

In the name of God



Varastegan Institute For Medical Sciences

Data Structure & Programming 1

PHP & MySql

for Health Information Technology students

Lecturer:

Mostafa Jahangir (PhD)

www.mjahangir.ir

mjahangir@varastegan.ac.ir

Farnaz Yahosseynipour

yahosseynipourf@varastegan.ac.ir

1397-1398

۱. مقدمه ای بر PHP

PHP چیست؟

شما در درس های قبلی زبان های اسکریپت نویسی HTML و CSS که از نوع زبان های سمت Client هستند را یاد گرفتید. البته فراموش نکنید که برای برنامه نویسی سمت Client باید زبان JavaScript را هم یاد بگیرید که این زبان احتمالا در درس برنامه نویسی و ساختمان داده ها ۲ آموزش داده می شود.

حالا نوبت آن است که یک زبان سمت Server را نیز یاد بگیرید تا بتوانید یک سایت کامل را طراحی و پیاده سازی کنید. زبان های مختلفی برای برنامه نویسی سمت Server وجود دارد مثل: PHP, JAVA, Python, ASP.Net و تعداد زیادی زبان دیگر. قرار است در این درس، زبان PHP را به شما آموزش بدهم. PHP که مخفف Hypertext Preprocessor به معنای پردازشگر ابرمتن است، یک زبان پر استفاده و قدرتمند است که اگر آن را یاد بگیرید، هر نوع وب سایتی را می توانید با آن طراحی و پیاده سازی کنید. در ضمن فرصت های شغلی خوبی هم برای کسانی که این زبان را یاد دارند در ایران و سراسر جهان فراهم است. WordPress هم به عنوان محبوب ترین سیستم مدیریت محتوا (CMS) هم که پیش از یاد گرفتید نیز با این زبان نوشته شده است.

فایل های PHP

همان طور که می دانید پسوند فایل های HTML و CSS به ترتیب html و css است. پسوند فایل های PHP هم php است. دقت کنید که بر خلاف فایل های HTML و CSS که در سمت Client و بر روی کامپیوتر کاربر اجرا می شوند، فایل های PHP، در سمت Server و بر روی Server اجرا می شوند ولی نتیجه اجرای آن بر روی کامپیوتر کاربر دیده می شود. برای ایجاد فایل های PHP و برنامه نویسی به این زبان، شما می توانید از Editor هایی مثل Bracket یا Dreamweaver که قبلا در آن ها کدهای HTML و CSS را نوشتید، استفاده کنید.

نصب PHP

حالا نوبت هم که باشد نوبت نصب PHP است. اگر یادتان باشد، HTML و CSS نیاز به نصب نداشت چون مرورگرهایی مانند Mozila Firefox، Internet Explorer، Chrom و ... کدهای HTML و CSS را اجرا می کنند. اما برای اجرای فایل های PHP دو تا کار شما می توانید انجام دهید:

۱. راه اول این است که یک Server واقعی تهیه کنید به این صورت که یک Domain و Host که PHP را پشتیبانی می کند، تهیه کنید و فایل های PHP را روی آن اجرا کنید. خوب این کار برای شروع یادگیری برنامه نویسی PHP ضرورت ندارد چون بالاخره هزینه دارد.

۲. راه دوم این است که به جای خرید فضا در Server واقعی، روی کامپیوتر خودتان یک Server مجازی درست کنید. استفاده از Server مجازی برای آموزش PHP خیلی خوب است. هزینه هم ندارد. وقتی شما برنامه تان را به صورت

کامل بر روی این Server مجازی درست کردید، بعد می توانید آن را بر روی یک Server واقعی بارگذاری و اجرا کنید تا همه از طریق اینترنت بتوانند به آن دسترسی داشته باشند.

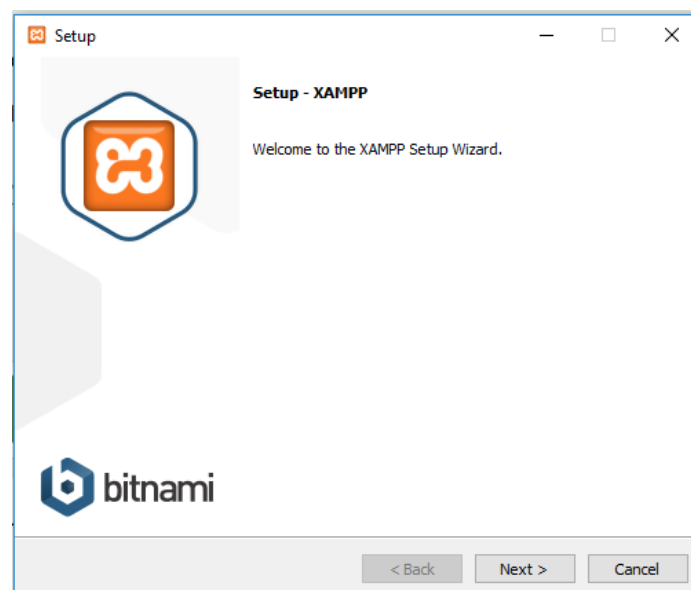
خب. ممکن است بپرسید که برای نصب Server مجازی باید چه کار کنیم؟ جواب این است که خیلی راحت است. شما می توانید نرم افزار XAMPP که یک سرور مجازی است را از سایت <https://www.apachefriends.org/index.html> دانلود و نصب کنید.



