CMSC335

Web Application Development with JavaScript

Debugging

Department of Computer Science

University of MD, College Park

Slides material developed by Ilchul Yoon, Nelson Padua-Perez

Debugging (Chrome)

- **Example:** Debugging.html
- **Step #1** - Right-click and select **Inspect** after loading the script
- **Step #2** - Select **Sources** from the menu (next to Console) and the file
  - You will see a page to the right with "Breakpoints", "Scope", "Call Stack", etc.
- **Step #3** - To set a breakpoint, click on a source line number to set a breakpoint
  - You will see the breakpoint set on the right pane and a thick blue line on the code. Unchecking a breakpoint on the pane leaves the breakpoint but disables it
- **Step #4** - To run the debugger, reload the script, and you will see the **light** blue line where the execution stopped
- **Step #5** - To examine local and global variables, expand the areas on the pane with the words "Local", "Script", and "Global"
  - For "Global" you will see **Window t**o the right
- **Step #6** - To resume, step over, step into, and step out, look to the top part of the right pane next to the code
- **Step #7** - To change between stack frames, select "Call Stack"

Debugging (Chrome) – debugger Statement

- You can add in your code the **debugger** statement, which will stop execution at that point
- To use the **debugger** statement
  - Add **debugger** to your code
  - Run the script
  - Right-click and select **Inspect**, and then **Sources**
  - Run the script
  - Execution will stop at the **debugger** statement
- **Example:** DebuggerStatement.html