LONG QUESTIONS OF WD-2

1.Write a note on jQuery.

- jQuery is a fast, small, cross-platform and feature-rich JavaScript library. It is designed to simplify the client-side scripting of HTML. It makes things like HTML document traversal and manipulation, animation, event handling, and AJAX very simple with an easy-to-use API that works on a lot of different type of browsers.
- The main purpose of jQuery is to provide an easy way to use JavaScript on your website to make it more interactive and attractive. It is also used to add animation.
- jQuery is a small, light-weight and fast JavaScript library. It is cross-platform and supports different types of browsers. It is also referred as ?write less do more? because it takes a lot of common tasks that requires many lines of JavaScript code to accomplish, and binds them into methods that can be called with a single line of code whenever needed. It is also very useful to simplify a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

- jQuery is a small, fast and lightweight JavaScript library.
- jQuery is platform-independent.
- jQuery means "write less do more".
- jQuery simplifies AJAX call and DOM manipulation.

2. Write a note on jQuery events.

- **jQuery Events**
- jQuery events are the actions that can be detected by your web application. They are used to create dynamic web pages. An event shows the exact moment when something happens.

These are some examples of events.

- A mouse click
- An HTML form submission
- A web page loading
- A keystroke on the keyboard
- Scrolling of the web page etc.

These events can be categorized on the basis their types:

Mouse Events

- click
- dblclick
- mouseenter
- mouseleave

Keyboard Events

- keyup
- keydown
- keypress

Form Events

- submit
- change
- blur
- focus

Document/Window Events

- load
- unload
- scroll
- resize

3. Write a note on effects of jQuery.

- jQuery is a popular JavaScript library that simplifies HTML document traversal and manipulation, as well as event handling, animation, and Ajax interactions. One of the most useful features of jQuery is its ability to create visual effects and animations on web pages. Here are some of the effects that can be achieved with jQuery:

- Hide and Show: jQuery provides methods for hiding and showing HTML elements. This can be useful for creating interactive interfaces, such as menus or pop-ups.

- Fade In and Fade Out: jQuery can animate the opacity of HTML elements, creating a smooth fade in or fade out effect. This effect can be used to gradually reveal or hide content on a web page.

- Slide: jQuery can animate the height or width of HTML elements, creating a sliding effect. This effect can be used to reveal or hide content, or to create interactive elements like accordion menus.

- Color: jQuery can animate the color of HTML elements, creating eye-catching effects. This effect can be used to highlight important elements on a web page or to create dynamic visual feedback.

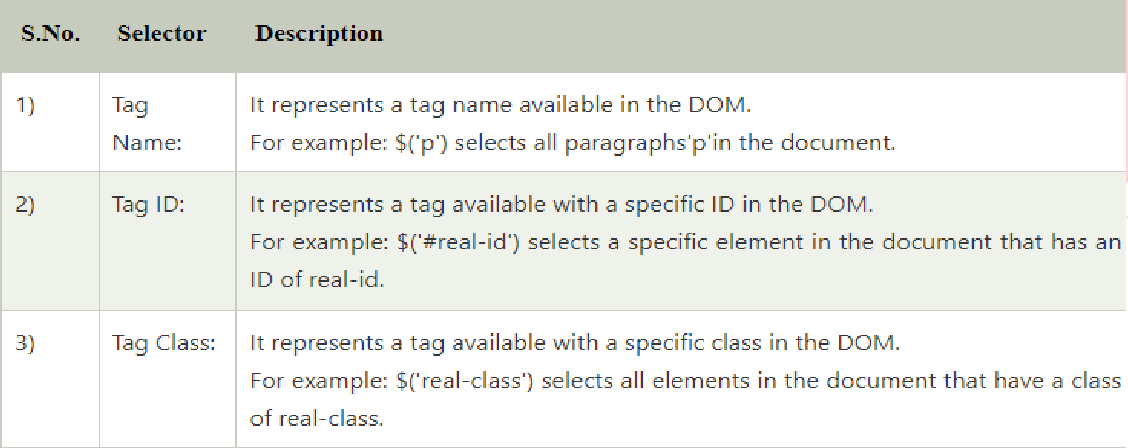
- Scroll: jQuery can animate the scroll position of a web page, creating a smooth scrolling effect. This effect can be used to create dynamic navigation or to provide a more engaging user experience.

