Chapter 5-part 1: PHP Arrays

What is an Array?

An array stores multiple values in one single variable:

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] ;
?>
```

```
<?php
$cars = array("Volvo", "BMW", "Toyota");



echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] ;
?>
```

How to create arrays ?

```
<?php
        $cars = array();
        $cars[0] = "Volvo " ;
        $cars[1] = "BMW " ;
        $cars[2] = "Toyota";
?>
```

In PHP, there are three types of arrays:

1. **Indexed arrays** - Arrays with a numeric index
2. **Associative arrays** - Arrays with named keys
3. **Multidimensional arrays** - Arrays containing one or more arrays

Example: Loop- Indexed Array

To get The Length of an Array – use the count() Function.

```
<?php
        $cars = array( "Volvo", "BMW", "Toyota");
        $arrlength =  count( $cars );

        for($x = 0; $x < $arrlength; $x++ ) {
                echo $cars[$x];
                echo "<br>";
        }
?>
```

Example: Associative Array

Associative arrays are arrays that use **named keys** that you assign to them.

```php
<?php
        $age = array("Petar"=>"35", "Joe"=>"43");

        // OR
        $age['Peter'] = "35";
        $age['Joe'] = "43";

        echo "Peter is " . $age['Peter'] . " years old ";
        echo "Joe is " . $age['Joe'] . " years old ";
?>
```

Example: Loop- Associative Array

To let a function return a value, use the return statement:

```
<?php

        $age = array("Petar"=>"35", "Joe"=>"43");
        foreach($age as $x => $x_value) {
                echo "Key=" . $x . ", Value=" . $x_value;
                echo "<br>";
        }

?>
```

Questions?