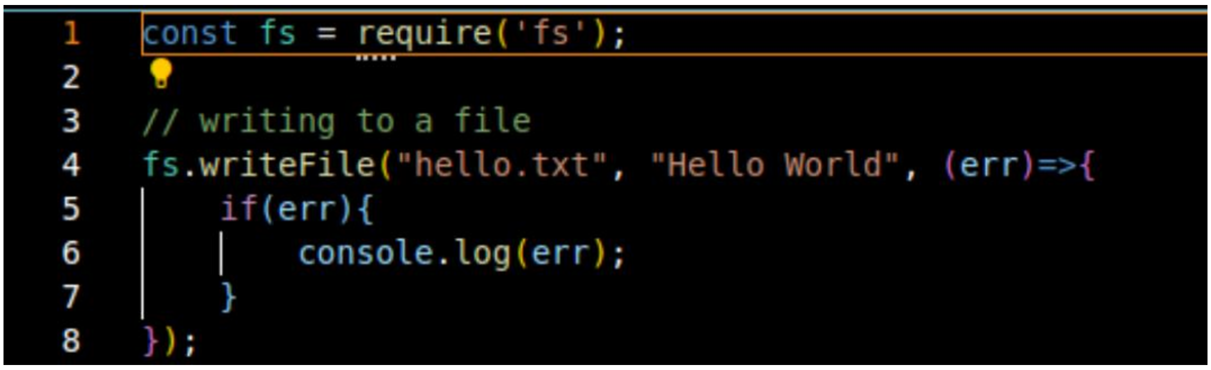
Date: 7 June 2024 Day: Friday

Overview:

The 3rd day of the internship was dedicated to exploring the fundamentals of Node.js, a powerful runtime environment built on Chrome's V8 JavaScript engine. Node.js allows developers to run JavaScript code outside of a web browser, making it suitable for server-side development.

Learning Objectives:

- **Basics of Node.js:**
  - Node.js is an open-source, cross-platform JavaScript runtime environment.
  - It enables developers to build scalable and high-performance applications.
- **Global Object and its Functionalities:**
  - In Node.js, the global object represents global variables and functions accessible throughout the application.
  - Key components include console, process, and require.
- **Working with the util Module:**
  - The util module provides utility functions for simplifying common tasks in Node.js.
  - Functions like util.format() for string formatting and util.promisify() for converting callback-based functions into promise-based ones were explored.



```
const fs = require('fs');

// writing to a file
fs.writeFile("hello.txt", "Hello World", (err)=>{
    if(err){
        console.log(err);
    }
});
```

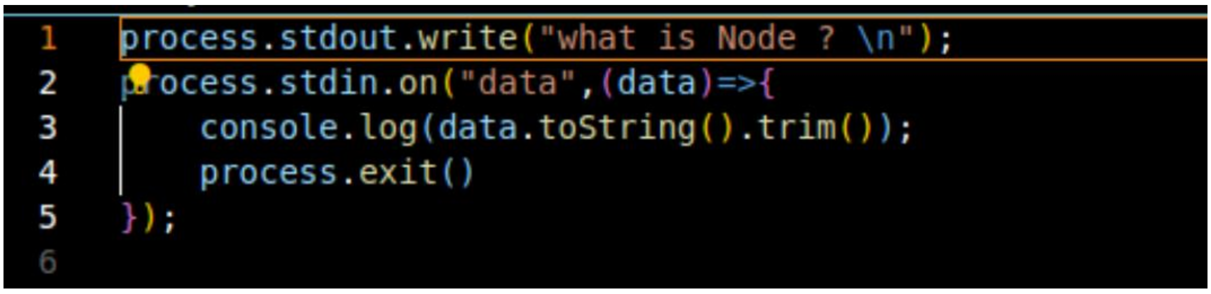
Activities and Insights:

- Exploring Node.js Basics
- Node.js was introduced as a runtime environment that allows JavaScript code execution outside of the browser, leveraging its event-driven, non-blocking I/O model for efficiency.

Understanding the Global Object:

The global object in Node.js encompasses various modules and utilities essential for application development. Notable components discussed include:

- **console Module:** Used for printing output to the console. Methods like console.log() and console.error() were highlighted for debugging and informational purposes.
- **process Object:** Provides information about the current Node.js process. It includes properties like process.env for environment variables and methods such as process.exit() for terminating the application.
- **require() Function:** Facilitates module loading within Node.js. It allows importing external modules and files into the current application scope.

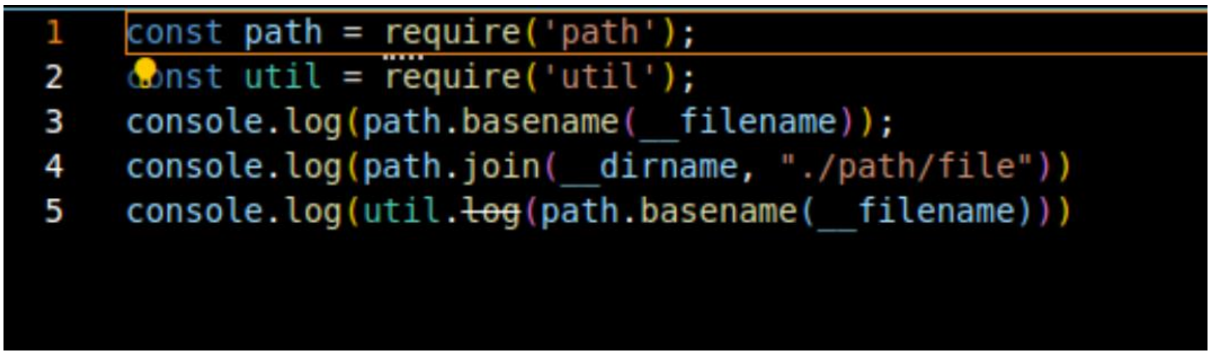


```javascript
process.stdout.write("what is Node ? \n");
process.stdin.on("data",(data)=>{
    console.log(data.toString().trim());
    process.exit()
});
```

Utilizing the util Module:

The util module was explored to streamline development tasks with built-in utility functions. Key functions studied include:

- **util.format():** Enables string formatting similar to printf-style formatting in C.
- **util.promisify():** Converts callback-based functions into promise-based functions, enhancing code readability and maintainability in asynchronous operations.



```javascript
const path = require('path');
const util = require('util');
console.log(path.basename(__filename));
console.log(path.join(__dirname, "./path/file"))
console.log(util.~~log~~(path.basename(__filename)))
```