Programming PHP

Santosh Kalwar, 07.04.2022

PHP Topics

General Discussions

Re-cap of pevious lesson

Object Oriented Programming in PHP

Practice / Classroom coding

What are classes in OOP?

A blue print

A definition

A description of something in your program

A class defines two things

Properties
- Variables
- Arrays
- Data

Methods
- Functions
- Behaviors

A class defines two things

Properties

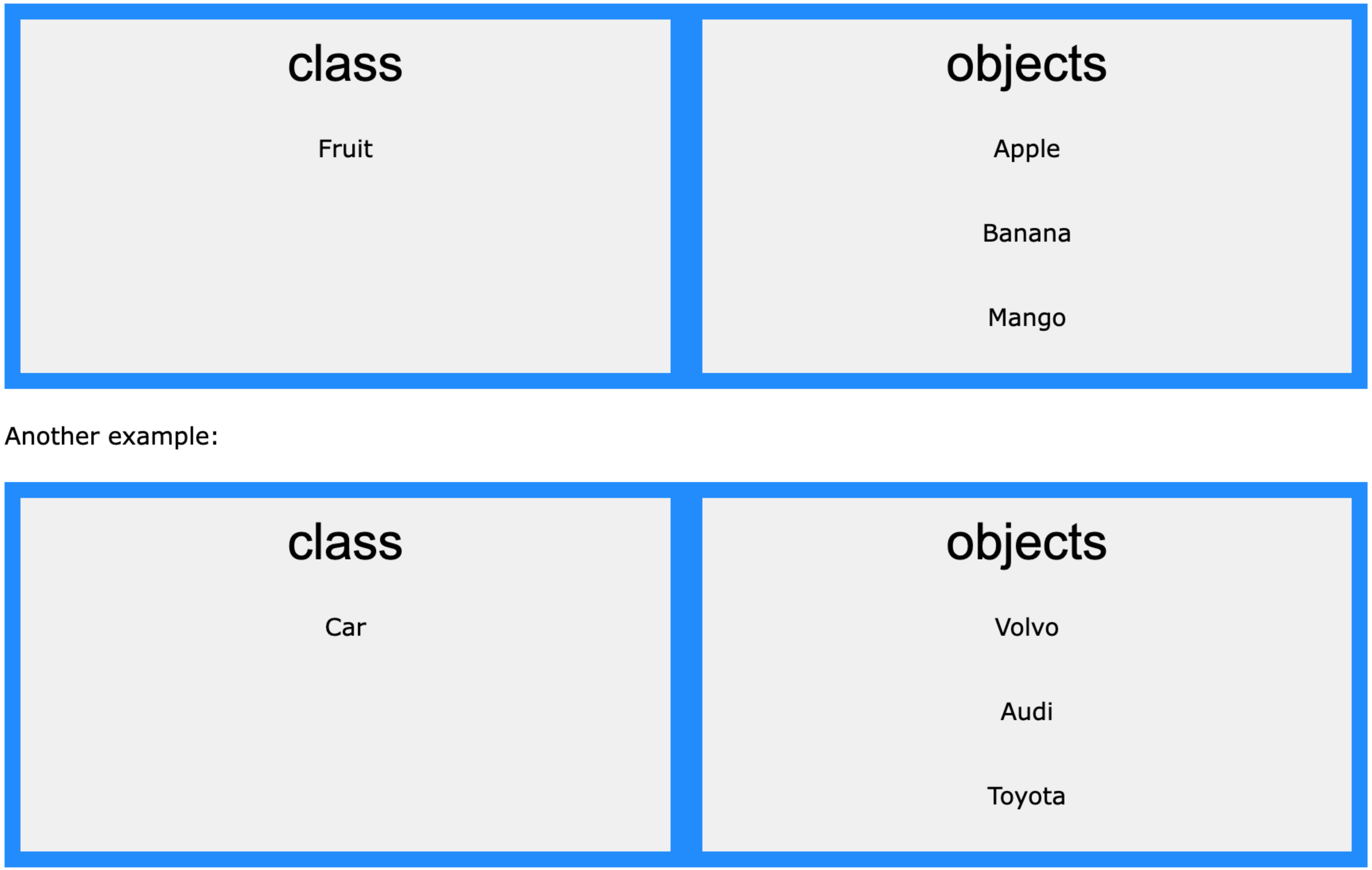
- Variables
- Arrays
- Data

Methods

- Functions
- Behaviors

So a class describes what exactly?

