

Final Exam: PHP

WEP 2024

{3h}

17 January 2024

BASIC SYNTAX

Syntax	Definition
\$	Use to declare variable. Example: \$username = "bopha";
echo()	Use to display output on web browser
print()	Use to display output on web browser
print_r()	Use for print array
var_dump()	Use for print array
error_log()	Use to check error on server side (terminal)
<?php ?>	PHP tags
<?= "something" ?>	Short hand to write <?php echo "something" ?>
foreach () { ... }	Looping : foreach(\$array as \$value) { }
foreach(): endforeach;	Just other ways to write foreach
if (condition) { ... }	Checking condition
if (condition): ... endif;	Other way to write if
isset()	is to check if a variable is set with a value and that value should not be null.
empty()	is to check if a given variable is empty. The difference with isset() is, isset has null check. is_null() is to check whether a variable is defined as null.

PHP GLOBAL VARIABLE

Syntax	Definition
<code>\$_GET[]</code>	<ul style="list-style-type: none">- Contains an array of variables received via the HTTP GET method.- A query string is data added at the end of a URL. In the link below, everything after the ? sign- In addition, in HTML form must be method attribute GET.
<code>\$_POST[]</code>	Contains an array of variables received via the HTTP POST method. In addition, in HTML form must be method attribute POST.
<code>\$_SERVER</code>	<code>\$_SERVER['REQUEST_METHOD']</code> Returns the request method used to access the page (such as POST or GET)

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
    // var_dump($_POST);  
    $name = $_POST['name'];  
    $age = $_POST['age'];  
    $gender = $_POST['gender'];  
}
```

PHP REGULAR EXPRESSION

Syntax	Definition
<code>"/ /"</code>	Pattern to start regular expression for example: <code>\$pattern = "/^[a-zA-Z\d]\$/";</code>
<code>pre_match(pattern, string)</code>	To match the pattern with our string input. <code>\$pattern = "/^[a-zA-Z\d]\$/";</code> <code>\$input = "Hello100";</code> <code>echo pre_match(\$pattern, \$input);</code>
<code>[a-z]</code>	Allow lowercase a to z
<code>[A-Z]</code>	Allow uppercase A to Z
<code>0 - 9</code>	Allow all numbers
<code>\d</code>	Allow all digit numbers the same 0 - 9
<code>\s</code>	Allow space
<code>\.\$@#%!</code>	Allow sign . \$ @ # % !
<code>[a-z]{3}</code>	Allow a to z lowercase only 3 characters
<code>[a-z]{2, 5}</code>	Allow a to z lowercase between 2 to 5 characters
<code>[a-z]{3, }</code>	Allow a to z lowercase from 3 characters

Example below:

```
<?php
$regexName = "/^[a-z\d]{1,8}$/";
$regexPass = "/^[a-zA-Z\s\d\.\@\#\!\%]+\$/";
$messageName = "";
$messagePass = "";
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (preg_match($regexName, $_POST['text'])) {
        $messageName = "😊";
    } else {
        $messageName = "😞";
    }
    if (preg_match($regexPass, $_POST['password'])) {
        $messagePass = "😊";
    } else {
        $messagePass = "😞";
    }
}
?>

<form method="post">
    <input type="text" name="text">
    <?= $messageName != ""? $messageName : ""?>
    <br>
    <input type="password" name="password">
    <?= $messagePass != ""? $messagePass : ""?>
    <br>
    <button>Submit</button>
</form>
```

Query for CRUD

Meaning	Query
Create / Add / Insert	INSERT INTO table_name (col1, col2, ..) VALUES(value1, value2,...);
Read/ Get data / Retrieve / select / show	SELECT * FROM table_name; SELECT * FROM table_name WHERE condition;
Delete / Destroy / Remove	DELETE FROM table_name WHERE condition;
Edit / update / change / modify data	UPDATE table_name SET col1=value1, col2=value2, ... WHERE condition;

CRUD Sample

SELECT

```
$statement = $db->prepare("SELECT * FROM items");
$statement->execute();
$data = $statement->fetchAll();
```

DELETE

```
$statement = $db->prepare("DELETE FROM items WHERE id = :id");
$statement->execute([
    ':id' => $id,
]);
```

UPDATE

```
$statement = $db->prepare("SELECT * FROM items WHERE id = :id");
$statement->execute([
    ':id' => $id,
]);
$data = $statement->fetch();
```

```
$statement = $db->prepare("UPDATE items SET item=:item, price=:price WHERE id=:id");
$statement->execute([
    ':item' => $item_name,
    ':price' => $price,
    ':id' => $id
]);
```

CREATE

```
-
$statement = $db->prepare("INSERT INTO items (item, price) VALUES(:item, :price)");
$statement->execute([
    ':item' => $item_name,
    ':price' => $price
]);
```