**AWT Practical**

Week - 3 Task

1. Change Color of HTML Table using PHP (Get rows/columns/color from php variables)

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="preconnect" href="https://fonts.googleapis.com">

  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700;800;900&display=swap" rel="stylesheet">

  <link rel="stylesheet" href="css/tableColor.css">

  <title>Table Color</title>

  <style>

    \* {

      margin: 0;

      padding: 0;

      box-sizing: border-box;

      font-family: "Poppins", sans-serif;

    }

    body {

      height: 100vh;

      width: 100vw;

      display: flex;

      flex-direction: column;

      justify-content: center;

      align-items: center;

      padding: 1em 3em;

      background-color: #ebebd3;

    }

    body h1 {

      margin-bottom: 3em;

      text-align: justify;

    }

    table {

      border-collapse: collapse;

    }

    tr,

    td {

      border: 1px solid #949494;

      padding: 8px;

    }

    body table td {

      padding: 0.5em 1em;

    }

  </style>

</head>

<body>

  <h1>Change Color of HTML Table using PHP (Get rows/columns/color from php variables)</h1>

  <table>

    <?php

    $rows = 6;

    $cols = 5;

    $color = '#fff';

    for ($i = 0; $i < $rows; $i++) {

      echo "<tr>";

      for ($j = 0; $j < $cols; $j++) {

        if (($i + $j) % 2 == 0) {

          $color = '#083d77';

        } else {

          $color = '#f4d35e';

        }

        echo "<td style='background-color: $color;'>($i, $j)</td>";

      }

      echo "</tr>";