4.What are the types of selectors in jQuery? Explain each

- **jQuery Selectors**
  - jQuery Selectors are used to select and manipulate HTML elements. They are very important part of jQuery library.
  - With jQuery selectors, you can find or select HTML elements based on their id, classes, attributes, types and much more from a DOM.
  - In simple words, you can say that selectors are used to select one or more HTML elements using jQuery and once the element is selected then you can perform various operation on that.
  - All jQuery selectors start with a dollor sign and parenthesis e.g. $(). It is known as the factory function.

The $() factory function

- Every jQuery selector start with thiis sign $(). This sign is known as the factory function. It uses the three basic building blocks while selecting an element in a given document.



S.No.	Selector	Description
1)	Tag Name:	It represents a tag name available in the DOM. For example: $('p') selects all paragraphs'p'in the document.
2)	Tag ID:	It represents a tag available with a specific ID in the DOM. For example: $('#real-id') selects a specific element in the document that has an ID of real-id.
3)	Tag Class:	It represents a tag available with a specific class in the DOM. For example: $('real-class') selects all elements in the document that have a class of real-class.

5.Explain jQuery methods for CSS manipulation.

- The common manipulation of classes includes actions like add class or remove class from the HTML tags.

The following classes are used for the manipulations :

- addClass()
- removeClass()
- toggleClass()

1. addClass() method: The purpose of addClass() function is to add one or more classes to the specified element or tag.

Syntax:

$('selector expression').addClass('class name');

2. removeClass() method: We use the removeClass() function to remove one or more or all classes from the specified element or tag.

Syntax:

$('selector expression').removeClass('class name');

3. toggleClass() method: We use the toggleClass() function to simultaneously add or remove the class to the specified element or tag.

Syntax:

$('selector expression').addClass('class name');

6. Explain jQuery insert methods .

- jQuery provides a number of methods for inserting content into HTML elements on a web page. These methods allow developers to dynamically add, remove, or replace HTML content based on user interactions or other events. Here are some of the most commonly used jQuery insert methods:

- .html(): This method sets or gets the HTML content of an element. It can be used to insert new HTML content into an element, or to retrieve the existing content for manipulation.

- .text(): This method sets or gets the text content of an element. It can be used to insert new text content into an element, or to retrieve the existing text content for manipulation.

- .append(): This method appends new content to the end of the selected elements. It can be used to add new HTML elements, text content, or other data to a web page.

- .prepend(): This method prepends new content to the beginning of the selected elements. It can be used to add new HTML elements, text content, or other data to a web page.

- .before(): This method inserts new content before the selected elements. It can be used to add new HTML elements, text content, or other data to a web page.

- .after(): This method inserts new content after the selected elements. It can be used to add new HTML elements, text content, or other data to a web page.

7.Explain jQuery methods to modify DOM.

➢ jQuery provides a number of methods to modify the Document Object Model (DOM) of a web page. The DOM is a tree-like structure that represents the HTML content of a web page, and can be manipulated using JavaScript and jQuery. Here are some of the most commonly used jQuery methods for modifying the DOM:

- .addClass(): This method adds one or more CSS classes to the selected elements. It can be used to dynamically change the styling of a web page based on user interactions or other events.

- .removeClass(): This method removes one or more CSS classes from the selected elements. It can be used to dynamically change the styling of a web page based on user interactions or other events.

- .attr(): This method sets or gets the value of an attribute on the selected elements. It can be used to dynamically modify the properties of HTML elements, such as the source of an image or the href of a link.

- .removeAttr(): This method removes an attribute from the selected elements. It can be used to dynamically modify the properties of HTML elements, such as removing the href from a link.

- .clone(): This method creates a copy of the selected elements, including all their child elements and attributes. It can be used to dynamically create new elements or to copy existing ones.

- .empty(): This method removes all child elements and content from the selected elements. It can be used to clear the content of an element, such as a form or a list.

- .remove(): This method removes the selected elements from the DOM. It can be used to remove elements that are no longer needed, such as form fields or list items.

8. Write features of AJAX.

- Ajax (Asynchronous JavaScript and XML) is a web development technique that allows web pages to dynamically update without reloading the entire page. Ajax enables web developers to make asynchronous requests to a server and update parts of a web page with new data without interrupting the user's experience. Here are some of the features of Ajax:

- Asynchronous data transfer: Ajax allows data to be transferred between the client and server asynchronously, meaning that the user can continue to interact with the web page while data is being loaded in the background.

