طراحی وب سایت

php

فصل دوم

مدرس : محمد امین‌زاده

زبان برنامه نویسی PHP چیست؟

PHP یک زبان تفسیرشونده است که اساس و بنیان آن طراحی و ایجاد صفحات پویای وب بوده است. به عبارتی دیگر، کسانی که وظیفه کدنویسی سایت را بر عهده دارند با استفاده از زبان PHP قادر خواهند بود وب اپلیکیشن‌هایی طراحی کنند که به صورت تعاملی می‌باشند. منظور از تعاملی بودن وجود قابلیت هایی مانند سطوح دسترسی مختلف برای کاربران، امکان ثبت‌نام در سایت، پنل کاربری ، ذخیره اطلاعات کاربران و بسیاری از امکانات دیگر است که امروزه در تمامی سایت‌های امروزی مشاهده می‌کنیم.

از نقاط قوت این زبان می‌توان به وجود فریمورک‌های بسیار قدرتمند و محبوبی هم چون لاراول و سیمفونی، دارا بودن بیش از ۸۰٪ از سهم بازار وب، متن باز و رایگان بودن، جامعه بزرگ و پشتیبانی خوب اشاره کرد. از طرفی PHP توانسته است خودش را به عنوان یکی از محبوب‌ترین زبان‌های برنامه نویسی در حوزه طراحی وب معرفی کند.

شروع برنامه نویسی با زبان PHP

کد زیر با استفاده از زبان PHP که درون کدهای HTML قرار گرفته، نوشته شده است:

یک اسکریپت PHP با <?php شروع می شود و با ?> پایان می یابد.

```
<!DOCTYPE html>
<html>
 <body>

  <?php
  echo "My first PHP script!";
  ?>

 </body>
</html>
```

متغیر در php

- ✓ در PHP، یک متغیر با علامت $ شروع می شود و به دنبال آن نام متغیر می آید.
- ✓ نام متغیر باید با یک حرف یا کاراکتر زیرخط شروع شود.
- ✓ نام متغیر نمی تواند با یک عدد شروع شود.
- ✓ نام متغیر فقط می‌تواند شامل نویسه‌های عددی و زیرخط باشد (A-z، 0-9و _)
- ✓ نام متغیرها به حروف بزرگ و کوچک حساس هستند ($age و $AGE دو متغیر متفاوت هستند)

```php
<?php
  $txt = "Hello world!";
  $x = 5;
  $y = 10.5;
?>
```

PHP echo and print

با PHP، دو راه اساسی برای دریافت خروجی وجود دارد: print و echo.

- echo مقدار بازگشتی ندارد در حالی که print مقدار بازگشتی ۱ دارد بنابراین می‌توان از آن در عبارات استفاده کرد.
- echo می‌تواند چندین پارامتر داشته باشد (اگرچه چنین استفاده‌ای نادر است) در حالی که print می‌تواند یک آرگومان داشته باشد.
- echo تا حدی سریعتر از print است.

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

```php
<?php
print "<h2>PHP is Fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

PHP Loops

اغلب هنگام نوشتن کد، می‌خواهید همان بلوک کد بارها و بارها به تعداد معینی اجرا شود. بنابراین، به جای اضافه کردن چندین خط کد تقریبا مساوی در یک اسکریپت، می توانیم از حلقه ها استفاده کنیم.

حلقه ها برای اجرای دوباره و دوباره همان بلوک کد استفاده می شوند، تا زمانی که یک شرط خاص درست باشد.

در PHP انواع حلقه های زیر را داریم:

while : تا زمانی که شرط مشخص شده درست باشد از طریق یک بلوک کد حلقه می زند

do...while : یک بار از طریق یک بلوک کد حلقه می زند و تا زمانی که شرط مشخص شده درست باشد حلقه را تکرار می کند.

for : از طریق یک بلوک کد به تعداد مشخصی بارها حلقه می زند

foreach : از طریق یک بلوک کد برای هر عنصر در یک آرایه حلقه می زند

PHP while Loop

حلقه while یک بلوک کد را تا زمانی که شرط مشخص شده درست باشد اجرا می کند.

مثال زیر اعداد ۱ تا ۵ را نشان می دهد:

```php
<?php
$x = 1;

while($x <= 5) {
  echo "The number is: $x <br>";
  $x++;
}
?>
```

PHP do while Loop

حلقه do...while همیشه بلوک کد را یک بار اجرا می‌کند، سپس شرط را بررسی می‌کند و در زمانی که شرط مشخص شده درست است، حلقه را تکرار می‌کند.

