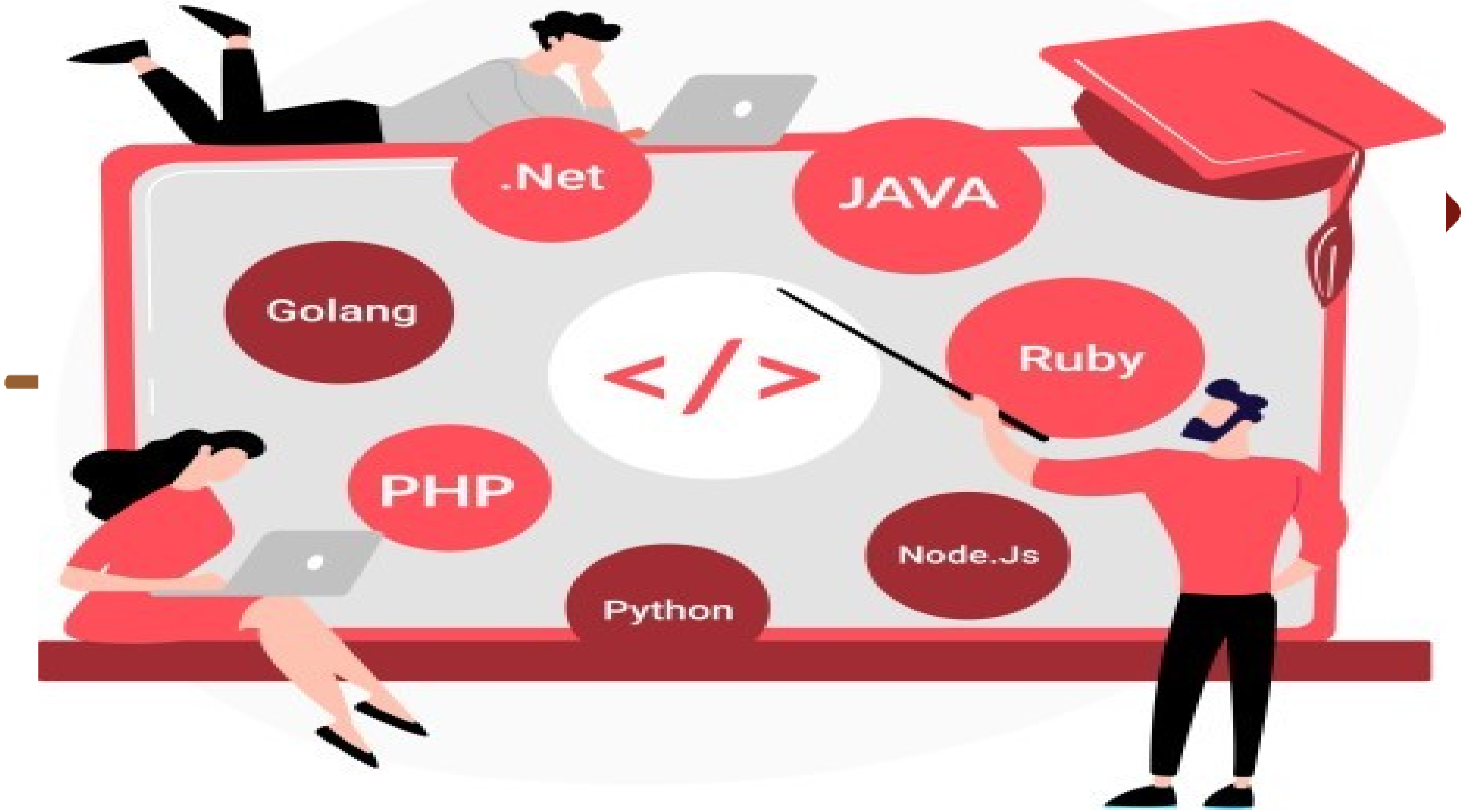
