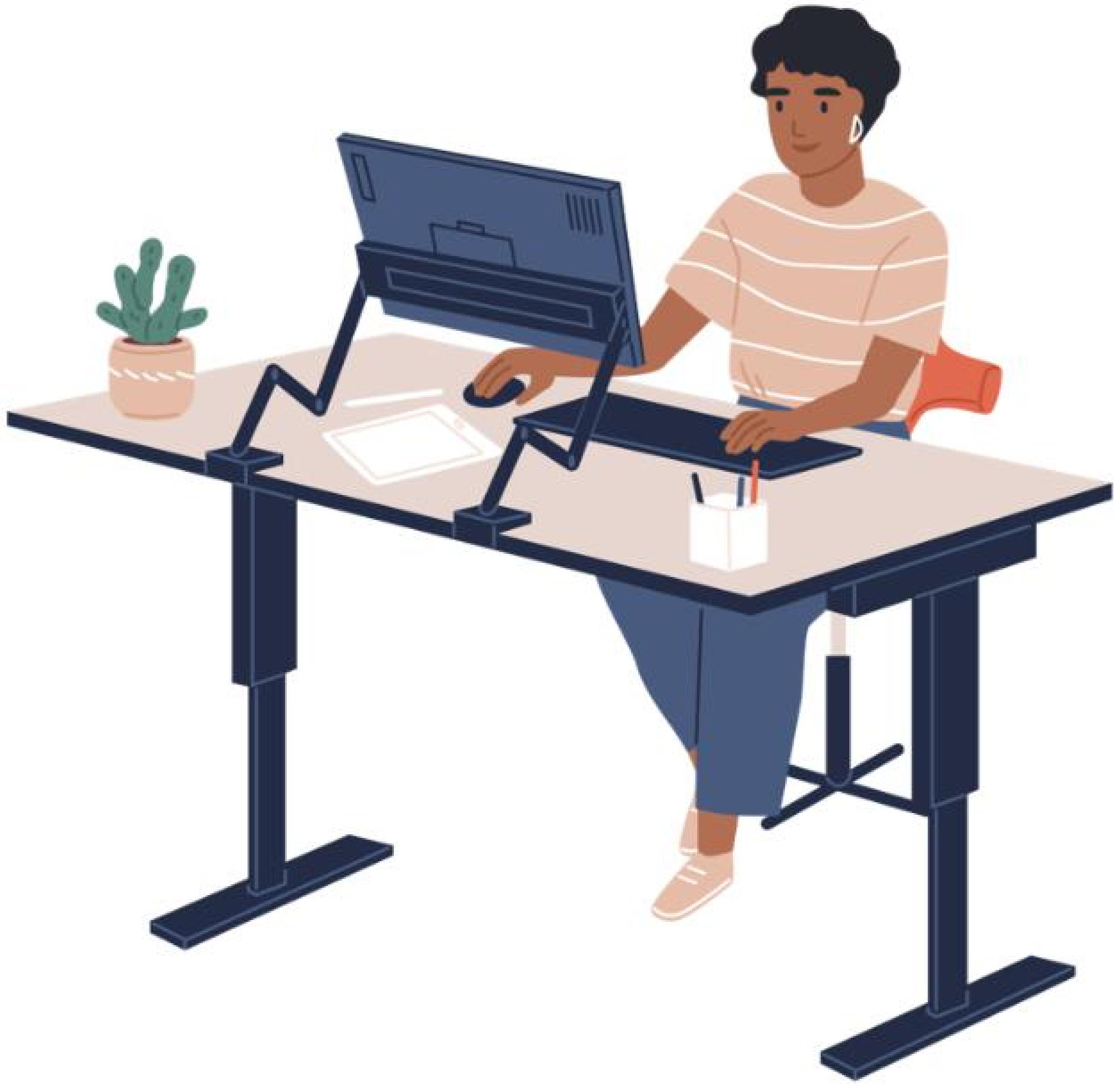
Learning Consolidation

Model Real World Data Using Objects



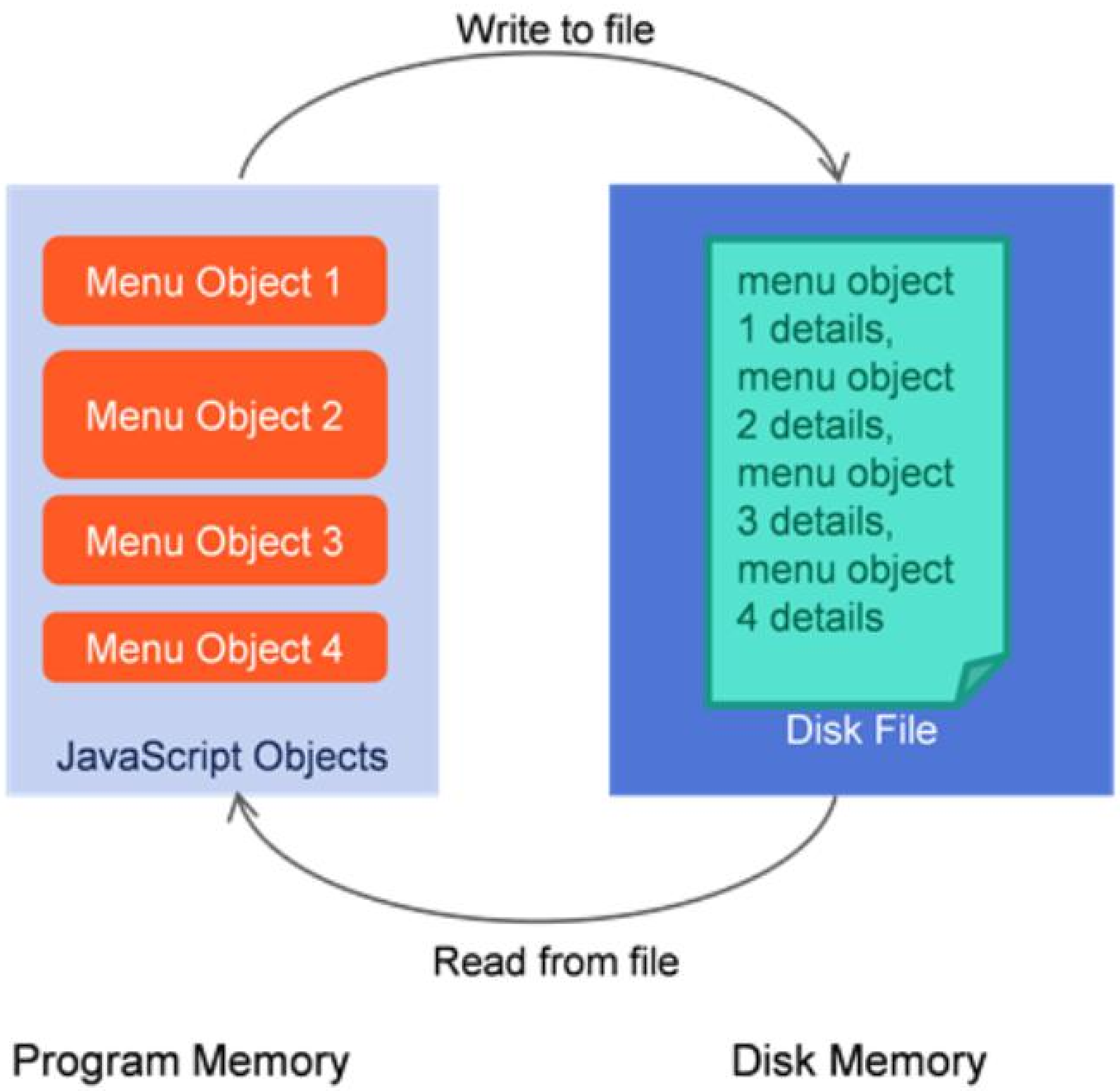
In this Sprint, you learned to:

- Explain the need to convert JavaScript object to JSON
- Model data using JSON – JavaScript Object Notation
- Explain different JSON constructs
- Perform conversions between JSON and JavaScript object
- Compare JSON and JavaScript object

Restaurant Menu Scenario

- For online customers, the restaurant's menu should always be updated.
- The menu enlists the food items offered by the restaurant.
- The menu is modified when new food items are added, and old ones are removed.
- In programs, data is stored in temporary memory, which gets destroyed when the program stops running.
- How can we ensure that menu items are always available?
- This is possible if the data is persisted or saved on a permanent basis.

Implement Persistence



- Persistence helps store and retrieve data from disk.
- Data in disk should be transportable, flexible and preferably in text format to permit storage of large values.
- Thus, we need to convert data into a lightweight textual format, before saving it to the disk.

Slide Note

Data Persistence requires data format conversion

What Is the Web?

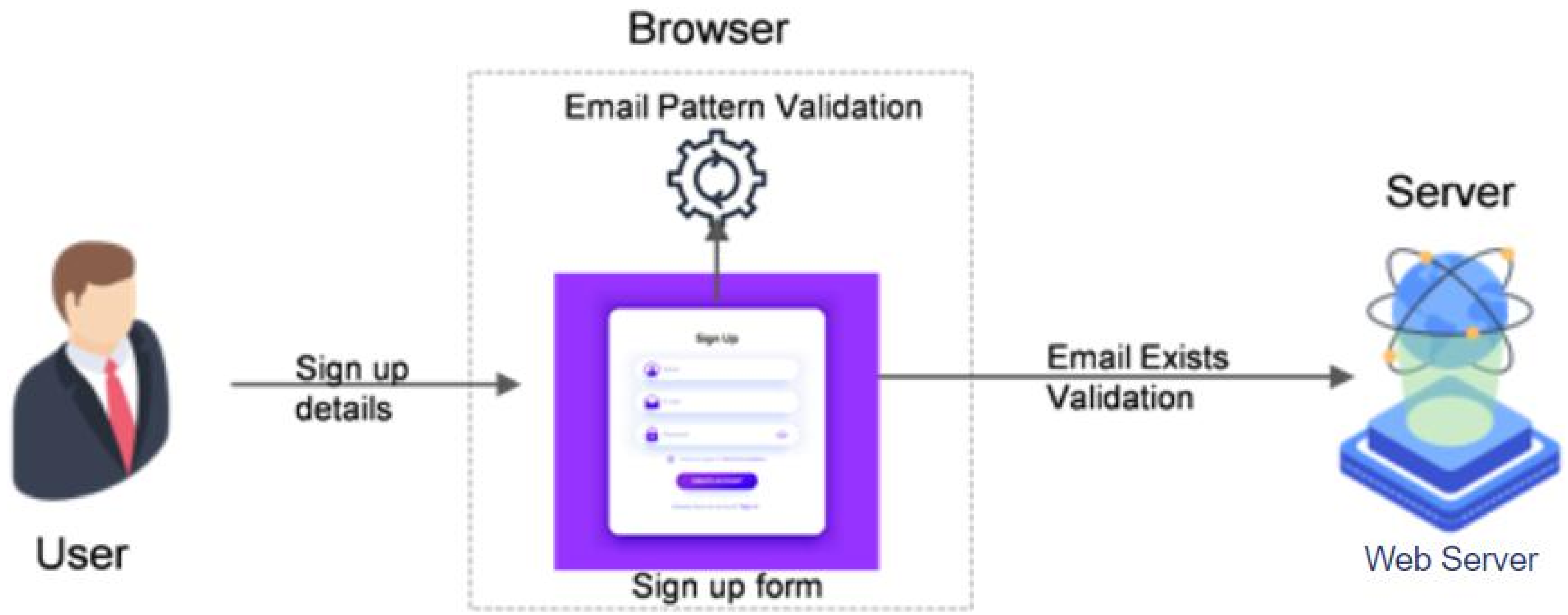
- The web is a platform of interconnected web pages.
- At the client (user's) end, some browsers provide access to the web pages.
- The data for the web pages are provided to the clients by a different machine called a web server.
- A web server is a machine that accepts requests, processes them and returns responses.
- Clients and servers are two different systems set up using different programming languages.
- Each language has its format for representing data.