- Partial page updates: Ajax enables developers to update parts of a web page dynamically, without reloading the entire page. This can help to improve the performance and user experience of a web application.

- Reduced server load: Ajax can help to reduce the load on the server by only sending and receiving the necessary data, rather than reloading the entire page with each request.

- Improved user experience: Ajax can help to improve the user experience of a web application by providing real-time updates and interactions.

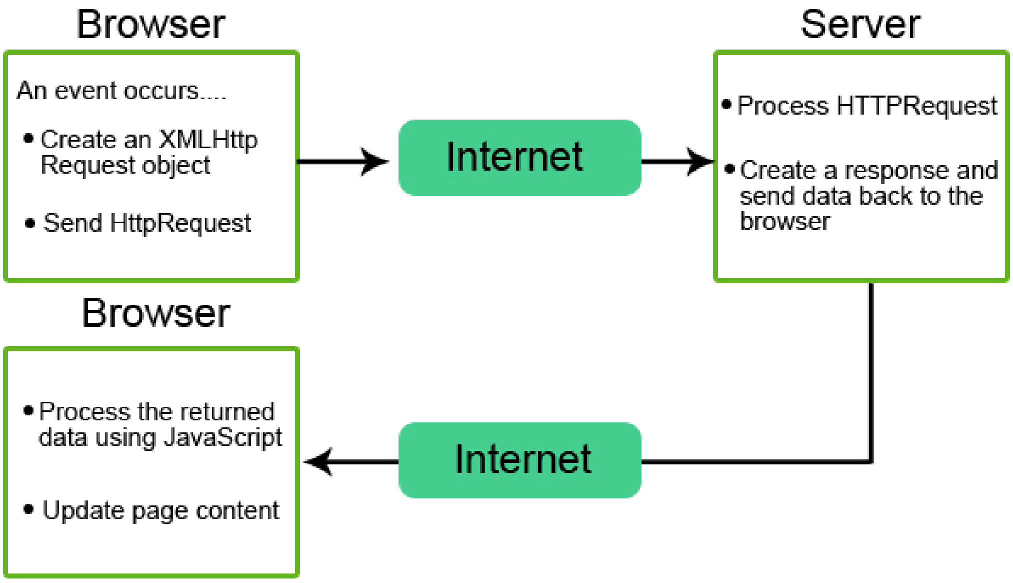
- Cross-platform compatibility: Ajax can be used on a variety of platforms and web browsers, making it a versatile and widely-used technology.

- Data exchange formats: Ajax supports a variety of data exchange formats, including XML, JSON, and HTML, making it flexible and adaptable to different data sources and APIs.

9. Explain working / architecture of AJAX with diagram.

- It creates more interactive techniques for faster and more efficient web applications by using JavaScript, XML, CSS, and HTML. For various web applications, Ajax uses various techniques like this.

- In Ajax, when the user needs to create content, XHTML is used while CSS is used for presenting the user request purpose; the document object model (DOM) and javascript both will be used to display the content dynamically.
- By using synchronous methods in web applications, information can be transmitted and received effectively, for example, when you fill a form and submit it. You will be automatically directed to the new server with that page information.
- After hitting the submit button in the background, javascript sends a request, and with the response generated, it will update to the current screen. In this process, the user will be unaware of the background XML code requests.
- XML is used as a format to generate and receive the server data in any format.
- While most web browsers are dependent on web server technology, it is independent of web server software.


Browser
An event occurs....
• Create an XMLHttp Request object
• Send HttpRequest
Internet
Server
• Process HTTPRequest
• Create a response and send data back to the browser
Internet
Browser
• Process the returned data using JavaScript
• Update page content

10. Explain various methods of XMLHttpRequest .

- XMLHttpRequest (XHR) is a JavaScript API used to send HTTP or HTTPS requests to a web server and receive responses from the server. There are various methods available in XMLHttpRequest, which can be used to interact with the server.