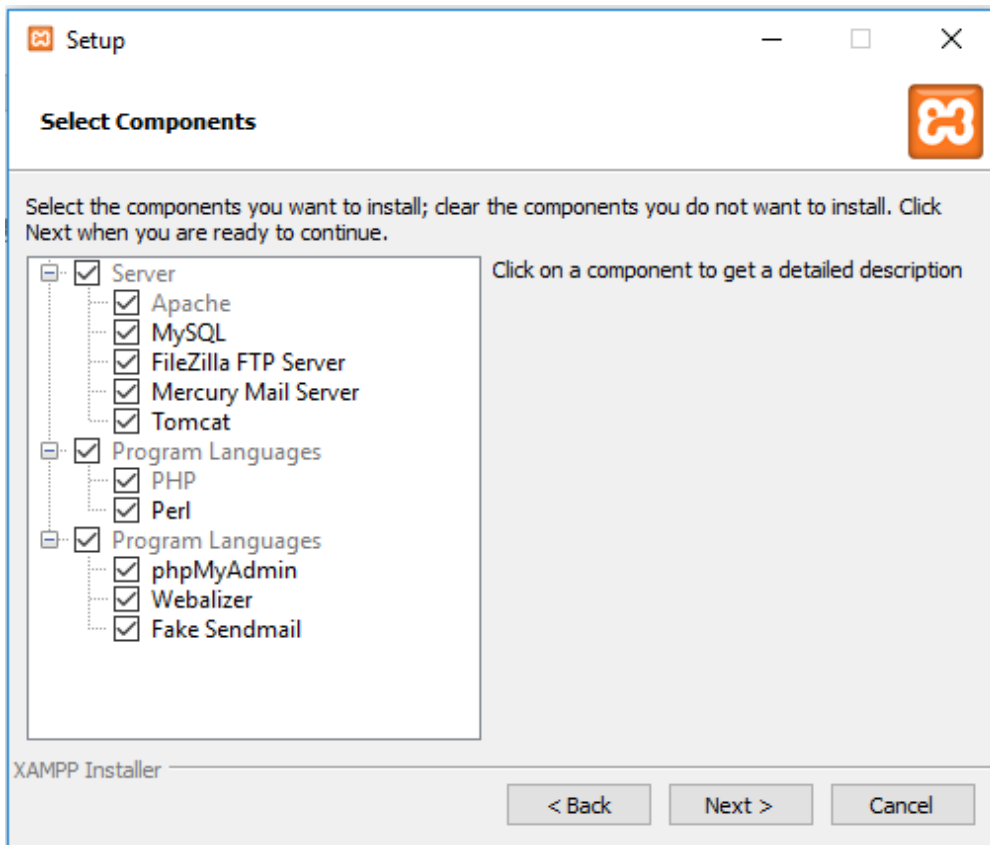
همان طور که در تصویر بالا می بینید، XAMPP را می توانید در Windows یا Linux یا OS X نصب کنید. پس بر اساس نوع سیستم عامل کامپیوترتان، روی لینک مربوطه کلیک کنید تا بر روی کامپیوترتان دانلود شود.

نصب Server مجازی XAMPP

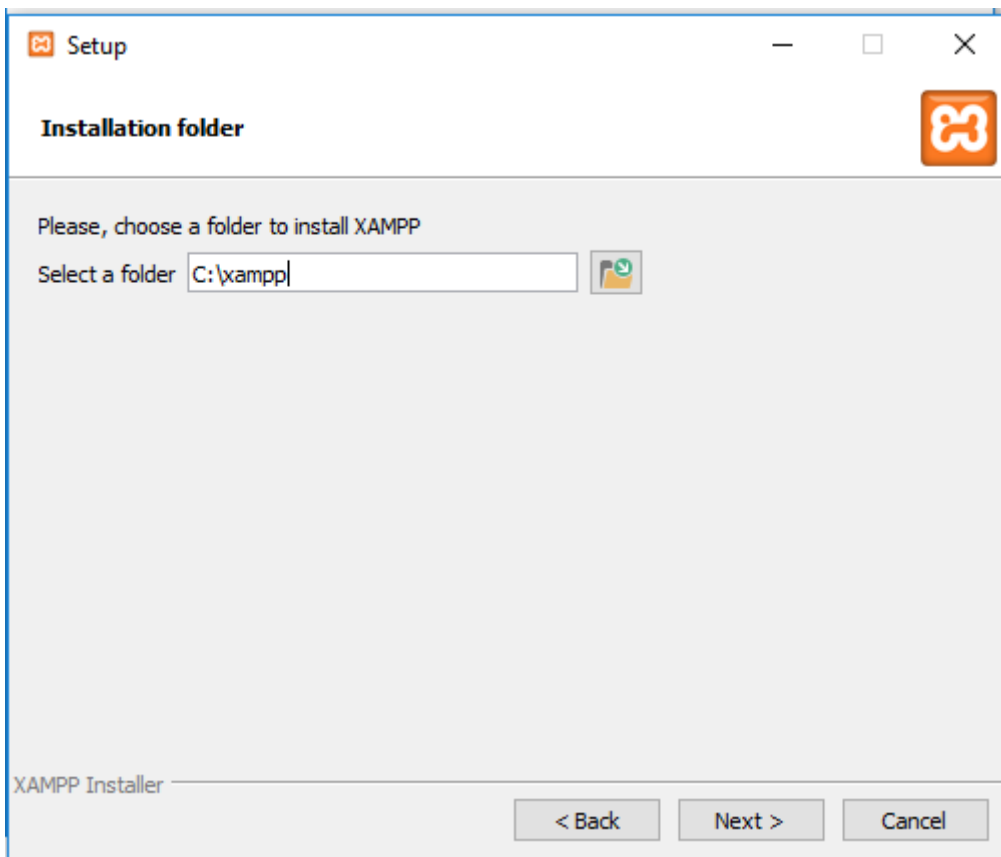
بعد از دانلود XAMPP نوبت نصب آن می رسد. تصویر زیر، اولین پنجره ای است که می بینید. روی Next کلیک کنید.