```php
<?php
$x = 1;

do {
  echo "The number is: $x <br>";
  $x++;
} while ($x <= 5);
?>
```

PHP for Loop

حلقه for زمانی استفاده می شود که از قبل می دانید اسکریپت چند بار باید اجرا شود.

مثال زیر اعداد ۰ تا ۱۰ را نشان می دهد:

```php
<?php
for ($x = 0; $x <= 10; $x++) {
  echo "The number is: $x <br>";
}
?>
```

PHP foreach Loop

حلقه foreach فقط روی آرایه ها کار می کند و برای حلقه زدن از طریق هر جفت کلید/مقدار در یک آرایه استفاده می شود.

مثال زیر مقادیر آرایه داده شده $colors را خروجی می‌دهد:

```php
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
  echo "$value <br>";
}
?>
```

اتصال به دیتابیس و فراخوانی داده

روش PDO جدیدترین روش اتصال به دیتابیس در PHP است. با PDO می‌توانیم به انواع مختلف دیتابیس متصل شویم. همچنین یکسری بررسی‌های امنیتی در هنگام اجرای کوئری‌ها انجام می‌دهد.

PDO روش‌های مختلفی را برای کار با Obejcts فراهم می‌کند و گزاره‌های تهیه شده را بازیابی می‌کند که باعث می‌شود کار توسعه‌دهنده بسیار راحت‌تر شود. این یک ابزار دسترسی به پایگاه داده در PHP است که از طریق آن دسترسی یکنواخت را از طریق چندین پایگاه داده امکان‌پذیر می‌کنیم.

اتصال به دیتابیس با PDO

می‌دانیم که در برنامه‌نویسی شی گرا، باید ابتدا یک شی از کلاس‌ها بسازیم. بنابراین برای ایجاد یک شئ PDO مشابه زیر عمل می‌کنیم:

متد سازنده PDO سه ورودی می‌گیرد که باید سه تای اول را مشخص کنیم. پارامتر اول، هاست (آدرس سرور دیتابیس) و در صورت نیاز نام دیتابیس را مشخص می‌کند. البته می‌توانیم اطلاعات دیگری نیز در این رشته متنی PHP تعیین کنیم.

پارامتر دوم و سوم، نام کاربری و رمز عبور اتصال به سرور دیتابیس را مشخص می‌کند.

```php
<?php
  $conn = new PDO();
?>
```

اتصال به دیتابیس با PDO

اگر به ساختار رشته ورودی اول دقت کنید، در ابتدا نوع دیتابیس را مشخص کرده‌ام. در اینجا از mysql استفاده می‌کنیم.

بعد از نوع دیتابیس، آدرس host و نام database را مشخص کرده‌ام.