    }

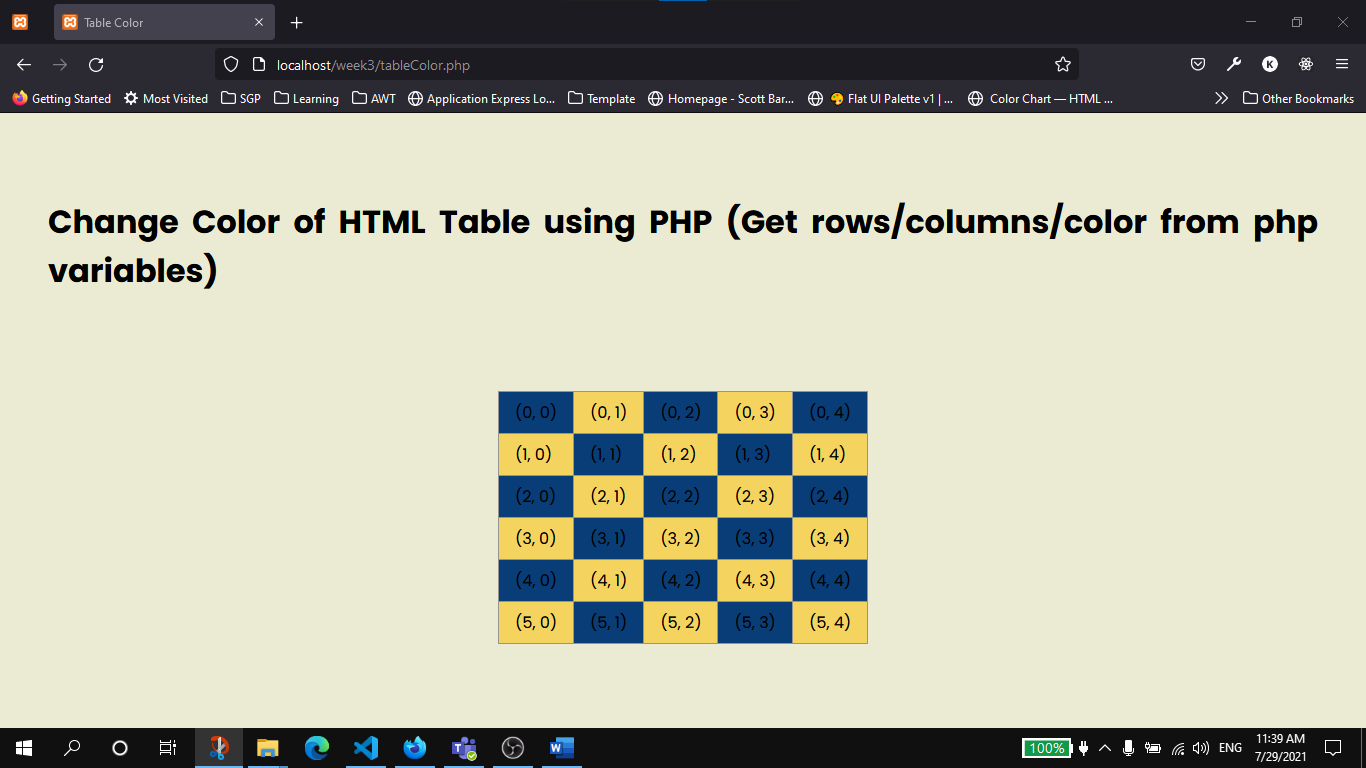
    ?>

  </table>

</body>

</html>

**Output:**



1. Get Password from PHP variable & Display following info in table

* Count Total Characters
* Count Total Number of digits
* Count Total Special Characters
* Display Password strength
* Not Acceptable
* Weak
* Strong

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="preconnect" href="https://fonts.googleapis.com">

  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700;800;900&display=swap" rel="stylesheet">

  <title>Password</title>

  <style>

    \* {

      margin: 0;

      padding: 0;

      box-sizing: border-box;

      font-family: "Poppins", sans-serif;

    }

    body {

      height: 100vh;

      width: 100vw;

      display: flex;

      flex-direction: column;

      justify-content: center;

      align-items: center;

      padding: 1em 3em;

      background-color: #ebebd3;

    }

    body h1 {

      margin-bottom: 3em;

      text-align: justify;

    }

    table {

      border-collapse: collapse;

    }

    tr,

    td {

      border: 1px solid #949494;

      padding: 8px;

    }

    ul {

      margin-bottom: 2em;

    }

    body table td {

      padding: 0.5em 1em;

    }

  </style>

</head>

<body>

  <p>Get Password from PHP variable & Display following info in tabl</p>

  <ul>

    <li>Count Total Characters</li>

    <li>Count Total Number of digits</li>

    <li>Count Total Special Characters</li>

    <li>Display Password strength

      <ul>

        <li>Not Acceptable</li>

        <li>Weak</li>

        <li>Strong</li>

      </ul>

    </li>

  </ul>

  <table>

    <?php

    $password = "krutik2001!@#";

    $totalChar = strlen($password);

    $totalDigits = preg\_match\_all("/[0-9]/", $password);

    $totalSpecialChar = preg\_match\_all("/[$&+,:;=?@#|'<>.-^\*()%!]/", $password) - $totalDigits;

    $passStrength = 'Not Acceptable';

    $color = '#e74c3c';

    if ($totalChar > 8 && ($totalDigits == 0 || $totalSpecialChar == 0)) {

      $passStrength = "Week";

      $color = '#f1c40f';

    }

    if ($totalChar > 8 && $totalDigits != 0 && $totalSpecialChar != 0) {

      $passStrength = "Strong";

      $color = '#2ecc71';

    }

    echo "<tr>

        <td>Total Characters</td>

        <td>$totalChar</td>

      </tr>

      <tr>

        <td>Total Digits</td>

        <td>$totalDigits</td>

      </tr>

      <tr>

        <td>Total Special Characters</td>

        <td>$totalSpecialChar</td>

      </tr>

      <tr>

        <td>Password Strength</td>

        <td style='background-color: $color;'>$passStrength</td>

      </tr>"

