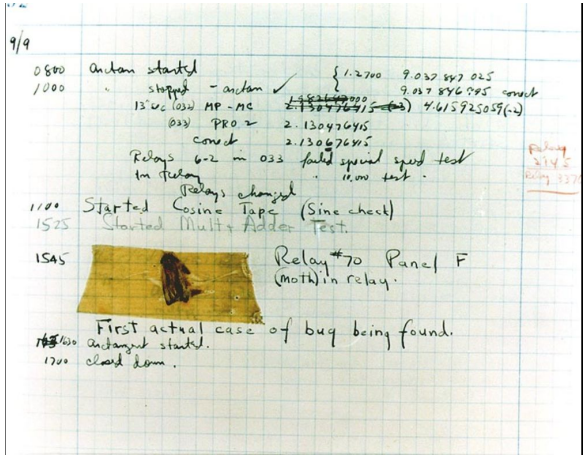
Debugging

In Java and JavaScript

History

The term was coined by computer scientist Grace Hopper in the 1940's. A moth was discovered inside Harvard's Mk. II computer where it was impeding the machine's normal operations.



Source: https://en.wikipedia.org/wiki/Debugging

Types of Bugs

Bugs fall into a few broad categories:

- **Syntax Errors**: Your code does not conform to the basic linguistic syntax required by the language.
  - Modern IDE's are good at catching these "live" as you are typing code.

- **Semantic Errors**: Your code conforms to the syntax, but it's not producing the expected outcome.
  - These are more challenging, **debuggers are great tools to dissect these problems**.

Debugging Vocab

- **Breakpoint**: By setting a breakpoint, you tell the debugger to pause the program at the indicated line.

- **Stepping Through**: Once the program is paused, we can tell the debugger to execute the program line by line.
  - **Stepping Into:** If the current line is on a method call, stepping into will take us to the code where the method is defined.
  - **Stepping Over**: If the current line is on a method call, stepping over will not take us to the method definition and we will remain in the current method.
  - **Step Return (aka Step Out)**: Returns execution to the calling method.

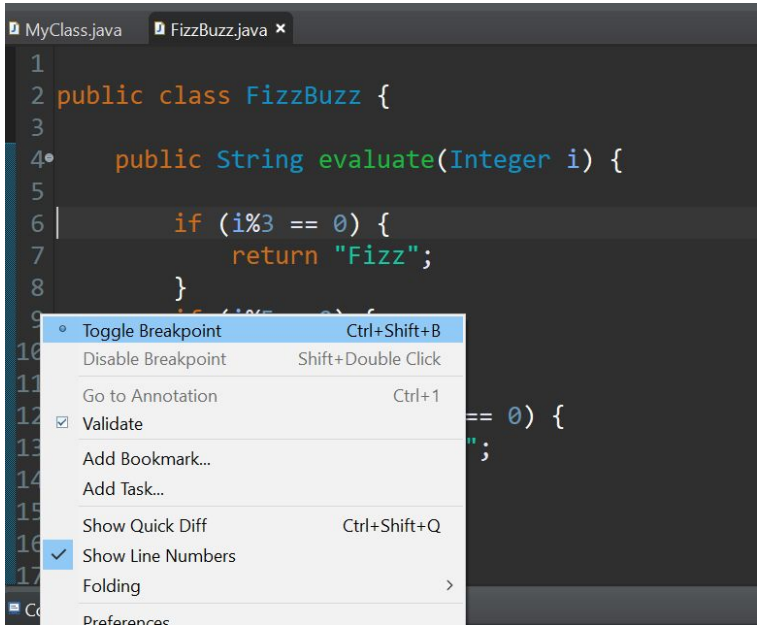
- **Resuming**: Stop stepping, go to the next breakpoint. If there are no more breakpoints, the code runs to completion.

Debugging in Eclipse

Most full featured IDE’s have a built in debugger and Eclipse is no different!