Think and Tell

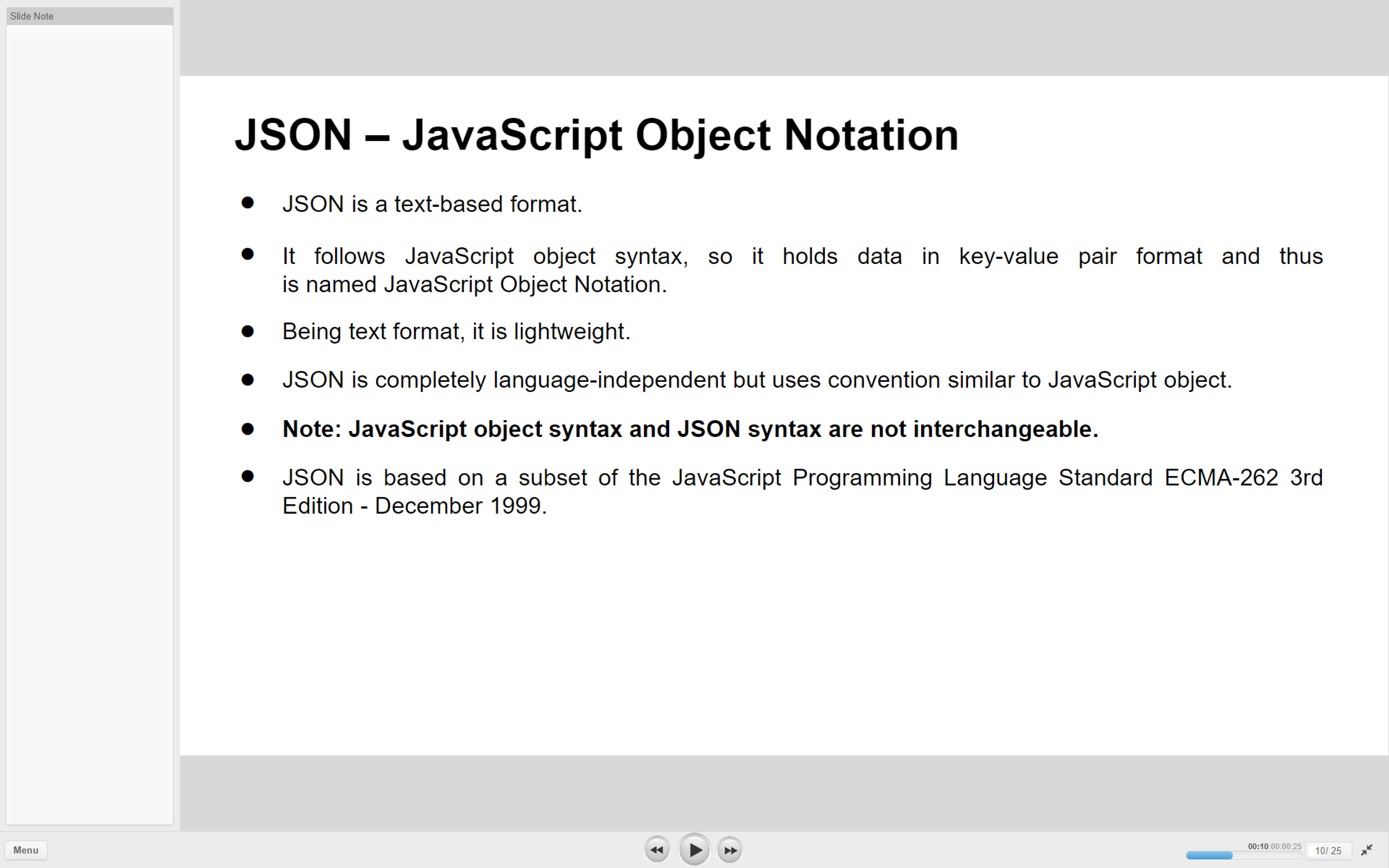
- Since there are different programming languages at the client and server ends, how does data travel from one end to the other?
- Is any conversion required to change data formats before transporting?

Data Interchange Over the Web

- On the web, multiple machines operate to process user requests.
- For example, the client machine captures sign-up details and validates email if it is in the correct format.
- The email is then sent to the server machine to ensure it is not already taken by another user.
- At both the ends, different programming languages are used to write code.
- So, the machines need to send and receive data in a format that both can understand.
- Data interchange between machines requires data format conversion.

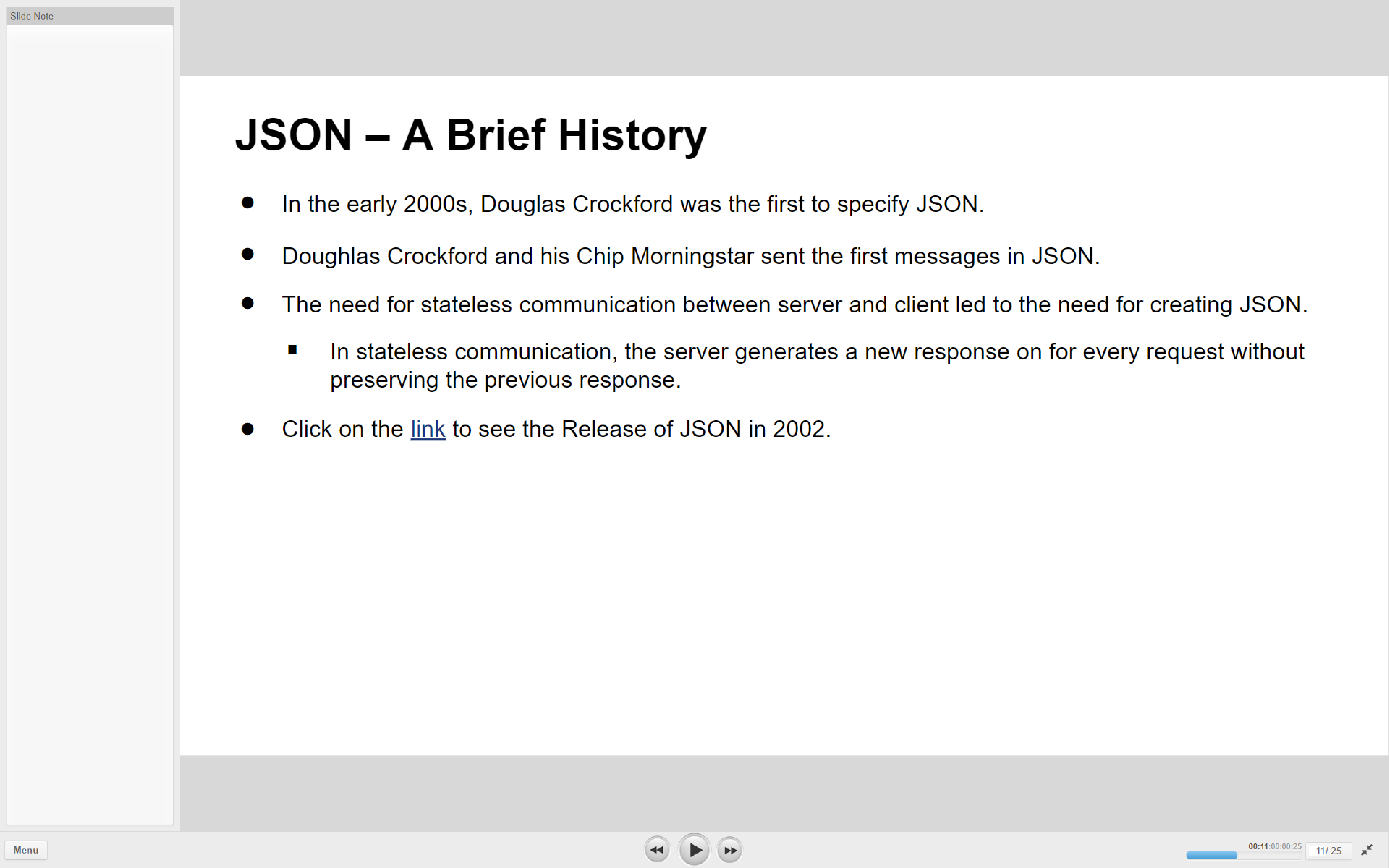


Data Interchange between machines requires data format conversion



JSON – JavaScript Object Notation

- JSON is a text-based format.
- It follows JavaScript object syntax, so it holds data in key-value pair format and thus is named JavaScript Object Notation.
- Being text format, it is lightweight.
- JSON is completely language-independent but uses convention similar to JavaScript object.
- **Note: JavaScript object syntax and JSON syntax are not interchangeable.**
- JSON is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999.