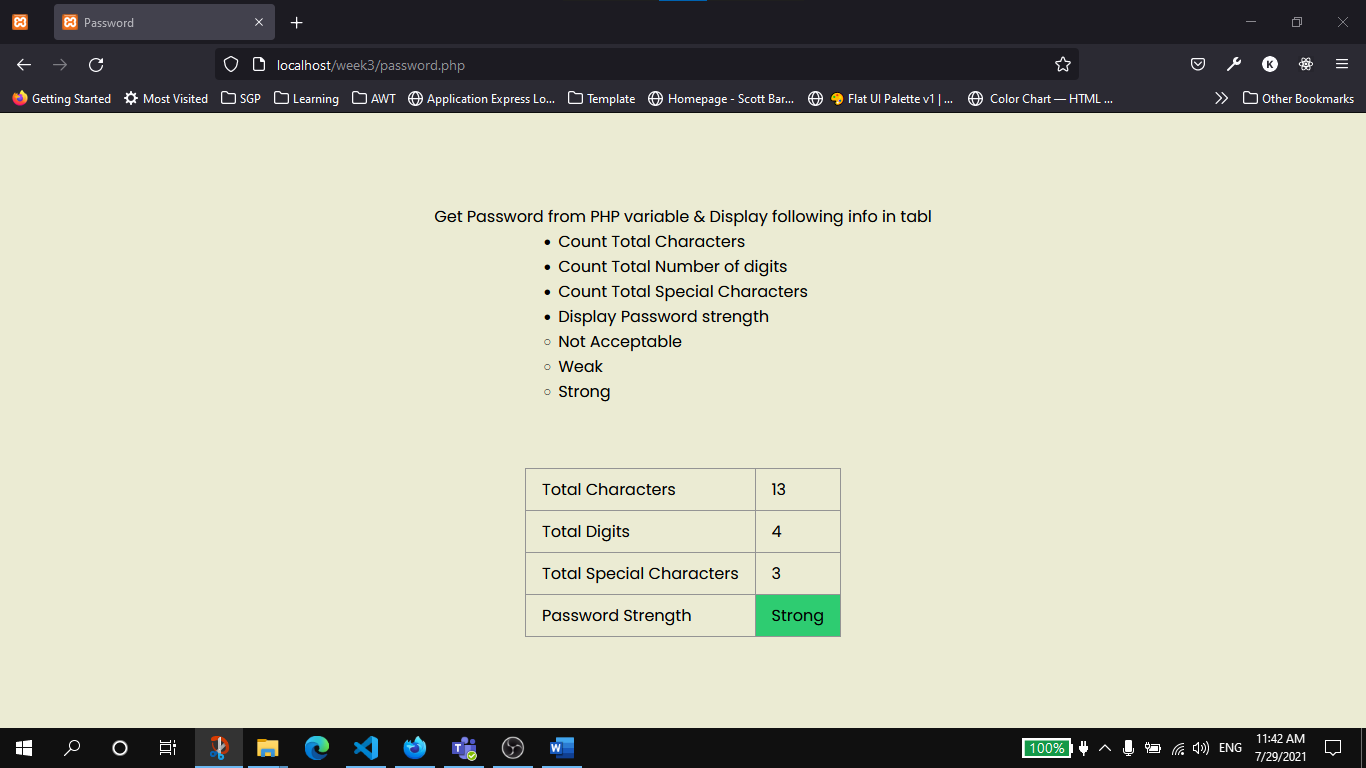
    ?>

  </table>

</body>

</html>

**Output:**



1. Explore and experiments with builtin string functions/methods - 20 minimum

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="preconnect" href="https://fonts.googleapis.com">

  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700;800;900&display=swap" rel="stylesheet">

  <title>String Functions/Methods</title>

  <style>

    \* {

      margin: 0;

      padding: 0;

      box-sizing: border-box;

      font-family: "Poppins", sans-serif;

    }

    body {

      display: flex;

      flex-direction: column;

*/\* justify-content: center; \*/*

      align-items: center;

      padding: 1em 3em;

      background-color: #ebebd3;

    }

    body h2 {

      margin-bottom: 2em;

      text-align: center;

    }

    table {

      border-collapse: collapse;

    }

    tr,

    td {

      border: 1px solid #000;

      padding: 8px;

    }

    tr:nth-child(even) {

      background-color: #cacaca;

    }

    ul {

      margin-bottom: 2em;

