

PHP Programming with My SQL : Gosselin (2nd Edition)

Best Coding Practices – Chapter 1

Topic	Best Practice
Formatting code	Format code consistently within a program Indent code five spaces for readability
Writing PHP code blocks	Use the Standard Script Delimiters <code><?php ... ?></code>
Using the echo statement	Use the <code>echo</code> construct exclusively Use lowercase characters for the <code>echo</code> keyword
Coding the echo statement	Do not use parentheses with the <code>echo</code> statement
Adding comments to PHP script	Add both line and block comments to your PHP code
Adding a line comment	Use the two forward slashes (<code>//</code>) to begin a line comment
Displaying array elements with built-in functions	Enclose the <code>print_r()</code> , <code>var_export()</code> , and <code>var_dump()</code> functions in beginning and ending <code><pre></code> tags

Best Coding Practices – Chapter 2

Topic	Best Practice
Formatting code	Format code consistently within a program Indent code five spaces for readability
Writing PHP code blocks	Use the Standard Script Delimiters <code><?php ... ?></code>
Using the echo statement	Use the <code>echo</code> construct exclusively Use lowercase characters for the <code>echo</code> keyword
Coding the echo statement	Do not use parentheses with the <code>echo</code> statement
Adding comments to PHP script	Add both line and block comments to your PHP code
Adding a line comment	Use the two forward slashes (<code>//</code>) to begin a line comment
Displaying array elements with built-in functions	Enclose the <code>print_r()</code> , <code>var_export()</code> , and <code>var_dump()</code> functions in beginning and ending <code><pre></code> tags
Writing a user-defined function	The name of a user-defined function should be descriptive of the task it will perform
Writing a function definition	Do NOT space between the function name and the opening parenthesis Key the opening curly brace on the same line as the function name Indent the function statements five spaces Key the closing curly brace on a separate line
Calling a function	Place the function definition above the calling statement
Writing a control structure	To differentiate a control structure from a function, space once after the conditional keyword before the parenthesis i.e. <code>if (...)</code> or <code>else (...)</code> Indent the statements to make them easier for the programmer to follow the flow Use curly braces around the statements to execute in a control structure if there is more than one statement.
Coding a switch statement	Include the optional <code>break</code> statement after the default case label in a <code>switch</code> statement
Naming a counter variable	If appropriate, name your counter variable <code>\$Count</code> or <code>\$Counter</code> or use the letters (<code>i, j, k, l, x, y, z</code>)

Best Coding Practices – Chapter 3

Topic	Best Practice
Formatting code	Format code consistently within a program Indent code five spaces for readability
Writing PHP code blocks	Use the Standard Script Delimiters <code><?php ... ?></code>
Using the echo statement	Use the <code>echo</code> construct exclusively Use lowercase characters for the <code>echo</code> keyword
Coding the echo statement	Do not use parentheses with the <code>echo</code> statement
Adding comments to PHP script	Add both line and block comments to your PHP code
Adding a line comment	Use the two forward slashes (<code>//</code>) to begin a line comment
Displaying array elements with built-in functions	Enclose the <code>print_r()</code> , <code>var_export()</code> , and <code>var_dump()</code> functions in beginning and ending <code><pre></code> tags
Writing a user-defined function	The name of a user-defined function should be descriptive of the task it will perform
Writing a function definition	Do NOT space between the function name and the opening parenthesis Key the opening curly brace on the same line as the function name Indent the function statements five spaces Key the closing curly brace on a separate line
Calling a function	Place the function definition above the calling statement
Writing a control structure	To differentiate a control structure from a function, space once after the conditional keyword before the parenthesis i.e. <code>if (...)</code> or <code>else (...)</code> Indent the statements to make them easier for the programmer to follow the flow Use curly braces around the statements to execute in a control structure if there is more than one statement.
Coding a switch statement	Include the optional <code>break</code> statement after the default case label in a <code>switch</code> statement
Naming a counter variable	If appropriate, name your counter variable <code>\$Count</code> or <code>\$Counter</code> or use the letters (<code>i, j, k, l, x, y, z</code>)
Ending an echo statement	Add a <code>\n</code> escape sequence at the end of each <code>echo</code> statement