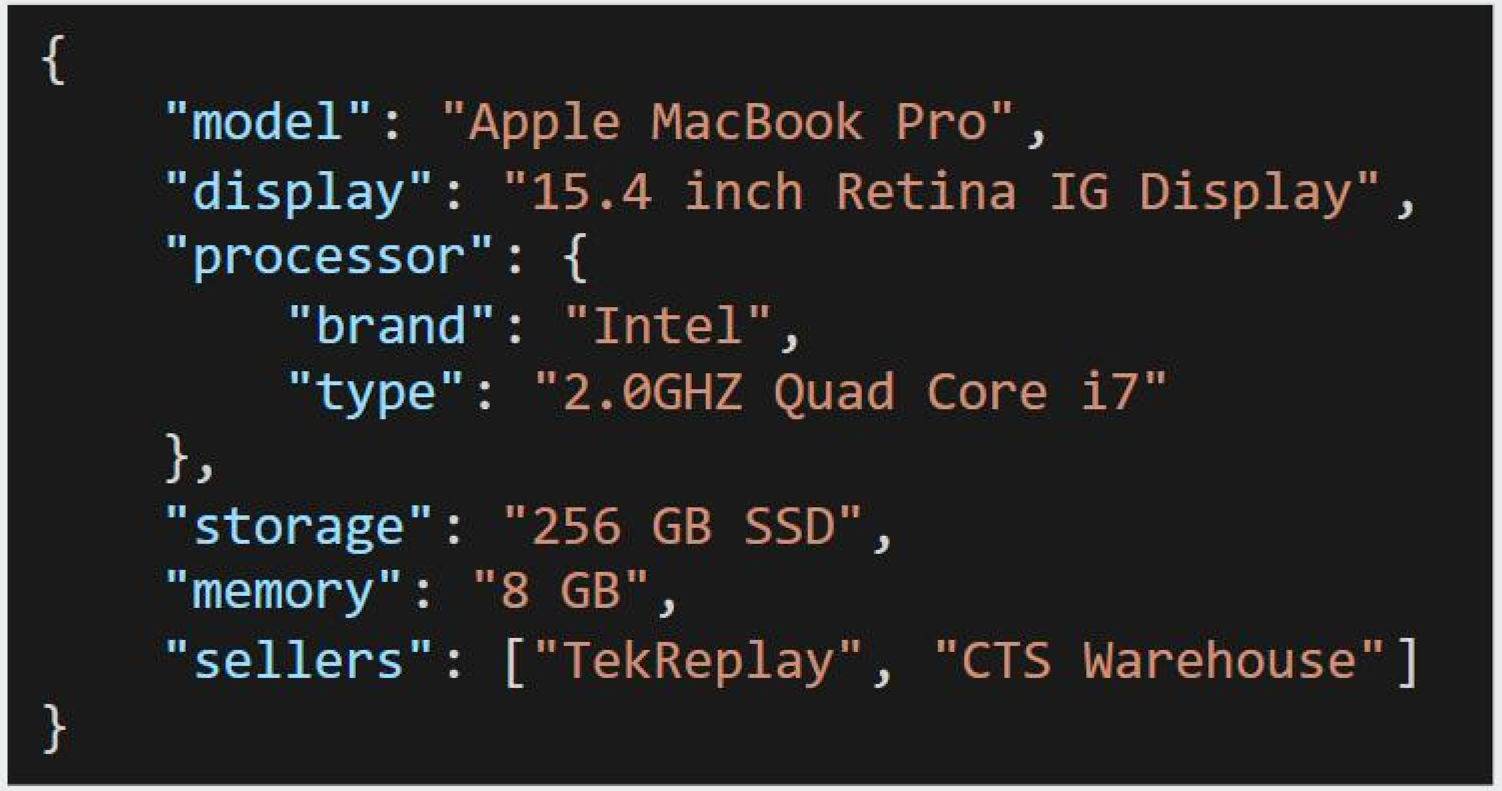


JSON – A Brief History

- In the early 2000s, Douglas Crockford was the first to specify JSON.
- Doughlas Crockford and his Chip Morningstar sent the first messages in JSON.
- The need for stateless communication between server and client led to the need for creating JSON.
  - In stateless communication, the server generates a new response on for every request without preserving the previous response.
- Click on the link to see the Release of JSON in 2002.

JSON Structure

- JSON is a string whose format resembles JavaScript object literal format.
- JSON can represent strings, numbers, Booleans, null, arrays and objects made up of these values.
- JSON data is in key-value pairs that are separated by commas.
- The key should be unique and contained in double quotes.
- The values should follow JavaScript literal syntax – strings in quotes, the number without quotes, Boolean values should be true or false, arrays should be in `[]` and the object in `{}`.
- A JSON string can be stored in a text file with an extension of `.json`.



```
{
    "model": "Apple MacBook Pro",
    "display": "15.4 inch Retina IG Display",
    "processor": {
        "brand": "Intel",
        "type": "2.0GHZ Quad Core i7"
    },
    "storage": "256 GB SSD",
    "memory": "8 GB",
    "sellers": ["TekReplay", "CTS Warehouse"]
}
```

Railroad Diagram – Simple JSON

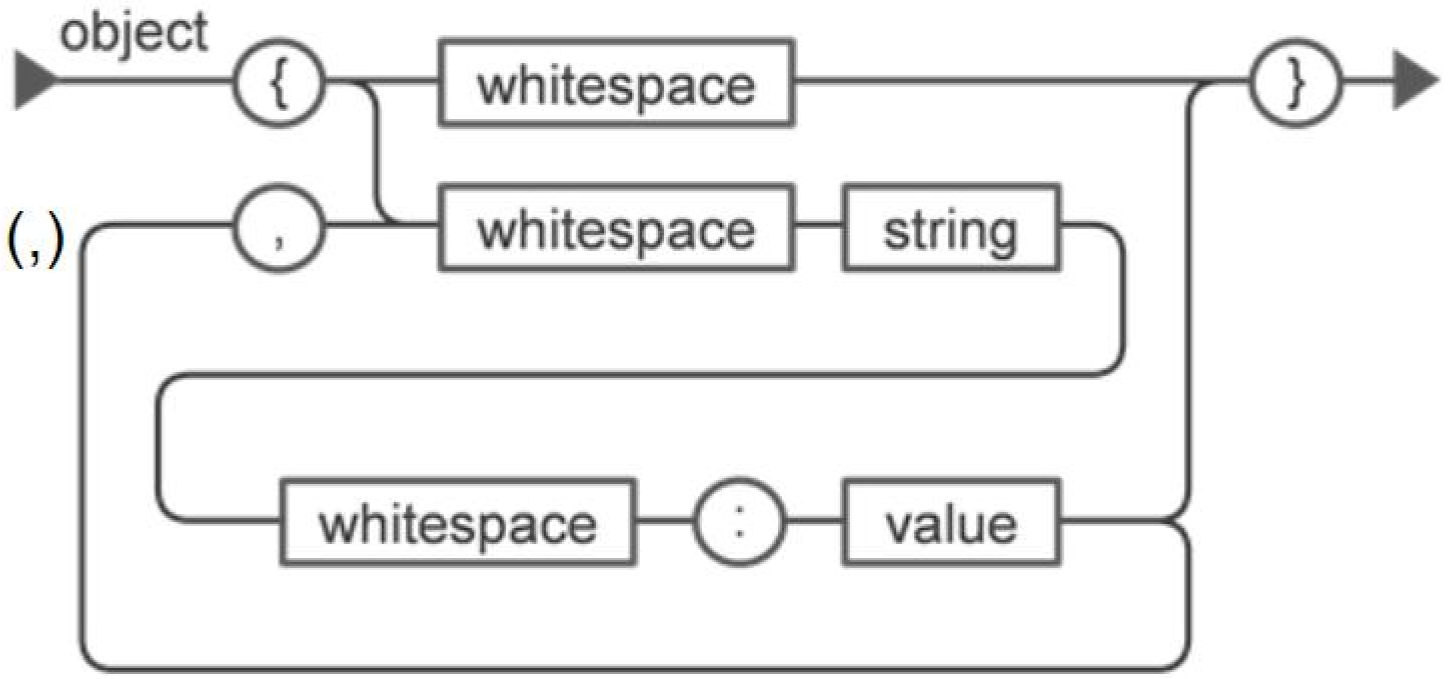
1. At least one whitespace enclosed in `{}`