an object



class	objects
Fruit	Apple Banana Mango

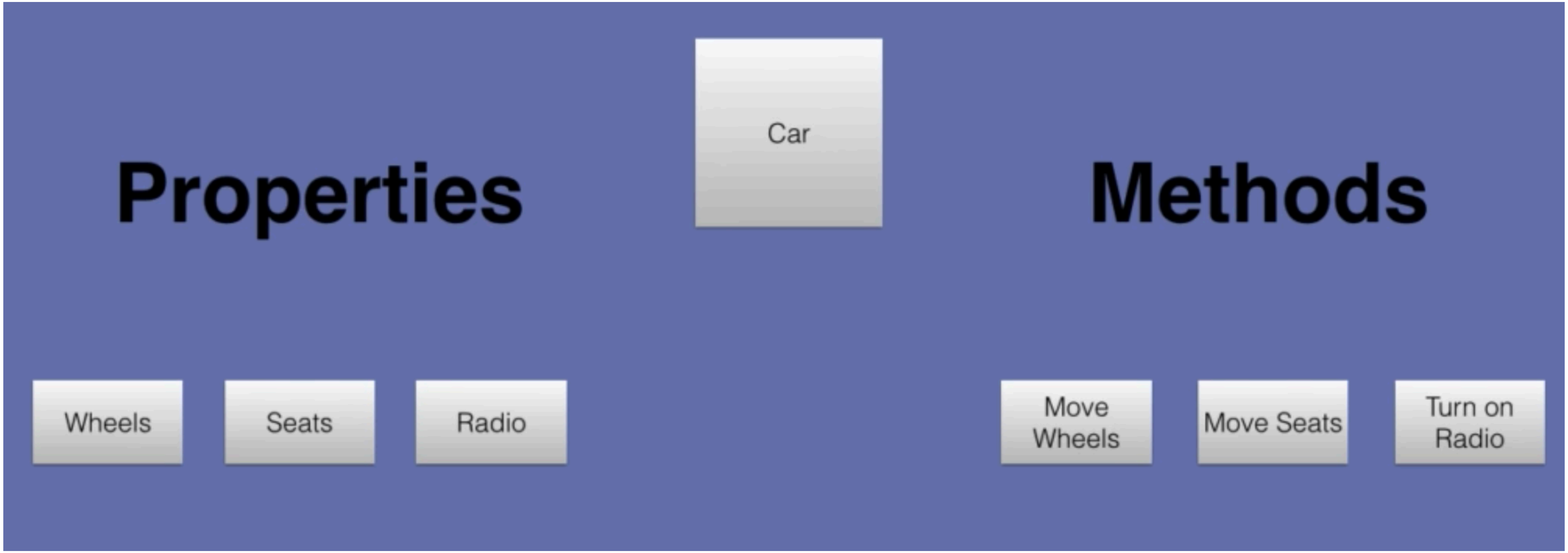
Another example:

class	objects
Car	Volvo Audi Toyota

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

And Object has Properties and Methods

e.g. object can be “Car”



Class in PHP

```php
<?php
class Fruit {
  // Properties
  public $name;
  public $color;

  // Methods
  function set_name($name) {
    $this->name = $name;
  }
  function get_name() {
    return $this->name;
  }
}
?>
```

Note: In a class, variables are called properties and functions are called methods!

Objects in PHP

Objects of a class is created using the `new` keyword.

```php
<?php
class Fruit {
  // Properties
  public $name;
  public $color;

  // Methods
  function set_name($name) {
    $this->name = $name;
  }
  function get_name() {
    return $this->name;
  }
}

$apple = new Fruit();
$banana = new Fruit();
$apple->set_name('Apple');
$banana->set_name('Banana');

echo $apple->get_name();
echo "<br>";
echo $banana->get_name();
?>
```

PHP - $this keyword

The $this keyword refers to the current object, and is only available inside methods.

```php
<?php
class Fruit {
  public $name;
}
$apple = new Fruit();
?>
```

So, where can we change the value of the $name property?

PHP - $this keyword

Two approaches:

1. Inside the class (by adding a set_name() method and use $this):

```
<?php
class Fruit {
  public $name;
  function set_name($name) {
    $this->name = $name;
  }
}
$apple = new Fruit();
$apple->set_name("Apple");

echo $apple->name;
?>
```

PHP - $this keyword

2. Outside the class (by directly changing the property value):

```php
<?php
class Fruit {
  public $name;
}
$apple = new Fruit();
$apple->name = "Apple";

echo $apple->name;
?>
```

PHP - instanceof

You can use the `instanceof` keyword to check if an object belongs to a specific class:

```php
<?php
$apple = new Fruit();
var_dump($apple instanceof Fruit);
?>
```

PHP - constructor

A constructor allows you to initialize an object's properties upon creation of the object.

If you create a `__construct()` function, PHP will automatically call this function when you create an object from a class.

```php
<?php
class Fruit {
  public $name;
  public $color;

  function __construct($name) {
    $this->name = $name;
  }
  function get_name() {
    return $this->name;
  }
}

$apple = new Fruit("Apple");
echo $apple->get_name();
?>
```

Notice that the construct function starts with two underscores (__)!

