

Internet Web Site Design

PHP Data Types

- Variables can store data of different types, and different data types can do different things.
- PHP supports the following data types:
 - String
 - Integer
 - Float (floating point numbers – also called double)
 - Boolean
 - Array
 - NULL
 - Object
 - Resource

PHP String

- A string is a sequence of characters, like "Hello world!".
- A string can be any text inside quotes. You can use single or double quotes:

PHP Code	Result
<pre><!DOCTYPE html> <html> <body> <?php \$x = "Hello world!"; \$y = 'Hello PHP!'; echo \$x; echo "
"; echo \$y; ?> </body> </html></pre>	Hello world! Hello PHP!

PHP Integer

- An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.

Rules for integers:

- An integer must have at least one digit.
- An integer must not have a decimal point.
- An integer can be either positive or negative.

Internet Web Site Design

- Integers can be specified in three formats: decimal (10-based), hexadecimal (16-based – prefixed with 0x) or octal (8-based – prefixed with 0).
- In the following example \$x is an integer. The PHP var_dump() function returns the data type and value:

PHP Code	Result
<pre><?php \$x = 40; //decimal number var_dump(\$x); echo '
'; \$x = 057; //octal number (equivalent to 47 decimal) var_dump(\$x); echo '
'; \$x = 0x4B; //hexadecimal number(equivalent to 75 decimal) var_dump(\$x); ?></pre>	<pre>int(40) int(47) int(75)</pre>

PHP Float

- A float (floating point number) is a number with a decimal point or a number in exponential form.
- In the following example \$x is a float. The PHP var_dump() function returns the data type and value:

PHP Code	Result
<pre><!DOCTYPE html> <html> <body> <?php \$x = 10.365; var_dump(\$x); ?> </body> </html></pre>	<pre>float(10.365)</pre>

PHP Boolean

- A Boolean represents two possible states: TRUE or FALSE.
 - \$x = true;
 - \$y = false;
- Booleans are often used in conditional testing. You will learn more about conditional testing later.

Internet Web Site Design

PHP Array

- An array stores multiple values in one single variable.
- In the following example \$cars is an array. The PHP var_dump() function returns the data type and value:

PHP Code	<pre><!DOCTYPE html> <html> <body> <?php \$cars = array("Volvo", "BMW", "Toyota"); var_dump(\$cars); ?> </body> </html></pre>
Result	<pre>array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" }</pre>

PHP NULL Value

- Null is a special data type which can have only one value: NULL.
- A variable of data type NULL is a variable that has no value assigned to it.
- If a variable is created without a value, it is automatically assigned a value of NULL.
- Variables can also be emptied by setting the value to NULL:

PHP Code	Result
<pre><!DOCTYPE html> <html> <body> <?php \$x = "Hello world!"; \$x = null; var_dump(\$x); ?> </body> </html></pre>	NULL

Internet Web Site Design

PHP 7 String Functions

- In this chapter we will look at some commonly used functions to manipulate strings.

Get The Length of a String

- The PHP `strlen()` function returns the length of a string.
- The example below returns the length of the string "Hello world!":

PHP Code	Result
<pre><!DOCTYPE html> <html> <body> <?php echo strlen("Hello"); ?> </body> </html></pre>	5

Count The Number of Words in a String

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

PHP Code	Result
<pre><!DOCTYPE html> <html> <body> <?php echo str_word_count("Hello World!"); ?> </body> </html></pre>	2

Reverse a String

- The PHP `strrev()` function reverses a string:

PHP Code	Result
<pre><!DOCTYPE html> <html> <body> <?php echo strrev("Hello World!"); ?> </body> </html></pre>	!dlroW olleH

Internet Web Site Design

Search For a Specific Text Within a String

- The PHP strpos() function searches for a specific text within a string.
- If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.
- The example below searches for the text "world" in the string "Hello world!":

PHP Code	Result
<pre><!DOCTYPE html> <html> <body> <?php echo strpos("Hello World!", "World"); ?> </body> </html></pre>	6

Note: The first character position in a string is 0 (not 1).

Replace Text Within a String

The PHP str_replace() function replaces some characters with some other characters in a string.

The example below replaces the text "world" with "Dolly":

PHP Code	Result
<pre><!DOCTYPE html> <html> <body> <?php echo str_replace("world", "Dolly", "Hello world!"); ?> </body> </html></pre>	Hello Dolly!

Internet Web Site Design

PHP 7 Operators

- Operators are used to perform operations on variables and values.
- PHP divides the operators in the following groups:
 - Arithmetic operators
 - Assignment operators
 - Comparison operators
 - Increment/Decrement operators
 - Logical operators
 - String operators
 - Array operators

Arithmetic Operators

- The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power

Assignment Operators

- The PHP assignment operators are used with numeric values to write a value to a variable.
- The basic assignment operator in PHP is "=". It means that the left operand gets set to the value of the assignment expression on the right.

