

# INLS 672

# Web Development 2

## PHP Conditional Statements



Joan Boone  
[jpboone@email.unc.edu](mailto:jpboone@email.unc.edu)

# Program Flow of Control

*Flow of control* is the order in which a program executes statements

- The default order of execution is sequential
- Each statement is executed in the order it appears in the program

Two ways that program flow can be altered

- Conditional statements that choose between 2 or more possible actions. Deciding which action to take is typically based on the result of a comparison or logical operator
- Looping (or iteration) statements perform an action until a stopping condition occurs

# Comparison Operators

```
==   ===   !=   <>   !==   >   <   >=   <=
```

## References

- PHP Manual: [Comparison Operators](#)
- w3schools: [PHP Operators](#)

# Logical Operators

```
and   or   xor   &&   ||   !
```

## References

- PHP Manual: [Logical Operators](#)
- w3Schools: [PHP Operators](#)

# Conditional Statement: **if...else**

Use curly brackets/braces **{...}** to enclose multiple statements for **if...else** conditions

Boolean expression that evaluates to true or false

```
if ($balance > 0) {  
    echo "You have earned interest";  
    $balance = $balance + ($interest_rate * $balance) / 12;  
}  
  
else {  
    echo "You will be charged a penalty.";  
    $balance = $balance - $overdrawn_penalty;  
}  
  
echo "Your new balance is $" . $balance;
```

# Conditional Statement: **if...elseif...else**

```
if ($balance > 0) {  
    echo "You have earned interest.";  
    $balance = $balance + ($interest_rate * $balance) / 12;  
}  
  
elseif ($balance == 0) {  
    echo "You need to deposit money ASAP!";  
}  
  
else {  
    echo "You will be charged a penalty.";  
    $balance = $balance - $overdrawn_penalty;  
}  
  
echo "Your new balance is $" . $balance;
```

An alternative is the **switch** statement that selects one of many blocks of code to be executed.

# Conditional Statement: **switch**

**if..elseif..else** version

```
if ($page == "Home")
    echo "You selected Home";

elseif ($page == "About")
    echo "You selected About";

elseif ($page == "News")
    echo "You selected News";

elseif ($page == "Login")
    echo "You selected Login";

elseif ($page == "Links")
    echo "You selected Links";

else
    echo "Unknown selection";
```

equivalent **switch** version

```
switch ($page)
{
    case "Home":
        echo "You selected Home";
        break;
    case "About":
        echo "You selected About";
        break;
    case "News":
        echo "You selected News";
        break;
    case "Login":
        echo "You selected Login";
        break;
    case "Links":
        echo "You selected Links";
        break;
    default:
        echo "Unknown selection";
        break;
}
```

# Purchase Order with Quantity Discount

The image displays four vertically stacked purchase order tables, each with a red box highlighting the 'Discounted Price' column. The tables show the following data:

Quantity	Price	Discounted Price (0%)	Subtotal	Tax (6%)	Purchase Total
30	160.00	160.00	4800.00	288.00	5088.00

Quantity	Price	Discounted Price (10%)	Subtotal	Tax (6%)	Purchase Total
200	160.00	144.00	28800.00	1728.00	30528.00

Quantity	Price	Discounted Price (20%)	Subtotal	Tax (6%)	Purchase Total
700	160.00	128.00	89600.00	5376.00	94976.00

Quantity	Price	Discounted Price (30%)	Subtotal	Tax (6%)	Purchase Total
2000	160.00	112.00	224000.00	13440.00	237440.00

A screenshot of a web browser showing a form for calculating a purchase order. The form includes fields for Quantity (30), Price (160), Tax rate (0.06), and a Submit button.

Use conditional statements to calculate Quantity Discount:

0%: <100

10%: 100-499

20%: 500-999

30%: 1000+

# Calculating Quantity Discount

## using `if..elseif..else`

```
// Get the form values from the POST array
$quantity = $_POST['qty'];
$price = $_POST['price'];
$taxrate = $_POST['taxrate'];
$discount = 0;

// Determine the discount rate
if ($quantity < 100) {
    $discount = 0;
}
elseif (($quantity >= 100) && ($quantity < 500)) {
    $discount = 0.10;
}
elseif (($quantity >= 500) && ($quantity < 1000)) {
    $discount = 0.20;
}
else {
    $discount = 0.30;
}

// Calculate discounted price, tax, and total
$discountedPrice = $price - ($price * $discount);
$subtotal = $quantity * $discountedPrice;
$taxamount = $subtotal * $taxrate;
$total = $subtotal + $taxamount;
```

# Combining input and output on same page

localhost/2021\_INLS672/PHPCon

Quantity

Price

Tax rate

**Submit**

localhost/2021\_INLS672/PHPCon

Quantity	Price	Discounted Price (0%)	Subtotal	Tax (6%)	Purchase Total
30	160.00	160.00	4800.00	288.00	5088.00

2 script version

1 script version

localhost/2021\_INLS672/PHP-Con

localhost/2021\_INLS672/PHP-Conditionals/purchase\_order\_discount\_one\_script.php

Quantity	Price	Tax rate	<b>Submit</b>		
30	160	0.06			
30	160.00	160.00	4800.00	288.00	5088.00

- Instead of putting the HTML in one file and handling the form processing in a separate PHP script, you can combine into one PHP script file
- Use the PHP function, `isset`, to test whether the submit button has been clicked

