+Q1. By convention, what name is used for the first key in a JSON schema?

- ☐ schema
- ☑ $schema
- ☐ JsonSchema
- ☐ JSONschema

Q2. Which code is the closest JSON equivalent of the data shown?

```
[
    'red',
    'blue',
    'green'
]
```

- [ ]

```
["red", "blue", "green"]
```

- [ ]

```
["red", "blue", "green"]
```

- [x]

```
{
  "color": "red",
  "color": "blue",
  "color": "green"
}
```

- [ ]

```
{
  "red": "red",
  "blue": "blue"
}
```

Q3. Which JavaScript method converts a JavaScript value to Json?

- ☐ JSON.parse()
- ☑ JSON.stringify()
- ☐ JSON.toString()
- ☐ JSON.objectify()

Q4. Which data type is NOT part of JSON standard?

- □ string
- □ number
- ☑ date
- □ array

Q5. Which term is commonly used to refer to converting data to JSON?

- □ unpacking
- ☑ serialization
- □ deserialization
- □ parsing

Q6. Which code uses the correct JSON syntax to encode the key/value pair shown?

```
accountNum: '000605802'
```

- ☑ "accountNum": "000605802"
- □ accountNum: "000605802"
- □ "accountNum": 000605802
- □ accountNum: 000605802

Q7. What character do you specify before a JSON control character when you want to use that control characters as a literal part of a string?

- □ /
- ☑ \
- □ :
- □ {

Q8. Which date type is part of the JSON standard?

- ☑ Boolean
- □ map
- □ promise
- □ function

Q9. Which key name is used to specify properties that must be included for JSON to be valid?

- ☐ important
- ☐ base
- ☐ core
- ☑ required

Q10. Which is the valid JSON equivalent of the data shown?

```
{
        photo: {
        width: 1600,
        height: 900,
        binaries: {
        url: 'https://www.example.com/images/34097349843',
        thumbnail: 'https://www.example.com/images/thumbs/34097349843'
        },
        animated: false,
        tags: [116, 943, 234, 38793],
        }
}
```

- [ ]

```
{
  "photo": {
    "width": 1600,
    "height": 900,
    "binaries": {
      "url": "https://www.example.com/images/34097349843",
      "thumbnail": "https://www.example.com/images/thumbs/34097349843"
    },
    "animated": "false",
    "tags": [116, 943, 234, 38793]
  }
}
```

- [ ]

```
{
  "photo": {
    "width": 1600,
    "height": 900,
    "binaries": {
      "url": "https://www.example.com/images/34097349843",
      "thumbnail": "https://www.example.com/images/thumbs/34097349843"
    },
    "animated": false,
    "tags": [116, 943, 234, 38793]
  }
}
```

- [x]

```
{
  "photo": {
    "width": 1600,
    "height": 900,
    "binaries": {
      "url": "https://www.example.com/images/34097349843",
      "thumbnail": "https://www.example.com/images/thumbs/34097349843"
    },
    "animated": false,
    "tags": [116, 943, 234, 38793]
  }
}
```

- [ ]

```
{
  "photo": {
    "width": "1600",
    "height": "900",
    "binaries": {
      "url": "https://www.example.com/images/34097349843",
      "thumbnail": "https://www.example.com/images/thumbs/34097349843"
    },
    "animated": "false",
    "tags": ["116", "943", "234", "38793"]
  }
}
```

Q11. How do you store several paragraphs of text as a string in JSON?

- ☑ Escape all whitespaces expect space characters.
- □ Escape line breaks.
- □ Escape paragraphs.
- □ Remove all whitespaces

Q12. What data type is represented by the value of the key/value pair shown?

```
loggedIn: true
```

- □ string
- ☑ Boolean
- □ number
- □ object

Q13. If you need to store the loggedIn status of a user in JSON as a boolean, what is the correct syntax?

- □ "loggedIn": (true)
- □ loggedIn: "true"
- ☑ "loggedIn": true
- □ loggedIn: {true}

Q14. What does JSON with padding (JSON-P) use to make a request?

- □ an `<origin>` element
- □ a header request
- ☑ the `<script>` element
- □ the XMLHttpRequest object

Q15. Which value is supported in the JSON specifications?

- □ undefined
- □ infinity
- □ NaN
- ☑ null

Q16. Which JavaScript method converts JSON to a JavaScript value?

