strongly disagree |
| 1 | ○ |
| 2 | ○ |
| 3 | ○ |
| 4 | ○ |
| 5 | ○ |
| | Strongly agree |

25. I would imagine that most people would learn to use this system very quickly *

*Mark only one oval.*

| | |
|---|---|
| | Strongly disagree |
| 1 | ○ |
| 2 | ○ |
| 3 | ○ |
| 4 | ○ |
| 5 | ○ |
| | Strongly agree |

26. I found the system very cumbersome to use *

*Mark only one oval.*

| | |
|---|---|
| | Strong disagree |
| 1 | ○ |
| 2 | ○ |
| 3 | ○ |
| 4 | ○ |
| 5 | ○ |
| | Strongly agree |

27. I felt very confident using the system *

*Mark only one oval.*

Strongly disagree

1 ○

2 ○

3 ○

4 ○

5 ○

Strongly agree

28. I needed to learn a lot of things before I could get going with this system *

*Mark only one oval.*

Strongly disagree

1 ○

2 ○

3 ○

4 ○

5 ○

Strongly agree

29. Space for comments on the JSONTalk tool:

## End of study

Thank you for taking part in this user study. The main aim of this study was to investigate whether the JSONTalk tool accurately represents the content of a JSON file, and whether the tool allows for quicker understanding of JSON file representation from a screen reader.

30. Space for questions or comments about the experiment: *

Please take a not of my email address (

2463548h@student.gla.ac.uk ) and don't hesitate to contact me if you have any further questions. Thank you for your help.