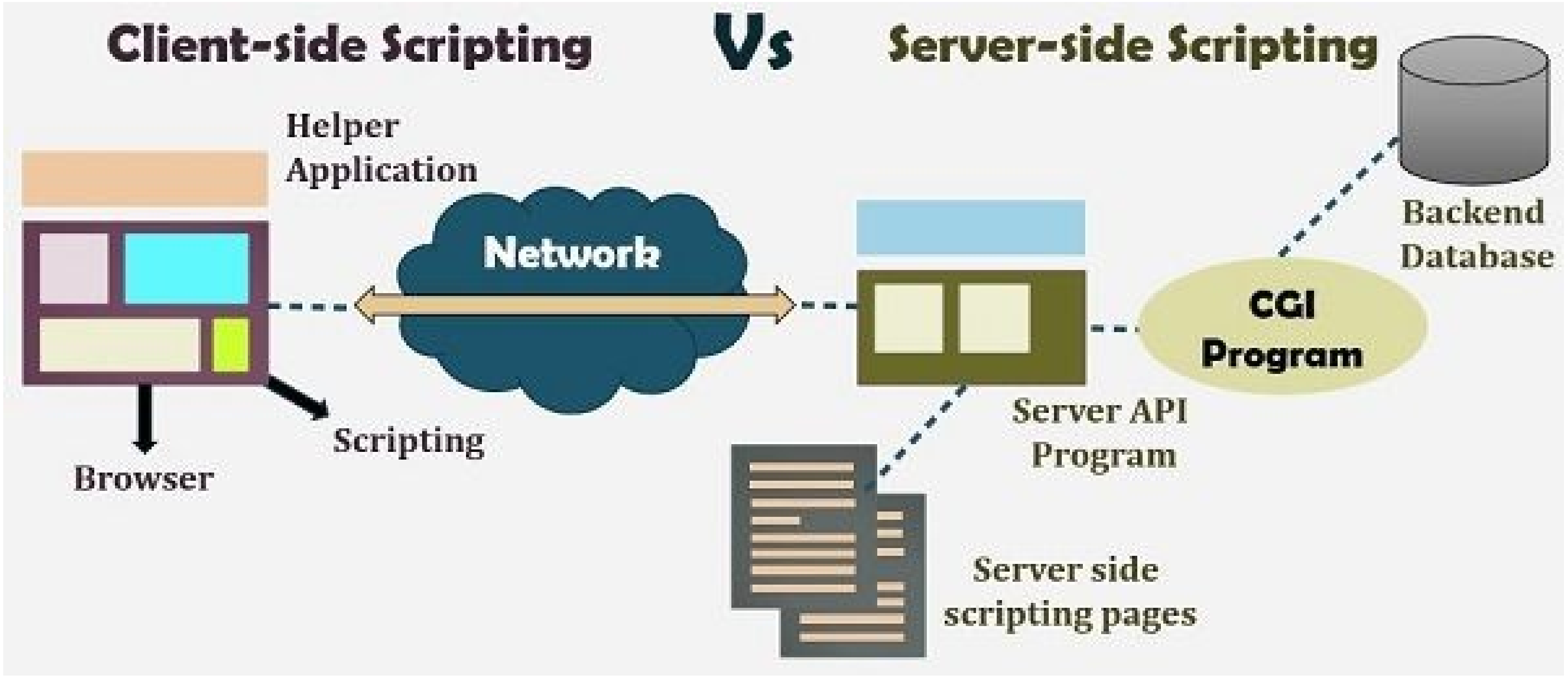
Web Technology II (BIT301)