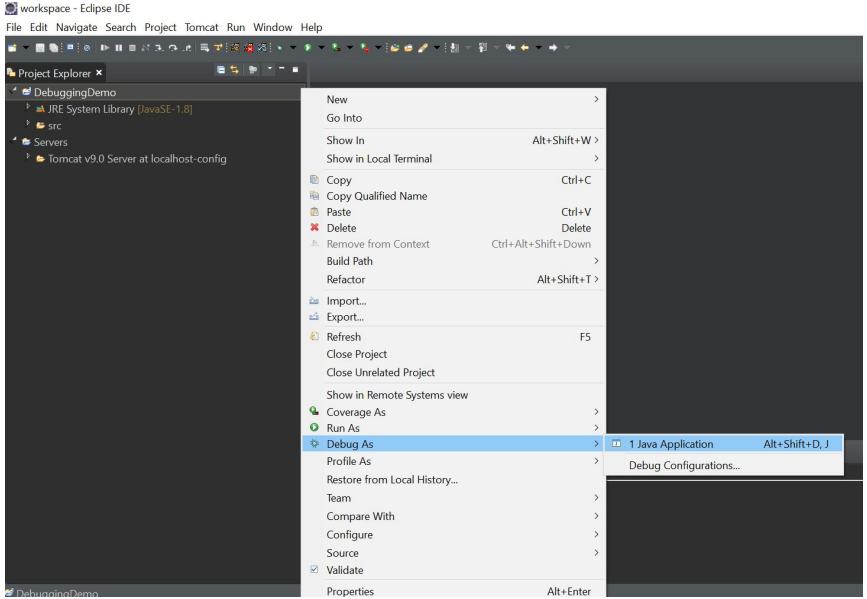
Eclipse Debugging Demo

Reference: Setting a Breakpoint



Right click on the line number indicator and set your breakpoint by clicking on "Toggle Breakpoint."

Reference: Debugging in Eclipse Cheat Sheet



Projects must be run in debug mode. Right click on your project or your entry point and select "Debug As"

Possible entry points:

- The class that has your main method.
- A unit test.

Reference: Accessing Step Controls

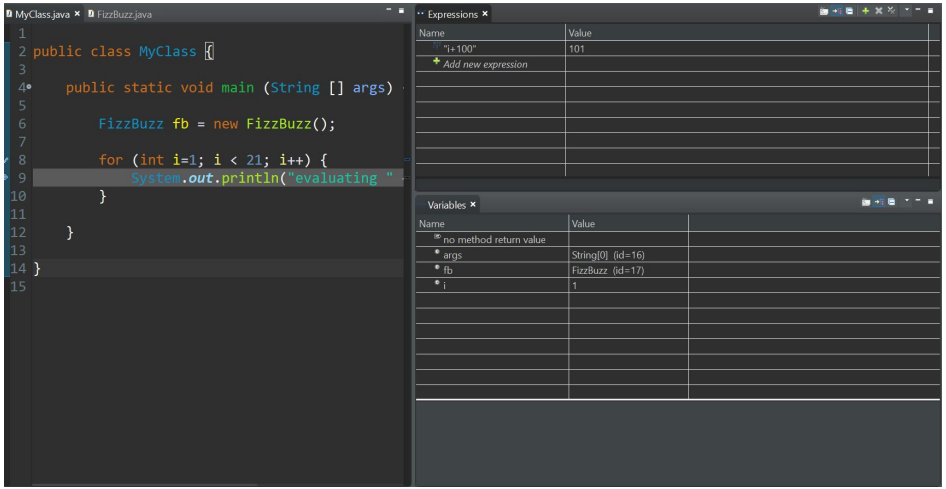
Resume

Step Into

Step Over

Step Return

Reference: Expressions & Variable Windows



You can turn on the variables or expression windows by going to:
Windows > Show View at the top and selecting them.

If you do not see them on the screen, Select:
WIndows > Show View > Other
You can then type and find Variables or Expressions.

