

Chapter 6

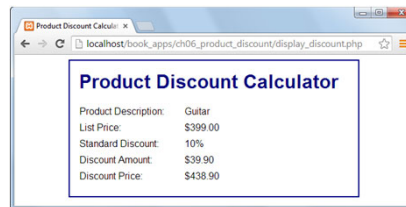
How to test and debug a PHP application



Murach's PHP and MySQL (3rd Ed.)

CE, Slide 1

The Discount application with a logic error



Murach's PHP and MySQL (3rd Ed.)

CE, Slide 4

Objectives

Applied

1. Test and debug your PHP applications.
2. Trace the execution of a PHP application with echo statements.
3. If you're using an IDE like NetBeans, set breakpoints, step through code, observe the changes in variables, and use the stack trace.



Murach's PHP and MySQL (3rd Ed.)

CE, Slide 2

The goal of testing

- To find all errors before the application is put into production.

The goal of debugging

- To fix all errors before the application is put into production.

Three test phases

- Check the user interface to make sure that it works correctly.
- Test the application with valid input data to make sure the results are correct.
- Test the application with invalid data or unexpected user actions. *Try to make the application fail.*



Murach's PHP and MySQL (3rd Ed.)

CE, Slide 5

Objectives (continued)

Knowledge

1. Distinguish between testing and debugging.
2. Distinguish between syntax, runtime, and logic errors.
3. Describe the use of breakpoints and stepping through code when you're using an IDE like NetBeans for debugging.



Murach's PHP and MySQL (3rd Ed.)

CE, Slide 3

The three types of errors that can occur

- syntax errors
- runtime errors
- logic errors



Murach's PHP and MySQL (3rd Ed.)

CE, Slide 6

PHP code that contains errors

```
// validate the list price entry
if ( $list_price = NULL || $list_price = FALSE ) {
    $error = "Please enter a valid number.";
} else {
    $error = ''
}
```

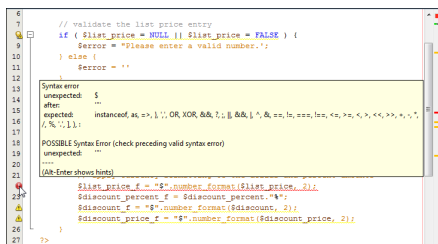


Problems with values

- Not checking that a value is the right data type before processing it.
- Using one equals sign instead of two when testing for equality.



The PHP code that contains errors in NetBeans



PHP with echo statements that trace the execution of the code

```
// calculate the future value
$future_value = $investment;
echo '$future_value: ' . $future_value . '<br>';
echo '$interest_rate: ' . $interest_rate . '<br>';
echo '$years: ' . $years . '<br>';
echo 'For loop for calculating future value is
starting...<br><br>';
for ($i = 1; $i <= $years; $i++) {
    $future_value += $future_value * $interest_rate;
    echo '$i: ' . $i . '<br>';
    echo '$future_value: ' . $future_value . '<br>';
}
```



Common syntax errors

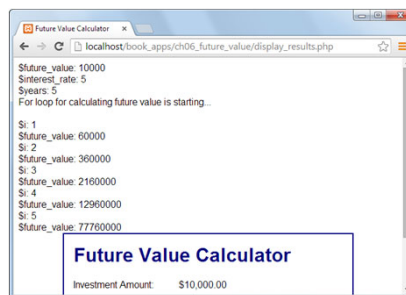
- Misspelling keywords.
- Forgetting an opening or closing parenthesis, bracket, brace, or comment character.
- Forgetting to end a PHP statement with a semicolon.
- Forgetting an opening or closing quotation mark.
- Not using the same opening and closing quotation mark.

Problems with variable names

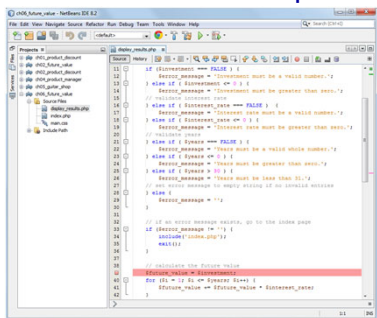
- Misspelling or incorrectly capitalizing a variable name.
- Using a keyword as a variable name.



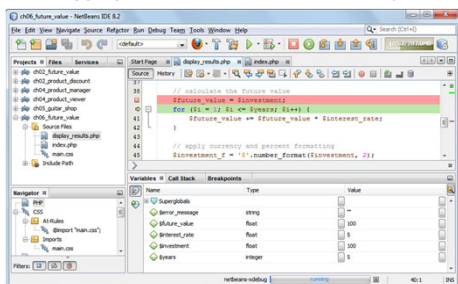
The data displayed in a browser



A code editor window with a breakpoint



A debugging session with variables displayed



A debugging session with a stack trace displayed

