

Variable Variables

This lesson introduces variable variables, also called dynamic variables, and discusses how you can use them in PHP.

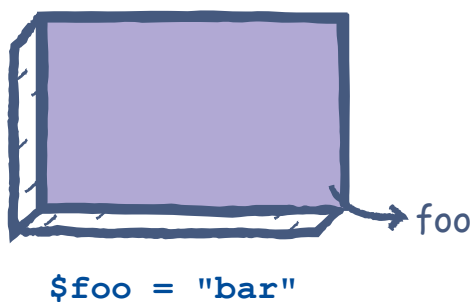
We'll cover the following ^

- What are Variable Variables?
- Implementation
- Quick Quiz

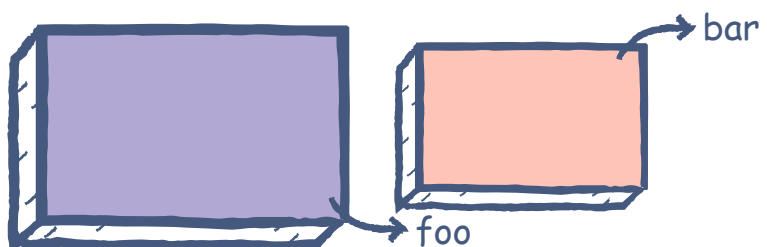
What are Variable Variables?

Using PHP, we can access data through dynamic variable names. The name of a variable can be stored in another variable, allowing it to be accessed dynamically. Such variables are known as *variable variables*.

To turn a variable into a *variable* variable, you put an extra **\$** sign in front of your variable. This method is illustrated in the following figure:

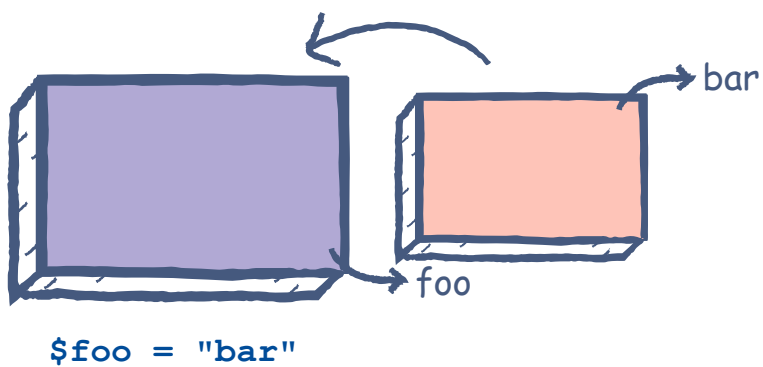


1 of 9

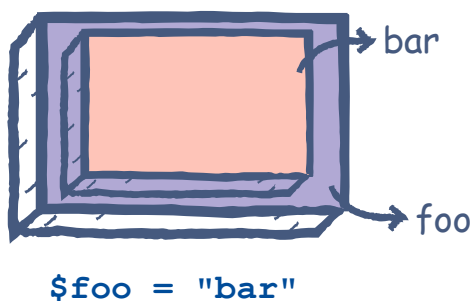


```
$foo = "bar"
```

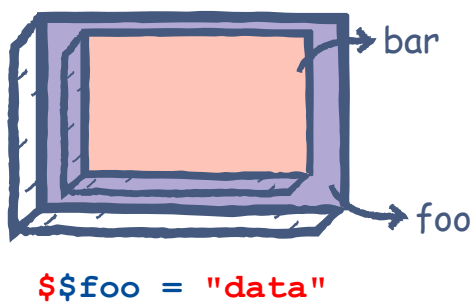
2 of 9



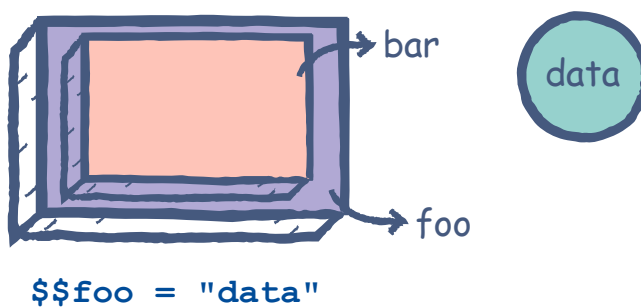
3 of 9



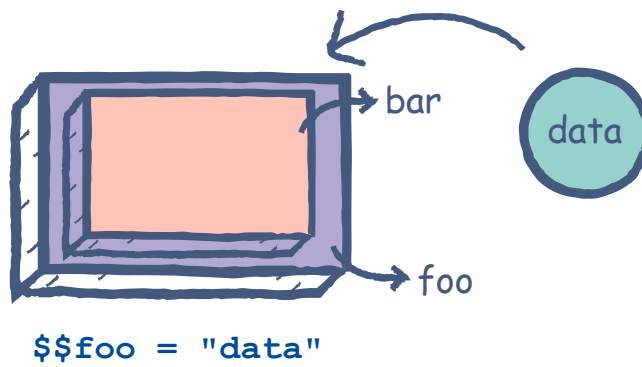
4 of 9



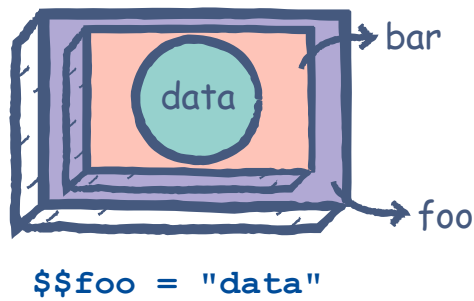
5 of 9



6 of 9

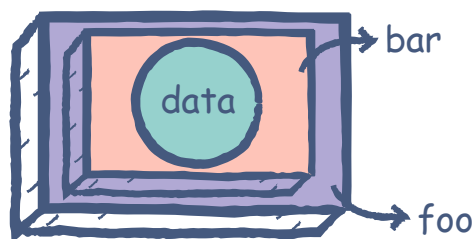


7 of 9



A variable containing a variable

8 of 9



9 of 9

—

☐

Implementation

The following code snippet shows how this is done in PHP:

```
<?php
$foo = "bar"; // foo has value "bar"
```



```

$$foo = "data"; // bar has value "data"
echo "\$foo:\t";
echo $foo; //prints bar

echo "\n";

echo "\${$foo}:\t";
echo ${$foo}; //prints data
echo "\n";

echo "\$$foo:\t";
echo $$foo; //prints data
echo "\n";

echo "\$bar:\t";
echo $bar; //prints data
echo "\n";
?>

```



Note: It is possible, but not required to put the variable name, say `$foo`, between `{}`.

The following examples are both equivalents and print the output “baz”:

```

<?php
$fooBar = 'baz';
$varPrefix = 'foo';

echo "\$fooBar:\t\t\t\t";
echo $fooBar; // Outputs "baz"
echo "\n";

echo "\${$varPrefix . 'Bar'}:\t\t"; //varPrefix will give value foo and the dot operator concatenates
                                     //Putting $ before fooBar makes it a variable variable.
echo ${$varPrefix . 'Bar'}; // Also outputs "baz"
?>

```



Using `{}` is only mandatory when the name of the variable is itself an expression, like this:

```

<?php
${$variableNamePart1 . $variableNamePart2} = $value;
?>

```



It is nevertheless recommended to always use `{}` because it's more readable.

Quick Quiz



What will the following code output?

```
<?php
$kangaroo = "Kanga";
$$kangaroo = "Roo";
echo "Hi, I'm ".$kangaroo." and this is ".$Kanga.".";
?>
```

[Retake Quiz](#)

Now that we've learnt about variables, let's move to PHP constants in the next lesson.