Instructor: Prakash Neupane


.Net
JAVA
Golang
</>
Ruby
PHP
Node.Js
Python

Introduction



Introduction

- Server-side scripting is a technique of programming for producing the code which can run software on the server side, in simple words any scripting or programming that can run on the web server is known as server-side scripting.
- The operations like customization of a website, dynamic change in the website content, response generation to the user's queries, accessing the database, and so on are performed at the server end.
-

Introduction

- Server-side scripting refers to the practice of running scripts on a web server to generate dynamic web content that is then sent to the client's browser.
- This is in contrast to client-side scripting, where scripts are executed directly in the browser to modify or enhance the user interface.
-

Introduction

- PHP:
- Purpose and Definition:
  - PHP stands for "Hypertext Preprocessor." It is a server-side scripting language primarily used for web development.
  - PHP is embedded within HTML code to create dynamic web pages and interact with databases.
- History and Origins:
  - PHP was created by Rasmus Lerdorf in 1994 as a collection of Perl scripts to track visits to his online resume.
  - It evolved into a more powerful scripting language and was released as "PHP/FI" (Personal Home Page/Forms Interpreter) in 1995.
  - The PHP/FI codebase was rewritten, leading to PHP 3 in 1998, which introduced the parser written in C.

Introduction

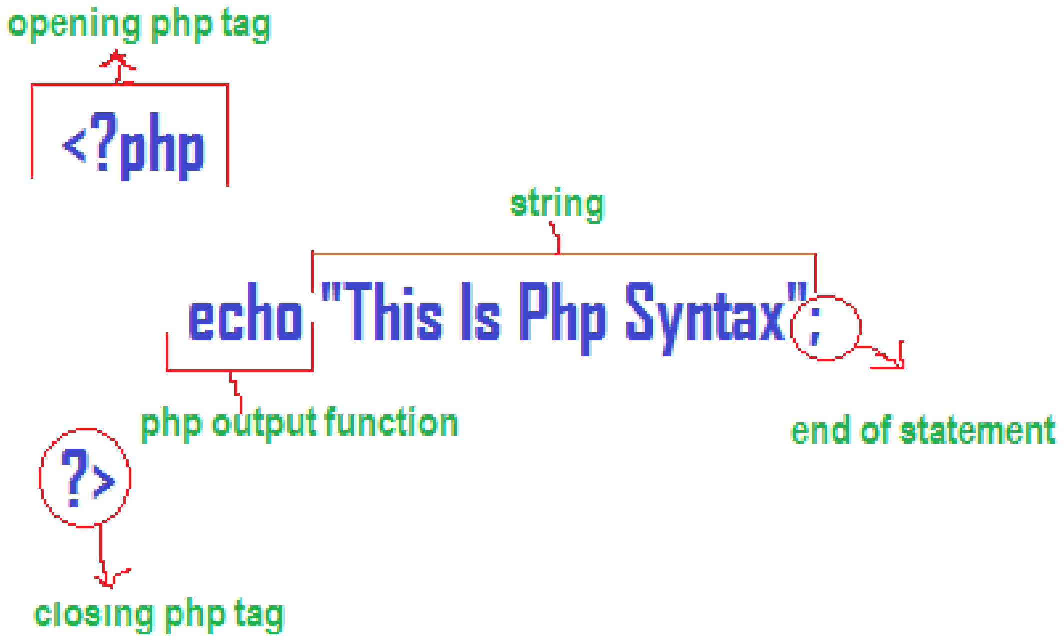
- PHP:
- Milestone Versions:
  - PHP 4, released in 2000, brought improved performance and support for object-oriented programming (OOP).
  - PHP 5, released in 2004, introduced significant OOP enhancements, including the Zend Engine, which improved performance and allowed for better memory management.
- Modern Development:
  - PHP 7, released in 2015, marked a substantial leap in performance and efficiency, with the Zend Engine 3. It also introduced features like scalar type declarations and return type declarations.
  - Subsequent PHP 7.x versions brought more improvements and features, emphasizing speed and security.

