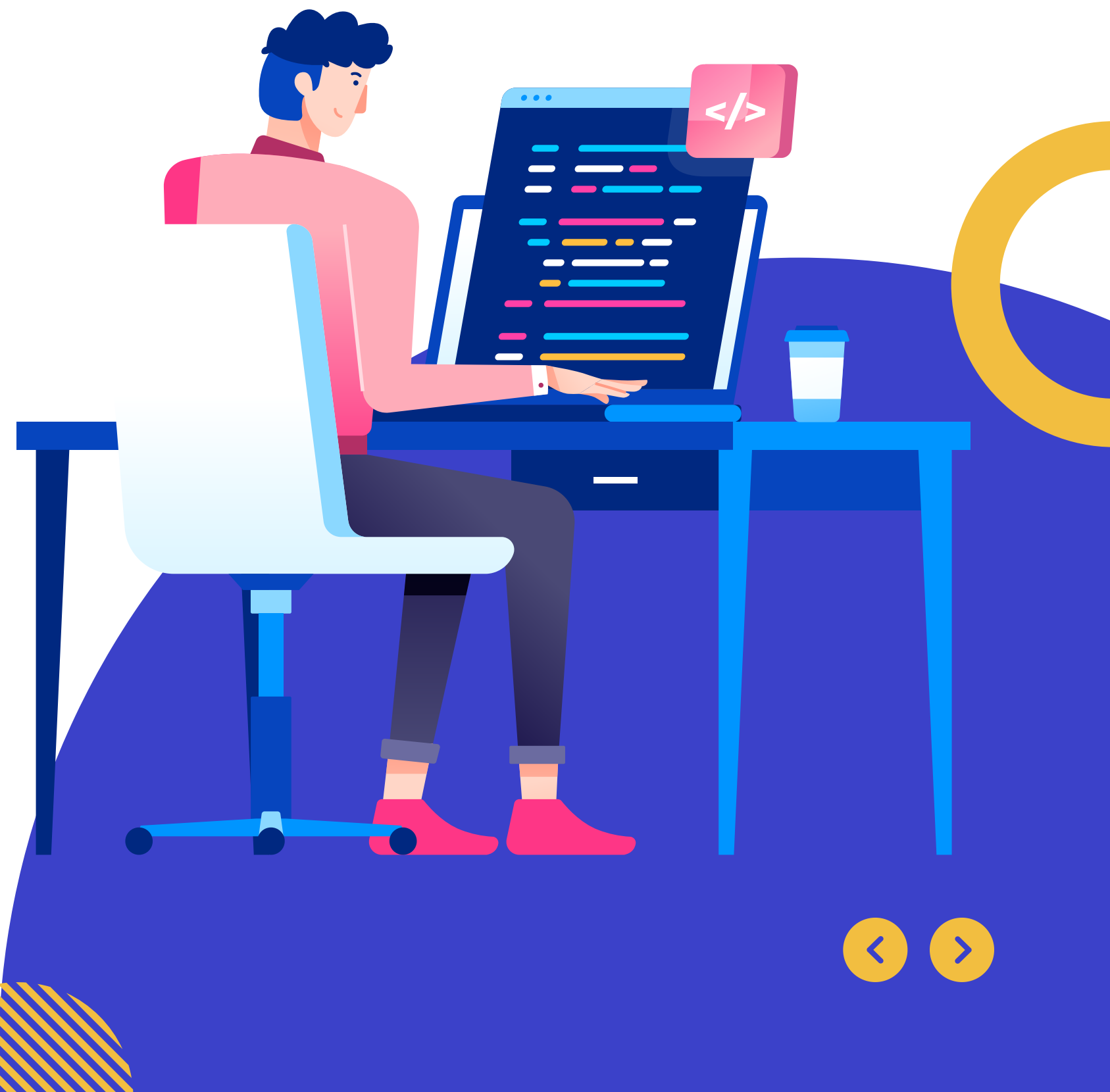


# PHP

## FUNDAMENTAL 01

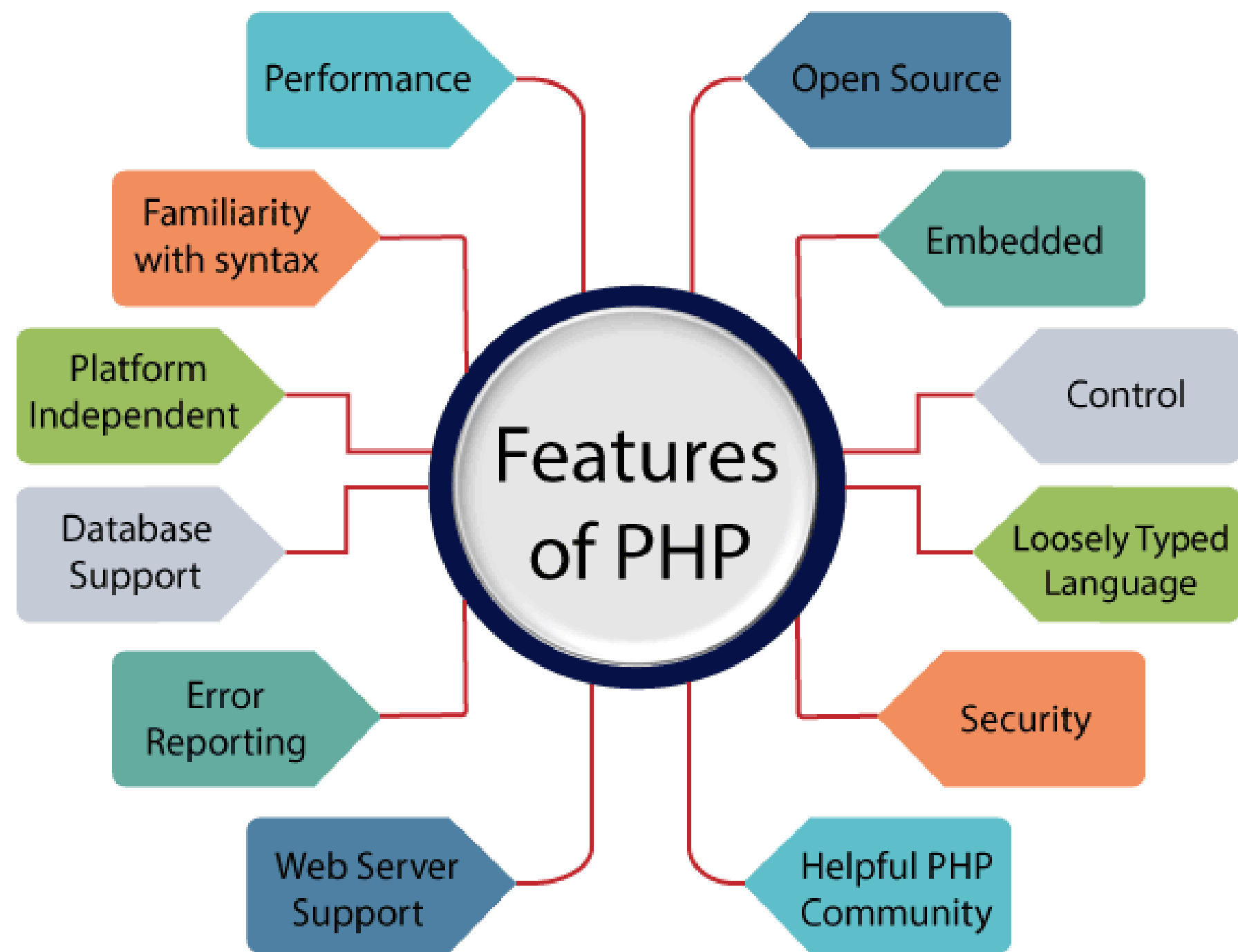


# PHP

PHP is an open-source, interpreted, and object-oriented scripting language.

