

পিএইচপি টিউটোরিয়াল

আনোয়ার হোসেন

৫.৬

টাইপ কাস্টিং কেন : পিএইচপিতে ডাটাবেজ বা ডাটা প্রসেস করার সময় ডাটার টাইপ পরিবর্তন দরকার হতে পারে। কোনো ফর্মে হিডেন মান থাকতে পারে, যেমন- আইডি নাম্বার। ডাটাবেজে কোয়েরি করার সময় সে মানটিকে ইন্টজার (integer) হিসেবে প্রয়োজন হলে ডাটা টাইপ পরিবর্তন করে নিতে হবে। ফরম সাবমিট করলে \$REQUEST ['hidden_id'] এভাবে বা কোনো পিএইচপি পেজে রিসিভ করলে সেটা string হিসেবে আসবে। string দিয়ে কোয়েরি করা যাবে না। কারণ, ডাটাবেজে মানটি integer হিসেবে দেয়া আছে। এই ধরনের বামেলা এড়িয়ে যেতে টাইপ কাস্টিং জরুরি। হিডেন মানটি নেয়ার জন্য নিচের মতো করা যেতে পারে-

```
<?php
$x = (int)$REQUEST['hidden_id'];//cast to integer
?>
```

এখন \$x-এর মান যদি এমন আসে, তবে কাস্টিং কওে '0' বা integer বানিয়ে নেয়া হবে। পিএইচপিতে স্ট্রিংয়ের ভূমিকা অনেক, তাই এ সম্পর্কে ভালোভাবে ধারণা রাখতে হবে। স্ট্রিং ব্যবহারের আগে বানিয়ে নিতে হবে। স্ট্রিং দুইভাবে ব্যবহার করা যেতে পারে। ফাংশনে সরাসরি ব্যবহার করা যেতে পারে, আবার কোনো ভ্যারিয়েবলে স্টোর করেও ব্যবহার করা যায়।

```
1. <?php
2. $my_string = "o merciful make me bold and brave!";
3. echo "o merciful make me bold and brave!";
4. echo $my_string;
5. ?>
```

উপরের উদাহরণে প্রথম স্ট্রিংকে ভ্যারিয়েবলে চুকিয়ে দেয়া হয়েছে এবং পরে স্ট্রিংকে ইকো করা হয়েছে। এ ক্ষেত্রে কোনো ভ্যারিয়েবলে মান স্টোর করা হয়নি। মনে রাখতে হবে কোনো স্ট্রিংকে একাধিক ব্যবহার করতে চাইলে সেটা কোনো ভ্যারিয়েবলে স্টোর করে রাখা উচিত।

ওপরের কোডটি লিখে সেভ করে রান করলে নিচের মতো আউটপুট আসবে।



নোট : ওপরে double quotes দিয়ে স্ট্রিং তৈরি করা হয়েছে। single quotes দিয়েও স্ট্রিং তৈরি করতে পারেন।

```
1. <?php
2. $my_string = 'o merciful make me bold and brave!';
3. echo 'o merciful make me bold and brave!';
4. echo $my_string;
5. ?>
```

তবে যদি কখনও স্ট্রিংয়ের ভেতরে single quotes ব্যবহারের প্রয়োজন হয়, তবে এভাবে করুন- echo 'PHP it's neat'

double quotes ব্যবহারে কিছু সুবিধা থাকায় আমরা এটাই এখন ব্যবহার করব।

পিএইচপি লেখার এই দুটি পদ্ধতি সব ল্যাপটপে একইরকম। পিএইচপিতে শক্তিশালী একটি টুল আছে, যা দিয়ে বহু লাইনের স্ট্রিং লেখা যায় কোনো quotation ব্যবহার করা ছাড়াই। টুলটির নাম হচ্ছে heredoc। এ ক্ষেত্রে কোডিং করার সময় একটু সতর্ক হতে হবে। এটা নিচের মতো করেও করতে হবে-

```
1. <?php
2. $my_string = <<<TEST
3. He will be succeeded here
4. and here after!
5. TEST;
6. echo $my_string;
7. ?>
```

এভাবে স্ট্রিং লেখার সময় কিছু বিষয় লক্ষ রাখতে হবে-

* <<< অথবা identifier আছে যেগুলো ব্যবহার করতে হবে heredoc শুরু করার পূর্বে। যেমনটা ওপরে করা হয়েছে TEST লিখে।

* শেষেও এটি ব্যবহার করতে হবে এবং সেমিকোলন দিয়ে শেষ করতে হবে।

* এটি নিজেই একটি লাইন হবে। ফাঁকা রেখে লাইন শুরু করা যাবে না।

* আউটপুট আসবে নিচের মতো।



সব প্রোগ্রামিং ল্যাপটপে অপারেটর আছে। সবসময় ব্যবহার হয় এ ধরনের অপারেটরগুলো শেখা খুব গুরুত্বপূর্ণ। যদি কোনো মার্কেটপ্লেসে (ওয়েব ডেভেলপমেন্ট, ফিল্ডওয়ার ডেভেলপমেন্ট) বা জেড ইত্যাদির পরীক্ষা দেন, তবে সবগুলো শিখতে হবে। কেননা, যেসব অপারেটর সাধারণত ব্যবহার হয় না, সেগুলোর পরেও প্রচুর প্রশ্ন থাকে। যাই হোক, সেগুলো এখন দেখা যাক-

