



Tishk International University  
IT Department  
Course Code: IT 349/A

# Web Programming

## Week #3 Introduction to PHP

Fall 2024  
Hemin Ibrahim, PhD  
[hemin.ibrahim@tiu.edu.iq](mailto:hemin.ibrahim@tiu.edu.iq)





---

# Overview

---



- Web servers
- What is PHP?
- Basic syntax in PHP
- Comments
- Variable
- Expressions and Operators
- Control Structures

---

# Objectives

---



- Understand the role of web servers in handling requests and serving web pages to users.
- Learn what PHP is and how it enables server-side scripting to create dynamic web pages.
- Familiarize yourself with the fundamental syntax and structure of PHP code.
- Understand how to declare and use variables to store and manipulate data in PHP.
- Explore the use of expressions and operators to perform calculations, and logical operations in PHP.
- Learn how to use control structures (if-else) to manage the flow of a PHP program based on conditions.

# URLs and web servers

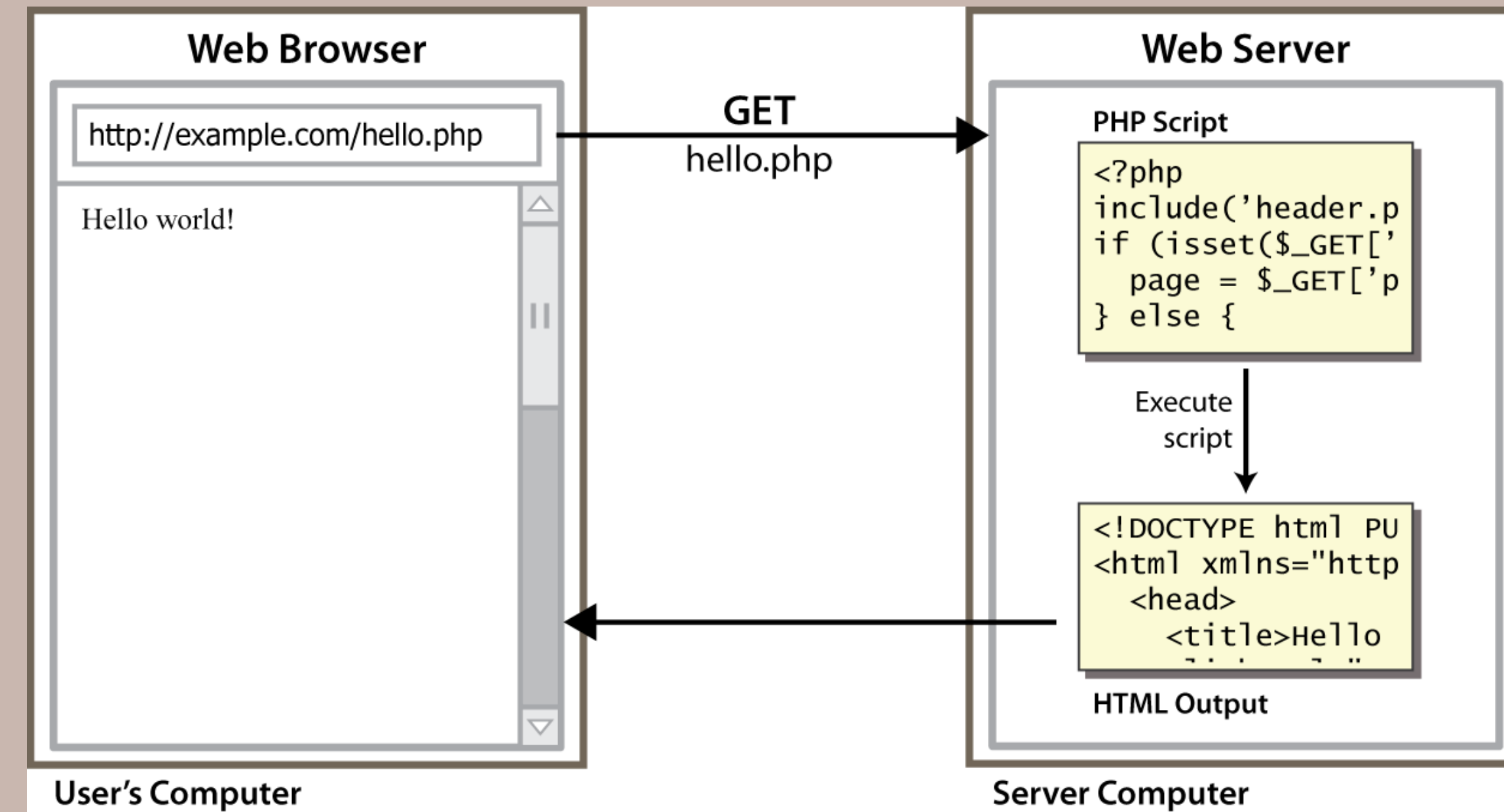


## http://server/path/file

- Usually when you type a URL in your browser:
  - your computer looks up the server's IP address using DNS
  - your browser connects to that IP address and requests the given file
  - the web server software (e.g. Apache) grabs that file from the server, and sends the contents to you