```
header('Content-Type: text/html; charset=UTF-8');
$servername = "localhost";
$dbname = "نام دیتابیس";
$dsn = "mysql:host=$servername; dbname=$dbname; charset=utf8";
$username = "root";
$password = "";
$connection = new PDO($dsn, $username, $password);
```

مدیریت خطای PDO

اگر به هر دلیلی امکان برقراری ارتباط با سرور دیتابیس وجود نداشته باشد، قطعه کد اسلاید قبل موجب یک خطای مهلک (fatal) شده و اجرای اسکریپت متوقف می‌شود. برای مدیریت این خطا، می‌توانیم از مدیریت استثنا در PHP استفاده کنیم.

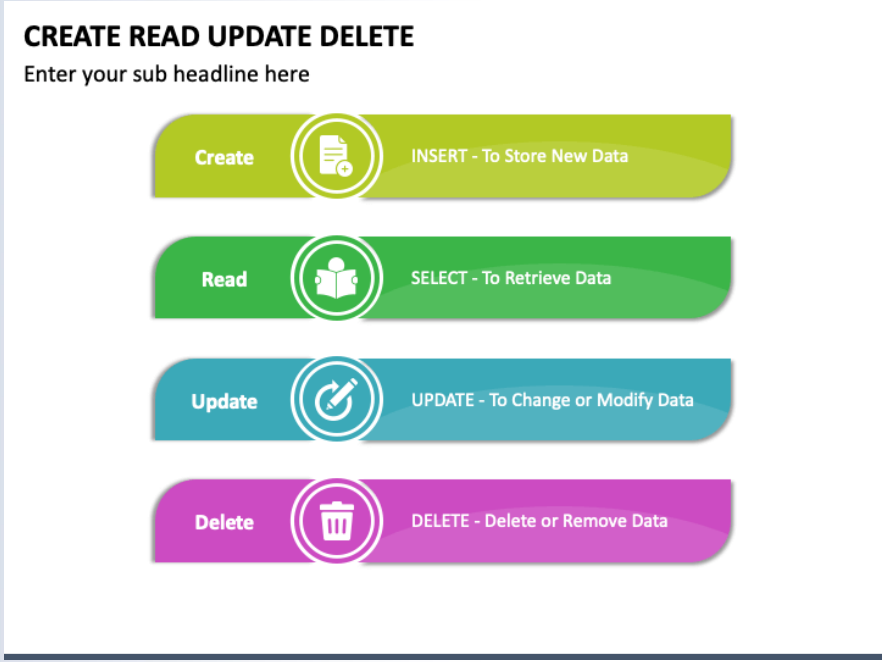
```
try {
   $connection = new PDO($dsn, $username, $password);

} catch (PDOException $e) {
   echo ' خطایی رخ داده است! ';
}
```

چهار دستور اصلی در دیتابیس

به طور کلی چهار دستور اصلی در دیتابیس وجود دارد که به وسیله‌ی آن‌ها، با دیتابیس ارتباط برقرار کرده و کار مورد نظر خود را انجام می‌دهیم. این چهار دستور به عبارت است از:

- Select
- Insert
- Delete
- Update



اجرای دستورهای PDO در PHP

اجرای کوئری با PDO

یکی از دستورهای پرکاربر در کار با دیتابیس به روش PDO در PHP متد ()query است. این متد یک ورودی گرفته و آن را مستقیماً در دیتابیس اجرا می‌کند.

دو دستور Select و Delete با متد ()query اجرا می‌شوند.

```
$res = $connection->query("SELECT * FROM نام جدول");
```

```
$connection->query("DELETE FROM نام جدول WHERE id='$id'");
```

ذخیره رکورد با PDO در PHP

اگر بخواهیم یک یا چند مقدار را از کاربر گرفته و از آن‌ها در دستور SQL استفاده کنیم، می‌توان از راهکار بهتری استفاده کرد. دستور Insert با متد ()prepare اجرا می‌شود.

بهتر است ابتدا کوئری مورد نظرمان را بنویسیم اما به جای تعریف مقادیر دریافتی از کاربر در کوئری، از علامت سوال (?) استفاده کنیم. اینطور به PDO می‌فهمانیم که قرار است به جای این علامت سؤال، یک مقدار جایگزین شود.

```
try {
   $connection = new PDO($dsn, $username, $password);
   $stmt = $connection->prepare("INSERT INTO نام جدول (نام ستون , نام ستون) VALUES (?,?)");
   $stmt->bindValue(1, $value1);
   $stmt->bindValue(2, $value2);
   $stmt->execute();

} catch (PDOException $e) {
 echo ' خطایی رخ داده است! ';
}
```

ویرایش رکورد با PDO در PHP

اگر بخواهیم یک یا چند مقدار را از کاربر گرفته و آن‌ها را در دستور SQL ویرایش کنیم، می‌توان از راهکار بهتری استفاده کرد. دستور Update با متد ()prepare اجرا می‌شود.

```
try {
   $connection = new PDO($dsn, $username, $password);
   $stmt = $connection->prepare("UPDATE نام جدول SET نام ستون = 'مقدار' WHERE id = 'مقدار' ");
   $stmt->execute();
} catch (PDOException $e) {
echo ' خطایی رخ داده است! ';
}
```

طراحی وب سایت

php

پایان فصل دوم

مدرس : محمد امین‌زاده