পিএইচপি টিউটোরিয়াল পর্ব-৪

আনোয়ার হোসেন

পিএইচপি ভেরিয়েবল

ভেরিয়েবলের সাথে আমরা অনেকেই পরিচিত। মাধ্যমিক গণিতে এই ভেরিয়েবলের সাথে আমাদের পরিচয় ঘটে। সেই যাই হোক, ভেরিয়েবলকে বলা যেতে পারে একটি পাত্র বা কন্টেইনার, যেখানে আমরা ডাটা বা তথ্য রাখতে পারি। যেমন– বলা যেতে পারে string ‘Hi Friends’ অথবা interger value 200। ভেরিয়েবলের সুবিধা হচ্ছে একবার কোনো তথ্য একটি ভেরিয়েবলে রেখে দেয়ার পর সেটা বারবার পিএইচপি কোডজুড়ে ব্যবহার করা যায়। পিএইচপিতে ভেরিয়েবলের আগে ‘$’ সাইনটি ব্যবহার বাধ্যতামূলক। অন্যথা হলে কোড কাজ করবে না। ভেরিয়েবলের মান যেকোনো প্রয়োজনে পরিবর্তন করা যায়। পিএইচপিতে ভেরিয়েবল case sensetive যেমন– $_address and $_ADDRESS দুটি আলাদা ভেরিয়েবল।

পিএইচপিতে ভেরিয়েবল লেখা হয় এভাবে–

$variable_name = Value;

উদাহরণ–

```
<?php
$hello = “This is a string”;
$a_number = 4;
$anotherNumber = 8;
echo $hello .”<br/>”;
$total = $a_number+$anotherNumber;
```

```
echo $total;
?>
```

আউটপুট

উপরের কোড লক্ষ করলে দেখা যাবে স্ট্রিংকে কোটেশনের ভেতরে রেখে $hello নামের ভেরিয়েবলের মাঝে রাখা হয়েছে এবং echo ব্যবহার করে ব্রাউজারে আনা হয়েছে। পরে $a_number ও $anotherNumber নামের দুটি ভেরিয়েবলের মধ্যে সংখ্যা রাখা হয়েছে এবং একটি অঙ্ক করা হয়েছে। অন্যান্য ল্যাঙ্গুয়েজে ডাটা টাইপ ঠিক করে দিতে হয়, কিন্তু পিএইচপি ‘Loosely Typed’ ল্যাঙ্গুয়েজ হওয়ার কারণে ভেরিয়েবলের টাইপ বা ধরন উল্লেখ না করলেও পিএইচপি নিজ থেকে ভেরিয়েবলের ডাটা টাইপ ঠিক করে দেয়।

ভেরিয়েবল নামকরণ পদ্ধতি

১. কোনো letter or “_” (underscore) দিয়ে শুরু করতে হবে।
২. নামের মাঝে alpha-numeric characters ও underscores. a-z, A-Z, 0-9, or _ থাকতে পারে।
৩. ভেরিয়েবলের নামে স্পেস থাকা যাবে না। একের বেশি নাম হলে “_” underscore ($my_string) অথবা বড় হাতের অক্ষরে ($myString) এভাবে লিখতে হবে।

```
<?php
$feedback = “anam”;
$domain = “@yahoo.com”;
$feedback = $feedback.$domain;
echo $feedback;
?>
```

concatenate (কনক্যাটেনেট) হচ্ছে একটি ডট (.) দিয়ে দুটি স্ট্রিংকে একসাথে করা। এই কাজটি অন্যভাবেও করা যায়–

```
<?php
$feedback = “anam”;
$domain = “@yahoo.com”;
$feedback .= $domain;
echo $feedback;
?>
```

এটার আউটপুট উপরেটির মতোই আসবে। অর্থাৎ anam@yahoo.com

নিচের কোডব্লক দুটির আউটপুটও হবে একইরকম–

```
<?php
$test = 0;
$test++;
echo $test;
?>
```

আউটপুট আসবে ১। উল্লেখ্য, $test ভেরিয়েবলের মান যদি ঠিক করে দেয়া না হয়, তবে পিএইচপি এর মান বাই ডিফল্ট NULL ধরে নেবে।

```
<?php
$test = 0;
$test = $test+1;
echo $test;
?>
```

এখানেও আউটপুট আসবে ১।

পিএইচপি ভেরিয়েবল স্কোপ ও কনস্ট্যান্ট ভেরিয়েবল স্কোপ

একটি ফাংশনে একটি ভেরিয়েবল থাকতে পারে। নিচের উদাহরণটি দেখুন–

```
<?php
$number = 8;
function calculation(){
$number  = 10;
$anothernumber = 20;
$addnumber   =  $number+$another-
number;
echo $addnumber;
}
echo “This \$number variable from out-
side of calculation() function and its value
is $number”;
echo “</br>”;
calculation();
?>
```

