A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

AAPP009-4-2 Web Development

INTRODUCTION TO PHP

Topic & Structure of The Lesson

- Client-side Scripting vs. Server-side Scripting
  - Comparison between HTML, JavaScript & PHP
- Introduction to PHP
- PHP Syntax
  - Comments
  - Variable
  - echo / print
  - Numeric expression
  - String
  - Operators
  - Conditional statements

Client-side Scripting vs. Server-side Scripting

- Server-side: Scripts run on the server, generate dynamic content, send static HTML to the browser
- Client-side: Scripts run in the user's browser, enhance interactivity

Introduction to PHP

- PHP runs on the web server, and the output is showing on user's browser.
- Combines scripting with HTML for dynamic content from the database
- PHP files have extension *.php
- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the web server, and the result is returned to the browser as plain HTML

PHP Syntax

- A PHP script can be placed anywhere in the document.
- A PHP script starts with **<?php** and ends with **?>**:

```
<?php
    // PHP code goes here
?>
```

- The default file extension for PHP files is ".php".
- A PHP file normally contains HTML tags, and some PHP scripting code.
- PHP statements end with a semicolon (;).

PHP Case Sensitivity

- PHP keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are not case-sensitive.
- However all variable names are case-sensitive.

```php
<?php
  ECHO "Hello World!<br>";
  echo "Hello World!<br>";
  EcHo "Hello World!<br>";
?>
```

Comments in PHP

Multiple Line

```
<?php
/*
This is a multiple-lines
comment block
that spans over multiple
lines
*/
?>
```

Single Line

```
<?php
// This is a single-line comment

# This is also a single-line comment
?>
```

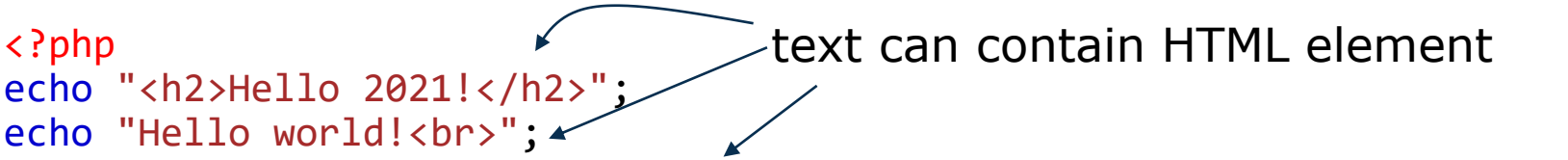
PHP Variables

- Integers (simple counting numbers)
- Floating point numbers (for fractions)
- Strings (text)
- Objects (groups of different things)
- Arrays (groups of things of the same type)
- We suggest you use **$nName** for numbers and **$sName** for strings.
- Ignore objects and arrays for the moment.
- PHP is case sensitive so **$nName** is not the same as **$nname**

PHP echo Statements

- echo used to output data to the screen.
- echo can be used with or without parentheses: echo or echo().

```php
<?php
echo "<h2>Hello 2021!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>
```



Numeric expressions

- Add $nTotal = $nAmt1 + $nAmt2;
  - Also subtract -, multiply *, divide /
- Comparisons
  - Equal $nAmt1 == $nAmt2,
  - Also Not Equal !=
  - Less than <, Greater than >
  - Less than or equal to <=
  - Greater than or equal to >=

Strings

- Double quoted strings:

```
$sFirstName="Chris";
$sLastName="Mayer";
print "Hello, $sFirstName $sLastName";
echo "Hello, " . $sFirstName ." ". $sLastName;
```

- This would print:
  Hello, Chris Mayer

- Special characters:
  \n newline, \$ dollar sign, \\ backslash, \" double quote

- Joining Strings:
  $sComplete = $sFirstName.$sLastName;

PHP Operators

- Arithmetic Operators

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

PHP Operators

- Comparison Operators

Operator	Name	Example
==	Equal	$x == $y
!=	Not equal	$x != $y
<>	Not equal	$x <> $y
>	Greater than	$x > $y
<	Less than	$x < $y
>=	Greater than or equal to	$x >= $y
<=	Less than or equal to	$x <= $y

Conditional statements

```
if (expression)
{  statement;
   statement;
}
else
{  statement;
   statement;
}
```

```php
if ($nCount<31)
{
    echo "okay!";
}
else
{
    phpinfo();
    /* Outputs information about
PHP's configuration */
}
```

More examples

```
if ($i==$j)
{
    print "equal $i  $j <br>\n";
    $i++;
}
else
{
    print "unequal $i  $j <br>\n";
    $i--;
}
```

Note
 puts a new line on your web page

\n puts a new line in the coding (i.e. if you open the browser's source view, it will have a new line)

For Loops

```
for (expression1; expression2; expression3)
{
   statement;
   statement;
}
```

```
for($i=0; $i<5; $i++)
{
   print "<p><font size=$i>Text Size $i</font></p>/n";
}
```

While Loops

```
while (expression)
{
    statement;
    statement;
}
```

```php
$i=1;
while ($i<6)
{
    echo '<p style="font-size:'.($i*10).'pt">Text
    Size'.$i.'</p>';
    $i++; // similar as $i=$i+1;
}
```

- Continues while the expression is TRUE
- Stops when it is FALSE

Quick Review Exercise

- Write PHP statements to convert 10 miles into kilometers, where 1 mile is equal to 1.60934 km. Declare a variable called ‘miles’ and assign it a value of 10. Then, print the conversion result in an HTML paragraph.

Summary

- We have discussed:
  - Some of what PHP can do,
  - What PHP looks like,
  - The programming environment,
  - The world according to a PHP script,
  - Examples of PHP statements.

Q & A