



Server-side Web Development 1: PHP contd.

Backend Web Development

by: Salahuddin ElKazak



Program Controls contd.

Array Finale

Lecture 2

Iterating through an Array

- Using `foreach`:

```
php

foreach ($array as $key => $value) {
    echo "$key: $value";
}
```

- Using `while`:

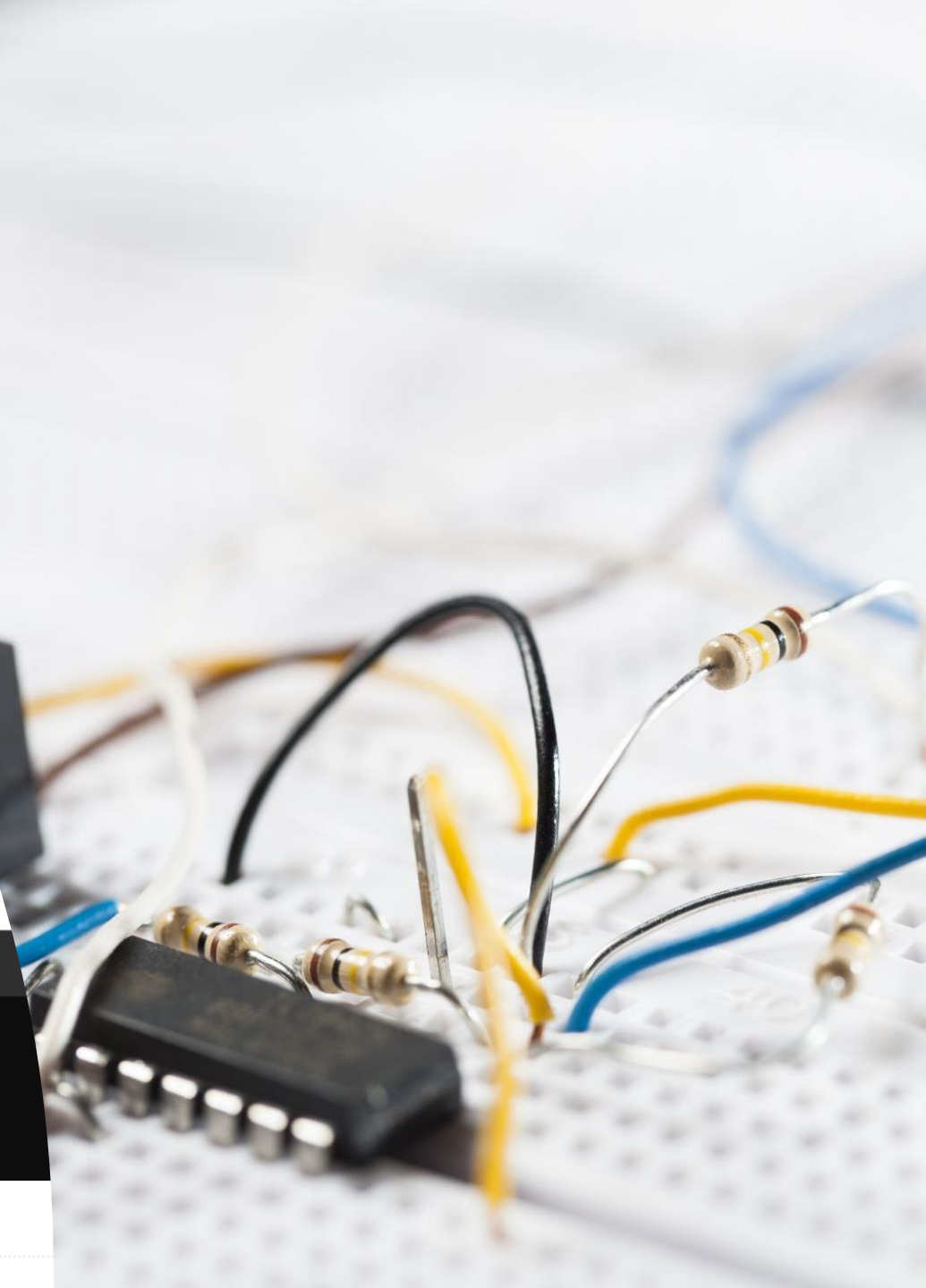
```
php

$i = 0;
while ($i < count($array)) {
    echo $array[$i];
    $i++;
}
```