2. Optionally add key-value pair

```
{
    "model": "Apple MacBook Pro"
}
```

3. Multiple key-value pairs must be separated by a comma (,)

```
{
    "model": "Apple MacBook Pro",
    "display": "15.4 inch Retina IG Display",
    "storage": "256 GB SSD",
    "memory": "8 GB",
}
```

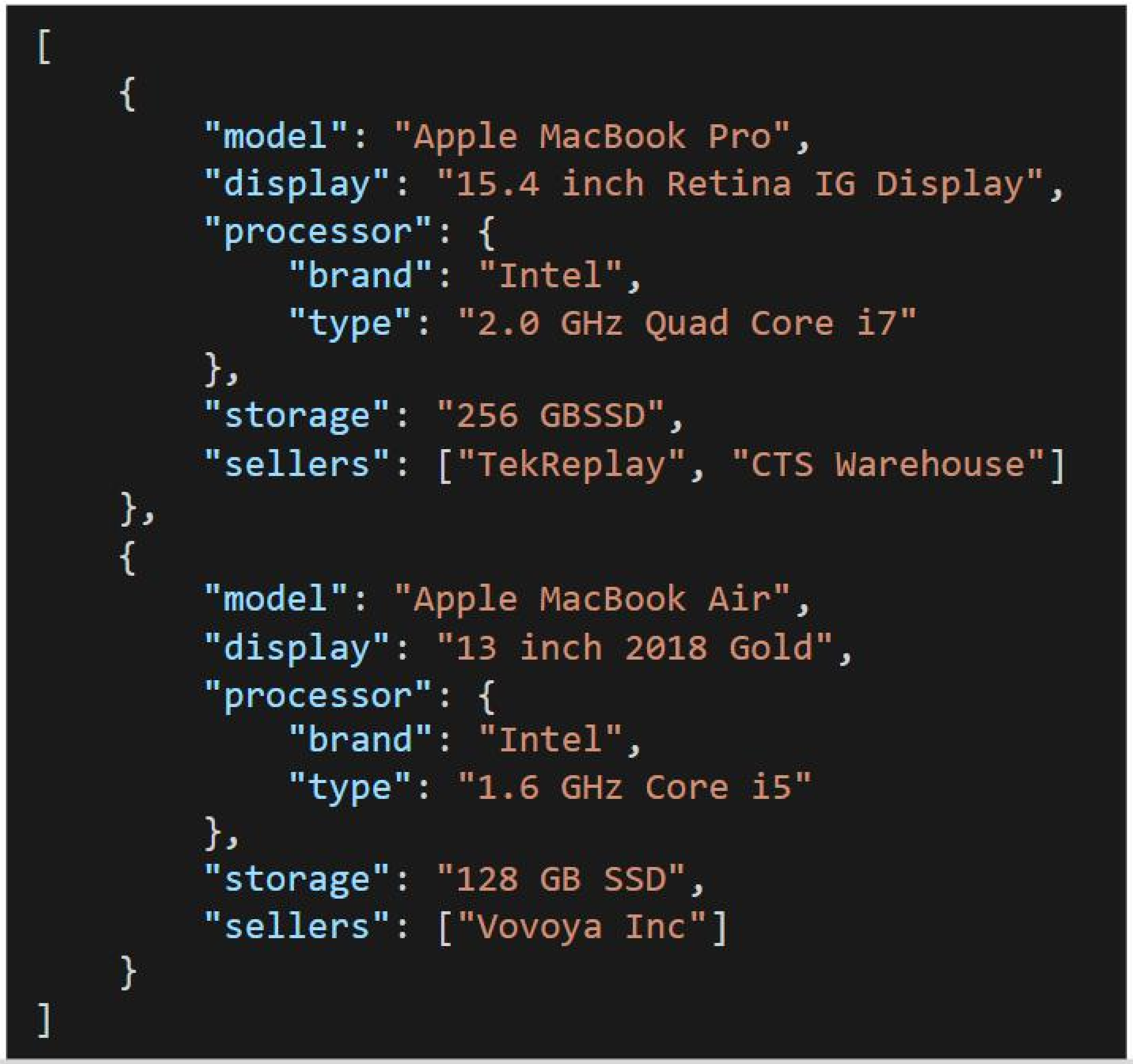


Note: The last key-value pair should not have a trailing comma.