- ☑ JSON.parse()
- □ JSON.stringify()
- □ JSON.toString()
- □ JSON.objectify()

Q17. Transferring JSON information from client to server and back again often includes HTTP requests. Match each method with the best choice describing its use.

```
1.Sends data to specific server to create or update information.
2.Sends data to specific server to create or update information without the risk
of creating the resource more than once.
3.Previews what the GET request response might be without the body of the text.
4.Learns the communication channels used by the target source.
5.Requests information from a specific source.
6.Removes information.
```

- □ 1.POST 2.DELETE 3.OPTIONS 4.HEADER 5.GET 6.PUT
- □ 1.POST 2.PUT 3.HEADER 4.OPTIONS 5.GET 6.DELETE

- □ 1.POST 2.DELETE 3.OPTIONS 4.HEADER 5.GET 6.PUT
- ☑ 1.POST 2.PUT 3.HEADER 4.OPTIONS 5.GET 6.DELETE

Q18. Which code uses valid JSON syntax for the text "Centennial Olympics?

- [ ]

```
{
"host": "Atlanta",
"year": 1996
/_ Centennial Olympics _/
}
```

- [ ]

```
{
"host": "Atlanta",
"year": 1996,
/_ Centennial Olympics _/
}
```

- [ ]

```
{
  "host": "Atlanta",
  "year": 1996
  // Centennial Olympics
}
```

- [x]

```
{
  "host": "Atlanta",
  "year": 1996,
  "comment": "Centennial Olympics"
}
```

Q19. Can trailing commas be used in objects and arrays?

- □ yes
- □ only if there is more than one item
- ☑ no
- □ only when arrays and objects contain more than 10 items

Q20. Which whitespace characters should be escaped within a string?

- □ All whitespace is allowed.

- ☐ double quotes, slashes new lines, and carriage returns
- ☑ new lines and carriage returns only
- ☐ double quotes only

Q21. Which is supported by YAML but not supported by JSON?

- ☐ nested
- ☑ comments
- ☐ arrays
- ☐ null values

Q22. Which is an invalid JSON value?

- ☐ `"'|=(_)(X 72(_)|\/||\*'"`
- ☐ `"|=(_)(X 72(_)|\/||\*"`
- ☐ `"|=(_)(X\" \"72(_)|\/||\*"`
- ☑ `"\s(_)(X 72(_)|\/||\*"`

Q23. How do you encode a date in JSON?

- ☐ Convert the date to UTC and enclose in quotes.
- ☑ Encode the date as string using the ISO-8601 date format.
- ☐ Wrap the date in double quotes.
- ☐ Add a "date" key to your object and include the date as string.

Q24. What's wrong with this JavaScript?

```
JSON.parse({"first": "Sarah", "last": "Connor"}) ;
```

- ☐ JSON should be wrapped with curly braces.
- ☐ JSON.parse() is missing an argument.
- ☑ The value supplied to JSON.parse() is not a string.
- ☐ Nothing is wrong with it.

Q25. What does this JavaScript code print?

```
printNullness(JSON.parse('{ "lemmings": [] }'));
printNullness(JSON.parse('{ "lemmings": null }'));
printNullness(JSON.parse('{ "lemmings": "null" }'));
```

```
function printNullness(testMe) {
  if (testMe.lemmings == null) console.log('null');
  else console.log('not null');
}
```

- ☑ A

```
not null
null
not null
```

- □ B

```
null
null
not null
```

- □ C

```
not null
null
null
```

- □ D

```
null
null
null
```

Q26. What tool might you use to validate your JSON?

- ☑ JSONLint
- □ ValidateJSON
- □ JSONFiddle
- □ TextEdit

Q27. What characters denote strings in JSON?

- ☑ double quotes
- □ smart (curly) quotes
- □ single or double quotes
- □ single quotes

Q28. Which code is a valid JSON equivalent of the key/value pair shown that also preserves the data type? `constant: 6.022E23`

- ☑ `"constant": "6.022E23"`
- □ `"constant": "6\.022\E23"`
- □ `constant: 6.022E23`
- □ `constant: "6.022E23"`

Reference **Explanation**: Scientific Notation is a valid JSON number. You can use `e` or `E`.

Q29. Which element is added to the DOM to create a JSON-P request?

- □ `<object>`
- ☑ `<script>`
- □ `<json>`
- □ `<meta>`

Q30. Why do so many APIs use JSON?

- □ Because it's object-based.
- □ Because it's a simple and adaptable format for sharing data.
- ☑ Because it's based on JavaScript.
- □ Because it is derived from SGML.