- Using `for`:

```
php

for ($i = 0; $i < count($array); $i++) {
    echo $array[$i];
}
```



Add/Update/Delete Elements

- **Adding/Updating:**

```
php
```

```
$array['key'] = 'value'; // Adds or updates
```

- **Deleting:**

```
php
```

```
unset($array['key']); // Removes an element
```

Check if a value exists

- Check if key exists:

```
php  
  
array_key_exists('key', $array); // True if key exists
```

- Check if value exists:

```
php  
  
in_array('value', $array); // True if value exists
```

- Check if a key is set and not null:

```
php  
  
isset($array['key']); // True if key exists and value is not null
```



PROTIP: Rarely go for switch-case

Switch...case

```
switch ($artType) {  
  case "PT":  
    $output = "Painting";  
    break;  
  case "SC":  
    $output = "Sculpture";  
    break;  
  default:  
    $output = "Other";  
}
```

Better Alternative (most of the time): associative map

```
$artTypes = [  
  "PT" => "Painting",  
  "SC" => "Sculpture"  
];  
  
$output = $artTypes[$artType] ??  
  "Other";
```



Classes & Objects

Lecture 3



Terminology

- **Object-Oriented Programming (OOP)** allows developers to work with items (objects) that have specific properties (attributes) and methods (functions).
- Objects are created from blueprints called **classes**, which define their properties and behavior.
- Once an object (or instance) is created from a class, it holds its own set of values for its properties and operates independently from the class.

Terminology ... and why?

- **Classes should be defined in their own files** and can be included in scripts using **include**, **require**, etc.
- New objects are created using the *new* keyword, allowing as many instances as memory permits.
- **Why OOP?**
 - Encapsulation
 - Abstraction
 - Inheritance
 - Polymorphism
 - Modularity





Defining Classes

```
<?php
// Define the Car class
class Car {
    // Properties (Attributes)
    public $make;
    public $model;
    public $year;
    // Constructor method
    public function __construct($make, $model, $year) {
        $this->make = $make;
        $this->model = $model;
        $this->year = $year;
    }
    // Method to display car information
    public function displayInfo() {
        return "This car is a {$this->year} {$this->make}
{$this->model}.";
    }
}
?>
```



Instantiating Objects

```
$car1 = new Car(...);
```

- Classes are blueprints, if you want to "build" them, you need to "instantiate" objects or make "new instants" of those classes.
- The keyword *new* is used in php to instantiate an object



Properties

```
public $make;  
public $model;  
public $year;
```

- Every object can have properties, like in real life all objects have attributes/properties that make them unique or define them.
- Some modifiers can be added to properties to 1) allow for changing them in certain ways, 2) limit changing them in others
- Properties are variables related associated with an object, e.g. a human can have a name, a height, a weight, that is a class-defined property. But if YOU, as an instantiated object, would have a "certain specific" name, height and weight e.g. Amal, 156, 54.
- Object/Instance properties are accessed using **`$this->property_name`**, inside an instance
- They can also be accessed using the **`$object_name->property_name`** if they are public



Method

```
public function displayInfo() {  
    return "This car is a {$this->year} {$this->make} {$this->model}.";  
}
```

- Functions in a class are defined in the same way as functions outside a class in PHP
- These functions, inside a class, are called methods.
- Methods in a class can be access the same way as properties like **`$this->displayInfo()`** or
- Outside the class (if they are public) using the **`$object_name->displayInfo()`**