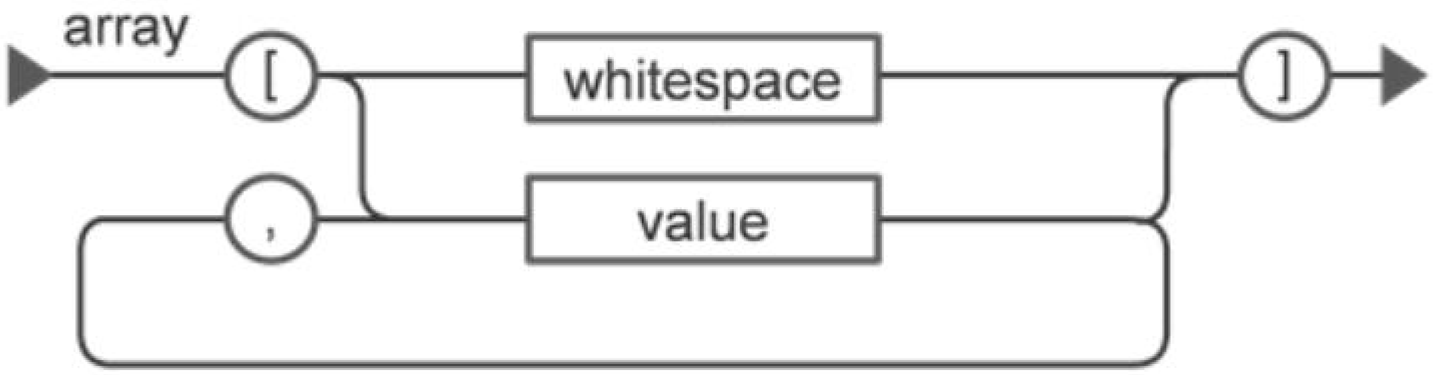
Railroad Diagram – JSON Array

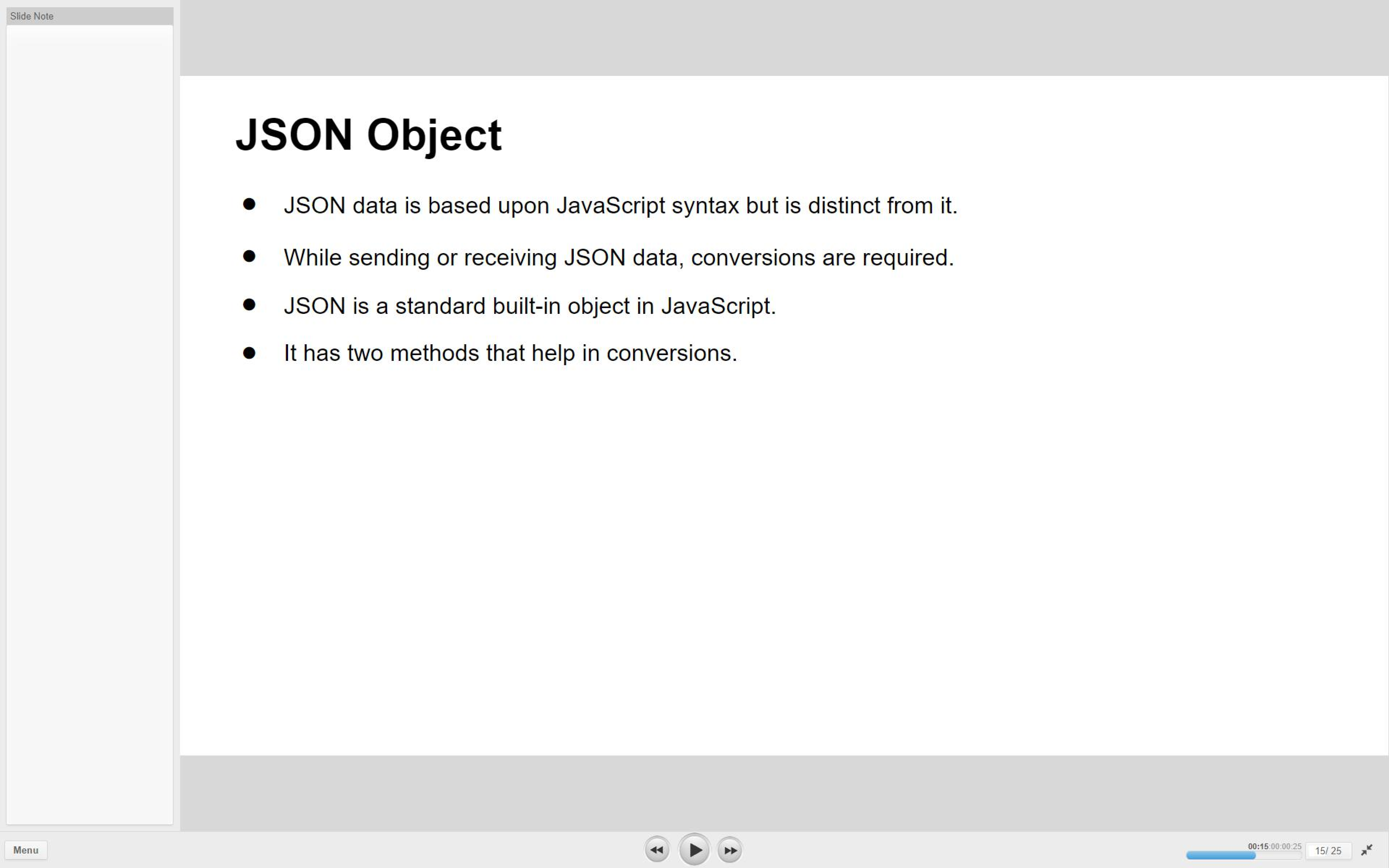
1. At least one white space enclosed in [ ]

2. Optionally add one or more JSON values separated by commas (,)



```
[
    {
        "model": "Apple MacBook Pro",
        "display": "15.4 inch Retina IG Display",
        "processor": {
            "brand": "Intel",
            "type": "2.0 GHz Quad Core i7"
        },
        "storage": "256 GBSSD",
        "sellers": ["TekReplay", "CTS Warehouse"]
    },
    {
        "model": "Apple MacBook Air",
        "display": "13 inch 2018 Gold",
        "processor": {
            "brand": "Intel",
            "type": "1.6 GHz Core i5"
        },
        "storage": "128 GB SSD",
        "sellers": ["Vovoya Inc"]
    }
]
```