- Some URLs actually specify *programs* that the web server should run, and then send their output back to you as the result: `https://lecture-notes.tiu.edu.iq/wp-login.php`
  - the above URL tells the server `lecture-notes.tiu.edu.iq` to run the file `wp-login.php` and send back its output.

# What is PHP?

- **PHP** stands for "PHP Hypertext Preprocessor"
- a server-side scripting language
- used to make web pages **dynamic**:
  - authenticate users
- PHP code can be embedded in HTML tags





# When you start coding in a new language without reading the documentation





# Hello, world!



The following contents could go into a file `hello.php`:

```
<?php  
  
?>
```

```
<?php  
    print "Hello, World!";  
    echo "Hello, World!";  
?>
```

- A block or file of PHP code begins with `<?php` and ends with `?>`
- PHP statements, function declarations, etc. appear between these endpoints.
- You can use “print” or “echo” for printing in PHP.



# PHP Basic Syntax

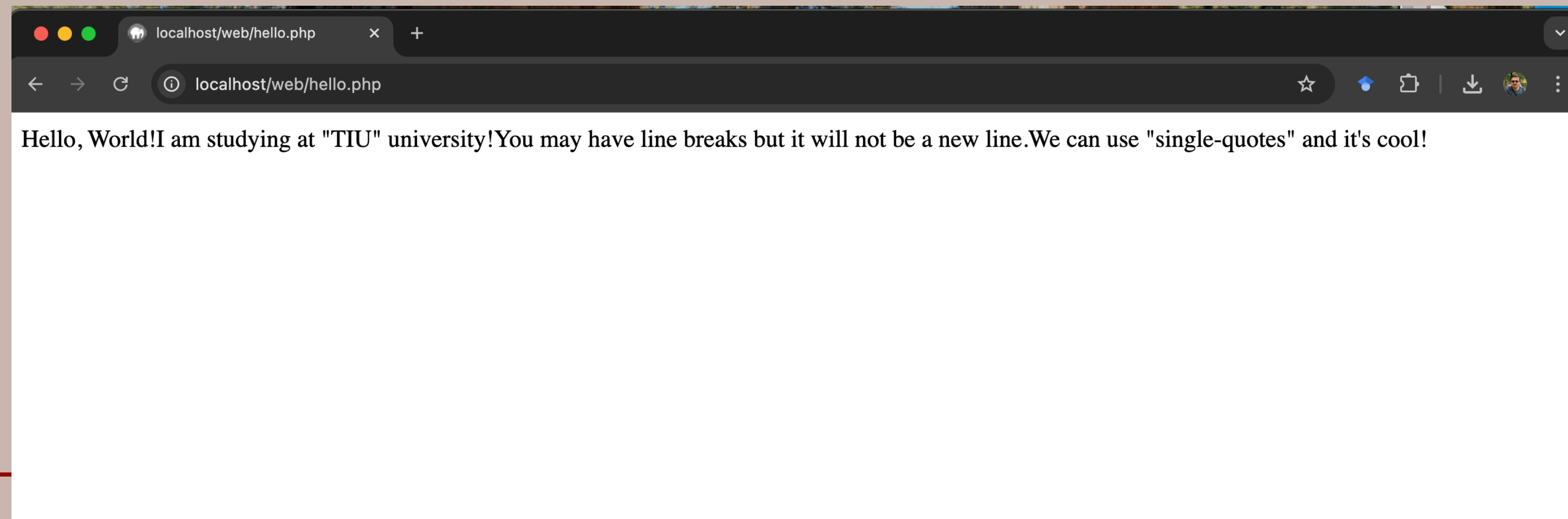


```
<?php
echo "Hello, World!";
echo "I am studying at \"TIU\" university!";

echo "You may have
line breaks but it will not be a new line.";

echo 'We can use "single-quotes" and it\'s cool!';
?>
```

