Tishk International University
Faculty of Applied Science
Information Technology Department

Variables and Pre-defined Functions

Web Programming

Lecture 4

Fall 2025

Course Code: IT349

Grade 3

Islam Abdulazeez
islam.abdulaziz@tiu.edu.iq

October 22, 2025

Outlines

- Introduction to Variables
- PHP Syntax and Case Sensitivity
- Variables and Data Types
- Expressions and Operators
- Built-in Number and String Functions
- Hashing Functions

Learning Outcomes

- At the end of today's session, you will be able to:

  - ✓ **Define PHP variables and syntax rules.**
  - ✓ **Identify PHP data types and their uses.**
  - ✓ **Apply variables and operators in simple programs.**
  - ✓ **Use built-in PHP functions for data processing.**

Variables

- A **variable** is a container used to store data that can change or be used later in your program

Variable Syntax

- Rules for PHP variables: $name

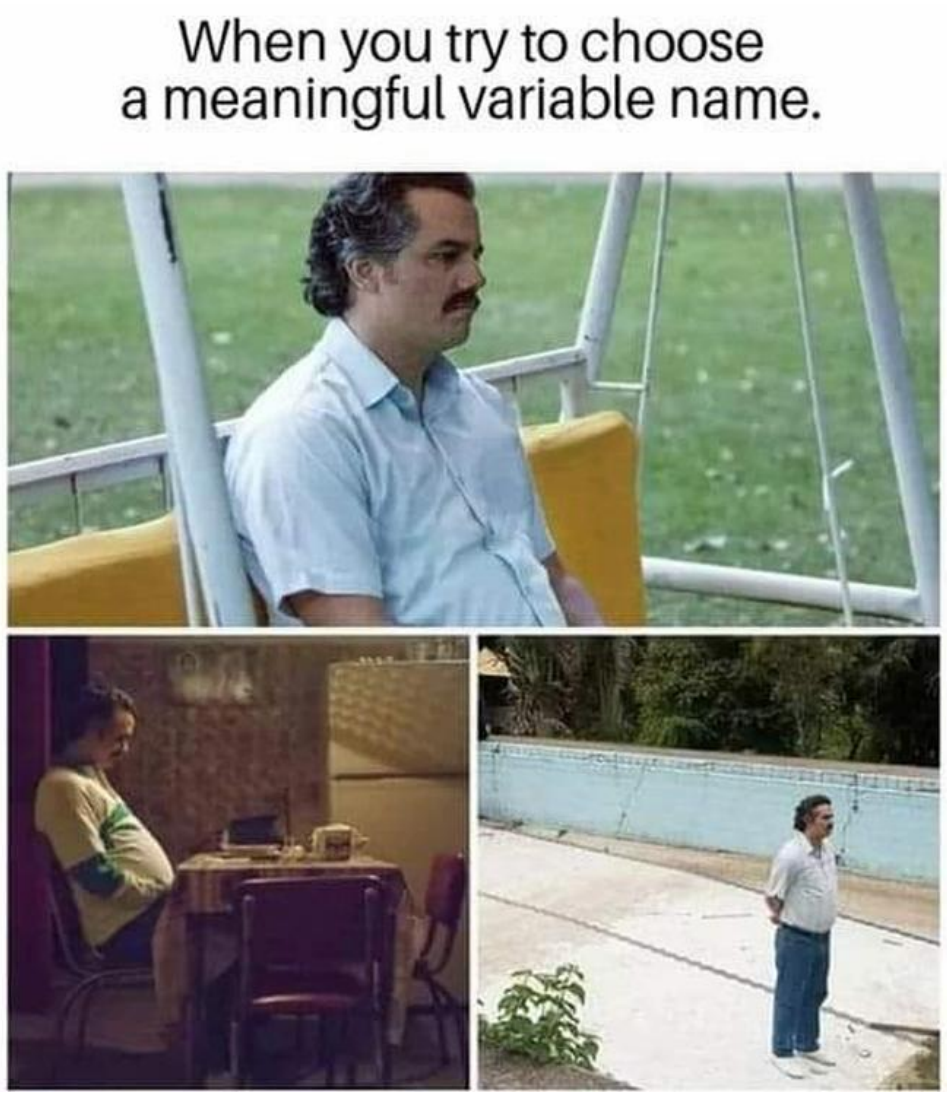
  - ✓ A variable starts with the $ sign, followed by the name of the variable
  - ✓ A variable name must start with a letter or the underscore character
  - ✓ A variable name cannot start with a number
  - ✓ A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
  - ✓ Variable names are case-sensitive ($age and $AGE are two different variables)