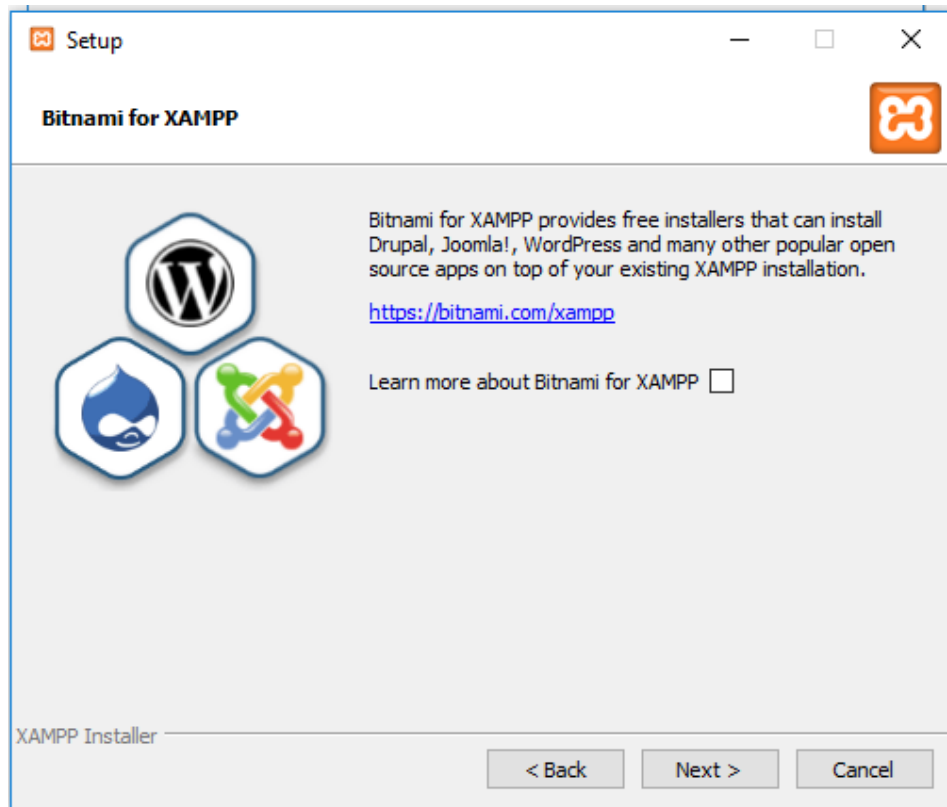
در صفحه دوم هم که نرم افزارهای مختلف موجود در XAMPP را نشان می دهد، بر روی Next کلیک کنید.



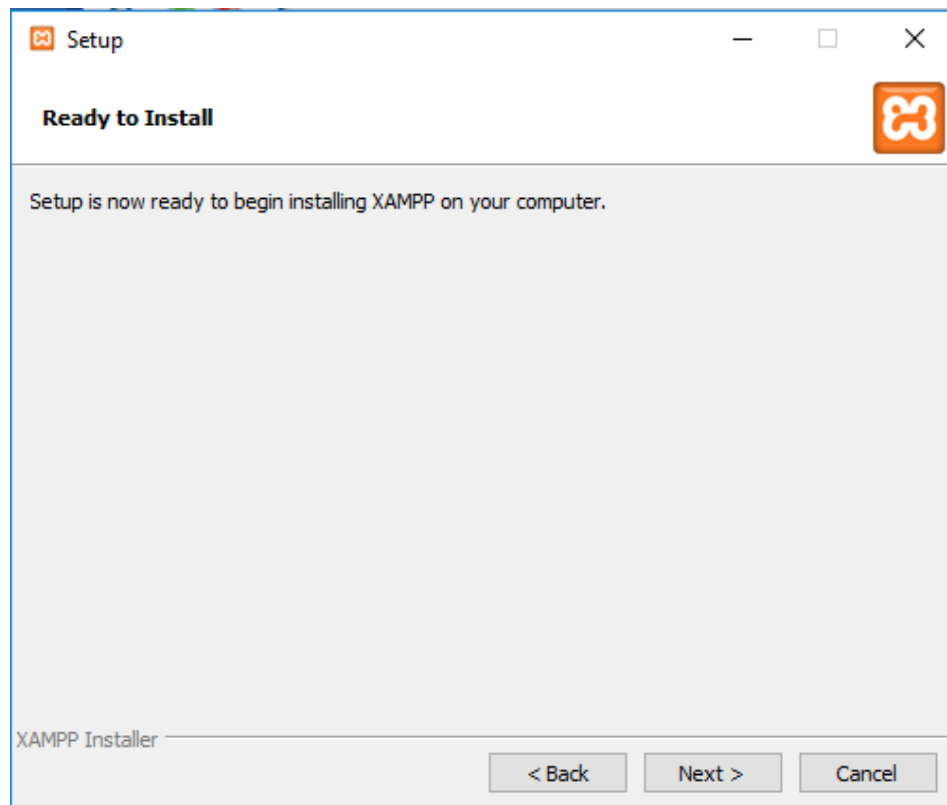
در صفحه سوم از شما می پرسد که مسیر نصب XAMPP کجا باشد که شما همان مسیر پیشفرض را تایید و دکمه Next را کلیک کنید.