- open() method: The open() method is used to set up the XMLHttpRequest object with the HTTP method, URL, and asynchronous flag. The syntax for this method is as follows:
- xhr.open(method, url, async)
  Here, method is the HTTP method to be used, such as GET, POST, PUT, DELETE, etc. url is the URL of the server, and async is a Boolean value that determines whether the request should be sent asynchronously or synchronously.

- send() method: The send() method is used to send the request to the server. It accepts an optional parameter, which can be used to send data along with the request. The syntax for this method is as follows:
- xhr.send([body])
  Here, body is the data to be sent with the request, such as form data, JSON data, etc.

- setRequestHeader() method: The setRequestHeader() method is used to set the value of an HTTP request header. This method must be called after the open() method and before the send() method. The syntax for this method is as follows:
- xhr.setRequestHeader(header, value)
  Here, header is the name of the header, such as Content-Type, Authorization, etc., and value is the value of the header.

- getResponseHeader() method: The getResponseHeader() method is used to get the value of an HTTP response header. This method must be called after the send() method and before the onload event. The syntax for this method is as follows:
- xhr.getResponseHeader(header)
  Here, header is the name of the header whose value is to be retrieved.

- getAllResponseHeaders() method: The getAllResponseHeaders() method is used to get all the HTTP response headers as a string. This method must be called after the send() method and before the onload event. The syntax for this method is as follows:
- xhr.getAllResponseHeaders():
  This method returns all the HTTP response headers as a string.

- abort() method: The abort() method is used to cancel the current request. This method can be called at any time, even before the request is sent. The syntax for this method is as follows:
- xhr.abort()
  This method does not return any value.

11.Explain various properties of XMLHttpRequest

- onload: When the request is received (loaded), it defines a function to be called.
- onreadystatechange: A function will be called whenever the readyState property changes.
- readyState: It defines the current state of the request or holds the current status of the XMLHttpRequest. There are five states of a request:
  - readyState= 0: It represents the Request not initialized.
  - readyState= 1: Establishment of server connection.
  - readyState= 2: Request has been received
  - readyState= 3: During the time of processing the request
  - readyState= 4: Response is ready after finishing the request
- responseText: It will return the data received by the request in the form of a string.
- responseXML: It will return the data received by the request in the form of XML data.
- status: It will return the status number of the request. (i.e. 200 and 404 for OK and NOT FOUND respectively).
- statusText: It will return the status text in form of a string. (i.e. OK and NOT FOUND for 200 and 404 respectively).

12. Explain XMLHttpRequest objects.

- XMLHTTPRequest object is an API which is used for fetching data from the server. XMLHTTPRequest is basically used in Ajax programming. It retrieve any type of data such as json, xml, text etc. It request for data in background and update the page without reloading page on client side. An object of XMLHTTPRequest is used for asynchronous communication between client and server. The $.ajax() method is used for the creation of XMLHTTPRequest object.

The $.ajax() does following steps in background:

- Send data from background.
- Receives the data from the server.

- Update webpage without reloading the page.
- Below we will see how to create XMLHTTPRequest object with $.ajax() method:

SYNTAX

```
var XHR = $.ajax({configs});
```

13.Write difference between Synchronous and Asynchronous web application.

S. No.	Synchronous Transmission	Asynchronous Transmission
1.	In Synchronous transmission, data is sent in form of blocks or frames.	In Asynchronous transmission, data is sent in form of bytes or characters.
2.	Synchronous transmission is fast.	Asynchronous transmission is slow.
3.	Synchronous transmission is costly.	Asynchronous transmission is economical.
4.	In Synchronous transmission, the time interval of transmission is constant.	In Asynchronous transmission, the time interval of transmission is not constant, it is random.
5.	In this transmission, users have to wait till the transmission is complete before getting a response back from the server.	Here, users do not have to wait for the completion of transmission in order to get a response from the server.
6.	In Synchronous transmission, there is no gap present between data.	In Asynchronous transmission, there is a gap present between data.
7.	Efficient use of transmission lines is done in synchronous transmission.	While in Asynchronous transmission, the transmission line remains empty during a gap in character transmission.
8.	The start and stop bits are not used in transmitting data.	The start and stop bits are used in transmitting data that imposes extra overhead.

S. No.	Synchronous Transmission	Asynchronous Transmission
9.	Synchronous transmission needs precisely synchronized clocks for the information of new bytes.	Asynchronous transmission does not need synchronized clocks as parity bit is used in this transmission for information of new bytes.

