

Chapter 6

How to test and debug a PHP application

Objectives

Applied

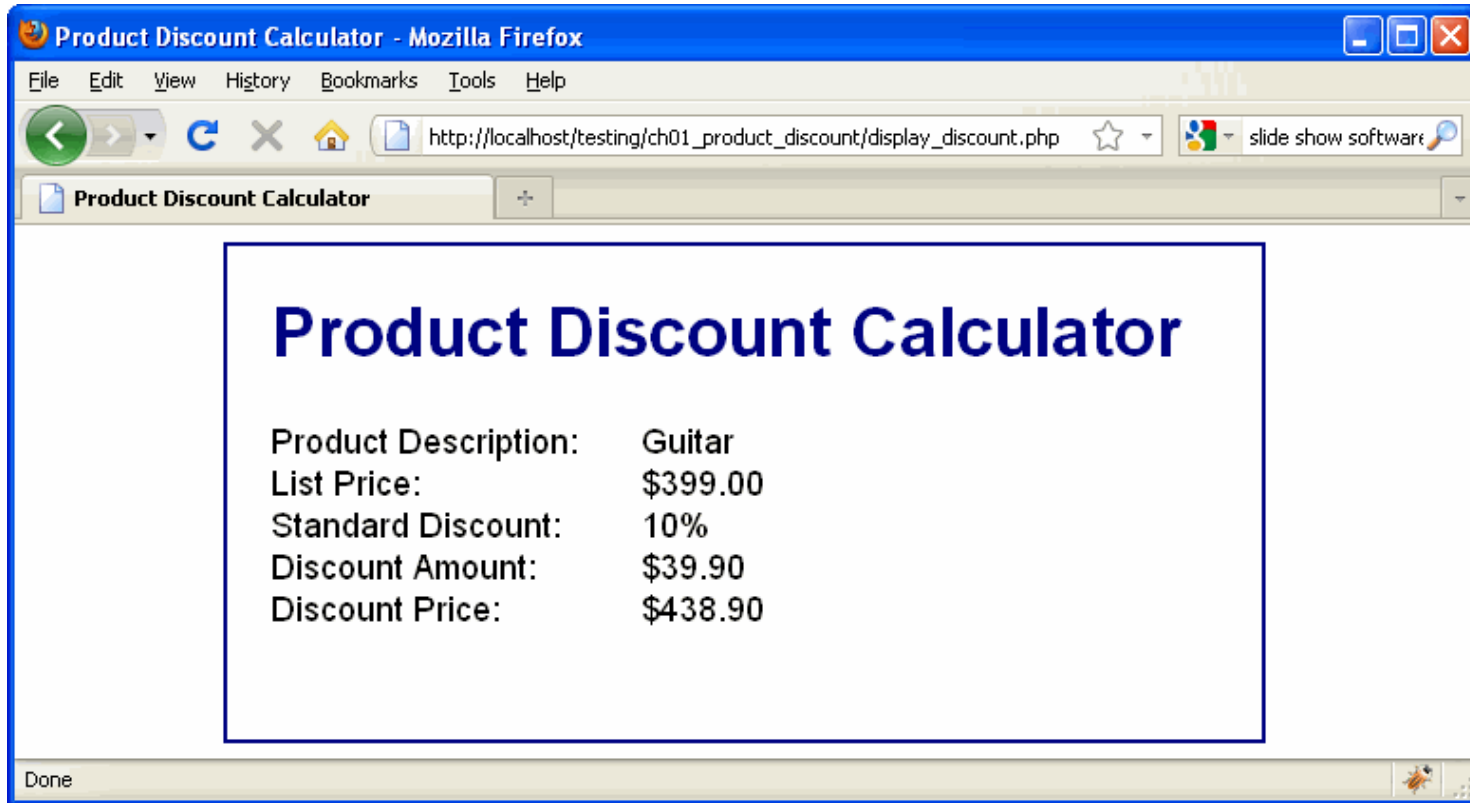
1. Test and debug your PHP applications.
2. Trace the execution of a PHP application with echo statements.
3. If you're using an IDE like NetBeans, set breakpoints, step through code, observe the changes in variables, and use the stack trace.

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.

The Discount application with a logic error



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.