Constructor

```
public function __construct($make, $model, $year) {  
    ...  
}
```

- When you first instantiate an object, you can use that same sentence to also assign properties to it or ... do more...That function that controls this behavior is called a constructor
- In PHP, the constructor method (*functions in a class are called methods*) has to be named "__construct" and you define it like you define a normal method with parameters and a function body



Visibility

```
class Fruit {  
    public $name;  
    protected $color;  
    private $weight;  
}
```

- Visibility/Access modifiers in OOP control how you want other code to access your class
- If you want the methods/properties to be easily and simply changed, you can make them public
- If you want the methods/properties to be only changed inside a class or one that "inherits" from it, use protected
- If you want the methods/properties to be only changed inside the same class, use private

- Lab Exercise: Managing a Collection of Books with Classes and Arrays
- **Objective:**
- In this lab, you will create and manage a collection of books using PHP classes and arrays. You'll learn how to define a class, create objects, store them in an array, and apply operations on the entire collection of objects.

- Part 1: Defining the Book Class
- Create a class named Book with the following properties:
 - title (string)
 - author (string)
 - price (float)
- Define a constructor in the Book class that initializes these properties.
- Add two methods to the class:
 - `getBookInfo()`: This method should return a string that shows the book's title, author, and price.
 - `applyDiscount($percent)`: This method should reduce the price of the book by a given percentage.

Part 2: Creating and Manipulating Book Objects

- In your main script, create an array to store multiple Book objects.
 - Add at least 4 books with different titles, authors, and prices.
- Write a function `applyDiscountToAll($books, $percent)` that applies a discount to all books in the array.
- **Before applying any discount:**
 - Print the details of all books using their `getBookInfo()` method.
- Apply a 15% discount to all books by calling `applyDiscountToAll($books, 15)`.
- **After applying the discount:**
 - Print the details of all books again to see the updated prices.

Lab Exercise 2

Recommended Solution Part 1

```
<?php
class Book {
    public $title;
    public $author;
    public $price;

    // Constructor to initialize the properties
    public function __construct($title, $author,
    $price) {
        $this->title = $title;
        $this->author = $author;
        $this->price = $price;
    }

    // Method to return book information
    public function getBookInfo() {
        return "Title: {$this->title}, Author:
    {$this->author}, Price: $".number_format($this-
    >price, 2);
    }

    // Method to apply a discount to the price
    public function applyDiscount($percent) {
        $this->price -= ($this->price * ($percent
    / 100));
    }
}

// Function to apply discount to all books in the
array
function applyDiscountToAll($books, $percent) {
    foreach ($books as $book) {
        $book->applyDiscount($percent);
    }
}
?>
```

Recommended Solution

Part 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Book Collection</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <div class="container my-5">
    <h1 class="text-center mb-4">Manage Book Collection</h1>
```

```
<?php
// Create book objects and store them in an array
$books = [
    new Book('The Great Gatsby', 'F. Scott Fitzgerald', 20.99),
    new Book('1984', 'George Orwell', 15.50),
    new Book('To Kill a Mockingbird', 'Harper Lee', 18.25),
    new Book('Moby-Dick', 'Herman Melville', 22.10)
];
// Function to display all book details
function displayBooks($books) {
    echo '<div class="row">';
    foreach ($books as $book) {
        echo '<div class="col-md-6 mb-3">';
        echo '<div class="card h-100">';
        echo '<div class="card-body">';
        echo '<h5 class="card-title">' . $book->title . '</h5>';
        echo '<p class="card-text">' . $book->getBookInfo() .
    }
    echo '</div>';
}
echo '</div>';

// Display books before discount
echo '<h3>Before Applying Discount</h3>';
displayBooks($books);

// Apply 15% discount to all books
applyDiscountToAll($books, 15);

// Display books after discount
echo '<h3 class="mt-4">After Applying 15% Discount</h3>';
displayBooks($books);
?>
</div>
</body>
</html>
```