অপারেটর : যেমন \$x + \$y; এখানে \$x, \$y এগুলো হচ্ছে অপারেণ্ড (Operand) আর প্লাস (+) চিহ্ন হচ্ছে অপারেটর। আরও অনেক অপারেটর আছে। যেমন- +, -, x, ÷, !, ++, --, |, and ইত্যাদি। অনেক অপারেটর আছে যারা শুধু একটি অপারেণ্ডের ওপর কাজ করে। যেমন- ++ (increment অপারেটর) বা ! (not অপারেটর) ইত্যাদি। এসব অপারেটরকে ইউনারি (Unary) অপারেটর বলে। অনেক অপারেটর আছে, যারা দুটি অপারেণ্ডের ওপর কাজ করে। যেমন- + (Addition বা plus অপারেটর) বা - (Subtraction বা minus অপারেটর)। এসব অপারেটরকে বাইনারি (Binary) অপারেটর বলে।

বেশিরভাগ অপারেটর বাইনারি অপারেটর। অনেক অপারেটর আছে, যারা তিনটি অপারেণ্ডের ওপর কাজ করে। যেমন- ? :, এসব অপারেটরকে টেনারি (Ternary) অপারেটর বলে। টেনারি অপারেটর একটাই।

অপারেটরের কিছু শ্রেণিবিভাগ আছে। যেমন- Arithmetic বা গাণিতিক অপারেটর।

ছোটবেলায় অঙ্ক করছিলেন মনে আছে? ওইসব অঙ্কে যোগ, বিয়োগ, গুণ, ভাগের জন্য +, -, x, ÷ চিহ্নগুলো যেভাবে ব্যবহার করেছিলেন ঠিক তেমনি।

| উদাহরণ | অপারেটরের নাম | ব্যাখ্যা |
|------------|---------------------------|--|
| -\$x | Negation | \$x-এর বিপরীত |
| \$x + \$y | Addition বা যোগ | \$x ও \$y-এর যোগ করতে ব্যবহার হয়েছে |
| \$x - \$y | Subtraction বা বিয়োগ | \$x থেকে \$y বিয়োগ করতে ব্যবহার হয়েছে |
| \$x \$y | Multiplication বা গুণ | \$x ও \$y -এর গুণ করা হয়েছে |
| \$x ÷ \$y | Division বা ভাগ | \$x-কে \$y দিয়ে ভাগ করা হয়েছে |
| \$x % \$y | Exponentiation বা সূচকীয় | \$x-কে \$y দিয়ে ভাগ করার পর অবশিষ্টাংশ |
| \$x ** \$y | | পিএইচপি ৫.৬ ভা স েন এ সে ছে । এখনও ব্যবহার ব্যাপকভাবে শুরু হয়নি |

```
01. <?php
02. $x = 15; $y = 3;
03. echo 'Negation of $x : ' . (-$x) . '<br/>';
04. echo 'Addition of $x and $y : ' . ($x + $y) . '<br/>';
05. echo 'Subtraction of $y from $x : ' . ($x - $y) . '<br/>';
06. echo 'Multiplication of $x and $y : ' . ($x * $y) . '<br/>';
07. echo 'Division of $x by $y : ' . ($x/$y) . '<br/>';
08. echo 'Remainder of $x divided by $y : ' . ($x % $y) . '<br/>';
09. ?>
```

আউটপুট

Negation of \$x : -15
Addition of \$x and \$y : 18
Subtraction of \$y from \$x : 12
Multiplication of \$x and \$y : 45
Division of \$x by \$y : 5
Remainder of \$x divided by \$y : 0

** মডুলাস করার আগে অপারেণ্ড দুটিকে পূর্ণসংখ্যা বানিয়ে নেয় (যদি দশমিক থাকে)। এরপর মডুলাস করে। আর ভাজ্যের (যেটাকে ভাগ করা হচ্ছে) চিহ্নই হবে ফলাফলের চিহ্ন। যেমন- যদি \$x = -১৫; হয় এবং \$y = ৪; হয় তাহলে (\$x % \$y-এর) ফলাফল হয় -৪।

** প্রথম বন্ধনী দেয়াতে আগে অপারেশন হয়েছে। এরপর স্ট্রিংয়ের সাথে কনক্যাট (concat) হয়েছে। বন্ধনী উঠিয়ে দিলে ভুল ফল আসবে সবগুলোতে।

** \$x = ১৫-এর অর্থ ১৫, সুতরাং \$y = (\$x = ১৫) + ৩; হলে \$y-এর মান বা আউটপুট হবে ১৮। কারণ, এখন \$y = ১৫+৩ হয়ে গেছে