    }

    div {

      display: flex;

      width: 70%;

      justify-content: space-between;

    }

    body table td {

      padding: 0.5em 1em;

    }

  </style>

</head>

<body>

*<!--*

*3) Explore and experiments with builtin string functions/methods  - 20 minimum*

*-->*

  <h2>PHP - Builtin String Functions/Methods</h2>

  <div>

    <table>

      <?php

      $str1 = 'krutik';

      $str2 = 'gadhiya';

      echo "<tr>

      <td style='text-align: center;' colspan='2'>str1 = $str1</td>

      </tr>";

      ?>

      <?php

*// strlen() lenght of the string*

      $temp = strlen($str1);

      echo "<tr>

      <td>strlen(\$str1);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// ucfirst() capatilise the first letter of first word*

      $temp = ucfirst($str1 . " " . $str2);

      echo "<tr>

      <td>ucfirst(\$str1.' '.\$str2);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// ucword() capatilise the first letter of every word*

      $temp = ucwords($str1 . " " . $str2);

      echo "<tr>

      <td>ucwords(\$str1.' '.\$str2);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// trim() remove the particular letter/word from the string*

      $temp = trim($str1, 'k');

      echo "<tr>

      <td>trim(\$str1, 'k');</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// substr() gets the substring from the given index*

      $temp = substr($str2, 2);

      echo "<tr>

      <td>substr(\$str1, 2);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// strstr() gets the substring from the first occurence of the given string*

      $temp = strstr($str1 . " " . $str2 . '!!!', 'gadhiya');

      echo "<tr>

      <td>strstr(\$str1.' '.\$str2.'!!!', 'gadhiya');</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// strcmp() compares the strings and give 0 (same str),1 (greater than) or -1 (less than)*

      $temp = strcmp($str1, $str2);

      echo "<tr>

      <td>substr(\$str1, \$str2);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// chop() Removes whitespace or other characters from the right end of a string*

      $temp = chop($str2, 'gad');

      echo "<tr>

      <td>substr(\$str2, 'gad');</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// bin2hex() Converts a string of ASCII characters to hexadecimal values*

      $temp = bin2hex($str2);

      echo "<tr>

      <td>bin2hex(\$str2);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// hex2bin() Converts a string of hexadecimal values to ASCII characters*

      $temp1 = hex2bin($temp);

      echo "<tr>

      <td>hex2bin($temp);</td>

      <td>$temp1</td>

      </tr>";

      ?>

    </table>

    <table>

      <?php

      echo "<tr>

      <td style='text-align: center;' colspan='2'>str2 = $str2</td>

      </tr>";

      ?>

      <?php

*// addslashes() Returns a string with backslashes in front of predefined characters*

      $temp = addslashes($str1 . '"' . $str2 . '"');

      echo "<tr>

      <td>addslashes(\$str1.' \" '.\$str2.' \" ');</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// chr() Returns a character from a specified ASCII value*

      $temp = chr(65);

      echo "<tr>

      <td>chr(65);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// crypt() One-way string hashing*

      $temp = crypt($str1 . " " . $str2, 'kk');

      echo "<tr>

      <td>crypt(\$str1.' '.\$str2, 'kk');</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// str\_shuffle() Randomly shuffles all characters in a string*

      $temp = str\_shuffle($str2);

      echo "<tr>

      <td>str\_shuffle(\$str2);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// echo() Output one or more string*

      echo "<tr>

      <td>echo(\$str2)</td>

      <td>$str2</td>

      </tr>";

      ?>

      <?php

*// lcfirst() Converts the first character of a string to lowercase*

      $temp = lcfirst("Krutik Gadhiya");

      echo "<tr>

      <td>lcfirst('Krutik Gadhiya');</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// str\_word\_count() Count the number of words in a string*

      $temp = str\_word\_count($str1 . ' ' . $str2);

      echo "<tr>

      <td>str\_word\_count(\$str1 . ' ' . \$str2);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// strrev() reverse the string*