Best Coding Practices – Chapter 4

Topic	Best Practice
Formatting code	Format code consistently within a program Indent code five spaces for readability
Writing PHP code blocks	Use the Standard Script Delimiters <code><?php ... ?></code>
Using the echo statement	Use the <code>echo</code> construct exclusively Use lowercase characters for the <code>echo</code> keyword
Coding the echo statement	Do not use parentheses with the <code>echo</code> statement
Adding comments to PHP script	Add both line and block comments to your PHP code
Adding a line comment	Use the two forward slashes (<code>//</code>) to begin a line comment
Displaying array elements with built-in functions	Enclose the <code>print_r()</code> , <code>var_export()</code> , and <code>var_dump()</code> functions in beginning and ending <code><pre></code> tags
Writing a user-defined function	The name of a user-defined function should be descriptive of the task it will perform

Writing a function definition	Do NOT space between the function name and the opening parenthesis Key the opening curly brace on the same line as the function name Indent the function statements five spaces Key the closing curly brace on a separate line
Calling a function	Place the function definition above the calling statement
Writing a control structure	To differentiate a control structure from a function, space once after the conditional keyword before the parenthesis i.e. <code>if (...) or else (...)</code> Indent the statements to make them easier for the programmer to follow the flow Use curly braces around the statements to execute in a control structure if there is more than one statement.
Coding a switch statement	Include the optional <code>break</code> statement after the default case label in a <code>switch</code> statement
Naming a counter variable	If appropriate, name your counter variable <code>\$Count</code> or <code>\$Counter</code> or use the letters (<code>i, j, k, l, x, y, z</code>)
Ending an echo statement	Add a <code>\n</code> escape sequence at the end of each <code>echo</code> statement
Submitting user input	Use “post” as the method to submit form data
Submitting a form to itself	Use <code>\$_SERVER["SCRIPT_NAME"]</code> to submit a form to the current script
Displaying dynamic content based on a URL token	Use the <code>\$_GET</code> autoglobal to retrieve hyperlink data

Best Coding Practices – Chapter 5

Topic	Best Practice
Formatting code	Format code consistently within a program Indent code five spaces for readability
Writing PHP code blocks	Use the Standard Script Delimiters <code><?php ... ?></code>
Using the echo statement	Use the <code>echo</code> construct exclusively Use lowercase characters for the <code>echo</code> keyword
Coding the echo statement	Do not use parentheses with the <code>echo</code> statement
Adding comments to PHP script	Add both line and block comments to your PHP code
Adding a line comment	Use the two forward slashes (<code>//</code>) to begin a line comment
Displaying array elements with built-in functions	Enclose the <code>print_r()</code> , <code>var_export()</code> , and <code>var_dump()</code> functions in beginning and ending <code><pre></code> tags
Writing a user-defined function	The name of a user-defined function should be descriptive of the task it will perform
Writing a function definition	Do NOT space between the function name and the opening parenthesis Key the opening curly brace on the same line as the function name Indent the function statements five spaces Key the closing curly brace on a separate line
Calling a function	Place the function definition above the calling statement
Writing a control structure	To differentiate a control structure from a function, space once after the conditional keyword before the parenthesis i.e. <code>if (...) or else (...)</code> Indent the statements to make them easier for the programmer to follow the flow Use curly braces around the statements to execute in a control structure if there is more than one statement.

