

INLS 672

Web Development 2

PHP Loops and Arrays



Joan Boone
jpboone@email.unc.edu

2 Basic Types of Repetition Structures

while loop

- Condition-controlled
- Causes the statements in the loop to repeatedly execute as long as a condition is true
- Something has to happen inside the loop to make the condition false and exit the loop
- **while** loops repeat (execute or iterate) an indefinite number of times

for loop

- Count-controlled
- Causes the statements in the loop to repeatedly execute a specific number of times.
- **for** loops repeat (execute or iterate) a definite number of times

Loops

Loops allow you to execute a set of PHP statements repeatedly in several ways

- **while**
- **do...while**
- **for**
- **foreach**

Often used with auto-increment/decrement operators

```
$total++; // same as $total = $total + 1;  
$count--; // same as $count = $count - 1;
```

References

- PHP Manual: [while](#), [do..while](#), [for](#), [foreach](#) loops
- w3schools: [PHP Loops](#)

while Loops

```
<?php
$upperLimit = '';
$squares = '';
$lowerLimit = 1;
if (isset($_POST['submit'])) {
    $upperLimit = $_POST['limit'];

    // keep printing squares until lower limit = upper limit
    while ($lowerLimit <= $upperLimit) {
        $squares = $squares . ($lowerLimit * $lowerLimit) . ' ';
        $lowerLimit++;
    }
}
?>
<form method="POST">
Print all the squares between 1 and
<input type="number" name="limit" required
      value=<?php echo $upperLimit ?>" min="1" max="500">
<input type="submit" name="submit" value="Go">
<br><br><?php echo $squares;?>
</form>
```



for Loop

```
<?php
$upperLimit = '';
$squares = '';

if (isset($_POST['submit'])) {
    $upperLimit = $_POST['limit'];

    for ($x=1; $x <= $upperLimit; $x++) {
        $squares = $squares .  pow($x, 2) . ' ';
    }
}
?>
<form method="POST">
Print all the squares between 1 and
<input type="number" name="limit" required
      value=<?php echo $upperLimit ?>" min="1" max="500">
<input type="submit" name="submit" value="Go">
<br><br>
<?php echo $squares;?>
</form>
```

localhost/2021_INLS672/PHP-Loops/for_loop_squares.php

Print all the squares between 1 and Go

1 4 9 16 25 36 49 64 81 100 121 144 169 196 225

Nested for Loop

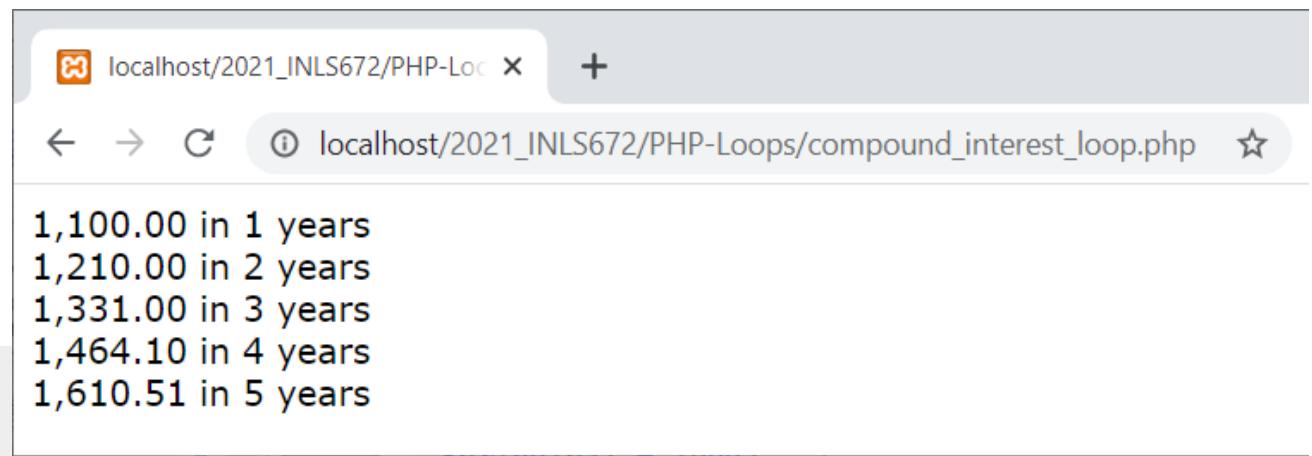
```
<?php
$rows = '';
$columns = '';
if (isset($_POST['submit'])) {
    $rows = $_POST['rows'];
    $columns = $_POST['columns'];
}
<form method="post">
    Enter number of rows <input name="rows" type="number" required
                           value="<?php echo $rows ?>">
    and columns <input name="columns" type="number" required
                           value="<?php echo $columns ?>">
    <input type="submit" name="submit" value="Draw Table">
</form>
<?php
if (isset($_POST['submit'])) {
    echo "<table>";
    for ($r = 1; $r <= $rows; $r++) {
        echo "<tr>";
        for ($c = 1; $c <= $columns; $c++) {
            echo "<td>" . $r . "," . $c . "</td>";
        }
        echo "</tr>";
    }
    echo "</table>";
}
?>
```