Introduction

- PHP:
- Current State and Usage:
  - PHP is among the most widely used server-side scripting languages, powering a substantial portion of the web.
  - Popular CMSs like WordPress, Joomla, and Drupal, as well as frameworks like Laravel and Symfony, are built using PHP.
  - The PHP community actively maintains and updates the language, with PHP 8 (released in 2020) introducing JIT (Just-In-Time) compilation and other enhancements.

Introduction

- PHP: Language Basics



```
<?php
echo "This Is Php Syntax";
?>
```

Introduction:PHP: Language Basics

- Lexical Structure:
  - the set of basic rules that governs how you write programs in that language.
  - It is the lowest-level syntax of the language
  - specifies such things as
    - what variable names look like,
    - what characters are used for comments, and
    - how program statements are separated from each other.

Introduction:PHP: Language Basics

- Lexical Structure:
  - Case Sensitivity
  - Statements and Semicolons
  - Whitespace and Line Breaks
  - Comments
  - Literals
  - Identifiers
  - Keywords

Introduction:PHP: Language Basics

- Lexical Structure:
  - Data Types
    - PHP provides eight types of values, or data types.
    - Four are scalar (single-value) types: integers, floating-point numbers, strings, and Booleans.
    - Two are compound (collection) types: arrays and objects.
    - The remaining two are special types: resource and NULL.

Introduction:PHP: Language Basics

- Lexical Structure:
  - Variables
    - Variables in PHP are identifiers prefixed with a dollar sign ($). For example:
      - $name
      - $Age
      - $_debugging
      - $MAXIMUM_IMPACT

Introduction:PHP: Language Basics

- Lexical Structure:
  - Variables
    - A variable may hold a value of any type.
    - There is no compile-time or runtime type checking on variables.
      - $what = "Fred";
      - $what = 35;
      - $what = array("Fred", 35, "Wilma");
    - There is no explicit syntax for declaring variables in PHP.
    - The first time the value of a variable is set, the variable is created in memory.
    - In other words, setting a value to a variable also functions as a declaration.

Introduction:PHP: Language Basics

- Lexical Structure:
  - Variables
    - For example, is this a valid complete PHP program?
      - $day = 60 * 60 * 24;
      - echo "There are {$day} seconds in a day.";
    - There are 86400 seconds in a day.

Introduction:PHP: Language Basics

- Lexical Structure:
  - Variables Variables
    - You can reference the value of a variable whose name is stored in another variable by prefacing the variable reference with an additional dollar sign ($).
    - For example:
      - $foo = "bar";
      - $$foo = "echo $bar will print me!";
    - After the second statement executes, the variable $bar has the value "echo $bar will print me!".

Introduction:PHP: Language Basics

- Lexical Structure:
  - Variable References
    - In PHP, references are how you create variable aliases or pointers.
    - To make $black an alias for the variable $white, use:
      - $black =& $white;
    - The old value of $black, if any, is lost.
    - Instead, $black is now another name for the value that is stored in $white.

Introduction:PHP: Language Basics

- Lexical Structure:
  - Variable Scope
    - The scope of a variable, which is controlled by the location of the variable’s declaration, determines those parts of the program that can access it.
    - There are four types of variable scope in PHP:
      - local,
      - global,
      - static, and
      - function parameters.