Coding a switch statement	Include the optional <code>break</code> statement after the default case label in a <code>switch</code> statement
Naming a counter variable	If appropriate, name your counter variable <code>\$Count</code> or <code>\$Counter</code> or use the letters (<code>i</code> , <code>j</code> , <code>k</code> , <code>l</code> , <code>x</code> , <code>y</code> , <code>z</code>)
Ending an <code>echo</code> statement	Add a <code>\n</code> escape sequence at the end of each <code>echo</code> statement
Submitting user input	Use “post” as the method to submit form data
Submitting a form to itself	Use <code>\$_SERVER["SCRIPT_NAME"]</code> to submit a form to the current script
Displaying dynamic content based on a URL token	Use the <code>\$_GET</code> autoglobal to retrieve hyperlink data

Best Coding Practices – Chapter 6

Topic	Best Practice
Formatting code	Format code consistently within a program Indent code five spaces for readability
Writing PHP code blocks	Use the Standard Script Delimiters <code><?php ... ?></code>
Using the <code>echo</code> statement	Use the <code>echo</code> construct exclusively Use lowercase characters for the <code>echo</code> keyword
Coding the <code>echo</code> statement	Do not use parentheses with the <code>echo</code> statement
Adding comments to PHP script	Add both line and block comments to your PHP code
Adding a line comment	Use the two forward slashes (<code>//</code>) to begin a line comment
Displaying array elements with built-in functions	Enclose the <code>print_r()</code> , <code>var_export()</code> , and <code>var_dump()</code> functions in beginning and ending <code><pre></code> tags
Writing a user-defined function	The name of a user-defined function should be descriptive of the task it will perform
Writing a function definition	Do NOT space between the function name and the opening parenthesis Key the opening curly brace on the same line as the function name Indent the function statements five spaces Key the closing curly brace on a separate line
Calling a function	Place the function definition above the calling statement
Writing a control structure	To differentiate a control structure from a function, space once after the conditional keyword before the parenthesis i.e. <code>if (...)</code> or <code>else (...)</code> Indent the statements to make them easier for the programmer to follow the flow Use curly braces around the statements to execute in a control structure if there is more than one statement.
Coding a switch statement	Include the optional <code>break</code> statement after the default case label in a <code>switch</code> statement
Naming a counter variable	If appropriate, name your counter variable <code>\$Count</code> or <code>\$Counter</code> or use the letters (<code>i</code> , <code>j</code> , <code>k</code> , <code>l</code> , <code>x</code> , <code>y</code> , <code>z</code>)
Ending an <code>echo</code> statement	Add a <code>\n</code> escape sequence at the end of each <code>echo</code> statement
Submitting user input	Use “post” as the method to submit form data
Submitting a form to itself	Use <code>\$_SERVER["SCRIPT_NAME"]</code> to submit a form to the current script
Displaying dynamic content based on a URL token	Use the <code>\$_GET</code> autoglobal to retrieve hyperlink data

Best Coding Practices – Chapter 7

Topic	Best Practice
Formatting code	Format code consistently within a program Indent code five spaces for readability
Writing PHP code blocks	Use the Standard Script Delimiters <code><?php ... ?></code>
Using the <code>echo</code> statement	Use the <code>echo</code> construct without parentheses Use lowercase characters for the <code>echo</code> keyword
Coding the <code>echo</code> statement	Do not use parentheses with the <code>echo</code> statement
Adding comments to PHP script	Add both line and block comments to your PHP code
Adding a line comment	Use the two forward slashes (<code>//</code>) to begin a line comment Do not use the <code>#</code> line comment
Displaying array elements with built-in print functions	Enclose the <code>print_r()</code> , <code>var_export()</code> , and <code>var_dump()</code> functions in beginning and ending <code><pre></code> tags.
Naming a function	Name the function something that represents its purpose
Writing a function definition	Do NOT space between the function name and the opening parentheses Key the opening curly brace on the same line as the function name Indent the function statements five spaces Key the closing curly brace on a separate line
Calling a function	Place the function definition above the calling statement
Writing a control structure	Space once after the conditional keyword before the parentheses i.e. <code>if (...)</code> or <code>else (...)</code>
Naming a counter variable	Name your counter variable <code>\$Count</code> or <code>\$Counter</code> or use the letters (<code>i, j, k, l, x, y, z</code>)
Coding a switch statement	Include the optional break statement after the default case label in a <code>switch</code> statement
Ending an <code>echo</code> statement	Add a <code>\n</code> escape sequence at the end of each <code>echo</code> statement
Submitting user input	Use “post” as the method to submit form data
Submitting a form to itself	Use <code>\$_SERVER["SCRIPT_NAME"]</code> to submit a form to the current script
Displaying dynamic content based on a URL token	Use the “get” method to retrieve hyperlink data
Keying MySQL commands	Key all SQL keywords in uppercase characters
Inserting values in a table column	Use the <code>COLUMNS</code> and <code>VALUES</code> syntax to insert data in a table