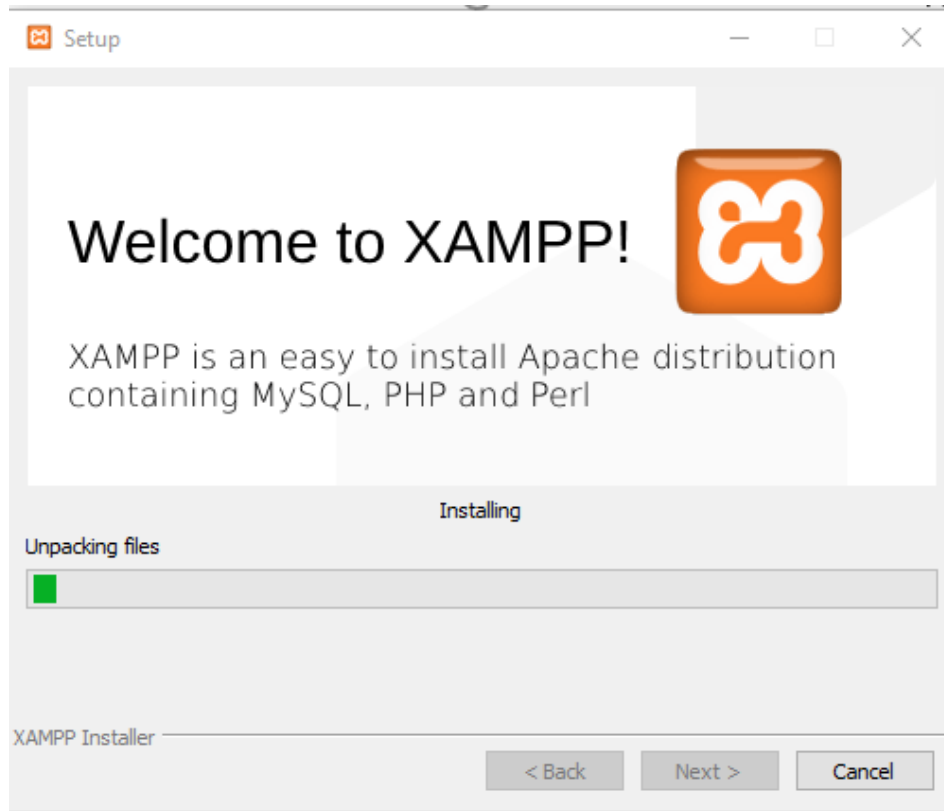
در صفحه چهارم هم بر روی **Next** کلیک کنید. فقط تیک جلوی گزینه **Learn more about ...** را بردارید. چون لازمش ندارید.



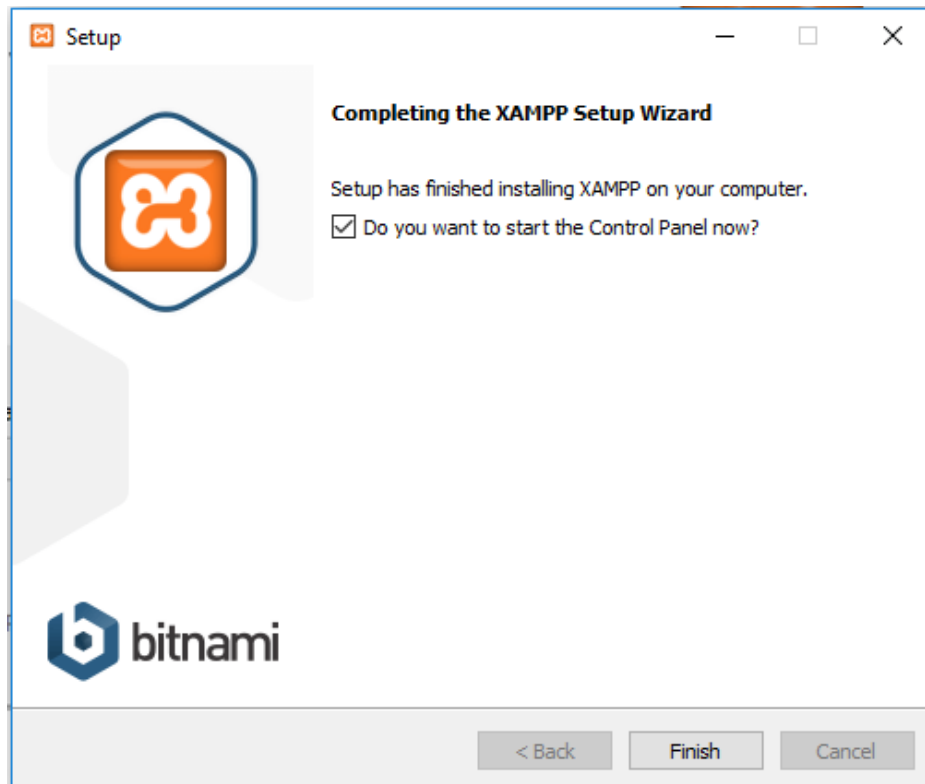
خب حالا در صفحه پنجم از شما می پرسد که آیا برای نصب XAMPP آماده اید که شما با کلیک کردن بر روی دکمه **Next** بهش بگویید بله.



