JavaScript 50 Q&A: HTTP, JSON, AJAX, and Promises

Q1 [Interview] What is HTTP?

Ans: HyperText Transfer Protocol used for communication over the web.

Q2 [Interview] Name some HTTP methods.

Ans: GET, POST, PUT, DELETE, PATCH

Q3 [Interview] What is status code 200?

Ans: Success.

Q4 [Interview] What is status code 404?

Ans: Not Found.

Q5 [Interview] What is JSON?

Ans: JavaScript Object Notation, a lightweight data format.

Q6 [Interview] How do you parse JSON?

Ans: Use JSON.parse().

```
JSON.parse('{"name":"John"}')
```

Q7 [Interview] How do you stringify an object?

Ans: Use JSON.stringify().

Q8 [Interview] What is AJAX?

Ans: Asynchronous JavaScript and XML, used for dynamic content loading.

Q9 [Interview] What is XMLHttpRequest?

Ans: An older API for AJAX.

Q10 [Interview] What is fetch()?

Ans: Modern promise-based AJAX API.

Q11 [Interview] What is a Promise?

Ans: An object representing the eventual completion or failure of an async operation.

Q12 [Interview] What does .then() do?

Ans: Handles resolved Promise result.

Q13 [Interview] What does .catch() do?

Ans: Handles rejected Promise result.

Q14 [Interview] What is async/await?

Ans: Syntax sugar for working with promises.

Q15 [Interview] Is fetch() async?

Ans: Yes, it returns a promise.

Q16 [DSA] Fetch data from API using fetch().

Ans:

```
fetch('https://api.example.com/data')
.then(res => res.json())
```

```
.then(data => console.log(data));
```

Q17 [DSA] POST data to API with fetch.

Ans:

```
fetch('/submit', {
  method: 'POST',
  body: JSON.stringify(data),
  headers: { 'Content-Type': 'application/json' }
})
```

Q18 [DSA] Create custom Promise that resolves in 2s.

Ans:

```
new Promise(res => setTimeout(() => res('Done'), 2000));
```

Q19 [DSA] Chain multiple fetch calls.

Ans:

Q20 [DSA] Handle fetch error with .catch().

Ans:

Q21 [DSA] Convert object to JSON before sending.

Ans:

Q22 [DSA] Display loading while waiting for fetch.

Ans:

Q23 [DSA] Abort fetch request using AbortController.

Ans:

Q24 [DSA] Retry fetch on failure.

Ans:

Q25 [DSA] Create wrapper function to fetch + parse JSON.

Ans:

Q26 [DSA] Validate API response before parsing.

Ans:

Q27 [DSA] Use async/await to fetch and return data.

Ans:

Q28 [DSA] Log status code of fetch.

Ans:

Q29 [DSA] Create loader using fetch().

Ans:

Q30 [DSA] Handle nested JSON fetch.

Ans:

Q31 [LeetCode] typeof JSON

Ans: 'object'

Q32 [LeetCode] typeof fetch

Ans: 'function'

Q33 [LeetCode] JSON.parse('{}')

Ans: returns {}

Q34 [LeetCode] Promise.resolve(5).then(x => x)

Ans: 5

Q35 [LeetCode] Promise.reject('error').catch(e => e)

Ans: 'error'

Q36 [LeetCode] fetch returns?

Ans: a Promise

Q37 [LeetCode] async function always returns?

Ans: a Promise

Q38 [LeetCode] await 10 returns?

Ans: 10

Q39 [LeetCode] typeof XMLHttpRequest

Ans: 'function'

Q40 [LeetCode] JSON.stringify([1,2])

Ans: '[1,2]'

Q41 [Conceptual] Is fetch() better than XHR?

Ans: Yes, modern and promise-based.

Q42 [Conceptual] Can JSON.parse() fail?

Ans: Yes, with invalid JSON.

Q43 [Conceptual] Are Promises synchronous?

Ans: No, they're async.

Q44 [Conceptual] Can .then() return another Promise?

Ans: Yes.

Q45 [Conceptual] Can await only be used in async function?

Ans: Yes.

Q46 [Conceptual] Can fetch handle timeout?

Ans: Only with AbortController.

Q47 [Conceptual] What happens if fetch fails?

Ans: Promise is rejected.

Q48 [Conceptual] Can JSON have comments?

Ans: No.

Q49 [Conceptual] Is JSON valid JavaScript?

Ans: Subset of JavaScript.

Q50 [Conceptual] What does JSON stand for?

Ans: JavaScript Object Notation.