

Slide 3.2: PHP syntax
Slide 3.4: PHP strings
[Home](#)



PHP Variables

All variables in PHP start with a \$ sign symbol. The script shows the variable type can be changed while program executes.

```
<html><body>
<?php
  $x = 10; echo "$x<br />";
  $x = "Hello!"; echo $x;
?>
</body></html>
```

PHP Is a Loosely Typed Language

In PHP a variable does not need to be declared before being set. PHP converts the variable to the correct data type automatically, depending on how they are set. To concatenate two or more variables together, use the dot (.) operator. There are two [string operators](#): the string concatenation operation (‘.’) and the concatenating assignment operator (‘.=’). The script shows variables do not need to be strings. They could be integer or real numbers or others.

```
<html><body>
<?php
  $txt = "Hello, World!";
  $num = 1000 + $x;
  echo $txt . " " . $num ;
?>
</body></html>
```

\$x =

Variable Naming Rules

- A variable name must start with a letter or an underscore ‘_’.

- A variable name can only contain alpha-numeric characters and underscores (a-Z, 0-9, and _).
- A variable name should not contain spaces. If a variable name should be more than one word, it should be separated with underscore such as `$my_string` or with capitalization such as `$myString`.

[Slide 3.2: PHP syntax](#)

[Slide 3.4: PHP strings](#)

[Home](#)