      $temp = strrev($str1);

      echo "<tr>

      <td>strrev(\$str1);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// strtolower() Converts a string to lowercase letters*

      $temp = strtolower($str1);

      echo "<tr>

      <td>strtolower(\$str1);</td>

      <td>$temp</td>

      </tr>";

      ?>

      <?php

*// strtoupper() Converts a string to uppercase letters*

      $temp = strtoupper($str2);

      echo "<tr>

      <td>strtoupper(\$str2);</td>

      <td>$temp</td>

      </tr>";

      ?>

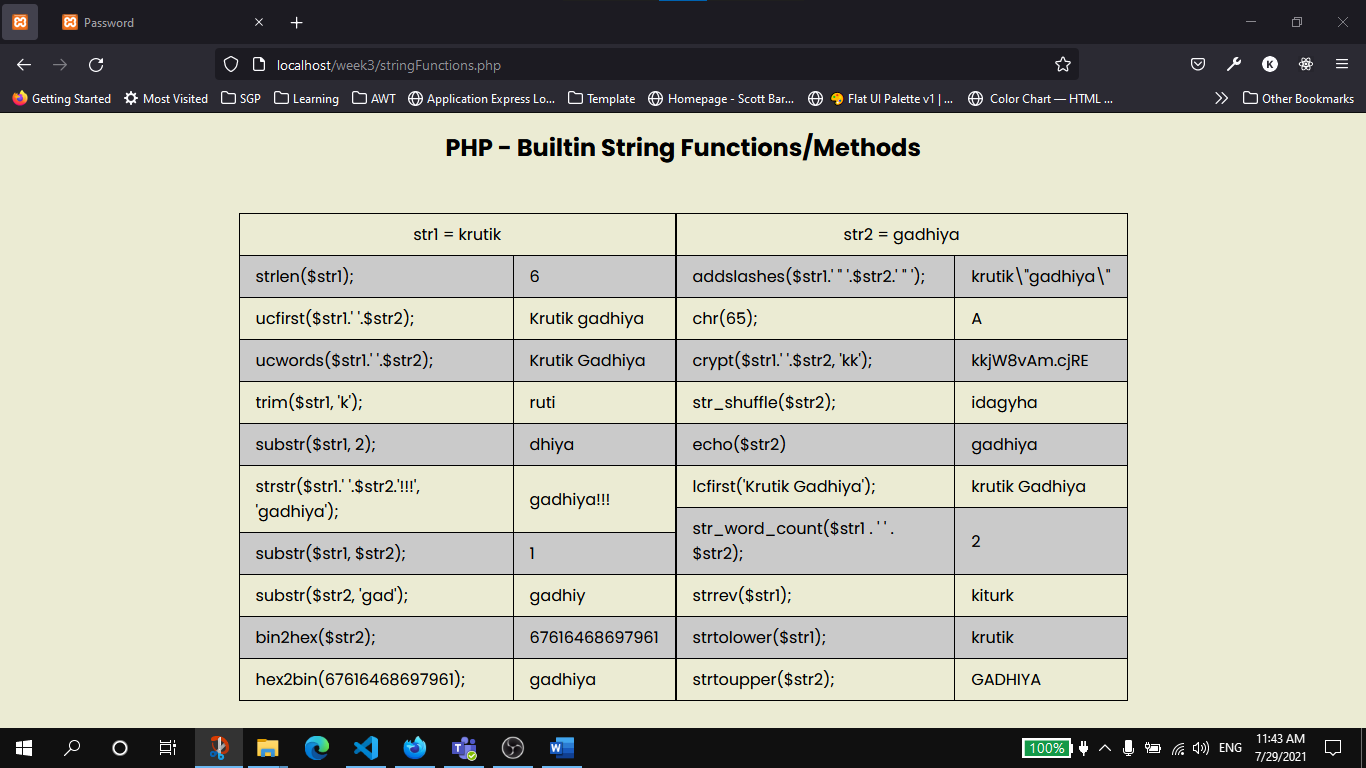
    </table>

  </div>

</body>

</html>

**Output:**



1. Experiments with following PHP Operators.

* Arithmetic operators
* Assignment operators
* Comparison operators
* Increment/Decrement operators
* Logical operators (AND, OR, NOT)
* String operators
* Conditional assignment operators
* Bitwise operators