আউটপুট

This $number variable from outside of calculation() function and its value is 8 30

এখানে লক্ষণীয়, ভেরিয়েবলের চিহ্ন $-এর পূর্বে \ বা ব্যাক স্ল্যাশ দেয়ার কারণে সেটি আর ভেরিয়েবল বলে গণ্য হয়নি, স্ট্রিংয়ের মতো আউটপুট দিয়েছে। এটিকে বলে এসকেপ ক্যারেক্টার (escape character)। ফাংশনের ভেতরে ও বাইরে $number নামে ভেরিয়েবল আছে। বাইরের ভেরিয়েবলের মান $number=8;। দুটি ভেরিয়েবলকে একই মনে হলেও আসলে এই দুটি সম্পূর্ণ আলাদা ভেরিয়েবল। কেননা, এদের একটি ফাংশনের ভেতরে ও অন্যটি বাইরে। ভেতরের ভেরিয়েবলটিকে বলে লোকাল ভেরিয়েবল ও বাইরেরটিকে বলে গ্লোবাল ভেরিয়েবল। লোকাল ভেরিয়েবলটি ব্যবহারে সীমাবদ্ধতা আছে, অর্থাৎ এটি শুধু ফাংশনের ভেতরে ব্যবহার করা যাবে, ফাংশনের বাইরে অ্যাক্সেস পাওয়া যাবে না। আর একটি ভেরিয়েবল শুধু একটি নির্দিষ্ট জায়গায় ব্যবহার করাই হলো ভেরিয়েবল স্কোপ।

গ্লোবাল ভেরিয়েবল

গ্লোবাল ভেরিয়েবল ব্যবহারে কোনো সীমাবদ্ধতা নেই। তাই এটিকে ফাংশনের যেকোনো জায়গায় ব্যবহার করা যায়। কোনো ফাংশনে গ্লোবাল ভেরিয়েবল ব্যবহারে পদ্ধতি আছে দুটি।

পদ্ধতি-১ ($GLOBALS[‘number’])

ধরুন, উপরের প্রোগ্রামে ব্যবহৃত গ্লোবাল ভেরিয়েবলটি যদি ফাংশনে ব্যবহার করতে চান, তাহলে নিচের মতো কোড লিখতে হবে–

```
$anothernumber = 20;
$addnumber    =  $number+$GLOB-
ALS[“number”];
echo $addnumber;
}
calculation();.
?>
```

ফাংশনের বাইরের $number ভেরিয়েবলটিকে ফাংশনের ভেতর $GLOBALS অ্যারে দিয়ে নিয়ে এসেছি। ফাংশনটি যদি আরও বড় হতো এবং অন্য কোথাও গ্লোবাল ভেরিয়েবলটি ব্যবহার করতে হতো, তাহলে প্রতিবারই $GLOBALS[“number”] লিখে ব্যবহার করা হতো।

(বাকি অংশ ৬৪ পৃষ্ঠায়)

পদ্ধতি-২ (global $number)

এই পদ্ধতিতে একটা মজার বিষয় আছে, একটু ভালোভাবে লক্ষ করুন। ধরুন, আগের মতোই ফাংশনের বাইরের $number (যেটা গ্লোবাল) ভেরিয়েবলটিকে ব্যবহার করার জন্য আমাদের শুধু $number-এর আগে global শব্দটি লিখে দিতে হবে। অর্থাৎ নিচের মতো global $number এবং এরপর থেকে যদি একাধিকবার এই ভেরিয়েবলটি ব্যবহার করতে হয়, তাহলে কীভাবে লিখবেন– global $number এভাবে? না। বরং এভাবে $number। এবার মজার কথাটি হলো নিচের কোডটি দেখুন–

```
<?php
$number = 8;.
function calculation(){
global $number;
$number  = 10;
$anothernumber = 20;
$addnumber  = $number+$number;
echo $addnumber;
}
calculation();
?>
```

উপরের কোডে (যেখানে $GLOBALS অ্যারে ব্যবহার করা হয়েছে) ৬ নম্বর লাইনে $number হচ্ছে লোকাল ভেরিয়েবল আর এই কোডে এই লাইনটি দিয়ে গ্লোবাল ভেরিয়েবলে নতুন নাম্বার ১০ assign করা হয়েছে। (কারণ, আগেই বলা হয়েছে global $number-কে ব্যবহার করতে এখন $number এভাবে লিখতে হবে)।