14.Explain important concept, features and working of Node JS.

- Node.js tutorial provides basic and advanced concepts of Node.js. Our Node.js tutorial is designed for beginners and professionals both.

  Node.js is a cross-platform environment and library for running JavaScript applications which is used to create networking and server-side applications.

- Our Node.js tutorial includes all topics of Node.js such as Node.js installation on windows and linux, REPL, package manager, callbacks, event loop, os, path, query string, cryptography, debugger, URL, DNS, Net, UDP, process, child processes, buffers, streams, file systems, global objects, web modules etc. There are also given Node.js interview questions to help you better understand the Node.js technology.

What is Node.js

- Node.js is a cross-platform runtime environment and library for running JavaScript applications outside the browser. It is used for creating server-side and networking web applications. It is open source and free to use. Many of the basic modules of Node.js are written in JavaScript. Node.js is mostly used to run real-time server applications.

  Node.js also provides a rich library of various JavaScript modules to simplify the development of web applications.

Features of Node.js

Following is a list of some important features of Node.js that makes it the first choice of software architects.

1. **Extremely fast:** Node.js is built on Google Chrome's V8 JavaScript Engine, so its library is very fast in code execution.
2. **I/O is Asynchronous and Event Driven:** All APIs of Node.js library are asynchronous i.e. non-blocking. So a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call. It is also a reason that it is very fast.
3. **Single threaded:** Node.js follows a single threaded model with event looping.
4. **Highly Scalable:** Node.js is highly scalable because event mechanism helps the server to respond in a non-blocking way.
5. **No buffering:** Node.js cuts down the overall processing time while uploading audio and video files. Node.js applications never buffer any data. These applications simply output the data in chunks.
6. **Open source:** Node.js has an open source community which has produced many excellent modules to add additional capabilities to Node.js applications.
7. **License:** Node.js is released under the MIT license.

16. Discuss various components of Node JS.

- Requests– Requests are of two types, incoming and outgoing. Incoming requests can either be complex which means blocking or it can be simple which corresponds to non-blocking. It relies on the task which is to be performed by the users in the web application or software.

- Node js Server– It is a server-side platform that takes requests from users after which it processes those requests and sends those responses to corresponding users.

- Event Queue– This is where the incoming client requests get stored and are then passed onto the Event loop one by one.

- Event loop– It is an indefinite loop where it receives requests, processes them, and returns the corresponding response to the clients.

- Thread pool– It is a pool of all the threads available in node js for carrying out tasks required to fulfill clients' requests.

- External resources– These are resources used for blocking clients' requests, computation, data storage, etc.

17. Explain the Characteristic and syntax of XML.

- XML, or Extensible Markup Language, is a markup language used to store and transport data. Here are some of the characteristics of XML:

- Self-describing: XML documents are self-describing, meaning that they contain information about their structure and content. The document itself includes information about the data it contains, such as the names and types of elements, attributes, and values.

- Hierarchical structure: XML documents have a hierarchical structure, with elements that can contain other elements, attributes, and text nodes. The hierarchy of elements defines the structure of the document and the relationships between different parts of the data.

- Extensible: XML is extensible, meaning that it can be easily customized and extended to meet the needs of specific applications. Developers can define their own custom elements and attributes, or use existing ones defined by industry or community standards.

- Platform-independent: XML is platform-independent, meaning that it can be used on any operating system or platform. It is widely supported by programming languages, databases, and web technologies.

- Human-readable: XML documents are human-readable, meaning that they can be easily understood and edited by humans. The use of tags and attributes to describe data makes it easy to interpret and modify XML documents using a text editor.

- Interoperable: XML is interoperable, meaning that it can be easily exchanged and processed by different applications and systems. The use of open standards and common protocols, such as HTTP and SOAP, make it possible to transmit and process XML data across different platforms and networks.

- Separation of content and presentation: XML separates the content and presentation of data, allowing developers to focus on the structure and meaning of the data without worrying about how it will be displayed. This separation of concerns makes it easier to create reusable and maintainable code.