# Combine HTML, PHP into one script

```
<body>
<?php
    // Initialize variables to 0
    $quantity = $subtotal = $taxamount = $total = ... 0;
    // If the "submit" variable exists, the form has been submitted -
    // get the value of the input fields in the form
    if (isset($_POST['submit'])) { ←
        // Get the form values from the POST array
        // Determine discount rate
        // Calculate discounted price,tax,total
    }
?>

<form method="post"> /* no action attribute */
    // Form input fields for quantity, price, tax rate, and Submit button
</form>
<table>
    // Output for quantity, price, discounted price, subtotal, tax amount, and total
</table>
</body>
```

Checks to see if there is a value in the POST array for 'submit'

If **true**, then get the form values and calculate results for the table

If **false**, just display the form and table with no values

# One more thing... Keeping User Input in the Form Fields

In this example, the Quantity, Price, and Tax rate values were entered, but after submitting, the form input fields are cleared.

Quantity      Price      Tax rate      Submit

Quantity	Price	Discounted Price (10%)	Subtotal	Tax (4%)	Purchase Total
100	50.00	45.00	4500.00	180.00	4680.00

In this example, the values entered by the user are kept in the form fields after the form is submitted.

Quantity      Price      Tax rate      Submit

Quantity	Price	Discounted Price (10%)	Subtotal	Tax (4%)	Purchase Total
100	50	0.04	4500.00	180.00	4680.00

You can keep the last values entered by the user by including the value attribute in the `<input>` field, and setting its value to the corresponding PHP variable.

```
<input type="number" id="qty" name="qty" size="10" min="1" max="5000"
       required value="php echo $quantity;?&gt;"&gt;

&lt;input type="number" id="price"    name="price"    size="10" min="1.00" max="5000.00"
       step=".10" required value="<?php echo $price;?&gt;"&gt;

&lt;input type="number" id="taxrate" name="taxrate" size="10" min="0.01" max="0.10"
       step="0.01" required value="<?php echo $taxrate;?&gt;"&gt;</pre
```

# Combining HTML and PHP into one script

PHP script files can contain both HTML and PHP

HTML can include PHP:

```
<td><?php echo $quantity ?></td>
```

PHP can generate HTML:

```
<?php echo "<td>" . $quantity . "</td>"?>
```

## How this works (PHP Manual: [PHP Tags](#) )

- When PHP parses a file, it looks for opening and closing tags (`<?php` and `?>`) that tell PHP to start and stop interpreting the code between them.
- Everything outside of a pair of opening and closing tags is ignored by the PHP parser, but it will start outputting whatever it finds until it finds another opening PHP tag.

# Example: Temperature Conversion

The screenshot shows a web browser window titled "Temperature Converter". The address bar indicates the page is at "localhost/2021\_INLS6...". The main content area displays a "Temperature Converter" application. It has a text input field labeled "Degrees:" containing the value "69". Below the input field are two buttons: "Celsius to Fahrenheit ▾" and a green "Convert" button. A horizontal line separates the input area from the output. The output is displayed in green text: "**69° Celsius is 156.20° Fahrenheit**". The browser interface includes standard controls like minimize, maximize, and close buttons at the top, and a toolbar with various icons below the address bar.

# Using `if...else` with a dropdown list

```
<h2>Temperature Converter</h2>
<form method="get">
  <div><label>Degrees: </label><input type="number" name="degrees" required
    value="php echo $degrees;?&gt;"&gt;&lt;br&gt;&lt;/div&gt;
  &lt;div&gt;
    &lt;select name="conversion"&gt;
      &lt;option value="ctf"
        &lt;?php if ($conversion_type == 'ctf') echo 'selected'; ?&gt;&gt;Celsius to Fahrenheit&lt;/option&gt;
      &lt;option value="ftc"
        &lt;?php if ($conversion_type == 'ftc') echo 'selected'; ?&gt;&gt;Fahrenheit to Celsius&lt;/option&gt;
    &lt;/select&gt;
    &lt;button type="submit" id="submit" name="submit"&gt;Convert&lt;/button&gt;
  &lt;/div&gt;
&lt;/form&gt;</pre
```

```
$degrees = $_GET['degrees'];
$conversion_type = $_GET['conversion'];

if (strcmp($conversion_type, 'ctf') == 0) { //convert Celsius to Fahrenheit
  $result = sprintf("%0.2f", $degrees * 9 / 5 + 32);
  $message = "$degrees° Celsius is $result° Fahrenheit"; }
else { // convert Fahrenheit to Celsius
  $result = sprintf("%0.2f", ($degrees - 32) * 5 / 9);
  $message = "$degrees° Fahrenheit is $result° Celsius"; }
```

temperature\_conversion.php

PHP functions:

PHP strcmp()

PHP sprintf()

# Quick debug: Inspecting GET/POST arrays

- `var_dump` displays structured information about a variable
- `print_r` prints slightly more readable information

array(3) { ["degrees"]=> string(2) "56" ["conversion"]=> string(3) "ctf" ["submit"]=> string(0) "" } **using var\_dump**  
Array ( [degrees] => 56 [conversion] => ctf [submit] => ) **using print\_r**

## Temperature Converter

Degrees:

Celsius to Fahrenheit

---

56° Celsius is 132.80° Fahrenheit

```
var_dump($_GET); echo "<strong> using var_dump</strong><br>";
print_r($_GET); echo "<strong> using print_r</strong><br>";
```