Best Coding Practices – Chapter 8

Topic	Best Practice
Formatting code	Format code consistently within a program Indent code five spaces for readability
Writing PHP code blocks	Use the Standard Script Delimiters <code><?php ... ?></code>
Using the <code>echo</code> statement	Use the <code>echo</code> construct without parentheses Use lowercase characters for the <code>echo</code> keyword
Coding the <code>echo</code> statement	Do not use parentheses with the <code>echo</code> statement
Adding comments to PHP script	Add both line and block comments to your PHP code
Adding a line comment	Use the two forward slashes (<code>//</code>) to begin a line comment Do not use the <code>#</code> line comment

Displaying array elements with built-in print functions	Enclose the <code>print_r()</code> , <code>var_export()</code> , and <code>var_dump()</code> functions in beginning and ending <code><pre></code> tags.
Naming a function	Name the function something that represents its purpose
Writing a function definition	Do NOT space between the function name and the opening parentheses Key the opening curly brace on the same line as the function name Indent the function statements five spaces Key the closing curly brace on a separate line
Calling a function	Place the function definition above the calling statement
Writing a control structure	Space once after the conditional keyword before the parentheses i.e. <code>if (...)</code> or <code>else (...)</code>
Naming a counter variable	Name your counter variable <code>\$Count</code> or <code>\$Counter</code> or use the letters (<code>i, j, k, l, x, y, z</code>)
Coding a switch statement	Include the optional break statement after the default case label in a switch statement
Ending an echo statement	Add a <code>\n</code> escape sequence at the end of each echo statement
Submitting user input	Use “post” as the method to submit form data
Submitting a form to itself	Use <code>\$_SERVER["SCRIPT_NAME"]</code> to submit a form to the current script
Displaying dynamic content based on a URL token	Use the “get” method to retrieve hyperlink data
Keying MySQL commands	Key all SQL keywords in uppercase characters
Inserting values in a table column	Use the COLUMNS and VALUES syntax to insert data in a table
Connecting to the server	Insert an <i>include</i> statement containing the connection information when connecting to the server rather than keying the connection information in the PHP script Save the <i>include</i> file above the www directory on the server
Handling an unsuccessful connection	Do NOT use the <code>die()</code> function to exit the script for unsuccessful connections; instead, use an <code>echo()</code> statement to notify the user
Displaying error messages	Do NOT use the <code>mysql_error()</code> function in production scripts

Best Coding Practices – Chapter 9

Topic	Best Practice
Formatting code	Format code consistently within a program Indent code five spaces for readability
Writing PHP code blocks	Use the Standard Script Delimiters <code><?php ... ?></code>
Using the echo statement	Use the echo construct without parentheses Use lowercase characters for the echo keyword
Coding the echo statement	Do not use parentheses with the echo statement
Adding comments to PHP script	Add both line and block comments to your PHP code
Adding a line comment	Use the two forward slashes (<code>//</code>) to begin a line comment Do not use the <code>#</code> line comment
Displaying array elements with built-in print functions	Enclose the <code>print_r()</code> , <code>var_export()</code> , and <code>var_dump()</code> functions in beginning and ending <code><pre></code> tags.
Naming a function	Name the function something that represents its purpose