Introduction:PHP: Language Basics

- Lexical Structure:
  - Arithmetic Operator

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

Introduction:PHP: Language Basics

- Lexical Structure:
  - Assignment Operator

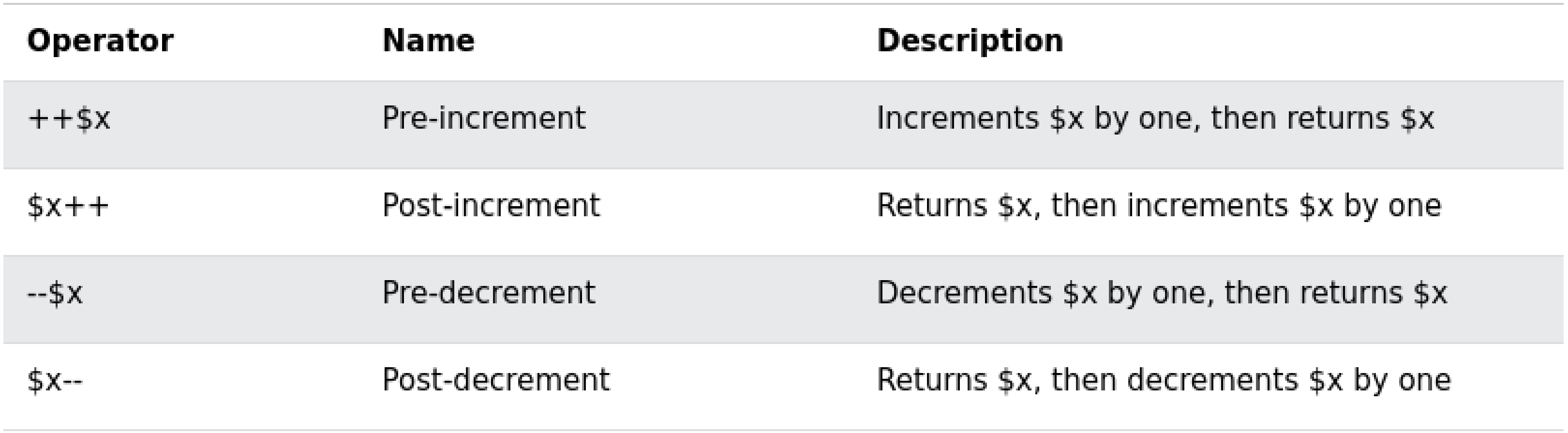
Assignment	Same as...	Description
x = y	x = y	The left operand gets set to the value of the expression on the right
x += y	x = x + y	Addition
x -= y	x = x - y	Subtraction
x *= y	x = x * y	Multiplication
x /= y	x = x / y	Division
x %= y	x = x % y	Modulus

Introduction:PHP: Language Basics
Comparison Operator

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

Introduction:PHP: Language Basics

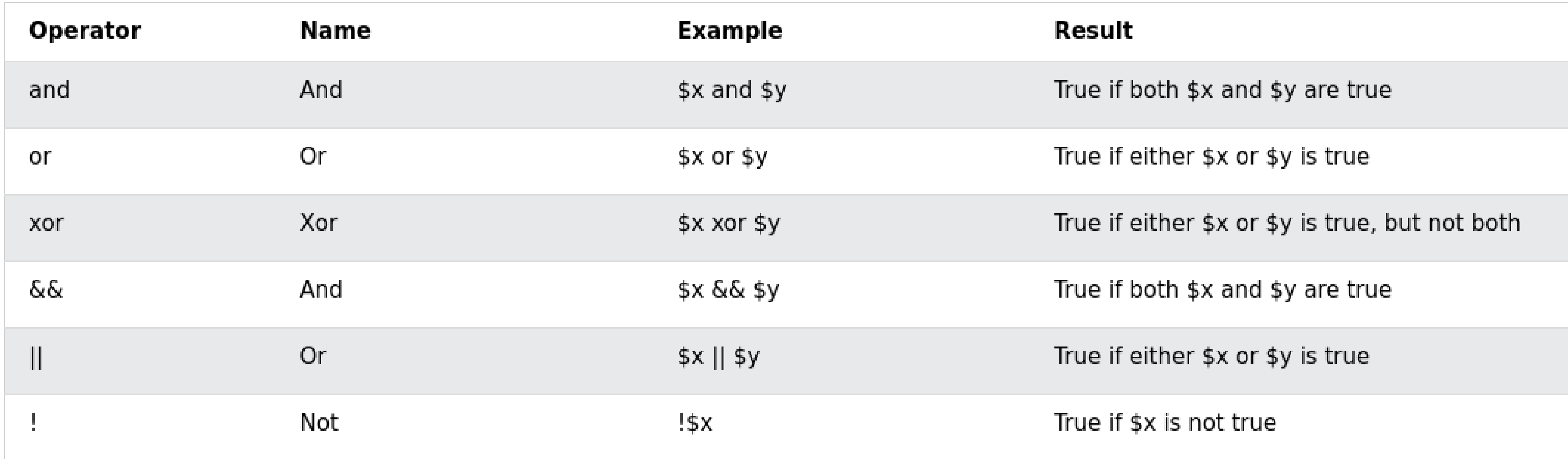
- Lexical Structure:
  - Increment and Decrement Operator