- Executed at the server-side
- Well suited for web development
- Faster than other scripting languages, for example, ASP and JSP
- Simple and easy to learn







# INSTALL PHP

XAMPP (Cross, Apache, MySQL, PHP, Perl) for Cross Platform: It includes some other components too such as FileZilla, OpenSSL, Webalizer, Mercury Mail, etc





●●● p1.php

```
1  <?php
2  echo "Hello World!";
3  print "I'm about to learn PHP!";
4  ?>
```

# PHP HELLO WORLD

The differences are small: echo has no return value while print has a return value of 1 so it can be used in expressions



 p1.php

```
1  <?php
2  $x = 15;
3  $y = 10;
4  $z=$x+$y;
5  echo $z;
6  ?>
```



# PHP VARIABLES

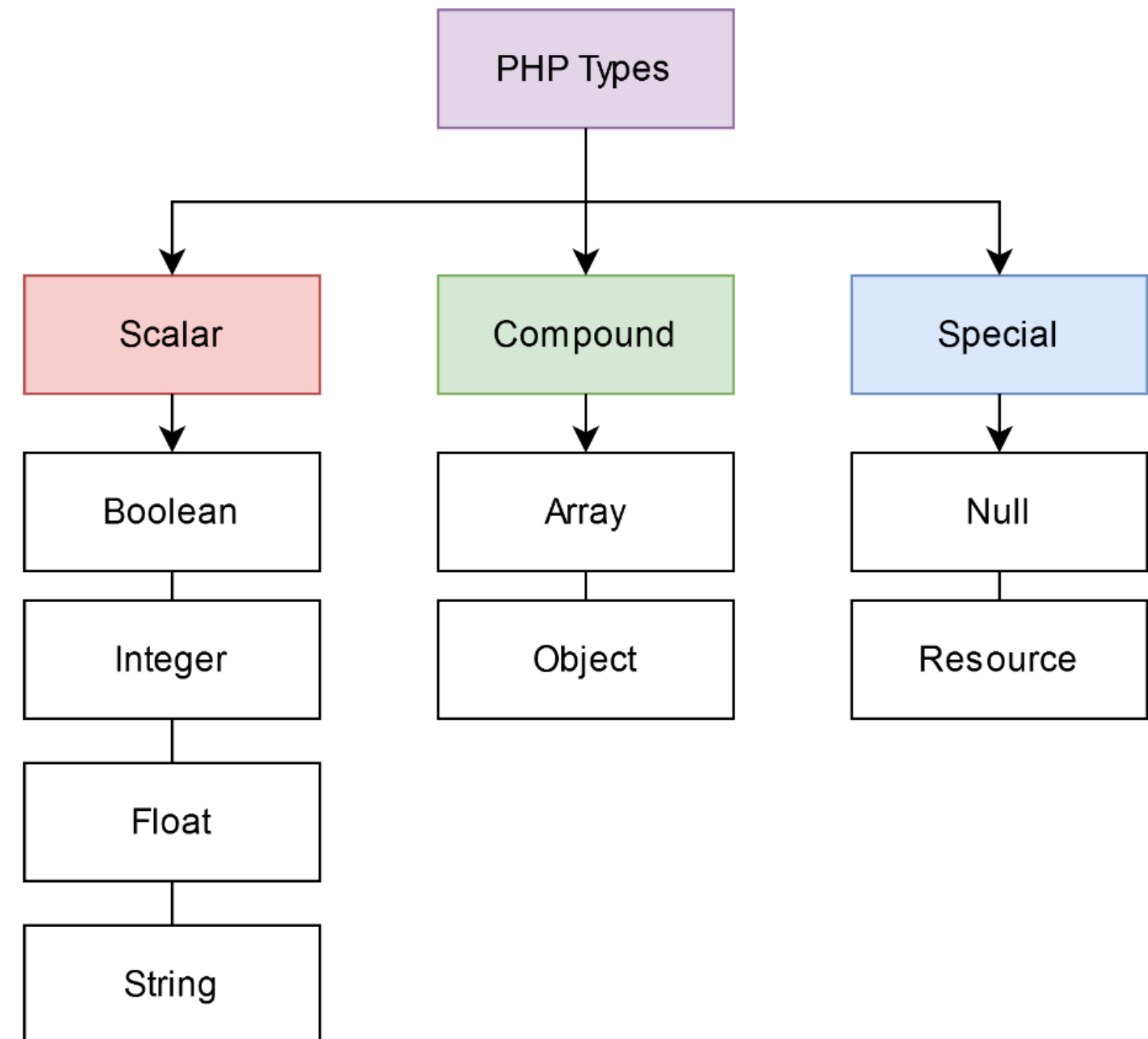
Variables are "containers" for storing information

