Demo : Debugging JavaScript Code with GitHub Copilot

Demonstrate how GitHub Copilot assists in debugging JavaScript code by identifying errors, suggesting fixes, and explaining the issues.

Prerequisites

- Visual Studio Code installed
- GitHub Copilot extension enabled
- GitHub Copilot subscription
- NodeJS 18 or later
- A sample JavaScript project

1. Create a Buggy JavaScript File. Open VS Code and create a file named buggy.js. Add the following faulty code:

```
function calculateTotal(price, tax) {
    total = price * tax + price;
    return total;
}
console.log(calculateTotal(100, 0.05));
```

2. Identify Errors Using Copilot. Enable Copilot Chat (if available) and ask:

```
What is wrong with this code?
```

Copilot’s Expected Response:

```
`total` is assigned without `let`, `const`, or `var` (causes an implicit
global variable).
Possible incorrect tax calculation (should it be price * (1 + tax)
instead?).
```

3. Fixing the Code with Copilot Hover over the error or use Copilot inline suggestions to generate a fix. Alternatively, ask:

```
How can I fix this function to avoid global variables and improve accuracy?
```

Copilot Suggested Fix:

```
function calculateTotal(price, tax) {
    let total = price * (1 + tax);
    return total;
}

console.log(calculateTotal(100, 0.05)); // Correct output: 105
```

4. Debugging with Copilot Explanations To understand why the fix works, ask Copilot:

```
Can you explain why using 'let' is important here?
```

Expected Copilot Explanation:

```
Using let ensures total is block-scoped and not a global variable.
The calculation is now accurate, using price * (1 + tax).
```

5. Experiment with Copilot Fix Suggestions Now, introduce another issue by modifying the function:

```
function calculateTotal(price, taxRate) {
    let total = price * (1 + taxRate / 100);
    return total.toFixed(2);
}

console.log(calculateTotal(100, "five"));
```

Ask Copilot:

```
What will happen if taxRate is not a number?
```

Copilot should highlight that passing "five" will result in NaN (Not a Number). Then, ask:

```
How can I ensure taxRate is always a valid number?
```

Copilot might suggest:

```
function calculateTotal(price, taxRate) {
    if (isNaN(taxRate) || typeof taxRate !== "number") {
        throw new Error("Invalid tax rate. Must be a number.");
    }
    let total = price * (1 + taxRate / 100);
    return total.toFixed(2);
}

console.log(calculateTotal(100, 5)); // Output: 105.00
```