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="preconnect" href="https://fonts.googleapis.com">

  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700;800;900&display=swap" rel="stylesheet">

  <title>Document</title>

  <style>

    \* {

      margin: 0;

      padding: 0;

      box-sizing: border-box;

      font-family: "Poppins", sans-serif;

    }

    body {

      display: flex;

      flex-direction: column;

*/\* justify-content: center; \*/*

      align-items: center;

      padding: 1em 3em;

      background-color: #ebebd3;

    }

    body h2 {

      margin-bottom: 2em;

      text-align: center;

    }

    table {

      border-collapse: collapse;

    }

    tr,

    td {

      border: 1px solid #000;

      padding: 8px;

    }

    tr {

      background-color: #ebebd3;

    }

    tr:nth-child(even) {

      background-color: #cacaca;

    }

    ul {

      margin-bottom: 2em;

    }

    div {

      display: flex;

      width: 70%;

      justify-content: space-between;

    }

    body table td {

      padding: 0.5em 1em;

    }

  </style>

</head>

<body>

*<!--*

*Experiments with following PHP Operators.*

*Arithmetic operators*

*Assignment operators*

*Comparison operators*

*Increment/Decrement operators*

*Logical operators (AND, OR, NOT)*

*String operators*

*Conditional assignment operators*

*Bitwise operators*

*-->*

  <h2>Operators in PHP</h2>

  <table>

    <tr>

      <td>Arithmetic Operator<br>(+, -, /, \*, %, \*\*)</td>

      <td>

        <table>

          <?php

          $x = 10;

          $y = 20;

          $temp = $x + $y;

          echo "<tr>

          <td>Addition</td>

          <td>$x + $y = $temp</td>

          </tr>";

          ?>

          <?php

          $temp = $x - $y;

          echo "<tr>

          <td>Subtraction</td>

          <td>$x - $y = $temp</td>

          </tr>";

          ?>

          <?php

          $temp = $x \* $y;

          echo "<tr>

          <td>Multiplication</td>

          <td>$x \* $y = $temp</td>

          </tr>";

          ?>

          <?php

          $temp = $x / $y;

          echo "<tr>

          <td>Division</td>

          <td>$x / $y = $temp</td>

          </tr>";

          ?>

          <?php

          $temp = $x % $y;

          echo "<tr>

          <td>Modulo</td>

          <td>$x % $y = $temp</td>

          </tr>";

          ?>

          <?php

          $temp = $x \*\* $y;

          echo "<tr>

          <td>Power/Exponent</td>

          <td>$x \*\* $y = $temp</td>

          </tr>";

          ?>

        </table>

      </td>

    </tr>

    <tr>

      <td>Assignment Operator<br>(=, +=, -=, \*=, /=, %=)</td>

      <td>

        <table>

          <?php

          $temp1 = $temp;

          $temp = $x;

          echo "<tr>

          <td>Simple Assignment</td>

          <td>\$temp = \$x: $temp</td>

          </tr>";

          ?>

          <?php

          $temp1 = $temp;

          $temp += $x;

          echo "<tr>

          <td> += </td>

          <td>$temp1 += $x: $temp</td>

          </tr>";

          ?>

          <?php

          $temp1 = $temp;

          $temp \*= $x;

          echo "<tr>

          <td> \*= </td>

          <td>$temp1 \*= $x: $temp</td>

          </tr>";

          ?>

          <?php

          $temp1 = $temp;

          $temp /= $y;

          echo "<tr>

          <td> /= </td>

          <td>$temp1 /= $y: $temp</td>

          </tr>";

          ?>

          <?php

          $temp1 = $temp;

          $temp -= $y;

          echo "<tr>

          <td> -= </td>

          <td>$temp1 -= $y: $temp</td>

          </tr>";

          ?>

          <?php

          $temp1 = $temp;

