

Strings

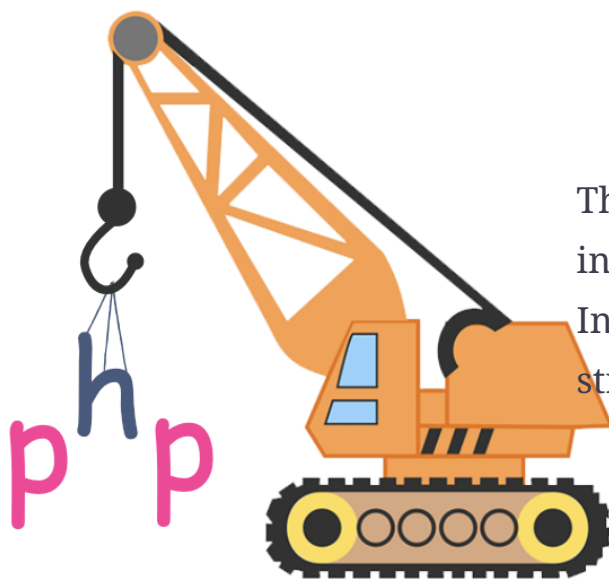
This lesson discusses string interpolation using examples.

We'll cover the following ^

- String Interpolation
- Curly Syntax

We have already covered the creation of strings in the [previous lesson](#). Here, we will discuss the various techniques to format and manipulate strings.

String Interpolation



The term interpolation refers to the insertion of a variable within a string. Interpolation works in double-quoted strings and the [here doc syntax](#) only.

```
<?php
$name = 'Joel';
echo "Hello $name, Nice to see you."; // $name will be replaced with `Joel`
?>
```



Notice that in line 2 of the above code, we used double quotes. If we use single quotes, the compiler will output `$name` as the raw text (without interpreting it). This is shown in the following code snippet:

```
<?php
$name = 'Joel';
echo 'Hello $name, Nice to see you.'; //does not interpret $name as Joel
?>
```



Curly Syntax

In PHP you can also use the **complex syntax**, or the **curly syntax** which requires that you wrap your variable within curly braces `{}`. This can be useful when embedding variables within the text and helps prevent possible ambiguity between text and variables. This is shown in the following code snippet:

```
<?php
$name = 'Joel';
// Example using the curly brace syntax for the variable $name
echo "We need more {$name}s to help us!\n";
?>
```



However you will get an error in the following case:

```
<?php
// This line will throw an error (as `$names` is not defined)
echo "We need more $names to help us!";
?>
```

The `{}` syntax only interpolates variables starting with a `$` into a string.

```
<?php
// trying to interpolate a PHP expression
echo "1 + 2 = {1 + 2}\n";

// using a constant
define("HELLO_WORLD", "Hello World!!");
echo "My constant is {HELLO_WORLD}.\n";

// Example using a function
function say_hello() {
    return "Hello!";
};
echo "I say: {say_hello()}";
?>
```





As you can see in the above example the expressions in **line 2**, **line 6** and **line 12** are written inside `{}` however the `{}` syntax does not evaluate arbitrary PHP expressions which is why the expressions are printed as it is on the console instead of the result of those expressions being printed.

In the next lesson we will discuss string operators.