PPT Program Assignment

Web Development Assignment - 5

Answer 1: Synchronous means events happen one after another in a specific order, while asynchronous means events can happen independently and at different times. Synchronous actions follow a timeline, while asynchronous actions can occur simultaneously or at different paces. Synchronous requires immediate responses, while asynchronous does not.

Answer 2: Web APIs are sets of rules that allow software applications to communicate over the internet. They enable developers to access features and data from web applications or services. APIs provide a standardized way to request and retrieve information or perform actions. They facilitate integration and enable the creation of powerful and interconnected software systems.

Answer 3: setTimeout is a JavaScript function that executes a callback function after a specified delay, allowing for a one-time delayed execution. setInterval, on the other hand, repeatedly executes a callback function at a defined interval until it is stopped. Both functions are used for scheduling actions in JavaScript.

Answer 4: In JavaScript, you can handle asynchronous code using various techniques:
1. Callbacks: Pass a function as a callback parameter to be executed once the asynchronous operation completes.
2. Promises: Use Promises to represent the eventual completion or failure of an asynchronous operation, and chain '.then()' and '.catch()' methods to handle the results.
3. Async/await: Use the 'async' keyword to define an asynchronous function and the 'await' keyword to pause execution until a Promise is resolved or rejected.
4. Event listeners: Attach event listeners to handle asynchronous events, such as user interactions or network responses.
5. Libraries and frameworks: Utilize libraries like Axios or frameworks like Node.js to simplify asynchronous code handling with built-in functionalities.
6. Error handling: Implement proper error handling techniques, such as try-catch blocks or error callback functions, to handle exceptions that may occur during asynchronous operations.

Answer 5: Callbacks are functions passed as arguments to other functions, which are then invoked at a later point in the program. They are commonly used in asynchronous programming to handle the result of an asynchronous operation.

Callback Hell refers to a situation where multiple nested callbacks are used, leading to code that becomes difficult to read and maintain. This occurs when callbacks are used as a means of handling sequential or dependent asynchronous operations, resulting in deeply nested code structures that are hard to follow and debug. It can be mitigated using techniques like Promises, async/await, or modularizing code.

Answer 6: Promises are JavaScript objects that represent the eventual completion or failure of an asynchronous operation. They are used to handle asynchronous code in a more organized and readable way. Promises have three essential methods:

1. .then(): This method is used to handle the successful completion of a Promise. It takes a callback function as an argument, which is invoked when the Promise is resolved. The callback function receives the resolved value as its parameter.

2. .catch(): This method is used to handle any errors that occur during the Promise execution. It takes a callback function as an argument, which is invoked when the Promise is rejected. The callback function receives the error as its parameter.

3. .finally(): This method is optional and is used to specify a callback function that will be executed regardless of whether the Promise is resolved or rejected. It is commonly used for cleanup operations or to perform actions that need to be done regardless of the Promise outcome.

Answer 7: The async keyword is used to define an asynchronous function in JavaScript, indicating that the function will contain asynchronous operations.

The await keyword is used within an async function to pause the execution of the function until a Promise is resolved or rejected. It allows for writing asynchronous code in a synchronous-like manner, making it easier to read and understand.

Answer 8: The purpose of a try and catch block in JavaScript is to handle exceptions and errors that may occur during the execution of a code block.

When potentially problematic code is placed within a try block, any exceptions thrown within that block are caught by the catch block. This allows you to gracefully handle errors, preventing the code from abruptly stopping and providing an opportunity to handle the error or execute alternative logic.

Answer 9: The fetch function in JavaScript is used to make HTTP requests and retrieve resources from a specified URL. It returns a Promise that resolves to the response object containing the server's response. You can handle the response using .then() and access the data returned by the server. fetch simplifies asynchronous network requests and is commonly used for API integration and data retrieval in web development.

Answer 10: To define an asynchronous function in JavaScript using async/await, you prepend the async keyword before the function declaration. Here's an example:

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

In the above code snippet, the fetchData function is defined as an asynchronous function using the async keyword. Within the function, the await keyword is used to pause the execution until the fetch operation completes and the response is received. The await keyword is also used to extract the JSON data from the response.