Grade Checker Source code in PHP

Are you a student who's curious about how to evaluate your grades in a subject? Or maybe you're a beginner programmer who's looking to learn more about conditional statements in PHP? Either way, you're in the right place! In this article, we'll take a closer look at a Grade Checker source code written in PHP that allows you to input your grade and receive feedback on your performance in a subject. We'll also dive into the if-else-if statement, a fundamental construct in programming that the code uses to evaluate the grade value and generate feedback messages based on multiple conditions. By the end of this article, you'll have a better understanding of how to use if-else-if statements in PHP and be equipped with the skills to create your own Grade Checker program. Let's get started!

Introduction

The source code presented in this post is a simple HTML form that allows users to input their grade and receive feedback on their performance in a subject. The form is powered by PHP scripting, which evaluates the input value and generates a feedback message based on the range of the grade. The code demonstrates the use of conditional statements to control the program flow based on a set of conditions.

In particular, the code uses an if-else-if statement to evaluate the grade value and determine the feedback message to display. An if-else-if statement is a programming construct that allows the program to perform different actions based on multiple conditions. The first if statement evaluates a condition, and if it is true, executes a block of code. If the condition is false, the program moves to the next elseif statement, which evaluates a new condition. If none of the conditions evaluate to true, the program executes an else statement or skips the block altogether.

The if-else-if statement is a fundamental construct in programming, and it's used extensively in various programming languages, including PHP, Java, C++, and Python. It's essential to understand the syntax and structure of the if-else-if statement to write efficient and robust programs. By mastering the if-else-if statement, developers can control program flow, handle errors and exceptions, and make complex decisions based on multiple conditions.

Objectives

Outcome-based objective for this lesson on if-else-if statements:

By the end of this lesson, the learner will be able to:

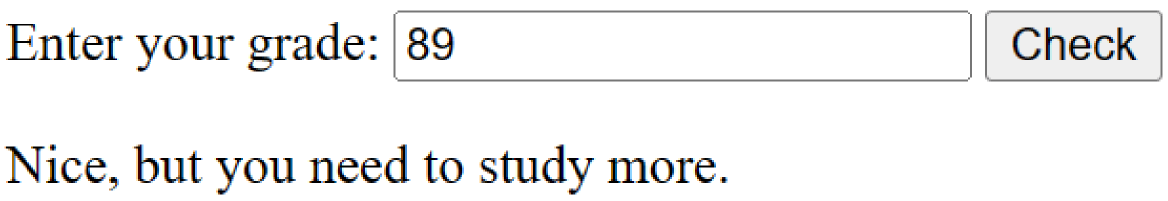
1. Understand the syntax and structure of the if-else-if statement.
2. Identify situations where the if-else-if statement is appropriate to use.
3. Write code that uses if-else-if statements to make complex decisions based on multiple conditions.

4. Apply best practices for using if-else-if statements, such as using logical operators, optimizing the order of the conditions, and handling edge cases.

Code Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Grade Checker</title>
</head>
<body>
    <form method="post">
        <label for="grade">Enter your grade:</label>
        <input type="number" id="grade" name="grade" required>
        <input type="submit" name="submit" value="Check">
    </form>
    <?php
        if(isset($_POST["submit"])) {
            $grade = $_POST["grade"];
            if($grade >= 90 && $grade <= 100) {
                echo "<p>Very Good</p>";
            } elseif($grade >= 80 && $grade <= 89) {
                echo "<p>Nice, but you need to study more.</p>";
            } elseif($grade >= 75 && $grade <= 79) {
                echo "<p>You barely pass the subject!</p>";
            } elseif($grade < 75) {
                echo "<p>You FAILED, exert more effort next time.</p>";
            }
        }
    ?>
</body>
</html>
```

Output



Enter your grade: 89 Check

Nice, but you need to study more.

Code Explanation

This is an HTML form that allows users to input their grade and receive feedback on their performance in a subject. Here's a breakdown of the code:

<!DOCTYPE html>: This declares the document type as HTML5.

<html>: This opening tag marks the beginning of the HTML document.

<head>: This opening tag marks the beginning of the document's head section, where metadata and other non-visible information about the document is defined.

<title>Grade Checker</title>: This specifies the title of the webpage, which appears in the browser's title bar.

</head>: This closing tag marks the end of the document's head section.

<body>: This opening tag marks the beginning of the document's body section, where visible content is defined.

<form method="post">: This opening tag starts a form, which allows users to input data. The method attribute specifies that the form data should be submitted using HTTP POST method.

<label for="grade">Enter your grade:</label>: This creates a label for the grade input field, making it more accessible.

<input type="number" id="grade" name="grade" required>: This creates an input field for the grade value. The type attribute is set to "number" to indicate that only numeric input is allowed. The id attribute sets a unique identifier for the input field, while the name attribute sets a name for the input field. The required attribute specifies that the input field must be filled out before the form can be submitted.

<input type="submit" name="submit" value="Check">: This creates a submit button for the form. The name attribute sets a name for the button, while the value attribute specifies the text that appears on the button.

</form>: This closing tag ends the form.

<?php ... ?>: This is a PHP code block, which allows server-side scripting to be embedded in the HTML document.

if(isset($_POST["submit"])): This checks if the submit button has been clicked and the form has been submitted.

$grade = $_POST["grade"];: This retrieves the value of the grade input field that was submitted through the form and assigns it to a variable called $grade.

if($grade >= 90 && $grade <= 100) { ... }: This checks the value of $grade and prints feedback based on the grade range. If the grade is between 90 and 100, the message "Very Good" is displayed. The elseif and else conditions check for other grade ranges and display different feedback messages accordingly.

echo "<p>...</p>";: This prints the feedback message within a paragraph element. The message is enclosed within double quotes, and any variables or HTML tags are concatenated using the dot operator.

</body>: This closing tag marks the end of the document's body section.

</html>: This closing tag marks the end of the HTML document.

Summary

In this lesson, we explored the Grade Checker program written in PHP, which evaluates a student's grade input and generates feedback messages based on multiple conditions using the if-else-if statement. We discussed the syntax and structure of the if-else-if statement and how it allows the program to perform different actions based on multiple conditions. We also identified best practices for using if-else-if statements, such as optimizing the order of the conditions and handling edge cases.

Here's a summary of the Grade Checker program in PHP:

- The program is a simple HTML form that allows users to input their grade and receive feedback on their performance in a subject.
- The form is powered by PHP scripting, which evaluates the input value and generates a feedback message based on the range of the grade.
- The code uses an if-else-if statement to evaluate the grade value and determine the feedback message to display.
- The if-else-if statement evaluates multiple conditions and executes different blocks of code based on the condition that evaluates to true.

By studying the Grade Checker program in PHP, learners can gain a better understanding of how to use if-else-if statements to make complex decisions based on multiple conditions.