The three types of errors that can occur

- syntax errors
- runtime errors
- logic errors

PHP code that contains errors

```
// validate the list price entry
if ( $list_price = '' ) {
    $error = "List price is a required field.";
} else if ( !is_numeric($list_price) ) {
    $error = 'List price must be a valid number.';
} else {
    $error = ''
}
}
```

The PHP code that contains errors in NetBeans

```
6
7 // validate the list price entry
8 if ( $list_price = '' ) {
9     $error = "List price is a required field.";
10 } else if ( !is_numeric($list_price) ) {
11     $error = 'List price must be a valid number.';
12 } else {
13     $error = ''
14 }
15
16 // calculate the discount
17 if ($error == '') {
18     $discount = $list_price * $discount_percent * 0.1;
19     $list_price_formatted = "$".number_format($list_price, 2);
20     $discount_percent_formatted = $discount_percent."%";
21     $discount_formatted = "$".number_format($discount, 2);
22     $discount_price_formatted = "$".number_format($discount_price, 2);
23 }
24 }
25
26
27 ?>
```

Syntax error:
expected: instanceof, as, =>, }, ', OR, XOR, AND, ?, ', ||, &&, |, ^, &, ==, !=, ===, !==, <=+, >=+, <, >, <<, >>, +, --
(Alt-Enter shows hints)

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.