          $temp %= $x;

          echo "<tr>

          <td> %= </td>

          <td>$temp1 %= $x: $temp</td>

          </tr>";

          ?>

        </table>

      </td>

    </tr>

    <tr>

      <td>Comparison Operator<br>( ==, ===, !=, <,>, <=,>= )</td>

      <td>

        <table>

          <?php

          if ($x == $y) {

            $temp = "same";

          } else {

            $temp = "different";

          }

          echo "<tr>

          <td>== (equals)</td>

          <td>\$x == \$y: $temp</td>

          </tr>";

          ?>

          <?php

          if ($x === $y) {

            $temp = "exactly same";

          } else {

            $temp = "exactly different";

          }

          echo "<tr>

          <td>=== (identical)</td>

          <td>\$x == \$y: $temp</td>

          </tr>";

          ?>

        </table>

      </td>

    </tr>

    <tr>

      <td>Logical Operator<br>(&& - and, || - or, ! - not, xor)</td>

      <td>

        <table>

          <?php

          echo "<tr>

          <td>&& - and</td>

          <td>\$x && \$y: true if both are true</td>

          </tr>";

          ?>

          <?php

          echo "<tr>

          <td>|| - or</td>

          <td>\$x || \$y: true if any one is true</td>

          </tr>";

          ?>

          <?php

          echo "<tr>

          <td>xor - exclusive or</td>

          <td>\$x xor \$y: if one true then true else false</td>

          </tr>";

          ?>

        </table>

      </td>

    </tr>

    <tr>

      <td>Increment-Decrement Operator<br>(++, --)</td>

      <td>

        <table>

          <?php

          $temp = $x++;

          echo "<tr>

          <td>++ (Increment)</td>

          <td>\$temp = \$x++: $temp</td>

          </tr>";

          ?>

          <?php

          $temp = $y--;

          echo "<tr>

          <td>-- (Decrement)</td>

          <td>\$temp = \$y--: $temp</td>

          </tr>";

          ?>

        </table>

      </td>

    </tr>

    <tr>

      <td>String Operator<br>(., .=)</td>

      <td>

        <table>

          <?php

          $str1 = 'kk';

          $str2 = 'gg';

          $temp1 = $str1 . $str2;

          echo "<tr>

          <td>. (string concatenation)</td>

          <td>\$temp1 = \$str1.\$str2: $temp1</td>

          </tr>";

          ?>

          <?php

          $temp1 .= $temp1;

          echo "<tr>

          <td>.= (concatenation assignment)</td>

          <td>\$temp1 .= \$temp1: $temp1</td>

          </tr>";

          ?>

        </table>

      </td>

    </tr>

    <tr>

      <td>Bitwise Operator<br>(&, |, ^, <<,>>, ~)</td>

      <td>

        <table>

          <?php

          $temp = $x & $y;

          echo "<tr>

          <td>& (bitwise and)</td>

          <td>\$temp = \$x & \$y: $temp</td>

          </tr>";

          ?>

          <?php

          $temp = $x | $y;

          echo "<tr>

          <td>| (bitwise or)</td>

          <td>\$temp = \$x | \$y: $temp</td>

          </tr>";

          ?>

          <?php

          $temp = $x ^ $y;

          echo "<tr>

          <td>^ (bitwise xor)</td>

          <td>\$temp = \$x ^ \$y: $temp</td>

          </tr>";

          ?>

          <?php

          $temp = $x << 2;

          echo "<tr>

          <td><< (shift left)</td>

          <td>\$temp = \$x << 2: $temp</td>

          </tr>";

          ?>

          <?php

          $temp = $y >> 2;

          echo "<tr>

          <td>>> (shift right)</td>

          <td>\$temp = \$y >> 2: $temp</td>

          </tr>";

          ?>

          <?php

          $temp = ~$x;

          echo "<tr>

          <td>~ (bitwise not)</td>

          <td>\$temp = ~ \$x: $temp</td>

          </tr>";

          ?>

        </table>

      </td>

    </tr>

  </table>

</body>

</html>

**Output:**