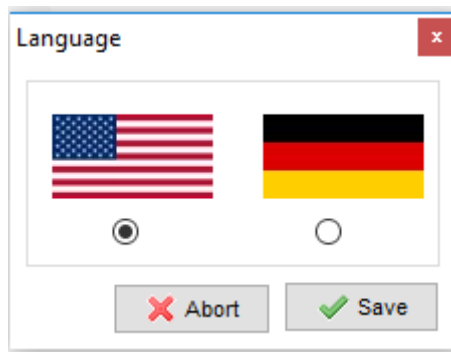
حالا شروع می کند به نصب کردن XAMPP.



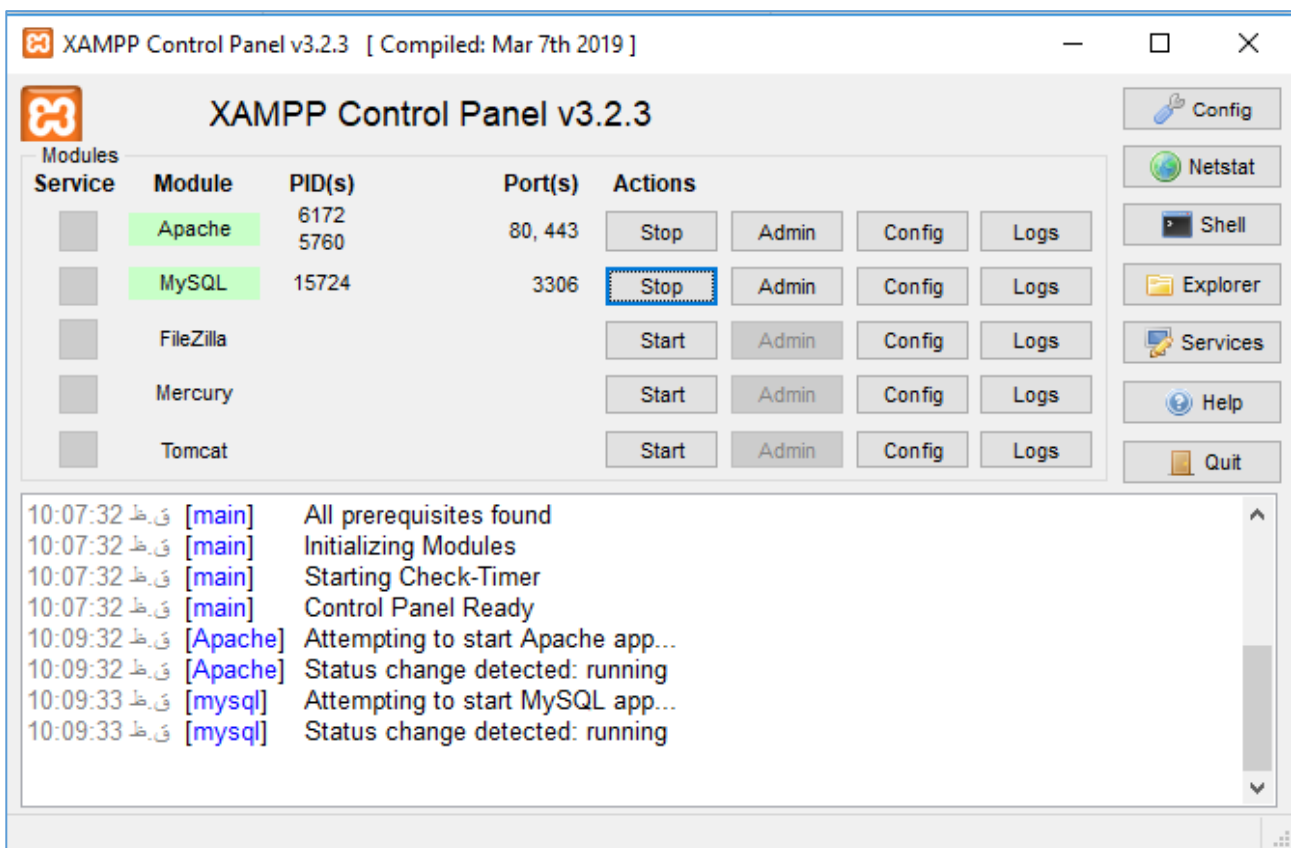
خب بالاخره نصب تمام شد. تیک گزینه **Do you want ...** را بگذارید بماند. نهایتاً روی **Finish** کلیک کنید.



بعد از کلیک بر روی **Finish** پنجره ای باز می شود که از شما می خواهد زبان مورد نظر خودتان را انتخاب کنید. اگر زبان آلمانی تان خوب است، آلمانی و گرنه همان زبان انگلیسی را انتخاب و روی دکمه **Save** کلیک کنید. 😊



خب حالا اصل کار یعنی Control panel نرم افزار XAMPP فعال می شود. در این صفحه دکمه Start گزینه های Apache و MySQL کلیک کنید تا مثل تصویر زیر فعال بشوند. Apache همان Server مجازی است که قبلا گفتیم، و MySQL هم پایگاه داده ای است که در PHP با آن کار می کنید.