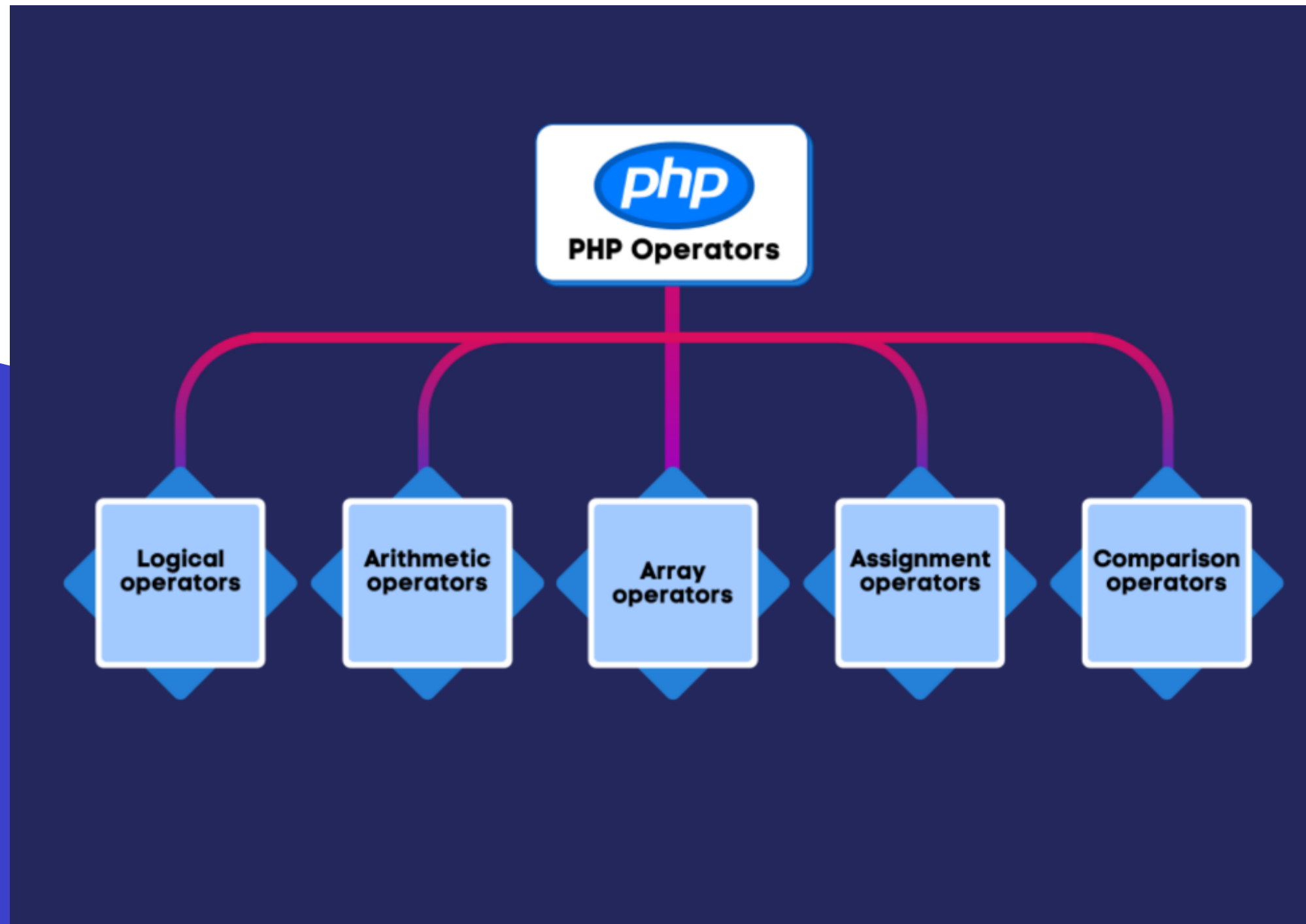
- Starts with the \$ sign
- Must start with a letter or the underscore character
- Cannot start with a number
- Only contain alpha-numeric characters and underscores
- Variable names are case-sensitive



# PHP DATA TYPES

Variables can store data of different types, and different data types can do different things





# PHP OPERATORS

PHP Operator is a symbol i.e used to perform operations on operands. In simple words, operators are used to perform operations on variables or values.