1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10	1,11	1,12	1,13	1,14
2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10	2,11	2,12	2,13	2,14
3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9	3,10	3,11	3,12	3,13	3,14
4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	4,10	4,11	4,12	4,13	4,14
5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9	5,10	5,11	5,12	5,13	5,14

Compound Interest Calculation

```
<?php
$principal = 1000;
$interest = .10;
$years = 5;
$index = 1;

while ($index <= $years)  {
    $principal = $principal + ($principal * $interest);
    echo number_format($principal,2) . ' in ' . $index . ' years <br>';
    $index++;
}
?>
```



A screenshot of a web browser window. The address bar shows the URL `localhost/2021_INLS672/PHP-Loops/compound_interest_loop.php`. The page content displays the following text:
1,100.00 in 1 years
1,210.00 in 2 years
1,331.00 in 3 years
1,464.10 in 4 years
1,610.51 in 5 years

A screenshot of a web browser window. The address bar shows 'localhost/2021_INLS672/PHPLoop'. The page contains a form with three input fields: 'Principal' (value: 10000), 'Interest' (value: 0.02), and 'Years' (value: 5). Below the form is a button labeled 'Calculate Compound Interest'. To the right of the form is a table titled 'Year' and 'Principal' showing the results of the calculation for each year from 1 to 5.

Year	Principal
1	10,200.00
2	10,404.00
3	10,612.08
4	10,824.32
5	11,040.81

Practice: Compound Interest with Form Input and Table Output

Update

[`compound_interest_table_draft.php`](#)

- Initialize variables from input fields
- Define a loop to calculate the compound interest and display the year and principal in a table row

Arrays

- Arrays are containers that allow you to store multiple values vs. a single-valued variable
- Arrays are ordered maps that associate values with keys. The key can be an integer or string, and the value can be any type
- Can be used to represent dictionaries, stacks, queues, collections
- One way to define and initialize an array

```
<?php  
  
$pizza_toppings = array('onion', 'tomato', 'cheese',  
                      'anchovies', 'ham', 'pepperoni');  
  
print_r($pizza_toppings);  
  
?>
```

Output of array where key is an integer:

```
Array ( [0] => onion [1] => tomato [2] => cheese [3] => anchovies [4] => ham [5] => pepperoni )
```

References

- PHP Manual: [Arrays](#) w3Schools: [PHP Arrays](#)

Many Ways to Define Arrays

- Simplest

```
$pasta = array('spaghetti', 'penne', 'macaroni');
```

- Element-by-element, where key is an integer

```
$pasta[0] = 'spaghetti';
```

```
$pasta[1] = 'penne';
```

```
$pasta[2] = 'macaroni';
```

- Element-by-element, where key is a string (associative)

```
$menu['breakfast'] = 'bacon and eggs';
```

```
$menu['lunch'] = 'roast beef';
```

```
$menu['dinner'] = 'lasagna';
```

- Add or modify by assignment

```
$pasta[3] = 'rigatoni';
```

```
$menu['lunch'] = 'chicken';
```

Looping through arrays: **for** and **foreach**

```
<html>
<body>
Pizza toppings:
<ul>

<?php
$pizza_toppings =
    array('onion', 'tomato', 'cheese', 'anchovies', 'ham', 'pepperoni');

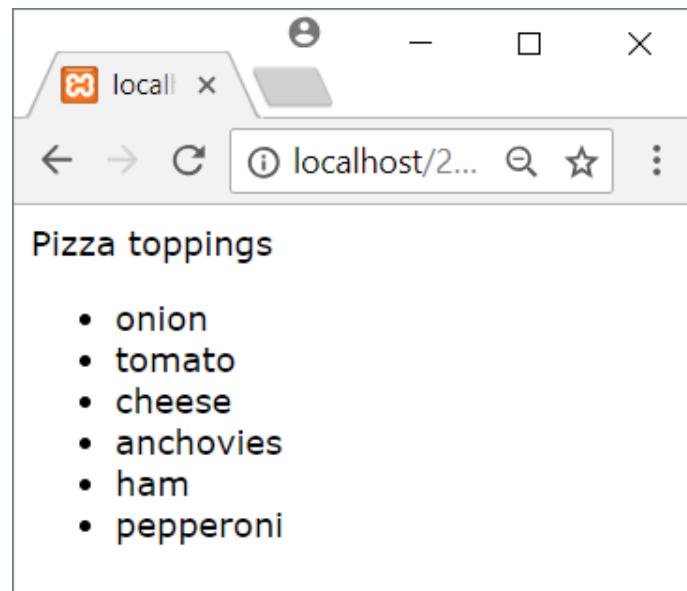
for ($x = 0; $x < count($pizza_toppings); $x++) {
    echo '<li>' . $pizza_toppings[$x] . '</li>';
}
?>

</ul>
</body>
</html>
```

count is a PHP function that returns the number of elements in the array

Alternative: **foreach** loop:

```
foreach ($pizza_toppings as $p) {
    echo '<li>' . $p. '</li>';
}
```