18. Compare JSON with XML.

JSON	XML
It is JavaScript Object Notation	It is Extensible markup language
It is based on JavaScript language.	It is derived from SGML.
It is a way of representing objects.	It is a markup language and uses tag structure to represent data items.
It does not provides any support for namespaces.	It supports namespaces.
It supports array.	It doesn't supports array.
Its files are very easy to read as compared to XML.	Its documents are comparatively difficult to read and interpret.
It doesn't use end tag.	It has start and end tags.
It is less secured.	It is more secured than JSON.
It doesn't supports comments.	It supports comments.
It supports only UTF-8 encoding.	It supports various encoding

19 .Explain JSON array and Object

- *Object in JSON* :

  In JSON an object is represented by a collection of Key-Value pairs. This collection of Key-Value pairs are grouped using { } (*opening and closing curly braces*}. Rules to writing an Object are

- *Key-Value pairs should be separated by a , (Comma)*
- *Each Object should **Start** with an **Opening {** (Opening Curly Brace)*
- Each Object should **End** with a ***Closing }*** *(Closing Curly Brace)*

An example here is the Person Object, discussed above. The Person object follows the rules mentioned for representing an Object

```
{
 "FirstName" : "Virender",
 "LastName"  : "Singh",
 "Age"       : 34,
 "Profession": "Engineer"
}
```

- *Array in JSON*

  Arrays are similar to Arrays that you know from any other programming language. In JSON an Array is collection of Values separated by Comma. Here are the rules to write an Array

- *An Array starts with an opening **[** (Bracket)*
- *An Array ends with a closing **]** (Bracket)*
- *Values in the Array are separated by **,** (Comma)*

To understand an Array let us add one more property to the ***Person Object***. Let us add hobby also, a Person can have multiple hobbies. This makes it suitable to represent hobbies as an Array. As shown in the *JSON* below

```
{
 "FirstName" : "Virender",
 "LastName"  : "Singh",
 "Age"       : 34,
 "Profession": "Engineer",
 "Hobbies"   : ["Videos games", "Computers", "Music"]
}
```

20.Explain query string in Node.js with example.

- **Node.js Query String**

  The Node.js Query String provides methods to deal with query string. It can be used to convert query string into JSON object and vice-versa.

  To use query string module, you need to use **require('querystring')**.

LONG QUESTIONS OF WD-2

- ## Node.js Query String Methods

The Node.js Query String utility has four methods. The two important methods are given below

Method	Description
querystring.parse(str[, sep][, eq][, options])	converts query string into JSON object.
querystring.stringify(obj[, sep][, eq][, options])	converts JSON object into query string.

- ## Node.js Query String Example 1: parse()

Let's see a simple example of Node.js Query String parse() method.

```
querystring = require('querystring');

const obj1=querystring.parse('name=sonoo&company=javatpoint');


console.log(obj1);
```

- ## Node.js Query String Example 2: stringify()

Let's see a simple example of Node.js Query String stringify() method.

File: query-string-example2.js

```
querystring = require('querystring');

const qs1=querystring.stringify({name:'sonoo',company:'javatpoint'});
console.log(qs1);
```

21.Explain features of JSON in detail with example.

- JSON, or JavaScript Object Notation, is a lightweight data interchange format that is easy for humans to read and write and easy for machines to parse and generate. Here are some of the key features of JSON:

- Simple syntax: JSON has a simple and straightforward syntax that is easy to understand and use. It consists of key-value pairs, where keys are always strings and values can be strings, numbers, arrays, objects, or Boolean values.

- Lightweight: JSON is a lightweight format, meaning that it has a small footprint and can be easily transmitted over the network. This makes it well-suited for use in web applications and other distributed systems.

- Easy to parse: JSON is easy to parse and generate using most programming languages. It can be quickly and easily converted to native data types in languages such as JavaScript, Python, and Java.

- Self-describing: JSON is self-describing, meaning that it includes information about its structure and content. The use of key-value pairs and nested objects makes it easy to understand the relationships between different parts of the data.

- Widely supported: JSON is widely supported by web browsers, servers, and programming languages. It is a popular choice for data interchange in web services and APIs, and is supported by many popular web frameworks and libraries.

- Extensible: JSON is extensible, meaning that it can be easily customized and extended to meet the needs of specific applications. Developers can define their own custom data types, or use existing ones defined by industry or community standards.

- Human-readable: JSON is human-readable, meaning that it can be easily understood and edited by humans. The use of simple syntax and whitespace makes it easy to interpret and modify JSON documents using a text editor.