# PHP OPERATORS

## Arithmetic Operators

Operator	Name	Example	Explanation
+	Addition	<code>\$a + \$b</code>	Sum of operands
-	Subtraction	<code>\$a - \$b</code>	Difference of operands
*	Multiplication	<code>\$a * \$b</code>	Product of operands
/	Division	<code>\$a / \$b</code>	Quotient of operands
%	Modulus	<code>\$a % \$b</code>	Remainder of operands
**	Exponentiation	<code>\$a ** \$b</code>	<code>\$a</code> raised to the power <code>\$b</code>



# PHP OPERATORS

## Bitwise Operators

Operator	Name	Example	Explanation
&	And	<code>\$a &amp; \$b</code>	Bits that are 1 in both <code>\$a</code> and <code>\$b</code> are set to 1, otherwise 0.
	Or (Inclusive or)	<code>\$a   \$b</code>	Bits that are 1 in either <code>\$a</code> or <code>\$b</code> are set to 1
^	Xor (Exclusive or)	<code>\$a ^ \$b</code>	Bits that are 1 in either <code>\$a</code> or <code>\$b</code> are set to 0.
~	Not	<code>~\$a</code>	Bits that are 1 set to 0 and bits that are 0 are set to 1
<<	Shift left	<code>\$a &lt;&lt; \$b</code>	Left shift the bits of operand <code>\$a</code> <code>\$b</code> steps
>>	Shift right	<code>\$a &gt;&gt; \$b</code>	Right shift the bits of <code>\$a</code> operand by <code>\$b</code> number of places



# PHP OPERATORS

## Assignment Operators

Operator	Name	Example	Explanation
=	Assign	<code>\$a = \$b</code>	The value of right operand is assigned to the left operand.
+=	Add then Assign	<code>\$a += \$b</code>	Addition same as <code>\$a = \$a + \$b</code>
-=	Subtract then Assign	<code>\$a -= \$b</code>	Subtraction same as <code>\$a = \$a - \$b</code>
*=	Multiply then Assign	<code>\$a *= \$b</code>	Multiplication same as <code>\$a = \$a * \$b</code>
/=	Divide then Assign (quotient)	<code>\$a /= \$b</code>	Find quotient same as <code>\$a = \$a / \$b</code>
%=	Divide then Assign (remainder)	<code>\$a %= \$b</code>	Find remainder same as <code>\$a = \$a % \$b</code>



# PHP OPERATORS

## Comparison Operators

Operator	Name	Example	Explanation
==	Equal	\$a == \$b	Return TRUE if \$a is equal to \$b
===	Identical	\$a === \$b	Return TRUE if \$a is equal to \$b, and they are of same data type
!==	Not identical	\$a !== \$b	Return TRUE if \$a is not equal to \$b, and they are not of same data type
!=	Not equal	\$a != \$b	Return TRUE if \$a is not equal to \$b
<>	Not equal	\$a <> \$b	Return TRUE if \$a is not equal to \$b
<	Less than	\$a < \$b	Return TRUE if \$a is less than \$b
>	Greater than	\$a > \$b	Return TRUE if \$a is greater than \$b
<=	Less than or equal to	\$a <= \$b	Return TRUE if \$a is less than or equal \$b
>=	Greater than or equal to	\$a >= \$b	Return TRUE if \$a is greater than or equal \$b
<=>	Spaceship	\$a <=> \$b	Return -1 if \$a is less than \$b Return 0 if \$a is equal \$b Return 1 if \$a is greater than \$b



# PHP OPERATORS

## Incrementing/Decrementing Operators

Operator	Name	Example	Explanation
++	Increment	++\$a	Increment the value of \$a by one, then return \$a
		\$a++	Return \$a, then increment the value of \$a by one
--	decrement	--\$a	Decrement the value of \$a by one, then return \$a
		\$a--	Return \$a, then decrement the value of \$a by one



# PHP OPERATORS

## Logical Operators

Operator	Name	Example	Explanation
and	And	\$a and \$b	Return TRUE if both \$a and \$b are true
Or	Or	\$a or \$b	Return TRUE if either \$a or \$b is true
xor	Xor	\$a xor \$b	Return TRUE if either \$ or \$b is true but not both
!	Not	! \$a	Return TRUE if \$a is not true
&&	And	\$a && \$b	Return TRUE if either \$a and \$b are true
	Or	\$a    \$b	Return TRUE if either \$a or \$b is true