Variable (Case Sensitivity)

- $name, $Name, and $NAME are different variables.

```php
<?php
$name = "Ahmad";
$Name = "Zhila";
$NAME = "Paywand";
?>
```

Meme ☺



Variables (Valid & Invalid)

Valid variables in PHP

Name
$first_name
$person
$address1
$_SERVER

Name	Valid?	Reason
$first name	No	Space
$first.name	No	dot
first_name	No	Dollar Sign?
$1address	No	Number
@email	No	Dollar Sign?
$first_name	Yes	☺
$Wêb	No	Accented character
_android	No	Dollar Sign?
$4	No	Can’t start with a num
$four	Yes	☺
$	No	Missing variable name

Types of Variables

- PHP provides eight types of values, or data types:
  - ✓ String
  - ✓ Integer
  - ✓ Float (floating point numbers - also called double)
  - ✓ Boolean
  - ✓ Array
  - ✓ Object
  - ✓ NULL
  - ✓ Resource

Variables (example)

```php
<?php
$name = "Dara";
$gpa = 3.2;
$course = "Web Programming";

echo "My name is $name, I love $course, and My GPA is $gpa";
?>
```

My name is Dara, I love Web Programming, and My GPA is 3.2

Integer

- An integer is a numeric data type that represents whole numbers, either positive or negative, without any fractional or decimal part.
- In PHP, the range of a 32-bit integer must lie between -2,147,483,648 and 2,147,483,647.
- In PHP, the range of a 64-bit integer must lie between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807.

Example

```php
<?php
    $posNum = 155;
    $negNum = -270;

    echo "Positive Number: $posNum <br>";
    echo "Negative Number: $negNum";
?>
```

// Positive Number: 155
// Negative Number: -270

Float & Double

- A floating-point number is a number that contains a decimal point.

```php
<?php
    $floatNum = 10.5;
    echo "The float number is: $floatNum <br>";

    $floatNum2 = -3.74;
    echo "The second float number is: $floatNum2";
?>
```

// The float number is: 10.5
// The second float number is: -3.74

Valid & Invalid Numbers

- Valid Numbers in PHP

Number	Type
1	Integer
1.0	Floating-point
1970	Integer
19.70	Floating-point
-1	Integer
-1.0	Floating-point

- Invalid Numbers in PHP

Number	Reason
1_3	Underscore
1970A	Contains a non-numeric character
1.2.4	More than one decimal point

Expressions and Operators

- An expression in PHP is a piece of code that can be evaluated to produce a value.

Operator	Operation
*	Multiplication
/	Division
%	Modulus
+	Addition
-	Subtraction
.	String concatenation
<, <=	Less than, less than or equal
>, >=	Greater than, greater than or equal
=	Assignment

Operator	Operation
!	Logical NOT
==	Value equality
!=	Inequality
===	Equality in type and value
&&	Logical AND
||	Logical OR
++	Increment
--	Decrement
+=, −=, *=, /=, .=	Assignment with operation

Equality and Identity

- Equality (==): It compares values only and checks whether they are equal after type conversion, if needed.
- Identity (===): It compares both the values and the data types, returning true only if they are exactly the same.

```php
<?php
$a = 10;
$b = "10";
?>
```

```
$a == $b      // True
$a === $b     // False
```

Expressions and Operators

- Addition (+)
- Subtraction (-)

```php
<?php
$num1 = 10;
$num2 = 5;
$sum = $num1 + $num2;

echo "The sum of $num1 and $num2 is: $sum";
?>
```

