

# DMIT2503 – Advanced Web Concepts

---

## OOP PHP Lesson 2

For this lesson, we will be working in the folder called *02-shapes*. Please save your files in this folder, then upload to your web server account (or local server will suffice). There is no database required for OOP lessons.

Create a file called *shapes.php* in the root of this folder. Create a PHP block at the top and a basic HTML template with a new PHP block below.

*Note: If your copy/paste from this, you may have to change certain characters, especially double quotes.*

```
<?php

?>

<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
<h1>OOP Shapes</h1>
<?php

?>
</body>
</html>
```

In the PHP block at the top, we will create a new Class. Remember, this Class will be a template for any new Object that we create from it.

We will also initialize some properties. Note that here we can also give them a default value. We may overwrite this value, but if not, then the default will apply.

```
<?php
class myShape {
    var $shape_height = 100;
    var $shape_width = 100;
    var $shape_color = '#EC0000';
}
?>
```

# DMIT2503 – Advanced Web Concepts

---

```
<?php
$square1 = new myShape();

?>
```

Upload this, and you will see nothing. Just checking for errors. We have however created a new object from the myShape class. Let's do something with it.

**First, let's create a simple square.**

Step 1: We will create some methods that set the values of some properties.

```
<?php
class myShape {
    var $shape_height = 100;
    var $shape_width = 100;
    var $shape_color = '#EC0000';

    function setHeight($value){
        $this->shape_height=$value;
    }
    function setWidth($value){
        $this->shape_width=$value;
    }
    function setColor($value){
        $this->shape_color=$value;
    }
}
?>
```

And then we can call those methods from the object.

```
<h1>OOP Shapes</h1>
<?php
$square1 = new myShape();
$square1->setHeight(300);
$square1->setWidth(300);
$square1->setColor('#00F');

?>
```

# DMIT2503 – Advanced Web Concepts

---

Ok, time to see something. Let's create a method that displays a square. Our object can then access that and we should see a square on our page.

```
<?php
// add this to our class
function displaysquare(){

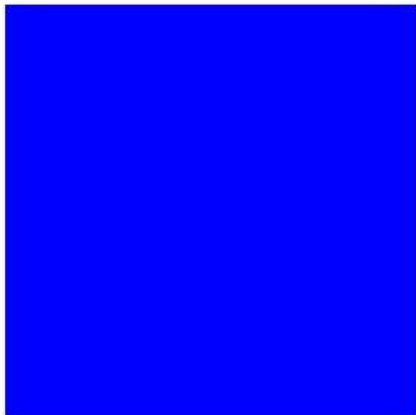
    $squareToDisplay = "\n<div style=\"height:\" . $this->shape_height . "px;";
    $squareToDisplay .= "width:\" . $this->shape_width . "px;";
    $squareToDisplay .= "background-color:\" . $this->shape_color . "\"\" . "> </div>";
    echo $squareToDisplay ;
}
?>
```

And then we can call those methods from the object.

```
// add this in your body PHP block for your object
$square1->displaysquare();
```

And voila, we should see a nice square on our page.

## OOP Shapes



# DMIT2503 – Advanced Web Concepts

---

## Student Activities

Here are some tasks to try:

1. In the same file, create a new instance of your class (a new object) called *rectangle1*. Make it rectangular (duh) and green.
2. Create a new method (function) in your class called *displaycircle()*. This can do everything that the *displaysquare* can do, but add some CSS to make the shape a circle. Try *border-radius:50%*.
3. Finally, create a separate folder and file for the class. Create a new document includes your class file and creates some shape.

**Note that what we call *properties* are really *variables*, and what we call *methods* are really *functions*. Much of OOP is just terminology and packaging what we already know in to a more organized and reusable style.**

Here's the whole file (without the separate class file above in step 3) in case you (or I) made a mistake along the way.

Cheers !

```
<?php
class myShape {
    var $shape_height = 100;
    var $shape_width = 100;
    var $shape_color = '#EC0000';

    function setHeight($value){
        $this->shape_height=$value;
    }
    function setWidth($value){
        $this->shape_width=$value;
    }
    function setColor($value){
        $this->shape_color=$value;
    }
    function displaysquare(){

        $squareToDisplay = "\n<div style=\"height:\" . $this->shape_height . "px";
        $squareToDisplay .= "width:\" . $this->shape_width . "px";
        $squareToDisplay .= "background-color:\" . $this->shape_color . "\"\" . "> </div>";
        echo $squareToDisplay ;
    }
    function displaycircle(){

        $circleToDisplay = "\n<div style=\"height:\" . $this->shape_height . "px";
```

# DMIT2503 – Advanced Web Concepts

---

```
$circleToDisplay .= "width:" . $this->shape_width . "px;";
$circleToDisplay .= "border-radius:50%;";
$circleToDisplay .= "background-color:" . $this->shape_color . "\" . "> </div>";
echo $circleToDisplay ;
}
}
?>

<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
    <h1>OOP Shapes</h1>
<?php

$square1 = new myShape();
$square1->setHeight(300);
$square1->setWidth(300);
$square1->setColor('#00F');
$square1->displaysquare();

$rect1 = new myShape();
$rect1->setHeight(200);
$rect1->setWidth(400);
$rect1->setColor('#0f0');
$rect1->displaysquare();

$circle1=new myShape();
$circle1->setHeight(300);
$circle1->setWidth(300);
$circle1->setColor('#f00');
$circle1->displaycircle();

?>
</body>
</html>
```