PHP - access modifiers

There are three access modifiers:

- `public` - the property or method can be accessed from everywhere. This is default
- `protected` - the property or method can be accessed within the class and by classes derived from that class
- `private` - the property or method can ONLY be accessed within the class

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

$mango = new Fruit();
$mango->name = 'Mango'; // OK
$mango->color = 'Yellow'; // ERROR
$mango->weight = '300'; // ERROR
?>
```

PHP - static data in classes

Static properties can be called directly - without creating an instance of a class.

Static properties are declared with the `static` keyword:

```php
<?php
class ClassName {
  public static $staticProp = “Programming PHP";
}
?>
```

To access a static property use the class name, double colon (::), and the property name:

```php
ClassName::$staticProp;
```

```php
<?php
class pi {
  public static $value = 3.14159;
}

// Get static property
echo pi::$value;
?>
```

PHP - static data in classes

Static methods can be called directly - without creating an instance of the class first.

Static methods are declared with the `static` keyword:

```
<?php
class ClassName {
  public static function staticMethod() {
    echo "Hello World!";
  }
}
?>
```

To access a static method use the class name, double colon (::), and the method name:

```
ClassName::staticMethod();
<?php
class greeting {
  public static function welcome() {
    echo "Hello World!";
  }
}
// Call static method
greeting::welcome();
?>
```

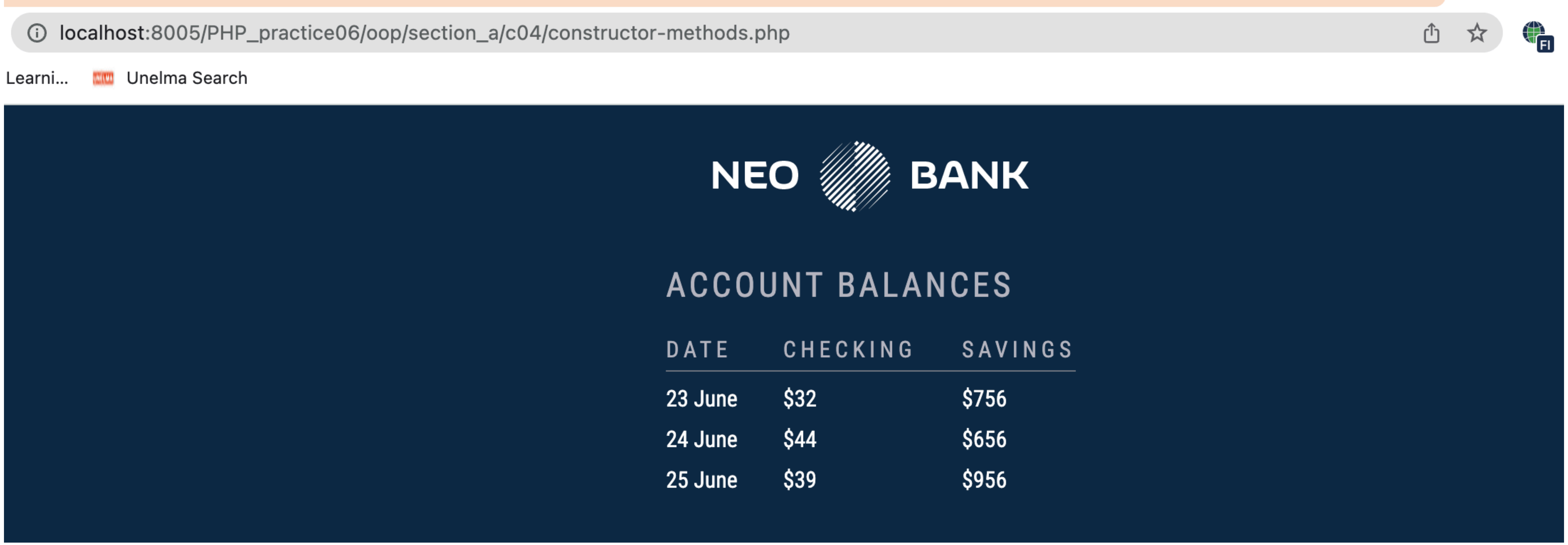
Chapter Summary

- Objects group together variables and functions that represent something from the world around us.
- In an object, the variables are called properties and the functions are called methods.
- A class is used as a template to create objects.
- Class definitions set out the properties and methods that each object created using that class will have.
- The __construct() method runs when an object is created. It can be used to put values into properties.
- $this access a property or method of *this* object.
- Properties can be declared as public (can be accessed by code outside the object) or protected (can only be used by code inside the object).
- Classes and objects help organize, re-use, maintain, and share code more effectively.

Practice / Classroom Coding

- **PHP_practice01/10.php**:
  - Create a simple class e.g. Dog in PHP
  - Set some properties, make methods, instantiate the class

- **PHP_practice06/oop/section_a/c04/constructor-methods.php**
  - Write PHP code using constructors with a class as shown below



localhost:8005/PHP_practice06/oop/section_a/c04/constructor-methods.php
Learni...
Unelma Search
NEO BANK

ACCOUNT BALANCES

DATE	CHECKING	SAVINGS
23 June	$32	$756
24 June	$44	$656
25 June	$39	$956