এরপর ১০ নম্বর লাইনে এই ভেরিয়েবলটির সাথে তাকেই আবার যোগ করা হয়েছে। কোড রান করালে আউটপুট আসবে 20। কাজেই global শব্দ ব্যবহার করে ভেরিয়েবল ব্যবহারের সময় সতর্ক থাকতে হবে। সবচেয়ে ভালো $GLOBALS[“number”] এভাবে ব্যবহার করুন। তাহলে আর কোনো শঙ্কা থাকে না।

আরেকটি গুরুত্বপূর্ণ জিনিস, নিচের কোডটি দেখুন, এখানে echo করলে আউটপুট কত আসবে– ১০, ২০ না ৩০?

```
<?php
$number = 10;
$number = 20;
$number = 30;
echo $number;
?>
```

উত্তর হচ্ছে ৩০। কারণ, প্রথমে $number-এর মান ছিল ১০। এর পরেরবার $number-এর মান assign করা হয়েছে ২০। আর সবশেষে ৩০ এবং শেষেরটা echo হয়েছে।

এই কোডের ওপরের কোডটিতে যখন global $number লেখা হয়েছে, তখন এর মান ছিল ৮ (কারণ, ফাংশনের বাইরে গ্লোবাল $number-এর মান ৮ আছে)। পরের লাইনে $number = ১০ দেয়াতে এর মান assign হয়েছে ১০ এবং এটিই এখন এর আসল মান কজ

ফিডব্যাক : hossain.anower009@gmail.com

পিএইচপি টিউটোরিয়াল

(৬৫ পৃষ্ঠার পর)

পদ্ধতি-২ (global $number)

এই পদ্ধতিতে একটা মজার বিষয় আছে, একটু ভালোভাবে লক্ষ করুন। ধরুন, আগের মতোই ফাংশনের বাইরের $number (যেটা গ্লোবাল) ভেরিয়েবলটিকে ব্যবহার করার জন্য আমাদের শুধু $number-এর আগে global শব্দটি লিখে দিতে হবে। অর্থাৎ নিচের মতো global $number এবং এরপর থেকে যদি একাধিকবার এই ভেরিয়েবলটি ব্যবহার করতে হয়, তাহলে কীভাবে লিখবেন– global $number এভাবে? না। বরং এভাবে $number। এবার মজার কথাটি হলো নিচের কোডটি দেখুন–

```
<?php
$number = 8;.
function calculation(){
global $number;
$number  = 10;
$anothernumber = 20;
$addnumber  = $number+$number;
echo $addnumber;
}
calculation();
?>
```

উপরের কোডে (যেখানে $GLOBALS অ্যারে ব্যবহার করা হয়েছে) ৬ নম্বর লাইনে $number হচ্ছে লোকাল ভেরিয়েবল আর এই কোডে এই লাইনটি দিয়ে গ্লোবাল ভেরিয়েবলে নতুন নাম্বার ১০ assign করা হয়েছে। (কারণ, আগেই বলা হয়েছে global $number-কে ব্যবহার করতে এখন $number এভাবে লিখতে হবে)।

এরপর ১০ নম্বর লাইনে এই ভেরিয়েবলটির সাথে তাকেই আবার যোগ করা হয়েছে। কোড রান করালে আউটপুট আসবে 20। কাজেই global শব্দ ব্যবহার করে ভেরিয়েবল ব্যবহারের সময় সতর্ক থাকতে হবে। সবচেয়ে ভালো $GLOBALS[“number”] এভাবে ব্যবহার করুন। তাহলে আর কোনো শঙ্কা থাকে না।

আরেকটি গুরুত্বপূর্ণ জিনিস, নিচের কোডটি দেখুন, এখানে echo করলে আউটপুট কত আসবে– ১০, ২০ না ৩০?

```
<?php
$number = 10;
$number = 20;
$number = 30;
echo $number;
?>
```

উত্তর হচ্ছে ৩০। কারণ, প্রথমে $number-এর মান ছিল ১০। এর পরেরবার $number-এর মান assign করা হয়েছে ২০। আর সবশেষে ৩০ এবং শেষেরটা echo হয়েছে।

এই কোডের ওপরের কোডটিতে যখন global $number লেখা হয়েছে, তখন এর মান ছিল ৮ (কারণ, ফাংশনের বাইরে গ্লোবাল $number-এর মান ৮ আছে)। পরের লাইনে $number = ১০ দেয়াতে এর মান assign হয়েছে ১০ এবং এটিই এখন এর আসল মান কজ

ফিডব্যাক : hossain.anower009@gmail.com