Operator	Same as	Description
$x = y$	$x = y$	The left operand gets set to the value of the expression on the right
$x += y$	$x = x + y$	Addition
$x -= y$	$x = x - y$	Subtraction
$x *= y$	$x = x * y$	Multiplication
$x /= y$	$x = x / y$	Division
$x \% = y$	$x = x \% y$	Modulus

Internet Web Site Design

Comparison Operators

– The PHP comparison operators are used to compare two values (number or string):

Operator	Name	Example	Result
==	Equal	<code>\$x == \$y</code>	Returns true if \$x is equal to \$y
===	Identical	<code>\$x === \$y</code>	Returns true if \$x is equal to \$y, and they are of the same type
!= or <>	Not equal	<code>\$x != \$y</code>	Returns true if \$x is not equal to \$y
!==	Not identical	<code>\$x !== \$y</code>	Returns true if \$x is not equal to \$y, or they are not of the same type
>	Greater than	<code>\$x > \$y</code>	Returns true if \$x is greater than \$y
<	Less than	<code>\$x < \$y</code>	Returns true if \$x is less than \$y
>=	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if \$x is greater than or equal to \$y
<=	Less than or equal to	<code>\$x <= \$y</code>	Returns true if \$x is less than or equal to \$y
<=>	Spaceship	<code>\$x <=> \$y</code>	Returns an integer less than, equal to, or greater than zero, depending on if \$x is less than, equal to, or greater than \$y. Introduced in PHP 7.

PHP Code	Result
<pre>!DOCTYPE html> <html> <body> <?php \$x = 5; \$y = 10; echo (\$x <=> \$y); // returns -1 because \$x is less than \$y echo "
"; \$x = 10; \$y = 10; echo (\$x <=> \$y); // returns 0 because values are equal echo "
"; \$x = 15; \$y = 10; echo (\$x <=> \$y); // returns +1 because \$x is greater than \$y ?> </body> </html></pre>	<p>-1 0 1</p>

Internet Web Site Design

Increment / Decrement Operators

- The PHP increment operators are used to increment a variable's value.
- The PHP decrement operators are used to decrement a variable's value.

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

Logical Operators

- The PHP logical operators are used to combine conditional statements.

Operator	Name	Example	Result
And	And	\$x and \$y	True if both \$x and \$y are true
Or	Or	\$x or \$y	True if either \$x or \$y is true
Xor	Xor	\$x xor \$y	True if either \$x or \$y is true, but not both
&&	And	\$x && \$y	True if both \$x and \$y are true
	Or	\$x \$y	True if either \$x or \$y is true
!	Not	!\$x	True if \$x is not true

String Operators

- PHP has two operators that are specially designed for strings.

Operator	Name	Example	Result
.	Concatenation	\$txt1 . \$txt2	Concatenation of \$txt1 and \$txt2
.=	Concatenation assignment	\$txt1 .= \$txt2	Appends \$txt2 to \$txt1

Array Operators

- The PHP array operators are used to compare arrays.

Operator	Name	Example	Result
+	Union	\$x + \$y	Union of \$x and \$y
==	Equality	\$x == \$y	Returns true if \$x and \$y have the same key/value pairs
===	Identity	\$x === \$y	Returns true if \$x and \$y have the same key/value pairs in the same order and of the same types
!=	Inequality	\$x != \$y	Returns true if \$x is not equal to \$y
<>	Inequality	\$x <> \$y	Returns true if \$x is not equal to \$y
!==	Non-identity	\$x !== \$y	Returns true if \$x is not identical to \$y

Internet Web Site Design

PHP Conditional Assignment Operators

The PHP conditional assignment operators are used to set a value depending on conditions:

Operator	Name	Example	Result
?:	Ternary	<code>\$x = expr1 ? expr2 : expr3</code>	Returns the value of <code>\$x</code> . The value of <code>\$x</code> is <code>expr2</code> if <code>expr1 = TRUE</code> . The value of <code>\$x</code> is <code>expr3</code> if <code>expr1 = FALSE</code>
??	Null coalescing	<code>\$x = expr1 ?? expr2</code>	Returns the value of <code>\$x</code> . The value of <code>\$x</code> is <code>expr1</code> if <code>expr1</code> exists, and is not NULL. If <code>expr1</code> does not exist, or is NULL, the value of <code>\$x</code> is <code>expr2</code> . Introduced in PHP 7

PHP Code	Result
<pre><!DOCTYPE html> <html> <body> <?php \$x= 5; \$y=10; echo \$z = (\$x==\$y) ? 100 : 200; echo("
"); \$y=5; echo \$z = (\$x==\$y) ? 100 : 200; ?> </body> </html></pre>	<p>200</p> <p>100</p>

PHP Code	Result
<pre><!DOCTYPE html> <html> <body> <?php \$x="Amjad"; echo \$user = \$x ?? "anonymous"; echo("
"); echo \$color = \$y ?? "red"; ?> </body></html></pre>	<p>Amjad</p> <p>red</p>