Alternative syntax for control structures

PHP offers an alternative syntax for some of its control structures; namely, `if`, `while`, `for`, `foreach`, `switch`, and `declare`. In each case, the basic form of the alternate syntax is to change the opening brace to a colon (:) and the closing brace to `endif;`, `endwhile;`, `endfor;`, `endforeach;`, `endswitch;`, or `enddeclare;`, respectively.

```
<?php
$a = 5;
if ($a == 5): ?>
A is equal to 5
<?php endif; ?>
```

ENTITY

```
A is equal to 5
```

In the above example, the HTML block "A is equal to 5" is nested within an `if` statement written in the alternative syntax. The HTML block would be displayed only if `$a` is equal to 5.

The alternative syntax applies to `else` and `elseif` as well. The following is an `if` structure with `elseif` and `else` in the alternative format:

```
<?php
$a = 5;
if ($a == 5):
    echo "a equals 5";
    echo "...";
elseif ($a == 6):
    echo "a equals 6";
    echo "!!!";
else:
```

```
    echo "a is neither 5 nor 6";
endif;
```

ENTITY

```
a equals 5...
```

> [!NOTE] Mixing syntaxes in the same control block is not supported.

> [!WARNING] Any output (including whitespace) between a `switch` statement and the first `case` will result in a syntax error. For example, this is invalid:
>
> ```
> <?php
> $foo = 1;
> switch ($foo): ?>
>     <?php case 1: ?>
>     Foo is 1
> <?php endswitch; ?>
> ```
>
> ENTITY
>
> ```
> Parse error: syntax error, unexpected T_INLINE_HTML "    ", expecting
> "endswitch" or "case" or "default" in script on line 4
> ```
>
> Whereas this is valid, as the trailing newline after the `switch` statement is considered part of the closing `?ENTITY` and hence nothing is output between the `switch` and `case`:
>
> ```
> <?php
> $foo = 1;
> switch ($foo): ?>
> <?php case 1: ?>
>     Foo is 1
> ```

```
<?php endswitch; ?>
```

ENTITY

```
    Foo is 1
```

See also while, for, and if for further examples.

□□□ #2
□ □□□ □□□ 2026-10-31 17:38:29 UTC
□ □□□ □□□ 2026-10-31 17:38:30 UTC