Output



# Case Sensitivity



- The names of user-defined classes and functions, as well as built-in constructs and keywords are case-insensitive.

```
<?php
    Echo "Hello, World!";
    ECHO "Hello, World!";
    echo "Hello, World!";
?>
```

All three lines are equivalent.



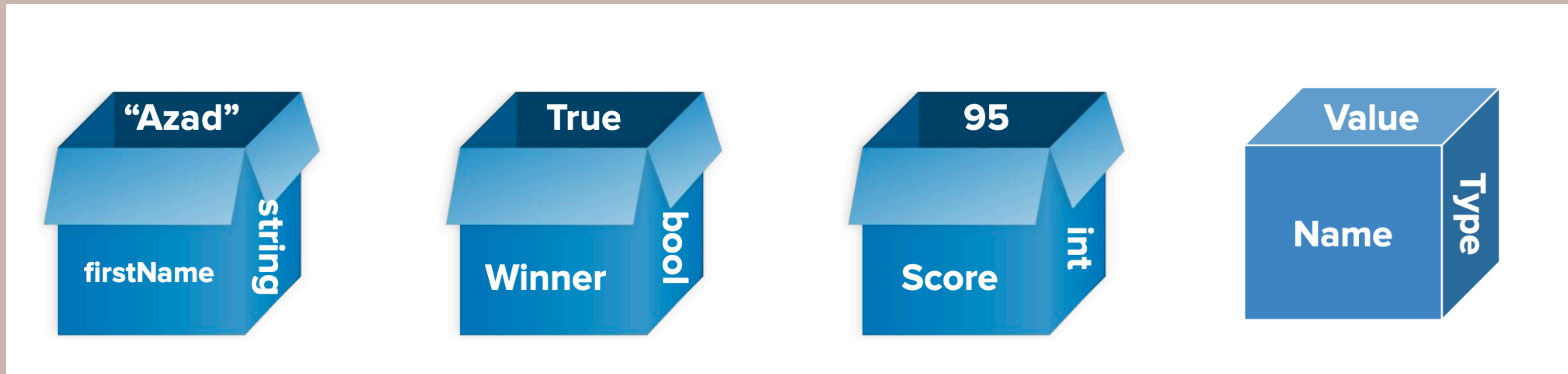
# Comments



```
<?php
# single-line comment
echo " Hello!";
// single-line comment

/*
multi-line comment
*/
?>
```

# Variables





# Variable Syntax

`$name = expression;`

- All variable names must be preceded by a dollar sign (\$).
- Following the dollar sign, the variable name must begin with either a letter (A–Z, a–z) or an underscore (\_). A number cannot immediately follow the dollar sign.
- The rest of the variable name can contain any combination of letters, underscores, and numbers.
- Each variable must have a unique name.
- Variable names are case-sensitive!

```
<?php
$user_name = "TIU";
$age = 16;
$drinking_age = $age + 5;
$this_class_rocks = true;
?>
```

# Case Sensitivity - Variables



- Variables, on the other hand, are case-sensitive.

```
<?php
    $name = "Aran";
    $Name = "Ahmed";
    $NAME = "Ali";
?>
```

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



When you try to choose a meaningful variable name.



# Variable Syntax



Valid variables in PHP

Name
\$first_name
\$person
\$address1
\$_SERVER

Name	Valid?	Reason
\$first name		
\$first.name		
first_name		
\$1address		
@email		
\$first_name		
\$Wêb		
_android		
\$4		
\$four		
\$		



---

# Types of Variables

---



PHP provides different types of values, or data types:

- Integer
- Float (Double)
- String
- Boolean
- Array
- Object
- NULL
- Resource
- Callable
- Iterable

# String type



- 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.
- Because strings are so common in web applications, PHP includes core-level support for creating and manipulating strings.

