

PHP - Bugs Debugging

A bug in a PHP code refers to an error in the program that leads to unexpected results or crash. A systematic approach towards the process of finding bugs before users do is called debugging. In this chapter, some important tips to trace bugs in a PHP code are given.

Programs rarely work correctly the first time. Many things can go wrong in your program that can cause the PHP interpreter to generate an error message. You have a choice about where those error messages go. The messages can be sent along with other program output to the web browser. They can also be included in the "web server error log".

To make error messages display in the browser, set the "display_errors" configuration directive to ON. Ensure that the following settings are enabled in the "php.ini" file.

```
display_errors=On  
display_startup_errors=On
```

You can also use the **ini_set() function** to override the "php.ini" configuration –

```
ini_set('display_errors', 1)  
ini_set('display_startup_errors', 1)
```

To send errors to the web server error log, set "log_errors" to ON. You can set them both to On if you want error messages in both places.

PHP defines some **constants** that you can use to set the value of **error_reporting** such that only errors of certain types get reported –

- E_ALL (for all errors except strict notices)
- E_PARSE (parse errors)
- E_ERROR (fatal errors)
- E_WARNING (warnings)
- E_NOTICE (notices)
- E_STRICT (strict notices)

While writing your PHP program, it is a good idea to use PHP-aware editors like BBEdit or Emacs. One of the special features of these editors is syntax highlighting. It changes the color of different parts of your program based on what those parts are. For example, strings are pink, keywords such as if and while are blue, comments are grey, and variables are black.

VS Code from Microsoft is also a good choice for editing PHP code. If you install VS Code extension Intelephense, you will get type hints and error message as you enter PHP statements in the editor window.

Another feature is quote and bracket matching, which helps to make sure that your quotes and brackets are balanced. When you type a closing delimiter such as "}", the editor highlights the opening "{" that it matches.

Points to Check while Debugging a Code

One needs to verify the following points while debugging a program code –

Missing Semicolons

Every PHP statement ends with a semicolon (;). PHP doesn't stop reading a statement until it reaches a semicolon. If you leave out the semicolon at the end of a line, PHP continues reading the statement on the following line.

Not Enough Equal Signs

When you ask whether two values are equal in a comparison statement, you need two equal signs (==). Using one equal sign is a common mistake.

Misspelled Variable Names

If you misspelled a variable then PHP understands it as a new variable. Remember: To PHP, \$test is not the same variable as \$Test.

Missing Dollar Signs

A missing dollar sign in a variable name is really hard to see, but at least it usually results in an error message so that you know where to look for the problem.

Troubling Quotes

You can have too many, too few, or the wrong kind of quotes. So check for a balanced number of quotes.

Missing Parentheses and curly brackets

They should always be in pairs.

Array Index

An array in PHP is a collection of items, each item assigned an incrementing index starting with 0.

Moreover, handle all the errors properly and direct all trace messages into system log file so that if any problem happens then it will be logged into system log file and you will be able to debug that problem.