// The sum of 10 and 5 is: 15

```php
<?php
$num1 = 10;
$num2 = 5;
$result = $num1 - $num2;
echo "The result is:" . $result;
?>
```

// The result is: 5

Expressions and Operators

- Multiplications (*)
- Division (/)
- Modulus (%)

```php
<?php
$num1 = 10;
$num2 = 5;

$multi = $num1 * $num2;
echo $multi . "<br>";          // 50

$div = $num1 / $num2;
echo $div . "<br>";            // 2

$mod = $num1 % $num2;
echo $mod;                     // 0
?>
```

Auto-Increment and Auto-Decrement

- In programming, one of the most common operations is to increase or decrease the value of a variable by one.

Operator	Name	
$var++	Post-increment	Returns $var, then increments $var by one
++$var	Pre-increment	Increments $var by one, then returns $var
$var--	Post-decrement	Returns $var, then decrements $var by one
--$var	Pre-decrement	Decrements $var by one, then returns $var

Examples

```php
<?php
    $num = 10;
    echo $num++ . "<br>";
    echo $num;
?>
```

// 10
// 11

```php
<?php
    $num = 10;
    echo ++$num . "<br>";
    echo $num;
?>
```

// 11
// 11

Example

```php
<?php
    $num = 10;
    echo $num++ . "<br>";
    echo ++$num;
?>
```

// 10

// 12

Examples

```php
<?php
    $num = 10;
    $num++;
    echo $num;
?>
```

// 11

```php
<?php
    $num = 10;
    ++$num;
    echo $num;
?>
```

// 11

Number Format

- number_format() is a built-in function used to format numbers and making numeric output more readable or appropriate.

```php
<?php
    echo number_format(271.5314, 3);
?>
```

// 271.531

Examples

```php
<?php
    echo number_format(271, 3);
?>
```

// 271.000

```php
<?php
    echo number_format(27153821);
?>
```

// 27,153,821

Round

- round() in PHP is a function that rounds a number to the nearest integer or to a specified number of decimal places.

```php
<?php
    echo round(4.7);
?>
```

// 5

```php
<?php
    $num = 481.2391;
    echo round($num);
?>
```

// 481

Random Numbers

- rand() is a function that generates a random integer.

```php
<?php
    echo rand();
?>
```

// 3454842

```php
<?php
    echo rand(0,10);
?>
```

// 7

Boolean

A Boolean value represents a truth value, it says whether something is true or not. Like most programming languages, PHP defines some values as true and others as false.

```php
<?php
    $x = true;
    $y = false;
?>
```

Boolean

In PHP, the following values all evaluate to **false**:

- The keyword **false**
- The integer **0**
- The floating-point value **0.0**
- The empty string **("")** and the string **"0"**
- An array with **zero elements**
- The **NULL** value

A value that is not false is true.

Strings

A string is any number of characters enclosed within a pair of either single (') or double (") quotation marks. Strings can contain any combination of characters that exist: letters, numbers, symbols, and spaces. Strings can also contain variables.

Strings

- Examples of valid string values:

```
"Hello, World!"
"Hello, $username!"
'Hello, $username!'
"4/2"
'How are you today?'
"21.10.2025"
"2000"
''
```

Strings

- There are cases, however, where you may run into problems. For example:

```
"I said, "How are you?""
```

Question: How can we solve this issue, and show the double quotation mark in the output?

Strings

- Solution 1:

```
'I said, "How are you?"'
```

- Solution 2:

```
"I said, \"How are you?\""
```

String Type

```php
$uni_name = "Tishk International University";
echo $uni_name[9];
```

- In PHP, strings can be accessed like arrays, where each character in the string has an index starting from 0.
- In this example, $uni_name[9] accesses the character at index 9 of the string "Tishk International University".
- If we count the indices from 0, the character at index 9 is “e"

Interpreted strings

- In PHP, interpreted strings are strings enclosed in double quotes (" ").

```php
<?php
 $uni_name = "TIU";
 echo "I am studying at $uni_name";
 echo 'I am studying at $uni_name';
?>
```

I am studying at TIU

I am studying at $uni_name

- Strings inside " " are interpreted.
- Strings inside ' ' are not interpreted.