Reading the Stack Demo

Debugging in Chrome

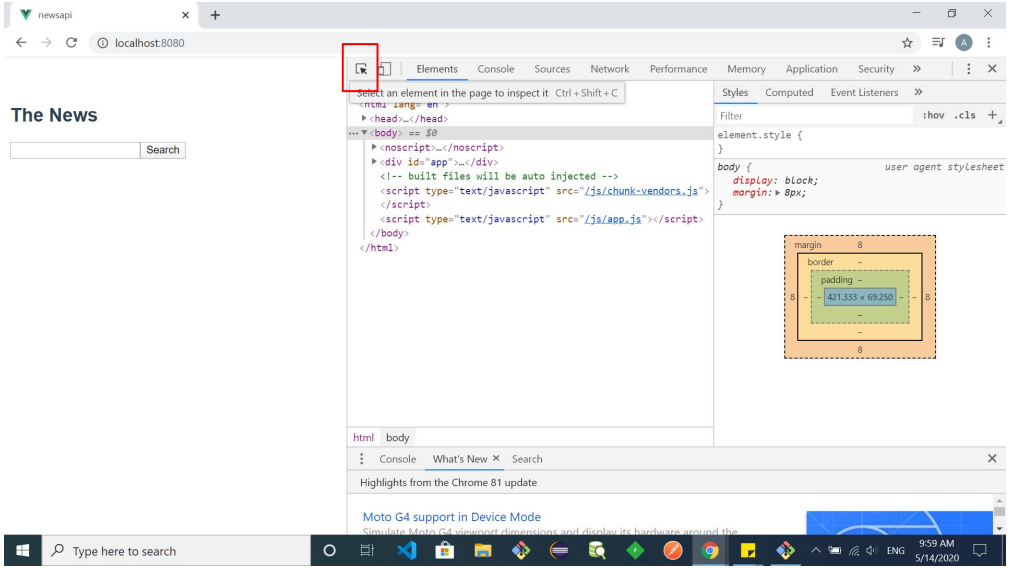
Chrome offers a very powerful set of tools to debug front end code, but today we will focus on just two:

- **Inspector**: To verify HTML elements.

- **Debugger**: To step through JS code. The vast majority of the concepts we discussed earlier with Eclipse carry over.

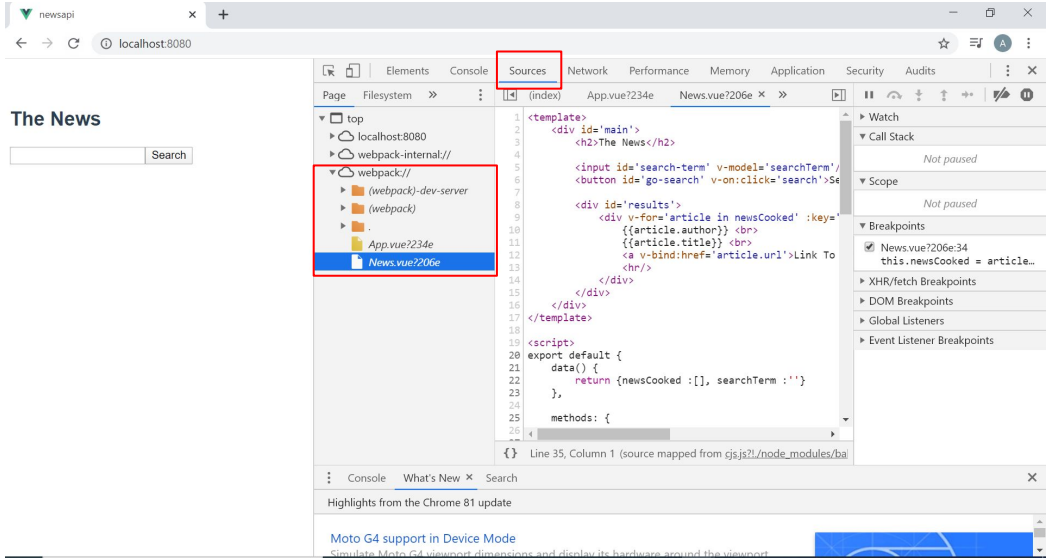
Chrome Debugging Demo

Reference: Inspector Tool



Press F12 to bring up the developer tools. The inspector can be accessed by clicking on the arrow against a square background.

Reference: Debugger



- *Press F12 to bring up the developer tools. The debugger can be accessed by clicking on the “Sources” tab.*
- *If you are using a framework like VUE, your component code will be under the webpacks section.*