Q31. You need to assemble a list of members, but your JSON is not parsing correctly. How do you change it?

```
{
  "avengers": [
    "iron man",
    "hulk",
    "thor",
    "black widow",
    "ant man",
    "spider man'
  ]
}
```

- ☑ A

```
{
  "avengers": ["iron man", "hulk", "thor", "black widow", "ant man", "spider man"]
}
```

- □ B

```
{
  'avengers': [
  {'iron man',
  'hulk',
  'thor',
  'black widow',
  'ant man',
  'spider man'}
  ]
}
```

- ☐ C

```
{
  "avengers": ["iron man", "hulk", "thor", "black widow", "ant man", "spider man"]
}
```

- ☐ D

```
{
  "avengers": {[
    "iron man",
    "hulk",
    "thor",
    "black widow",
    "ant man",
    "spider man"
  ]}
}
```

Q32. When building dynamic web applications using AJAX, developers originally used the _ data format, which has since been replaced by JSON.

- ☑ XML
- ☐ GRAPHQL
- ☐ REST
- ☐ SOAP

Q33. How is a true boolean value represented in JSON?

- ☐ TRUE
- ☐ "true"
- ☐ 1
- ☑ true

Q34. Which array is valid JSON?

- ☐ ['tatooine', 'hoth', 'dagobah']

- □ [tatooine, hoth, dagobah]
- □ ["tatooine", "hoth", "dagobah",]
- ☑ ["tatooine", "hoth", "dagobah"]

Q35. What data type is encoded in the outermost component of the JSON shown?

```
[
  {
    "year": 2024
  },
  {
    "location": "unknown"
  }
]
```

- □ object
- ☑ array
- □ number
- □ string

Q36. Which is ignored by JSON but treated as significant by YAML?

- □ trailing commas
- □ trailing decimals
- ☑ whitespace
- □ leading zeroes

Reference

Q37. When you need to set the value of a key in JSON to be blank, what is the correct syntax for the empty value?

- □ FALSE
- □ 0
- □ ""
- ☑ null

Q38. How do you assign a number value in JSON?

- □ Escape the number with a backslash.

- □ Enclose the number in double quotes.
- □ Enclose the number in single quotes.
- ☑ Leave the number as is.

Q39. Which code uses the correct JSON syntax for encoding a key/value pair with a null value?

- ☑ "lastVisit": "null"
- □ lastVisit: null
- □ "lastVisit": 0
- □ "lastVisit": undefined

Note: None of the above options appear to be correct. A correct answer would be `"lastVisit": null`

Q40. Which reference to the Unicode character U+1F602 complies with the JSON standard?

- □ 128514
- □ d83dde02
- ☑ \uD83D\uDE02
- □ &#128514;

Q41. Which code uses the correct JSON syntax for a key/Value pair containing a string?

- ☑ "largest": "blue whale"
- □ largest: 'blue whale'
- □ 'largest': 'blue whale'
- □ largest: "blue whale"

Q42. Which key name is used to specify data type in a JSON schema?

- □ data
- □ schemadata
- □ schematype
- ☑ type

Q43. Which code is a valid JSON equivalent of the key/value pair shown that also preserves the original value?

- □ "UPC": \043875
- □ UPC: "043875"
- □ "UPC": 043875
- □ UPC: '043875'

NONE OF THESE ARE CORRECT A CORRECT ANSWER WOULD LOOK LIKE `"UPC": "043875"`

Q44. Which data format is a JSON schema written in?

- □ markdown
- □ YAML
- □ XML
- ☑ JSON

Q45. Which code is valid JSON equivalent of the key/value pair shown that also preserves the data type?

- ☑ "variance": "-0.0823"
- □ variance: "-0.0823"
- □ "variance": "-0.0823"
- □ variance: -0.0823

Q46. With what character should key/value pairs be separated?

- ☑ colon
- □ space
- □ semicolon
- □ comma

Q47. What character separates keys from values?

- ☑ :
- □ ->
- □ ::

- □ .

Q48. If a string contains line breaks, what should you do?

