



Previous examples have used 'echo' to output from the php, technically echo is a 'language construct' which allows Developers to build commands to be executed, the main difference between a language construct and a function, is for purposes of this course, the way they are used, functions require brackets to be placed after the 'call' to the function, whereas language constructs require, where necessary, quotations.

The echo construct is a very important component because it gives the Developers the ability to output from the php values which have been determined within it, for example a simple word, a variable or even a HTML statement.



If echo is used to output a single string, then that string must be enclosed by double or single quotes, it is important to note that while the following would output exactly the same result, the use of single quotes with variables or escape characters may not produce the expected result, the differing results are examined in the following slides ...

```
<?php
```

```
    echo "here";
```

```
    echo 'here';
```

```
?>
```

Note also that each echo command is terminated by a semi-colon, ';'. This is the case for all processing in PHP.

## PHP Structure – Concatenating Echo

---



Even though the previous echo example was written on two lines, the output actually came out as follows ...

*herehere*

To place the output with a space between, the Developer merely had to place a space after the first word or before the second ...

```
<?php
```

```
    echo "here ";
```

```
    echo 'here';
```

```
?>
```

## PHP Structure – Concatenating Echo

---



In reality, the Developer may have to output a whole series of different datatypes on one line, to achieve this they can use the concatenation symbol ... a full stop or period, this effectively glues different strings or variables together ...

Here the Developer is joining the word 'here' with a space followed by another 'here', more complicated examples will be demonstrated later in the course !

```
<?php
```

```
    echo "here" . " " . "here";
```

```
?>
```

## PHP Structure – Use of Echo with Escape Characters

---



Echo can be used with escape characters providing the string is enclosed with double quotes not single, in this example the Developer wishes to display the results on two lines using the same string and the '\n' escape character, unfortunately, this can only be tested when writing to files ...

```
<?php
```

```
    echo "This is on more than\none line";
```

```
?>
```

If however the Developer had used single quotes the results are unexpected in HTML ...

# PHP Structure – Use of Echo with Escape Characters

---



```
<?php
```

```
    echo 'This is on more than\none line';
```

```
?>
```

Produces ...

*This is on more than\none line*

Single quotes will amongst other things, ignore escape characters.

## PHP Structure – Use of Echo with Escape Characters

---



On occasions the Developer may wish to display a backslash in the output, to achieve this a second backslash is used this effectively making the backslash a literal, or of course use single quotes ...

```
<?php
```

```
    echo "To write a line feed we use \\n";  
    echo 'To write a line feed we use \n';
```

```
?>
```

This outputs ...

```
To write a line feed we use \nTo write a line feed we use \n
```

## PHP Structure – Use of Echo with Escape Characters

---



One more comment about the use of double quotes with echo, it will allow the use of single quotes within the output with using any escape characters to make it literal ...

```
<?php
```

```
    echo "She's leaving home, bye bye";
```

```
?>
```

*She's leaving home, bye bye*



## PHP Structure – Use of Echo with Escape Characters

---



If the process needs double quotes within the statement then a slash can be used ...

```
<?php
```

```
    echo "She said \"yes\", I say \"no\"";
```

```
?>
```

*She said "yes", I say "no"*

The reverse can be said of using single quotes ... please refer to the following slide for the examples ...

# PHP Structure – Use of Echo with Escape Characters

---



```
<?php
```

```
    echo 'She\'s leaving home, bye bye';
```

```
?>
```

Produces

*She's leaving home, bye bye*

```
<?php
```

```
    echo 'She said "yes", I say "no" ';
```

```
?>
```

Produces

*She said "yes", I say "no"*

# PHP Structure – Use of Echo with Calculations

---



PHP can be used to perform calculations with numbers ...

```
<?php  
    echo 15 / 3;  
?>
```

Results in ...

5

The common calculation symbols are +, -, / and \*

% will display the modulus of a calculation, 10%3 will return 1

## PHP Structure – Use of Echo with Calculations

---



All computer systems suffer from 'precedence' in that a division will always be performed before addition, in this example the Developer wanted 10 and 15 to be added together then divided by 5, however the result proves how precedence can cause unexpected results ...

```
<?php
```

```
    echo 10 + 15 / 3;
```

```
?>
```

```
15
```

## PHP Structure – Use of Echo with Calculations

---



What has happened in the previous example is that the 15 was divided by 3 before being added to the 10, to make sure that calculation are displayed correctly, the Developer must use brackets which will force the program to evaluate what is in the them first before performing any other calculations ...

```
<?php
```

```
    echo (10 + 15) / 3;
```

```
?>
```

```
8.33333333333333
```