خب این بار اول بود که بعد از نصب Control panel را باز می کردید. بعدا هم هر موقع خواستید با PHP برنامه نویسی کنید باید اول این Control panel را باز کنید و Apache و MySQL را Start بزنید. هر موقع هم کارتان تمام شد، بی زحمت این دو را Stop کنید.

اجرای اولین کد PHP

خب الان نوبت هم باشد نوبت اجرای اولیید کد PHP است. برای این کار نرم افزار Bracket را باز کنید. و کد زیر را در آن بنویسید. همان طور که مشاهده می کنید، می توان کدهای PHP را در لابه لای کدهای HTML نوشت. در برنامه زیر، کدی که به رنگ قرمز نوشته شده است، کد PHP است.

```

<html dir="rtl">
<head>
  <meta charset="utf-8">
  <title>آموزش PHP</title>
  <link rel="stylesheet" href="styles.css"/>
</head>
<body>
  <?php
  echo "سلام به سایت من خوش آمدید";
  ?>
</body>
</html>

```

همانطور که مشاهده می کنید کدهای PHP با تگ <?php شروع می شوند و با تگ >? بسته می شوند. حواستان باشد که همیشه کدهای PHP را حتما در بین این دو تگ بنویسید تا اجرا شوند. خب، حالا فایل را با نام و پسوند index.php در درایو C، پوشه xampp، پوشه htdocs، اینجا یک پوشه درست کنید به نام mysite، بعد وارد این پوشه بشوید و فایل تان را در این پوشه ذخیره کنید. یعنی مسیر زیر:

C:\xampp\htdocs\mysite\index.php

بعد که این فایل را ذخیره کردید، برای اجرای آن مرورگر را باز کنید. البته حواستان باشد که قبل از آن از control panel نرم افزار XAMPP، گزینه های Apache و MySQL را فعال کرده باشید. در مرورگر آدرس زیر را وارد کنید تا خروجی برنامه را ببینید.

http://localhost/mysite/index.php

احتمالا این سوال برای شما پیش آمده است که localhost در آدرس بالا چیست. ببینید. همان طور که قبلا گفتم، فایل های PHP باید روی Server اجرا شوند. در Server های واقعی شما باید آدرس اینترنتی سایت مربوطه را وارد کنید تا بتوانید به فایل مربوطه دسترسی داشته باشید ولی در Server مجازی آدرس دسترسی به فایل های PHP شما، localhost است.

متغیرها (Variables)

متغیر جایی است که در آن داده‌ها ذخیره می‌شوند. در PHP، علامت \$ باید قبل از نام متغیر بیاید. در برنامه زیر سه متغیر تعریف شده‌اند. حتماً حواستان هست که وقتی می‌خواهید یک مقدار متنی را در یک متغیر بریزید از "" استفاده کنید.

```
<body>
<?php
$txt = "خوش آمدید";
$x = 5;
$y = 10.5;
echo $txt;
echo $x;
echo $y;
?>
</body>
```

در اینجا چند قاعده برای نامگذاری متغیرها در PHP را برای شما می‌گوییم:

۱. نام معنادار برای متغیرتان انتخاب کنید. مثلاً برای نام از \$fname یا برای سن از \$age استفاده کنید تا وقتی تعداد متغیرهایتان زیاد شد بدانید هر متغیر برای چه منظوری استفاده شده است.
۲. نام متغیر بعد از علامت \$ حتماً باید با یک حرف (\$fname) و یا _ (\$_age) شروع شود. اصلاً متغیرتان را با عدد شروع نکنید. مثلاً \$1stname نامگذاری اشتباهی است.
۳. در نامگذاری متغیر، بعد از علامت \$ فقط می‌توانید از حروف انگلیسی (A-Z)، اعداد (0-9) و _ استفاده کنید و استفاده از کاراکترهای دیگر مجازی نیست. ☹️
۴. نام متغیرها حساس به بزرگی و کوچکی کاراکترها (case-sensitive) است. یعنی متغیر \$AGE با \$age فرق می‌کند و دو متغیر جداگانه محسوب می‌شوند.

دستور echo

همان‌طور که در برنامه‌های قبلی مشاهده کردید، دستور echo برای نمایش متن یا مقادیر متغیرها در خروجی استفاده می‌شود. در برنامه زیر، متن به صورت ترکیب با کدهای HTML مثل h2 و
 آورده شده است.

```
<?php
echo "<h2>وب سایت آموزش برنامه نویسی</h2>";
echo "<br>به سایت خوش آمدید";
?>
```

در برنامه زیر، نحوه نمایش مقادیر متغیرها توسط دستور echo آورده شده است.

```
<?php
$txt1 = "یادگیری برنامه نویسی وب";
$txt2 = "علوم پزشکی و ارستگان";
$x = 5;
$y = 4;
echo "<h2>" . $txt1 . "</h2>";
echo "<br>" . $txt2 . " یادگیری برنامه نویسی وب در";
echo $x + $y;
```

- در ادامه می‌خواهیم برنامه‌ای بنویسیم که زمانی که کاربر نام و نام خانوادگی خود را وارد می‌کند بعد از کلیک بر روی دکمه ارسال، یک پیام خوش آمدگویی شامل "نام و نام خانوادگی کاربر + به سایت خوش آمدید" نمایش داده شود. همان طور که مشاهده کردید، با استفاده از دستور :

```
<form action="program1.php" method="post">
```

مشخص می‌کنیم که فرم طراحی شده ما در HTML با فایلی با نام program1.php ارتباط برقرار می‌کند و دستورات مربوط به PHP را اجرا می‌کند.