# String type



```
<?php
$uni_name = "Tishk International University";
echo $uni_name[11];
?>
```

- 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[11]` accesses the character at index 11 of the string "Tishk International University".
- If we count the indices from 0, the character at index 11 is "n"



# Interpreted strings



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

Output: I am studying at TIU

```
<?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;
?>
```

Output?

# Interpreted strings



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

- Strings inside " " are **interpreted**
  - Variables that appear inside them will have their values inserted into the string
- Strings inside ' ' are not interpreted

# String Concatenation Operator



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

```
<?php
$first_name = "Alan";
$last_name = "Baker";
echo $first_name . " " . $last_name;
?>
```

This will produce the following result: **Alan Baker**



# String Concatenation Operator



```
<?php
$first_name = "Alan";
$last_name = "Baker";
echo "My Name is " . $first_name . " " . $last_name;
?>
```

This will produce the following result: **My Name is Alan Baker**

# strlen()



The strlen() function is used to find the length of a string. Let's find the length of "Tishk International University".

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

Output: 30

# str\_word\_count()



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

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

Output: 3



# str\_replace()



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

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

Output: Hello Kurdistan!

# 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.

```
<?php
    $string = "Hello World!";
    echo strtolower($string); // hello world!
    echo strtoupper($string); // HELLO WORLD!
    echo ucfirst("hello world"); // Hello world
    $string = "hello world!";
    echo ucwords($string); /// Hello World!
```

```
?>
```

# Integer



- Integer means numeric data with a negative or positive sign.
- It holds only whole numbers, i.e., numbers without fractional part or decimal points.
- The range of an integer must be lie between - 2,147,483,648 and 2,147,483,647 i.e., -  $2^{31}$  to  $2^{31}$ .

```
<?php
    $a = 115;
    $b = -109;
    echo "The positive number: ".$a ." and the negative number: ".$b;
?>
```



# Float & Double



- A floating-point number is a number with a decimal point.

```
<?php

    // Declaring a float
    $floatNumber = 3.14;
    echo $floatNumber;
    // Output: 3.14

    // Declaring a double (same as float in PHP)
    $doubleNumber = 123.456;
    echo $doubleNumber;
    // Output: 123.456

?>
```

# Valid Numbers

- A floating-point number is a number with a decimal point.

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	?
1970A	?
1.2.4	?

# Expressions and Operators



- An expression is a bit of PHP code that can be evaluated to produce a value

Operator	Operation	Operator	Operation
*	Multiplication	!	Logical NOT
/	Division	==	Value equality
%	Modulus	!=	Inequality
+	Addition	===	Equality in type and value
-	Subtraction	&&	Logical AND
.	String concatenation		Logical OR
<, <=	Less than, less than or equal	++	Increment
>, >=	Greater than, greater than or equal	--	Decrement
=	Assignment	+=, -=, *=, /=, .=	Assignment with operation



# Equality and Identity



- Equality (==): Compares **values only**. It checks whether the values are equal after type conversion if needed
- Identity (===): Compares both **values and data types**. It returns true only if both the value and type are exactly the same.

```
<?php
    $a = 5;           // Integer
    $b = "5";        // String

    if ($a == $b) {
        echo "Values are Equal\n";
    }

    if($a=== $b) {
        echo "Values and types Equal\n";
    }

?>
```

# Expressions and Operators



- Addition (+):
- Subtraction (-):

```
<?php  
  
$a = 5;  
$b = 3;  
$result = $a - $b;  
echo $result; // 2  
  
?>
```

```
<?php  
  
$a = 5;  
$b = 3;  
$result = $a + $b;  
echo $result; // 8  
  
?>
```

# Expressions and Operators



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

```
<?php
```

```
$a = 5;  
$b = 3;  
$multi = $a * $b;  
echo $multi; // 15
```

```
$div= $a / $b;  
echo $div; // 1.6666666666666667
```

```
$mod = $a % $b;  
echo $mod; // 2
```

```
?>
```



# 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

Operator	Name	
<code>\$var++</code>	Post-increment	Returns \$var, then increments \$var by one
<code>++\$var</code>	Pre-increment	Increments \$var by one, then returns \$var
<code>\$var--</code>	Post-decrement	Returns \$var, then decrements \$var by one
<code>--\$var</code>	Pre-decrement	Decrement \$var by one, then returns \$var

---

Thank You

---