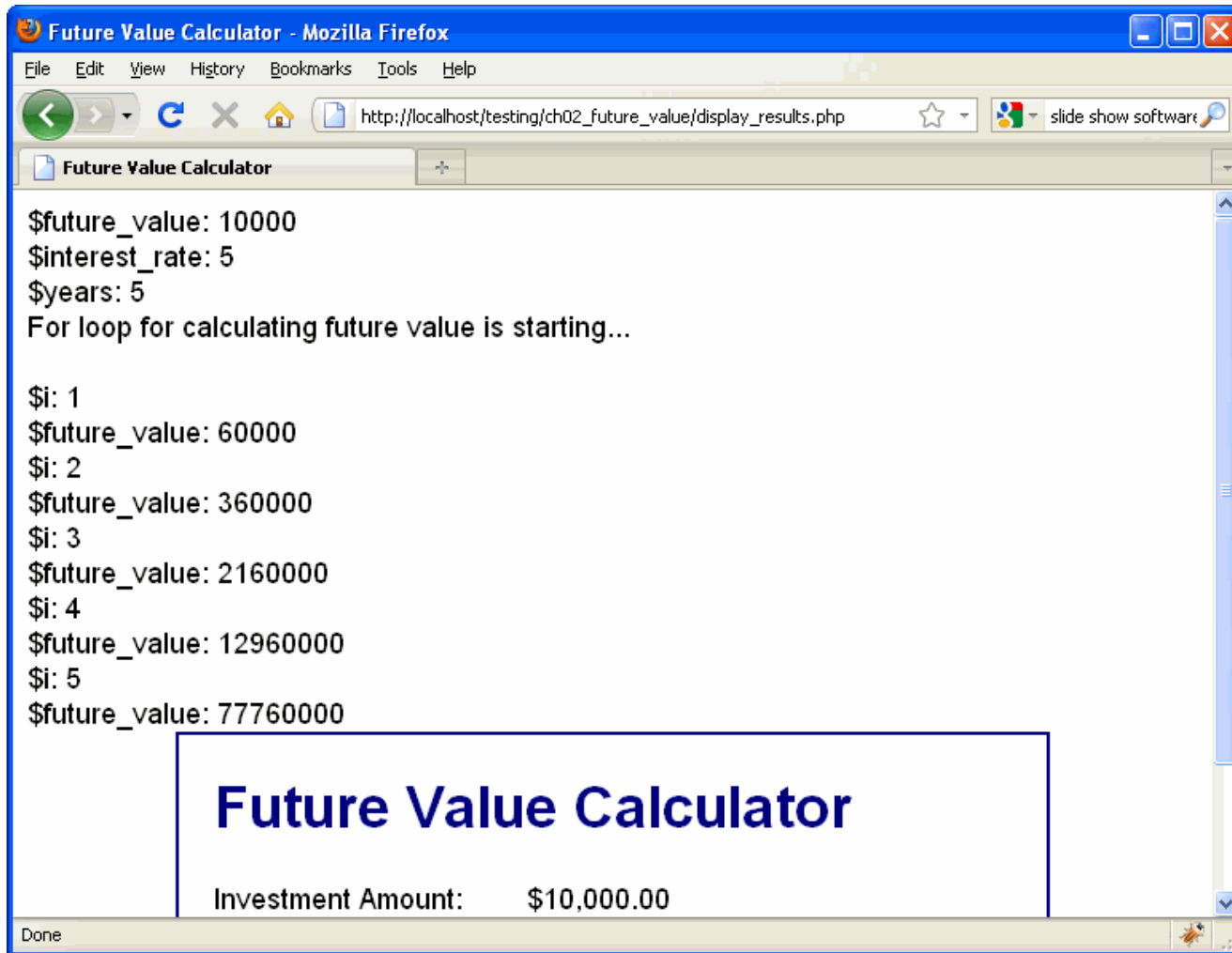
Problems with values

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

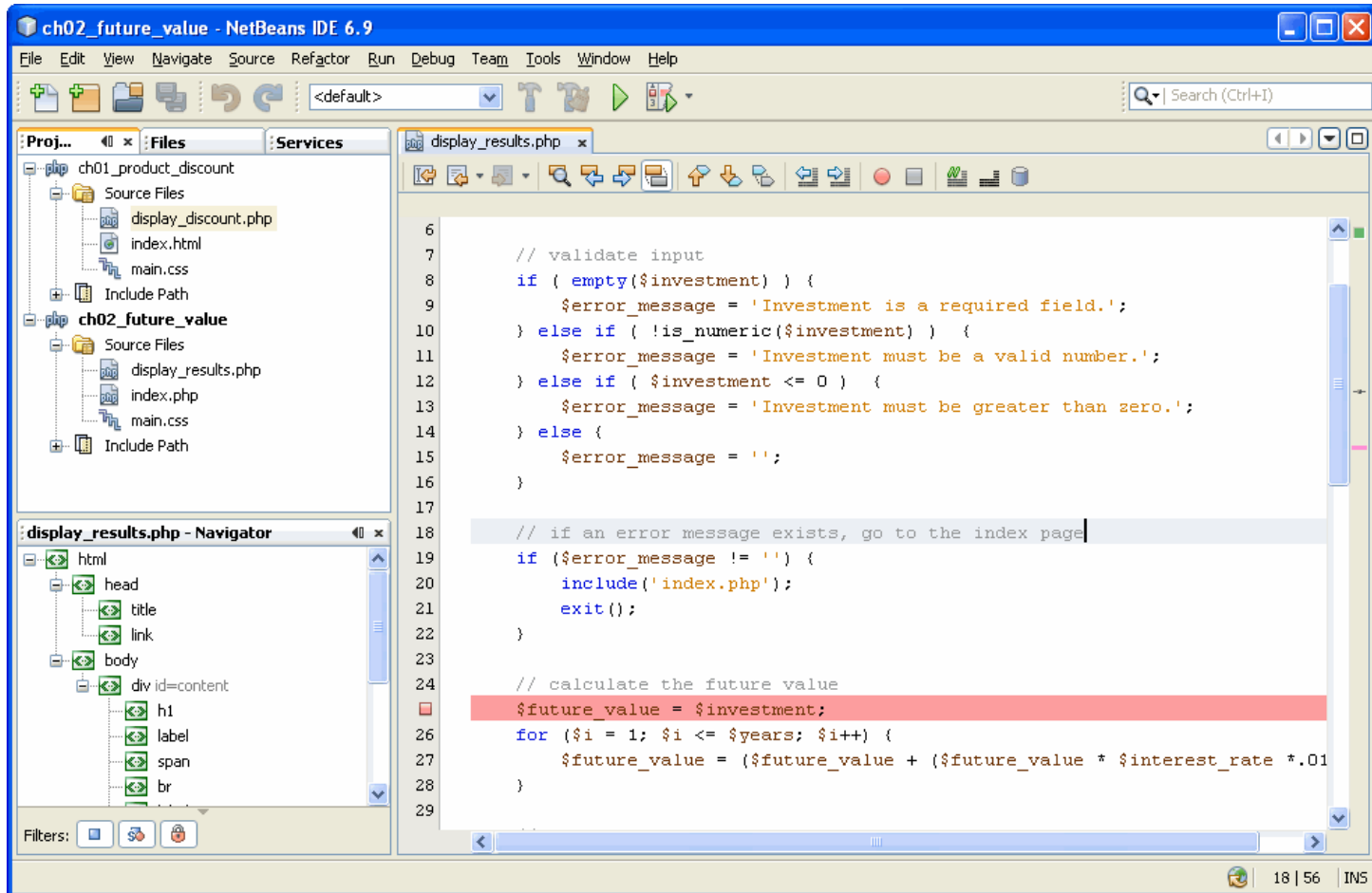
PHP with echo statements that trace the execution of the code

```
// 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 +
            ($future_value * $interest_rate));
    echo '$i: ' . $i . '<br />';
    echo '$future_value: ' . $future_value . '<br />';
}
```

The data displayed in a browser



A code editor window with a breakpoint



A debugging session with variables displayed

The screenshot shows the NetBeans IDE 6.9 interface during a debugging session. The main editor displays the following PHP code:

```
23
24 // calculate the future value
25 $future_value = $investment;
26 for ($i = 1; $i <= $years; $i++) {
27     $future_value = ($future_value + ($future_value * $interest_rate *.01
28 )
29 }
30 // apply currency and percent formatting
31 $investment_f = '$'.number_format($investment, 2);
32 $yearly_rate_f = $interest_rate.'%';
33 $future_value_f = '$'.number_format($future_value, 2);
```

The Variables window shows the following variables and their values:

Name	Type	Value
Superglobals		
error_message	string	
future_value	float	11576.25
i	integer	4
interest_rate	string	5
investment	string	10000
years	string	5

The Call Stack and Breakpoints windows are also visible at the bottom of the IDE.

A debugging session with a stack trace displayed