**** در پایان هر برنامه خروجی برنامه مورد نظر قرار گرفته است. ****

آموزش کار با متغیر سراسری \$_POST در PHP

از متغیر سراسری \$_POST برای جمع‌آوری اطلاعات یک فرم HTML پس از ارسال یا در اصطلاح submit آن به روش "method="post" استفاده می‌شود. همچنین از متغیر \$_POST می‌توان برای ارسال متغیرها نیز استفاده کرد.

کد HTML (برنامه ۱)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<form action="program1.php" method="post">
نام : <br>
<input type="text" name="fname"><br>
نام خانوادگی : <br>
<input type="text" name="lname"><br>
<input type="submit" value="ارسال">
</form>
</body>
</html>
```

```
1 ▼ <html dir="rtl">
2 ▼ <head>
3   <meta charset="utf-8">
4 </head>
5 ▼ <body>
6 ▼ <form action="program1.php" method="post">
7   نام : <br>
8   <input type="text" name="fname"><br>
9   نام خانوادگی : <br>
10  <input type="text" name="lname"><br>
11  <input type="submit" value="ارسال">
12 </form>
13 </body>
14 </html>
```

کد PHP (برنامه ۱)

در دستور `echo` با استفاده از تعریف متغیرهای `fname` و `lname` که به ترتیب نام و نام خانوادگی را شامل می شود و متن مورد نظر خود را نوشته تا بدین ترتیب خروجی مورد نظر ما نمایش داده شود. قراردادن نقطه (.) بین متغیرهای `fname` و `lname` باعث می شود که عبارت این دو متغیر به همدیگر متصل گردد. توجه داشته باشید که `$_POST` که یکی از روش های ارسال اطلاعات بوده و می بایست تمام حروف به صورت بزرگ نوشته شود.

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<?php
  echo $_POST["fname"]. " ". $_POST["lname"]. "به سایت خوش آمدید.";
?>
</body>
</html>
```

```
1 <html dir="rtl">
2 <head>
3   <meta charset="utf-8">
4 </head>
5 <body>
6
7 <?php
8   echo $_POST["fname"]. " ". $_POST["lname"]. "به سایت خوش آمدید.";
9 ?>
10
11 </body>
12 </html> |
```

خروجی

- برنامه بعدی که می خواهیم بنویسیم برنامه ای است که از کابر اطلاعاتی را دریافت کرده و با توجه به اطلاعات وارد شده رسید نوبت دهی را چاپ می کند.

ابتدا فرم مورد نظر خود را با استفاده از دستورات HTML نوشته و مانند برنامه قبلی با استفاده از دستور `$_POST` متغیرهای خواسته شده را فراخوانی می کنیم. به نحوه نوشتن دستورات دقت نمایید که متغیرها می بایست حتما داخل " " قرار گیرد.

کد HTML (برنامه ۲)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
```

```

</head>
<body>
<form action="program2.php" method="post">
  <fieldset>
    <h4> فرم نوبت دهی آنلاین </h4>
    نام و نام خانوادگی:
    <input type="text" name="name"><br>
    جنسیت:
    <input type="radio" name="gender" value="مرد"> مرد
    <input type="radio" name="gender" value="زن"> زن<br>
    سن:
    <input type="text" name="age"><br>
    پزشک مورد درخواست:
    <select name="doctors">
      <option value="دکتر عظیمی متخصص قلب">دکتر عظیمی متخصص قلب
      <option value="دکتر صاحبی متخصص اطفال">دکتر صاحبی متخصص اطفال
      <option value="دکتر محمودی متخصص پوست">دکتر محمودی متخصص پوست
      <option value="دکتر نخعی متخصص گوش و حلق و بینی">دکتر نخعی متخصص گوش و حلق و بینی
    </select><br>
    تاریخ ویزیت:
    روز: <input type="text" name="day"><br>
    ماه: <input type="text" name="month"><br>
    سال: <input type="text" name="year"><br>
    <input type="submit" value="ارسال"><br>
  </fieldset>
</form>
</body>
</html>

```

```

1 <html dir="rtl">
2 <head>
3   <meta charset="utf-8">
4 </head>
5 <body>
6 <form action="program2.php" method="post">
7   <fieldset>
8     <h4>فرم نوبت دهی آنلاین</h4>
9     نام و نام خانوادگی:
10    <input type="text" name="name"><br>
11    جنسیت:
12    <input type="radio" name="gender" value="مرد"> مرد
13    <input type="radio" name="gender" value="زن"> زن<br>
14    سن:
15    <input type="text" name="age"><br>
16    پزشک مورد درخواست:
17    <select name="doctors">
18      <option value="دکتر عظیمی متخصص قلب">دکتر عظیمی متخصص قلب
19      <option value="دکتر صاحبی متخصص اطفال">دکتر صاحبی متخصص اطفال
20      <option value="دکتر محمودی متخصص پوست">دکتر محمودی متخصص پوست
21      <option value="دکتر نخعی متخصص گوش و حلق و بینی">دکتر نخعی متخصص گوش و حلق و بینی
22    </select><br>
23    تاریخ ویزیت:
24    روز: <input type="text" name="day"><br>
25    ماه: <input type="text" name="month"><br>
26    سال: <input type="text" name="year"><br>
27    <input type="submit" value="ارسال">
28  </fieldset>
29 </form>
30 </body>
31 </html>

```

کد PHP (برنامه ۲)

