**Adding a Breakpoint in Code**

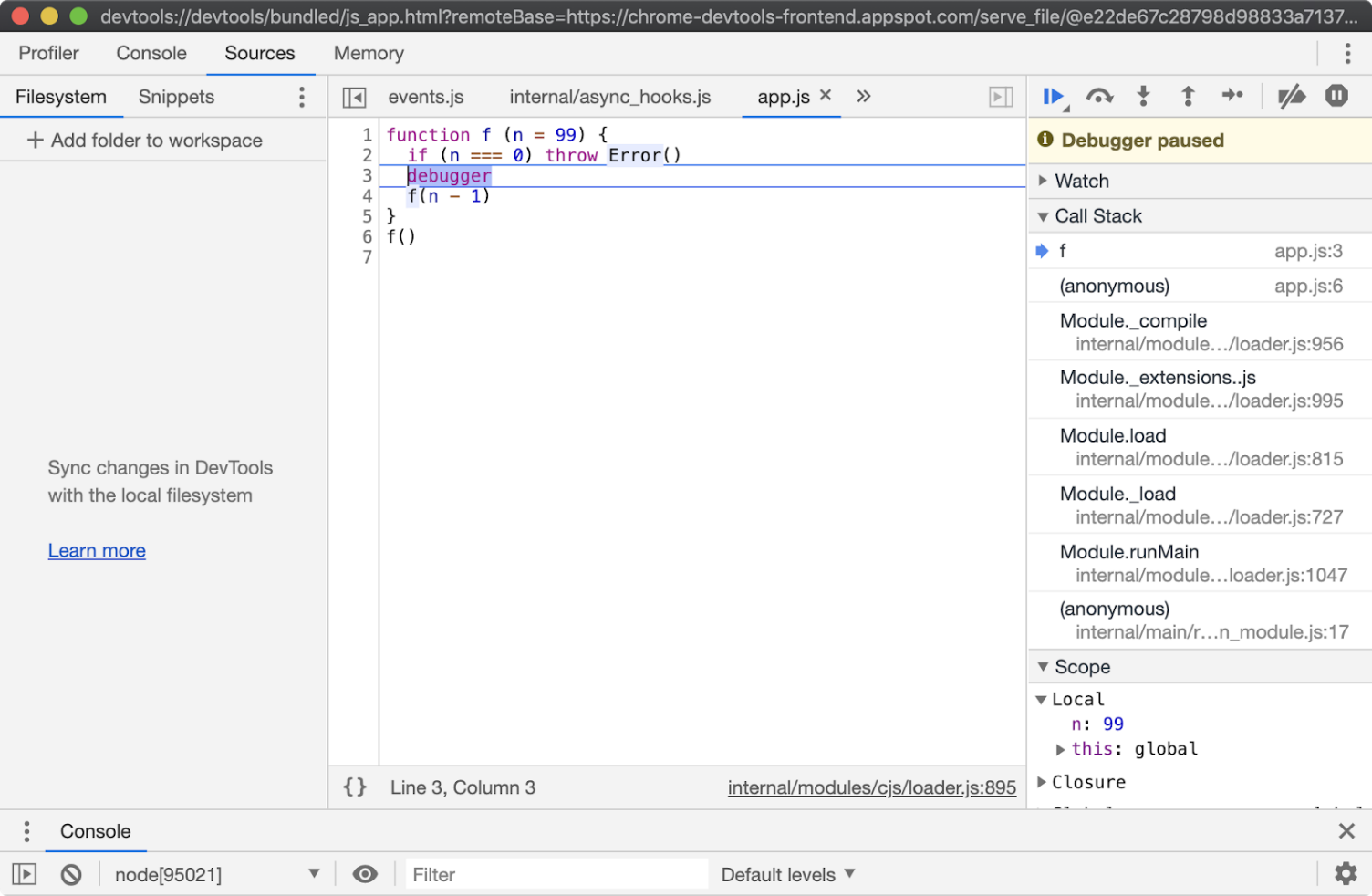
In some scenarios it can be easier to set a breakpoint directly in the code, instead of via the Devtools UI.

The **debugger** statement can be used to explicitly pause on the line that the statement appears when debugging.

Let's edit **app.js** to include a **debugger** statement on line 3:

**function f (n = 99) {  
  if (n === 0) throw Error()  
  debugger  
  f(n - 1)  
}  
f()**

This time, start **app.js** in Inspect mode, that is with the **--inspect** flag instead of the **-inspect-brk** flag. Once Chrome Devtools is connected to the inspector, the "Sources" tab of Devtools will show that the application is paused on line 3:



Using the **debugger** statement is particularly useful when the line we wish to break at is buried somewhere in a dependency tree: in a function that exists in a required module of a required module of a required module and so on.

When not debugging, these **debugger** statements are ignored, however due to noise and potential performance impact it is not good practice to leave **debugger** statements in code.