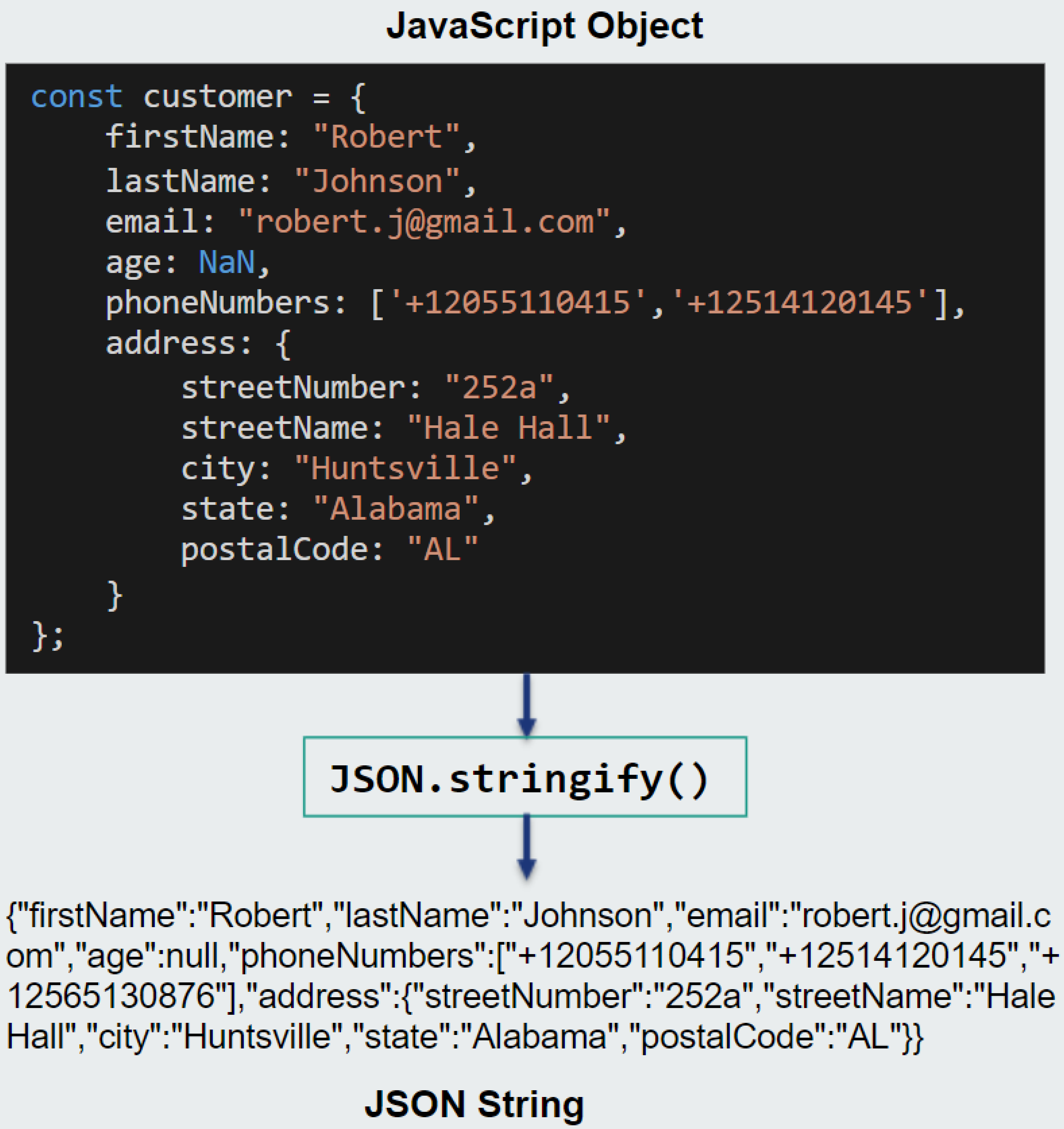


JSON Object

- JSON data is based upon JavaScript syntax but is distinct from it.
- While sending or receiving JSON data, conversions are required.
- JSON is a standard built-in object in JavaScript.
- It has two methods that help in conversions.

Stringify JavaScript Object

- The `JSON.stringify()` method converts a JavaScript object or value to a JSON string.
- If the operation is successful, the method returns a JSON string.
- If the operation fails, the method returns `undefined` value.
- Boolean, Number and String objects are converted to the corresponding primitive values during stringification.



JavaScript Object

```
const customer = {
    firstName: "Robert",
    lastName: "Johnson",
    email: "robert.j@gmail.com",
    age: NaN,
    phoneNumbers: ['+12055110415','+12514120145'],
    address: {
        streetNumber: "252a",
        streetName: "Hale Hall",
        city: "Huntsville",
        state: "Alabama",
        postalCode: "AL"
    }
};
```

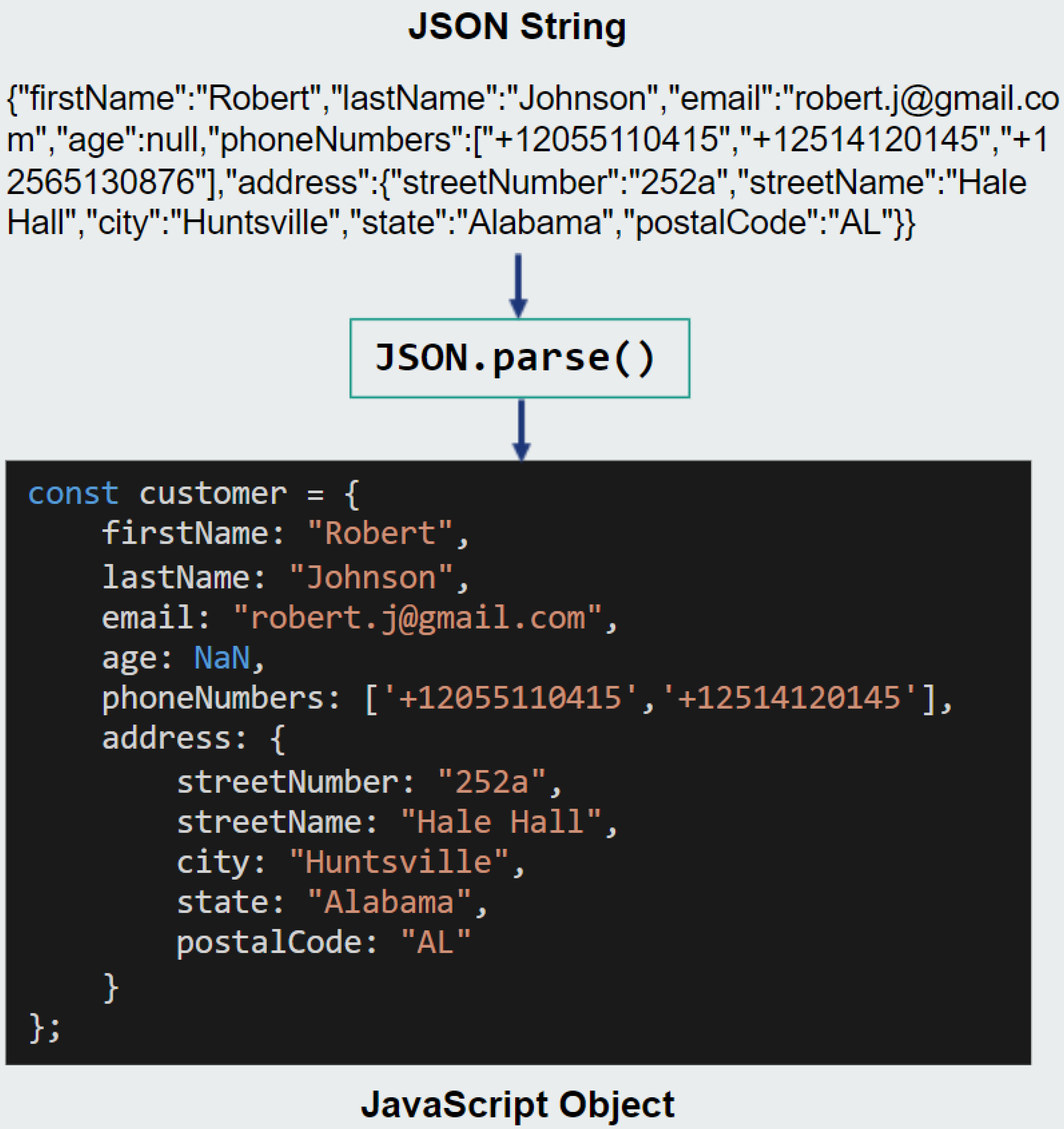
`JSON.stringify()`

```
{"firstName":"Robert","lastName":"Johnson","email":"robert.j@gmail.com","age":null,"phoneNumbers":["+12055110415","+12514120145","+12565130876"],"address":{"streetNumber":"252a","streetName":"Hale Hall","city":"Huntsville","state":"Alabama","postalCode":"AL"}}
```

JSON String

Parse JSON to JavaScript Object

- The `JSON.parse()` method parses a JSON string and constructs a JavaScript object as described by the string.
- Parsing is the process of analyzing strings of symbols in a computer language. It confirms the rules of languages are implemented.
- The value returned by the method could be an object, array, string, number, Boolean, or null value corresponding to the given JSON text.