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

Introduction:PHP: Language Basics

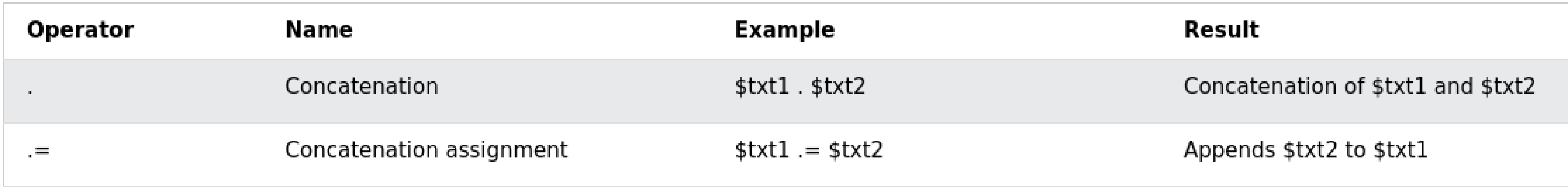
- Lexical Structure:
  - Logical Operator



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

Introduction:PHP: Language Basics

- Lexical Structure:
  - String Operator



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

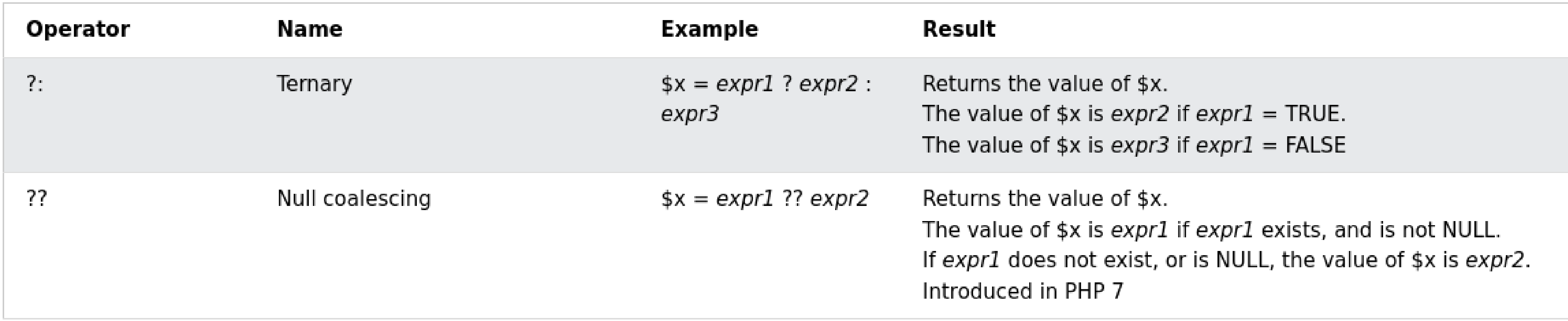
Introduction:PHP: Language Basics

- Lexical Structure:
  - Array Operator

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

Introduction:PHP: Language Basics

- Lexical Structure:
  - Conditional Assignment Operator



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

Introduction:PHP: Language Basics

- Lexical Structure:
  - Operator Precedence and Associativity
  - Implicit Casting