- □ Wrap line breaks in single quotation marks
- □ Make no changes, because whitespace is allowed
- □ Replace any line breaks with `\r`
- ☑ Replace any line breaks with `\n`

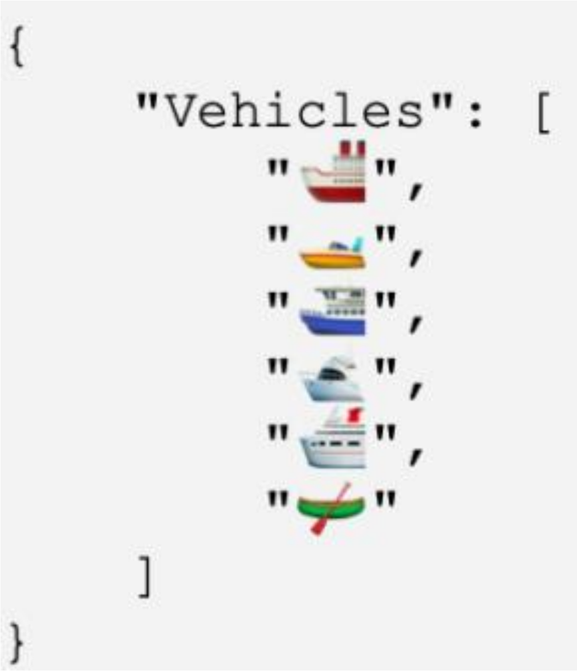
Q49. Which number types are available in javascript but not supported in json?

- □ Fractional and Transcendental
- □ Infinity or Rational
- □ Rational and Irrational
- ☑ Infinity or NaN

Q50. How should a date value be stored in JSON?

- □ As a string with quotes
- □ As a string without quotes
- □ As a string in ISO 8583 format
- ☑ As a string in ISO 8601 format

Q51. Is the JSON code shown here valid?



```
{
    "Vehicles": [
        "🚢",
        "🚤",
        "⛴",
        "🛥",
        "🛳",
        "🛶"
    ]
}
```

- ☐ No, emojis must be escaped with a backslash.
- ☐ No, JSON strings are not allowed to use emojis.
- ☐ No, emojis should never be enclosed with double quotes.
- ☑ Yes, emojis are valid characters because JSON strings are unicode.

Q52. What data structure do you use to encode ordered information?

- ☐ list
- ☑ array
- ☐ struct
- ☐ indexed hash

Q53. What are valid values in JSON?

- ☐ arrays, strings, numbers, true/false
- ☐ hashes, arrays, strings, numbers, booleans, null
- ☐ arrays, objects, lists, strings, numbers, booleans
- ☑ objects, arrays, strings, numbers, booleans, null

Q54. Which key format is valid JSON?

- ☐ key: "value"

- ☑ "key": "value"
- ☐ key, "value"
- ☐ 'key': 'value'

Q55. What data type is encoded in the outermost component of the JSON shown?

```
{
  "result": [
    {
      "year": 2024
    },
    {
      "location": "unknown"
    }
  ]
}
```

- ☐ number
- ☐ array
- ☐ string
- ☑ object

Q56. What's wrong with this JSON object?

```
{
  "series": "Star Trek",
  "episodes": "The Trouble with Tribbles"
}
```

- ☐ Key/value pairs should be separated by semicolons.
- ☑ Keys should be enclosed with double quotes.
- ☐ Key/value pairs should always have a trailing comma.
- ☐ Keys and values should be enclosed with single quotes.

Q57. How should comments be formatted in JSON?

- ☐ Wrap the comments in single quotes and place it at the bottom of the file.
- ☐ Wrap the comments in double parentheses.
- ☐ Escape comments by placing two slashes at the start of the comment.
- ☑ JSON does not support comments.

Q58. How would you make the following string valid in JSON?

```
"name": 'bb-8'
```

- □ name: "bb-8"
- □ "name": 'bb-8'
- ☑ "name": "bb-8"
- □ "name": bb-8

Q59. How are values separated in JSON?

- □ with brackets
- □ with colons
- ☑ with commas
- □ with parentheses

Q60. With which programming language is JSON best used?

- ☑ any language, as JSON is language agnostic.
- □ Ruby
- □ JavaScript
- □ Python

Q61. What values can arrays contain?

- □ only numbers
- ☑ any valid JSON value
- □ only strings and numbers
- □ only strings

Q62. What technique can be used to represent complex objects with JSON?

- □ reserialization
- ☑ nesting
- □ memorization
- □ minimization

Q63. What characters are used to enclose an array?

- ☐ []
- ☑ {}
- ☐ ""
- ☐ ()

Q64. What is the official MIME type for JSON?

- ☐ text/json
- ☐ text/javascript
- ☑ application/json
- ☐ data/json

Q65. What character cannot be part of string within JSON without additional formatting?