Adding elements to an array

```
<html>
<body>
Pasta
<ul>

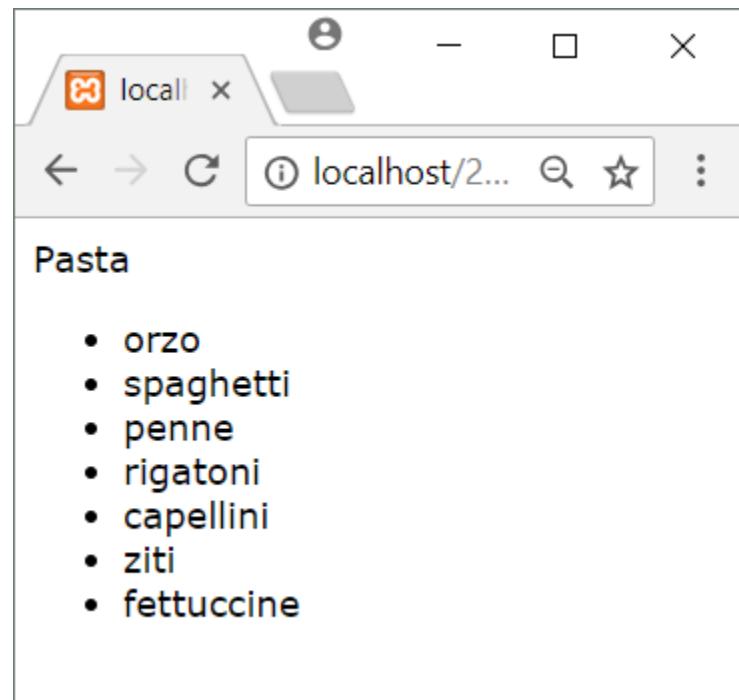
<?php
$pasta = array();
array_push($pasta, 'spaghetti');
array_push($pasta, 'penne');
array_push($pasta, 'rigatoni');

$pasta[] = 'capellini';
$pasta[] = 'ziti';
$pasta[] = 'fettuccine';

array_unshift($pasta, 'orzo');

foreach ($pasta as $p) {
    echo '<li>' . $p . '</li>' ; }
?>

</ul>
</body>
</html>
```



Useful Array Functions

array_pop() removes an element off the end of an array

array_push() pushes an element on to the end of an array

array_shift() removes an element from the beginning of an array

array_unshift() add an element to the beginning of an array

array_search() search an array for a value and return key if found

array_rand() pick a random entry out of an array

explode() splits a string into an array, based on a delimiter

implode() creates a string from array elements

sort() sorts alphabetically (or numerically), ascending

shuffle() randomizes elements in an array

in_array() checks if a value is in an array

And many, many more... PHP Manual: [Array Functions](#)

“Associative” Arrays

- PHP does not distinguish between indexed (key as integer) and associative arrays (key as string).
- How to define and initialize

```
<?php  
    $fruits = array('red' => 'apple', 'yellow' => 'banana',  
                    'purple' => 'plum', 'green' => 'grape') ;  
    print_r($fruits);  
?>
```

Output: Array ([red] => apple [yellow] => banana [purple] => plum [green] => grape)

Multidimensional Arrays

Arrays of arrays (w3schools: [PHP Multidimensional arrays](#))

```
<$cars = array  (  
    array("Volvo",22,18),  
    array("BMW",15,13),  
    array("Saab",5,2),  
    array("Land Rover",17,15)  );
```

Using Arrays with multiple selection input types in forms

check_boxes.php

The image displays two separate browser windows side-by-side, both showing the results of a PHP form submission.

check_boxes.php Browser Window:

- URL: localhost/2021_INLS672/PHP-Loops/check_boxes.php
- Form Fields:
 - Supreme
 - Vegetarian
 - Meat Lovers
 - Cheese
- Buttons:
 - A "Select" button.
- Output:

Your selections are

Vegetarian
Cheese

dropdown_multi_select.php Browser Window:

- URL: localhost/2021_INLS672/PHP-Loops/dropdown_multi_select.php
- Form Fields:
 - A dropdown menu containing the following options:
 - Supreme
 - Vegetarian
 - Meat Lovers
 - Cheese
 - A "Select" button.
- Output:

Your selections are

Supreme
Cheese
- Text at bottom right: dropdown_multi_select.php