JSON String

{"firstName":"Robert","lastName":"Johnson","email":"robert.j@gmail.com","age":null,"phoneNumbers":["+12055110415","+12514120145","+12565130876"],"address":{"streetNumber":"252a","streetName":"Hale Hall","city":"Huntsville","state":"Alabama","postalCode":"AL"}}

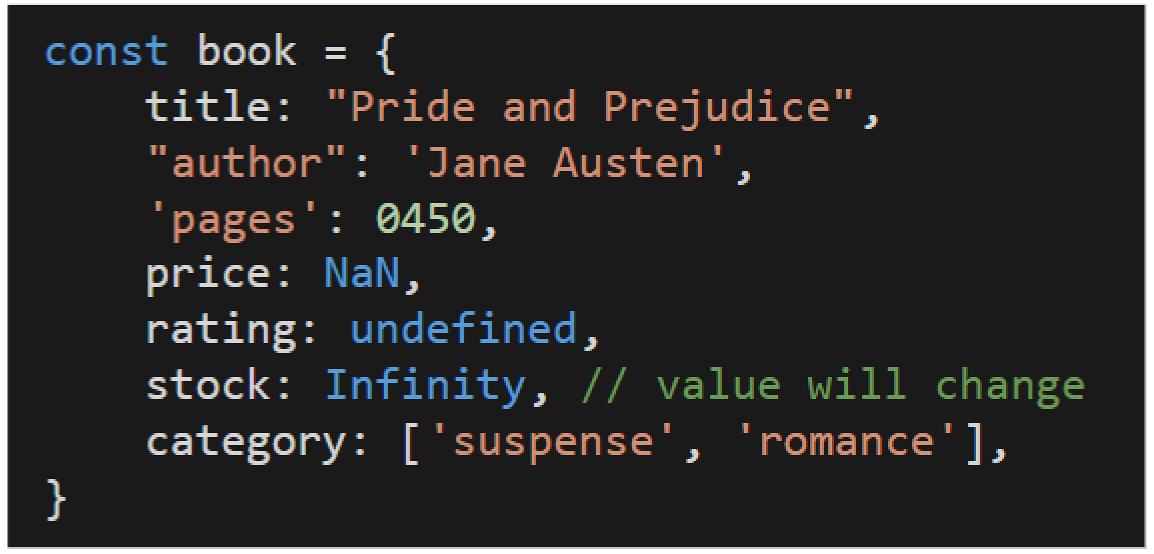
```
const customer = {
    firstName: "Robert",
    lastName: "Johnson",
    email: "robert.j@gmail.com",
    age: NaN,
    phoneNumbers: ['+12055110415','+12514120145'],
    address: {
        streetNumber: "252a",
        streetName: "Hale Hall",
        city: "Huntsville",
        state: "Alabama",
        postalCode: "AL"
    }
};
```

JavaScript Object

Comparing JavaScript Object With JSON

JavaScript

- The property names in objects need not be quoted.
- Trailing commas are permitted in JavaScript object literal.
- `Undefined`, `NaN` and `Infinity` are valid JavaScript values.
- Object literals allow comments.

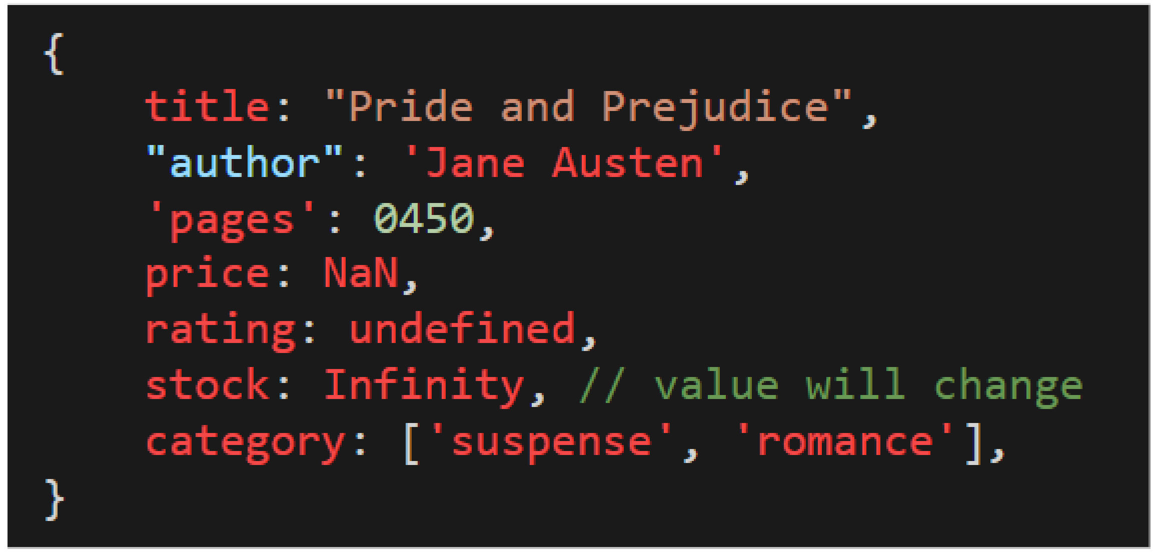


```
const book = {
    title: "Pride and Prejudice",
    "author": 'Jane Austen',
    'pages': 0450,
    price: NaN,
    rating: undefined,
    stock: Infinity, // value will change
    category: ['suspense', 'romance'],
}
```

Valid JavaScript Object

JSON

- The property names must be double-quoted strings.
- Trailing commas are forbidden.
- `Undefined`, `NaN` and `Infinity` are not valid JSON values.
- Comments are not supported by JSON data.



```
{
    title: "Pride and Prejudice",
    "author": 'Jane Austen',
    'pages': 0450,
    price: NaN,
    rating: undefined,
    stock: Infinity, // value will change
    category: ['suspense', 'romance'],
}
```

Invalid JSON String

Slide Note

Self-Check

State True or False.

JSON structure is based on JavaScript object and thus it supports all the primitive and non-primitive data types supported by JavaScript language.

1. True

2. False

Self-Check: Solution

State True or False.

JSON structure is based on JavaScript object and thus it supports all the primitive and non-primitive data types supported by JavaScript language.

1. True

2. **False**

Explanation:

Although JSON follows JavaScript object notation, it does not support all the data types supported by JavaScript.
JSON does not support types such as `Function`, `Undefined` and `Date`.

Self-Check

Identify the error in JSON data shown below:

```
{
    "name": "John",
    "age": 30, // age must not be greater than 60
    "city": "New York"
}
```

1. Keys must be in single quotes.
2. String values must be in single quotes.
3. JSON data cannot have comments.
4. No error.

Slide Note

Self-Check

How do you convert JSON to JavaScript object?

1. Use the `convert()` method of JSON object.
2. Use `parse()` method of JSON object.
3. Use `stringify()` method of JSON object.
4. Use `toString()` method of JSON object.