# PHP OPERATORS

## Array Operators

Operator	Name	Example	Explanation
+	Union	<code>\$a + \$y</code>	Union of <code>\$a</code> and <code>\$b</code>
<code>==</code>	Equality	<code>\$a == \$b</code>	Return TRUE if <code>\$a</code> and <code>\$b</code> have same key/value pair
<code>!=</code>	Inequality	<code>\$a != \$b</code>	Return TRUE if <code>\$a</code> is not equal to <code>\$b</code>
<code>===</code>	Identity	<code>\$a === \$b</code>	Return TRUE if <code>\$a</code> and <code>\$b</code> have same key/value pair of same type in same order
<code>!==</code>	Non-Identity	<code>\$a !== \$b</code>	Return TRUE if <code>\$a</code> is not identical to <code>\$b</code>
<code>&lt;&gt;</code>	Inequality	<code>\$a &lt;&gt; \$b</code>	Return TRUE if <code>\$a</code> is not equal to <code>\$b</code>





●●● p1.php

```
1  <?php
2  echo "Hello World!";
3  //print "I'm about to learn PHP!";
4  /*
5  Anything placed
6  within comment
7  */
8  ?>
```

# PHP COMMENTS

There are two ways to use single line comments in PHP.







●●● p1.php

```
1  <?php
2  $num=12;
3  if($num<100){
4  echo "$num is less than 100";
5  }
6  ?>
```

# PHP IF STATEMENT

- PHP if statement allows conditional execution of code. It is executed if condition is true.
- If statement is used to executes the block of code exist inside the if statement only if the specified condition is true.



p1.php

```
1  <?php
2      $marks=69;
3      if ($marks<33){
4          echo "fail";
5      }
6      else if ($marks>=34 && $marks<50) {
7          echo "D grade";
8      }
9      else if ($marks>=50 && $marks<65) {
10         echo "C grade";
11     }
12     else if ($marks>=65 && $marks<80) {
13         echo "B grade";
14     }
15     else if ($marks>=80 && $marks<90) {
16         echo "A grade";
17     }
18     else if ($marks>=90 && $marks<100) {
19         echo "A+ grade";
20     }
21     else {
22         echo "Invalid input";
23     }
24     ?>
```

# PHP IF-ELSE-IF STATEMENT

- The PHP if-else-if is a special statement used to combine multiple if?.else statements.
- So, we can check multiple conditions using this statement.



●●● p1.php

```
1  <?php
2      $num=20;
3      switch($num){
4      case 10:
5          echo("number is equals to 10");
6          break;
7      case 20:
8          echo("number is equal to 20");
9          break;
10     case 30:
11         echo("number is equal to 30");
12         break;
13     default:
14         echo("number is not equal to 10, 20 or 30");
15     }
16  ?>
```

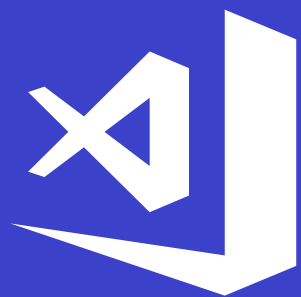
# PHP SWITCH

- PHP switch statement is used to execute one statement from multiple conditions.
- It works like PHP if-else-if statement.



# VISUAL STUDIO CODE POWER TIPS

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.



ostad

# VISUAL STUDIO CODE POWER TIPS

- Font scaling with mouse wheel
- Setting key bindings
- Duplicating lines quickly
- Move down the line quickly
- Visual Studio Code Minimap
- Move quickly to a specified line
- Hiding or showing sidebars



ostad

# VISUAL STUDIO CODE POWER TIPS

- Splitting the editor
- Quickly navigate between active tabs
- Creating nested folders quickly
- Use of the Multiple Cursor Feature
- Faster incrementing and decrementing
- Opening file in new tab
- Quickly navigate through different files





ostad

# VISUAL STUDIO CODE POWER TIPS

- Quickly navigate to previous open tabs
- Accessing the integrated terminal
- Installing better looking monospace fonts
- Formatting Code



ostad

# VISUAL STUDIO CODE POWER TIPS

- Essential Extensions for PHP
- Code Runner Extension
- Creating Live Templates
- Displaying Hints in Snippets

