

PHP Loops and Functions - Written by Anthony Adams

Last week we covered if, else, else if. These are great for when we need to run different code depending on different input or situations.

However, what if we need to keep checking for something? For example: What if we need to keep checking for data in a database and keep that check going until the condition is met? For that we need to use loops.

A loop will keep doing something until we tell it to stop. There are different types and we will look at them 1 by 1.

There are 4 types of loops:

While

Do-while

For

Foreach

While Loop

A while loop will loop continually as long as the condition is true.

Note that we would obviously want to avoid creating a forever loop. If we removed the incremental ++ operator, the loop would go on forever and the page would never stop loading. If this was running on a server, eventually the php.ini *should* kill the process.

A while loop could be used in the case where you needed to cycle through some data from a database. Maybe you have an image gallery and you need to cycle through all the images constantly for example.

In the example below the num variable is equal to 1. While (the number is less than or equal to 10) {echo out the loop number and add 1} Once we get to 10 the loop will be false and thus it will stop.

```
//while loop
$num = 1;
while ($num <=10) {
    echo "Loop Number<br> " .
$num++;
}
```

The while loop has its limitations though. If the condition starts as false, it will never run, not even once.

Output:

```
Loop Number
1Loop Number
2Loop Number
3Loop Number
4Loop Number
5Loop Number
6Loop Number
7Loop Number
8Loop Number
9Loop Number
10
```

Do-while Loop

A do-while loop on the other hand will run the

code at least 1 time regardless of if the condition is true or false. Then it will check against the condition and run until the conditions are met.

In the case below the variable newnumber is = to 10, do will loop until the condition is met, that condition being while the variable newnumber is = or greater then 20.

```
//do while loop
$newnum = 10;
    do {
        echo "New Loop Number
<br>" . $newnum++ ;
    }
    while ($newnum <= 20);
```

Output:

New Loop Number
10New Loop Number
11New Loop Number
12New Loop Number
13New Loop Number
14New Loop Number
15New Loop Number
16New Loop Number
17New Loop Number
18New Loop Number
19New Loop Number
20

For Loop

A for loop will cycle through a block of code a specific number of times.

In the case below. We declare the variable year as 2010; We state that if the year is less then or = to 2023; Add 1 more year incrementally. Once the condition is met (it getting to 2023) it will stop.

```
//for loop
    for ($year = 2010; $year <=
2023; $year++){
```

```
        echo 'The Year is ' .  
$year . '<br>';  
    }
```

Output:

The Year is 2010
The Year is 2011
The Year is 2012
The Year is 2013
The Year is 2014
The Year is 2015
The Year is 2016
The Year is 2017
The Year is 2018
The Year is 2019
The Year is 2020
The Year is 2021
The Year is 2022
The Year is 2023

Foreach Loop

The foreach loop is used to iterate over an arrays. If you need to go through each element of an array the foreach loop is best practice.

In the code below we just have an array which is a list of cars. Suppose we need to go through the list, we can use foreach to go through the

list of cars and convert the plural to singular and list them individually.

```
$cars = array('Genesis', '240SX',  
'300ZX', 'Silvia', 'EVO' );  
    foreach ($cars as $car) {  
        echo $car;  
        echo '<br>';  
    }
```

Output:

```
Genesis  
240SX  
300ZX  
Silvia  
EVO
```

Break

To break out of a loop you can use the **break;** keyword into the structure. This will break you out of a for, foreach, while, do-while or switch structure.

Functions

In php there are:

Built in functions - These are pre built functions in PHP, which PHP has over 1000 of them. There is no way you will remember them all, rather you can reference them when you need to:

https://www.w3schools.com/php/php_ref_overview.asp

Alternatively you can just google “php built in functions” and find many results for different lists.

An example:

This echos out the current date.

```
echo gmdate("l jS \of F Y h:i:s A")
```

User defined functions - These are any function you make. The syntax:

```
function functionName() {
```

```
    code to be executed;
}
```

Always give your function a descriptive name. Never give your user defined function the same name as a built in function.

An example:

Below we make a function named 'num_nums' and make the arguments num1= and num2=. In the code to be executed, we create a variable of \$value which is the sum of the math. Then we return the result. Later we can echo out the result.

```
function num_nums($num1 = 10, $num2
= 5) {
    $value = $num1 * $num2;
    return $value; //return
gets the result but does not print
it
}

echo num_nums();
```

HTML in PHP

You can echo out HTML structure in PHP:

Refer to the file `html-structure.php`