DEBUGGING IN Javascript

GIRLSCRIPT EDUCATION OUTREACH PROGRAM

TOPICS TO COVER:

1. What is Debugging?
2. Errors
3. Console
4. BreakPoints
5. Debugger Keyword and Window
6. Watches and Call Stack
7. Execution Keys
8. Easy way of debugging the code
9. Debugging Tools

What Is Debugging?

- Searching for (and fixing) errors in programming code is called code debugging.
- Debugger Provides the facility to go through the program at the runtime.
- Debugging is not easy. But fortunately, all modern browsers have a built-in JavaScript debugger.
- Best Way to solve a problem is to avoid them :)

Errors

EvalError :- An error has occurred in the eval() function (newer versions of JavaScript do not throw EvalError. Use SyntaxError instead.)

RangeError :- A number "out of range" has occurred

ReferenceError :- An illegal reference has occurred

SyntaxError :- A syntax error has occurred

TypeError :- A type error has occurred

URIError :- An error in encodeURI() has occurred

Console

`console.assert()`

Log a message and stack trace to console if the first argument is `false`.

`console.clear()`

Clear the console.

`console.debug()`

Outputs a message to the console with the log level `"debug"`.

`console.error()`

Outputs an error message.

`console.count()`

Log the number of times this line has been called with the given label.

`console.group()`

Creates a new inline group, indenting all following output by another level.

`console.log()`

For general output of logging information.

`console.table()`

Displays tabular data as a table.

`console.warn()`

Outputs a warning message.

BreakPoints

In the debugger window, you can set breakpoints in the JavaScript code for checking your code at runtime.

At each breakpoint, JavaScript will stop executing, and let you examine JavaScript values and you can debug your code at any point.

After examining values, you can resume the execution of code (typically with a play button at the upper side of the window).

‘DEBUGGER’ Keyword and Window

The debugger statement stops the execution of JavaScript, and calls (if available) the debugging function.

Using the debugger statement has the same function as setting a breakpoint in the code.

Watches And Call Stack

`Watch` – shows current values for any expressions.

`Call Stack` – shows the nested calls chain.

`Scope` – current variables.

`Local` shows local function variables.

`Global` has global variables (out of any functions).

Execution Keys

- Add Breakpoint of debugger
- Resume for continuing the execution
- step :- run the next command
- step over:- run next command but don’t go into a function
- Step into:- That’s similar to “Step”, but behaves differently in case of asynchronous function calls.
- Continue the execution till the end of the current function
- Enable of Disable Breakpoints
- Change the Values in console and check your debugged code.

Easy Way of Debugging The Code

Keep practicing to make you debugging skills more stronger

Debugging Tools

- Chrome Developer Tools (Best Tool ever)
- Firefox Developer Tools
- Safari Developer Menu
- Postman for debugging request and responses of API
- ESLint