Golden Rules of Debugging

Don't fix what isn't broken. Don't make random changes.

- Be sure that interactive prompts ask for appropriate input.
- If you don't understand how a bug or feature works, write a four or five line test program to understand what it does. If the overall program turns out to be based on mistakes then it is best to start over from the beginning. *Put the science back in computer science.*

When you're sure that everything you're doing is right, and your program still doesn't work, one of the things you're sure of is wrong.

Take a closer look:

- Make sure you close all HTML tags, { with }, (with) and /* with */.
- A complicated command that contains a control structure (e.g. nested ifs) or a compound statement (e.g. multiple and-or-not conditions) should be broken down into multiple procedures/functions/steps – try setting a variable/flag and set it using simple structures or logic.
- Don't forget semicolon after statements; it is a form of action.
- Don't make any assumptions about the input. Instead, look at your logic and check the values at different points. For example, if you skip over or get stuck in a loop, verify that the values are what you expect going into the loop, at the end of the loop and when you leave the loop.
- Make a stripped-down skeleton version of your program to check the overall logic and structure. Is it doing what you expect it to do?

Distinguish between Debugging and Testing

- *Debugging* is what you do before you consider a program complete.
- *Testing* is what a program user does to verify your program does what it is supposed to do. Unit testing is when you are the user, but remember, you know how you think. Your program needs to work with anyone.

Your cleverness is in making a working program, not obscuring your design.

You're not programming 80's robots with 4MB RAM limitations.

- The golden rule of space is that a program that works but uses lot of space (has many lines of code) is always better than the one that is uses few lines but doesn't work at all.