






LEARN JAVASCRIPT

 Challenge Your JavaScript Knowledge with Our Quiz! 
JavaScript Web Storage and Asynchronous

JavaScript Web Storage and Asynchronous	1
JavaScript Web Storage and Asynchronous Operations	2
1. Web Storage: Local and Session Storage	2
Example: Storing data in Local Storage	2
Example: Retrieving data from Local Storage	2
Example: Using Session Storage	3
2. Asynchronous Operations: Promises and Async/Await	3
Example: Using Promises	3
Example: Async/Await	3
Quiz Questions and Answers	4
Q1: What is the main difference between Local Storage and Session Storage in web browsers?	4
Q2: How do you remove an item from Local Storage in JavaScript?	4
Q3: What does the fetch function in JavaScript return?	5
Q4: What will happen if you try to JSON.parse() a value retrieved from Local Storage?	5
Q5: What is the purpose of using async before a function in JavaScript?	5

 Whether you're a seasoned pro or just starting out, this quiz is a great way to assess your understanding of JavaScript's nuances and functionalities. Dive in and see how you stack up in the world of web development. 

 Here's a sneak peek at what you can expect:

- Do you know the difference between Local Storage and Session Storage?
- Ever wondered how `async` and `await` enhance JavaScript's capabilities?
- What about the intricacies of Promises and their states?

JavaScript Web Storage and Asynchronous Operations

JavaScript's Web Storage API and asynchronous operations are fundamental in creating dynamic, user-friendly web applications. Web Storage allows data to be stored locally on the user's browser, while asynchronous operations enable the execution of long-running tasks without blocking the main thread.

1. Web Storage: Local and Session Storage

Example: Storing data in Local Storage

```
localStorage.setItem('username', 'Alice');
```

Explanation: This stores a key-value pair in the browser's local storage. The data persists even after the browser is closed.

Example: Retrieving data from Local Storage

```
let username = localStorage.getItem('username'); // 'Alice'
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Explanation: This retrieves the value associated with the key 'username' from local storage.

Example: Using Session Storage

```
sessionStorage.setItem('sessionKey', '12345');  
let sessionValue = sessionStorage.getItem('sessionKey'); // '12345'
```

Explanation: Session Storage is similar to Local Storage but is cleared when the page session ends (e.g., when the tab is closed).

2. Asynchronous Operations: Promises and Async/Await

Example: Using Promises

```
let promise = new Promise((resolve, reject) => {  
  setTimeout(() => resolve("Data Loaded"), 3000);  
});
```

```
promise.then(data => console.log(data)); // Logs "Data Loaded" after 3 seconds
```

Explanation: A Promise represents a future value. It can be in one of three states: pending, resolved, or rejected. Here, after 3 seconds, the promise is resolved with the message "Data Loaded".

Example: Async/Await

```
async function fetchData() {  
  let response = await fetch('https://api.example.com/data');  
  let data = await response.json();  
}
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
return data;
}
fetchData().then(data => console.log(data));
```

Explanation: async and await make working with promises easier and more readable. The fetchData function asynchronously fetches data and waits for it to be resolved before proceeding.

Quiz Questions and Answers

Q1: What is the main difference between Local Storage and Session Storage in web browsers?

- A) Local Storage is larger
- B) Session Storage is temporary and gets cleared after the session ends
- C) Local Storage can store complex data types

Answer: B) Session Storage is temporary and gets cleared after the session ends

Q2: How do you remove an item from Local Storage in JavaScript?

- A) localStorage.deleteItem('key')
- B) localStorage.removeItem('key')
- C) localStorage.clearItem('key')

Answer: B) localStorage.removeItem('key')

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Q3: What does the fetch function in JavaScript return?

- A) A JSON object
- B) A Promise
- C) An Array

Answer: B) A Promise

Q4: What will happen if you try to JSON.parse() a value retrieved from Local Storage?

- A) It converts the string back to its original JavaScript object
- B) It causes an error
- C) It returns null

Answer: A) It converts the string back to its original JavaScript object

Q5: What is the purpose of using async before a function in JavaScript?

- A) To indicate that the function runs asynchronously
- B) To automatically catch any errors in the function
- C) To speed up the function execution

Answer: A) To indicate that the function runs asynchronously

These concepts and examples provide a foundation for understanding how web storage and asynchronous operations work in JavaScript, essential for building modern, efficient web applications.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>