Writing a function definition	Do NOT space between the function name and the opening parentheses Key the opening curly brace on the same line as the function name Indent the function statements five spaces Key the closing curly brace on a separate line
Calling a function	Place the function definition above the calling statement
Writing a control structure	Space once after the conditional keyword before the parentheses i.e. <code>if (...)</code> or <code>else (...)</code>
Naming a counter variable	Name your counter variable <code>\$Count</code> or <code>\$Counter</code> or use the letters (i, j, k, l, x, y, z)
Coding a switch statement	Include the optional break statement after the default case label in a switch statement
Ending an echo statement	Add a <code>\n</code> escape sequence at the end of each echo statement
Submitting user input	Use “post” as the method to submit form data
Submitting a form to itself	Use <code>\$_SERVER["SCRIPT_NAME"]</code> to submit a form to the current script
Displaying dynamic content based on a URL token	Use the “get” method to retrieve hyperlink data
Keying MySQL commands	Key all SQL keywords in uppercase characters
Inserting values in a table column	Use the COLUMNS and VALUES syntax to insert data in a table
Connecting to the server	Insert an <i>include</i> statement containing the connection information when connecting to the server rather than keying the connection information in the PHP script Save the <i>include</i> file above the www directory on the server
Handling an unsuccessful connection	Do NOT use the <code>die()</code> function to exit the script for unsuccessful connections; instead, use an <code>echo()</code> statement to notify the user
Displaying error messages	Do NOT use the <code>mysql_error()</code> function in production scripts
Initiating a session	Insert the <code>session_start()</code> function directly after the opening script delimiters on each page that requires sessions

PHP Best Coding Practices – Chapter 10

Topic	Best Practice
Formatting code	Format code consistently within a program Indent code five spaces for readability
Writing PHP code blocks	Use the Standard Script Delimiters <code><?php ... ?></code>
Using the echo statement	Use the echo construct without parentheses Use lowercase characters for the echo keyword
Coding the echo statement	Do not use parentheses with the echo statement
Adding comments to PHP script	Add both line and block comments to your PHP code
Adding a line comment	Use the two forward slashes (<code>//</code>) to begin a line comment Do not use the <code>#</code> line comment
Displaying array elements with built-in print functions	Enclose the <code>print_r()</code> , <code>var_export()</code> , and <code>var_dump()</code> functions in beginning and ending <code><pre></code> tags.
Naming a function	Name the function something that represents its purpose
Writing a function definition	Do NOT space between the function name and the opening parentheses Key the opening curly brace on the same line as the function name

	Indent the function statements five spaces Key the closing curly brace on a separate line
Calling a function	Place the function definition above the calling statement
Writing a control structure	Space once after the conditional keyword before the parentheses i.e. <code>if (...) or else (...)</code>
Naming a counter variable	Name your counter variable <code>\$Count</code> or <code>\$Counter</code> or use the letters (<code>i, j, k, l, x, y, z</code>)
Coding a switch statement	Include the optional break statement after the default case label in a switch statement
Ending an echo statement	Add a <code>\n</code> escape sequence at the end of each echo statement
Submitting user input	Use “post” as the method to submit form data
Submitting a form to itself	Use <code>\$_SERVER["SCRIPT_NAME"]</code> to submit a form to the current script
Displaying dynamic content based on a URL token	Use the “get” method to retrieve hyperlink data
Keying MySQL commands	Key all SQL keywords in uppercase characters
Inserting values in a table column	Use the COLUMNS and VALUES syntax to insert data in a table
Connecting to the server	Insert an <i>include</i> statement containing the connection information when connecting to the server rather than keying the connection information in the PHP script Save the <i>include</i> file above the www directory on the server
Handling an unsuccessful connection	Do NOT use the <code>die()</code> function to exit the script for unsuccessful connections; instead, use an <code>echo()</code> statement to notify the user
Displaying error messages	Do NOT use the <code>mysql_error()</code> function in production scripts
Initiating a session	Insert the <code>session_start()</code> function directly after the opening script delimiters on each page that requires sessions