- ☐ /
- ☑ "
- ☐ -
- ☐ :

Q66. What is the purpose of JSON?

- ☑ to provide a simple way to serialize and deserialize data between different sources
- ☐ to provide a way to store data for machine learning
- ☐ to provide an archival solution for data warehousing
- ☐ to provide a way for JavaScript to run other languages

Q67. Can you assume the order of keys in an object?

- ☑ No, but you can sort the keys with `JSON.sort()`.
- ☐ Yes, object keys are always ordered alphabetically.
- ☐ Yes, but only when the object is first parsed.
- ☐ No, you can never assume object keys will be in order.

Q68. With what character should object keys be enclosed?

- □ *
- □ #
- □ ^
- ☑ "

Q69. What is the maximum number of key/value pairs that JSON supports?

- □ 10,000
- □ 1000
- ☑ There is no defined limit.
- □ 1 million

Q70. What is the recommended term used to refer to multiple resources?

- □ a swarm
- ☑ multiple resources
- □ a collection
- □ a group

Q71. Does JSON support signed numbers?

- □ yes, but only if they are enclosed in quotes
- □ only with integers
- ☑ no
- □ yes

Q72. What two nonnumerical characters can numbers contain?

- ☑ dash and dot
- □ dash and comma
- □ comma and exclamation point
- □ dot and comma

Q73. What JavaScript method is used to load JSON data?

- □ JSON.ingest()
- □ JSON.convert()

- □ JSON.read()
- ☑ JSON.parse()

Q74. How are the values in an array separated?

- □ with right arrows
- □ with colons
- □ with semicolons
- ☑ with commas

Q75. What is the minimum number of values in an array?

- ☑ zero
- □ one
- □ two
- □ three

Q76. In JSON, a set of brackets ([]) is used to denote _, whereas curly braces ({}) denote _.

- ☑ arrays; objects
- □ objects; functions
- □ arrays; functions
- □ key/value pairs; arrays

Q77. How does JSON represent truth, falsity and nullness?

- ☑ true, false, null
- □ true, false
- □ TRUE, FALSE, NULL
- □ true, false, null

Q78. If an object key contains spaces, how can you access its value in JavaScript?

- □ Remove spaces from the key before accessing the value.
- □ Use dot notation to access the value.

- □ Use object key index to access the value.
- ☑ Use bracket notation to access the value.

Q79. When parsing JSON, "caching" is a method used to _.

- □ remove duplicate data from a server
- ☑ temporarily store data for faster access
- □ store excess information in chunks to be reviewed later
- □ clog up a web browser with useless information

Q80. What is the only valid whitespace character within a string that does not require an escape character?

- □ line break
- ☑ space
- □ tab
- □ return

Q81. What technique can you use to safely encode very large numbers?

- ☑ Store the number as a string
- □ Convert the number into an exponent.
- □ Round the number to the nearest 10-digit number.
- □ Split the number into smaller parts.

Q82. Which code is a valid JSON equivalent of the key/value pair shown that also preserves the data type?

```
variance: -0.0823
```

- □ variance: "-0.0823"
- □ variance: -0.0823
- ☑ "variance": "-0.0823"
- □ "variance": "-0.0823"

Reference data types in json

Q83. Based on the JSON object shown, which JavaScript code would you use to find out what is in this vehicle's glove box while assigning the glove box contents to a new variable?

```
myVehicleContents = {
  van: {
    inside: {
      'glove box': 'maps',
      'passenger seat': 'crumbs',
    },
    exterior: {
      trunk: 'jack',
    },
  },
};
```

- ☐ A

```
myVehicleContents.van.inside['glove box'] = gloveBoxContents;
```

- ☑ B

```
var gloveBoxContents = myVehicleContents.van.inside['glove box'];
gloveBoxContents;
```

- ☐ C

```
var gloveBoxContents = myVehicleContents.van.glove box.contents;
gloveBoxContents;
```

- ☐ D

```
myVehicleContents.van.inside['inside'] = gloveBoxContents;
```

Source

Q84. What characters are used to define an object?

- ☐ ||
- ☑ {}
- ☐ ()
- ☐ []

Q85. The JSON structure features nested objects and arrays. Sometimes the data containing these features exists in relational databases. How is the relational structure different that JSON?

- □ It has a flat architecture.
- □ It has a rewritable structure.
- ☑ It has an array-only structure.
- □ It has a table structure.

Q86. What is supported by YAML but not supported by JSON?

- □ arrays
- □ null values
- ☑ comments
- □ nested