String Concatenation Operator

- To concatenate two string variables together, use the dot (.) operator.

```php
<?php
$first_name = "Paywand";
$last_name = "Mustafa";
echo $first_name . " " . $last_name;
?>
```

This will produce the following output: Paywand Mustafa

Example

```php
<?php
 $uni_name = "TIU";
 echo "I am studying at $uni_name";
 echo 'I am studying at $uni_name';
 echo "I am studying at ".$uni_name;
 echo 'I am studying at '.$uni_name;
?>
```

// I am studying at TIU

// I am studying at $uni_name

// I am studying at TIU

// I am studying at TIU

Example

```php
<?php
 $first_name = "Dler";
 $last_name = "Ahmad";
 echo "My name is " . $first_name . " " . $last_name;
?>
```

This will produce the following output: My name is Dler Ahmad

strlen()

- The strlen() function is used to find the length of a string.

```php
<?php
 $uni_name = "Tishk International University";
 echo strlen($uni_name);
?>
```

// 30

str_word_count()

- The PHP str_word_count() function counts the number of words in a string.

```php
<?php
 $uni_name = "Tishk International University";
 echo str_word_count($uni_name);
?>
```

```
// 3
```

str_replace()

- The PHP str_replace() function replaces specified characters or substrings with other characters or substrings in a string.

```php
<?php
 echo str_replace("Design", "Programming", "Web Design is fun!");
?>
```

// Web Programming is fun!

More String Functions

- strtolower(): Make a string lowercase.
- strtoupper(): Make a string uppercase.
- ucfirst(): Make a string's first character uppercase.
- ucwords(): Uppercase the first character of each word in a string.

Examples

```php
<?php
$coffeeLevel = "Super strong";
echo strtolower($coffeeLevel);
echo strtoupper($coffeeLevel);

echo ucfirst("hello world");

$favouriteFruit = "i like pineapple";
echo ucwords($favouriteFruit);
?>
```

```
// super strong
// SUPER STRONG

// Hello world

// I Like Pineapple
```

strpos()

- The strpos() function is used to find the position of the first occurrence of a substring within a string.

```php
<?php
    echo strpos("Hello World!", "World");
?>
```

// 6

substr()

- substr() is a string function that is used to extract a part of a string.

  substr(string, start, length)

  - string → the original string
  - start → starting index (0-based)
  - length → (optional) number of characters to extract

Examples

```php
<?php
    echo substr("PHP is my favorite language", 7);
?>
```

// my favorite language

```php
<?php
    echo substr("PHP is my favorite language", 7, 11);
?>
```

// my favorite

Hashing

- Message Digest 5 or md5() is a cryptographic hash function, a mathematical algorithm that takes an input (like a string or file) and produces a fixed length output.

```php
<?php
    $pass = "12345";
    echo md5($pass);
?>
```

// 827ccb0eea8a706c4c34a16891f84e7b

Hashing

- SHA-1 (Secure Hash Algorithm 1) is another cryptographic hash function, similar in purpose to MD5 but slightly more secure.

```php
<?php
    $uni = "TIU";
    echo sha1($uni);
?>
```

// c903596c8cca36b78834139172d6458d9f5a2565

Examples

```php
<?php
    $uni = "TIU";
    echo md5($uni);
    echo sha1($uni);
?>
```

// f2ea286934636861b2ee4d50e5d8c848

//c903596c8cca36b78834139172d6458d9f5a2565

Lab Assesments and Next Session's Topic

Lab Assessments

- Lab Exercises
- Quiz 2 (Theoretical)

Next Session's Topic

- Control Structures and User-defined Functions

References

- Tatroe, K., & MacIntyre, P. (2020). Programming PHP: Creating dynamic web pages (4th ed.). O'Reilly Media.
- Ullman, L. (2016). PHP for the web: Visual QuickStart guide (5th ed.). Peachpit Press.
- PHP Group. (n.d.). PHP: Hypertext Preprocessor — official documentation. Retrieved October 19, 2025, from https://www.php.net

Thank You!