```

<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<h4>رسید نوبت دهی </h4>
<table>
  <tr>
    <td> نام و نام خانوادگی </td>
    <td><?php echo $_POST["name"] ?></td>
  </tr>
  <tr>
    <td> جنسیت </td>
    <td><?php echo $_POST["gender"] ?></td>
  </tr>
  <tr>
    <td> سن </td>
    <td><?php echo $_POST["age"] ?></td>
  </tr>
  <tr>
    <td> پزشک مورد درخواست </td>
    <td><?php echo $_POST["doctors"] ?></td>
  </tr>
  <tr>
    <td> تاریخ ویزیت </td>
    <td><?php echo $_POST["year"]."/".$_POST["month"]."/". $_POST["day"] ?></td>
  </tr>
</table>
</body>
</html>

```

```

1 <html dir="rtl">
2 <head>
3   <meta charset="utf-8">
4 </head>
5 <body>
6 <h4>رسید نوبت دهی</h4>
7 <table>
8   <tr>
9     <td> نام و نام خانوادگی</td>
10    <td><?php echo $_POST["name"] ?></td>
11  </tr>
12  <tr>
13    <td>جنسیت</td>
14    <td><?php echo $_POST["gender"] ?></td>
15  </tr>
16  <tr>
17    <td>سن</td>
18    <td><?php echo $_POST["age"] ?></td>
19  </tr>
20  <tr>
21    <td>پزشک مورد درخواست</td>
22    <td><?php echo $_POST["doctors"] ?></td>
23  </tr>
24  <tr>
25    <td>تاریخ ویزیت</td>
26    <td><?php echo $_POST["year"]."/".$_POST["month"]."/". $_POST["day"] ?></td>
27  </tr>
28 </table>
29
30 </body>
31 </html>

```

خروجی :

رسید نوبت دهی:

نام و نام خانوادگی: علی محمدی

جنسیت: مرد

سن: 19

پزشک مورد درخواست: دکتر نخعی متخصص گوش و حلق و بینی

تاریخ ویزیت: 98/03/25

فرم نوبت دهی آنلاین

نام و نام خانوادگی:

جنسیت:
 مرد زن

سن:

پزشک مورد درخواست:

تاریخ ویزیت:

دستور شرطی if

نخستین عبارت کلیدی که در دستورات شرطی استفاده می شود "اگر" است که در برنامه نویسی به صورت "if" به کار می رود، با دستور if درستی یا نادرستی پاسخ ارزیابی شرطی را در مقایسه بررسی می کنیم.

If (condition)

قبل از اجرای دستور if ابتدا شرط بررسی می شود، اگر شرط برقرار باشد (یعنی درست باشد) کد اجرا می شود. شرط، یک عبارت مقایسه ای است. می توان از عملگرهای مقایسه ای برای تست درست یا اشتباه بودن شرط استفاده کرد. شما می توانید چندین دستور را در داخل دستور if بنویسید. کافیست که از یک آکولاد {} برای نشان دادن ابتدا و انتهای دستورات استفاده کنید. همه دستورات بین آکولاد جزء بدنه دستور if هستند. نحوه تعریف چند دستور در داخل بدنه if به صورت زیر است:

```
If (condition)
{
Statement1;
Statement2;
.
.
.
Statement N;
}
```

برنامه ۳: در برنامه زیر می خواهیم با استفاده از دستور if برنامه ای بنویسیم که در صورتی که میزان قند خون بین بازه اعداد ۷۲ تا ۱۰۸ وارد شود وضعیت قند خون را نرمال اعلام کند.

کد HTML (برنامه ۳)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<form action="program3.php" method="post">
  <fieldset>
    <br> : میزان قند خون خود را وارد کنید
    <input type="text" name="FBS"><br>
    <input type="submit" value="ارسال">
  </fieldset>
</form>
</body>
</html>
```

```
1 ▼ <html dir="rtl">
2 ▼ <head>
3   <meta charset="utf-8">
4 </head>
5 ▼ <body>
6 ▼ <form action="program3.php" method="post">
7 ▼   <fieldset>
8     <br> : میزان قند خون خود را وارد کنید
9     <input type="text" name="FBS"><br>
10    <input type="submit" value="ارسال">
11    </fieldset>
12  </form>
13 </body>
14 </html> |
```

کد PHP (برنامه ۳)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<h4> : نتیجه آزمایش قند خون </h4>
<?php
  if ($_POST["FBS"]>=72 and $_POST["FBS"]<=108){
    echo "وضعیت قند خون نرمال است ";
  }
}
```

```

?>
</body>
</html>

```

```

1 <html dir="rtl">
2 <head>
3   <meta charset="utf-8">
4 </head>
5 <body>
6 <h4>نتیجه آزمایش قند خون</h4>
7 <?php
8   if ($_POST["FBS"]>=72 and $_POST["FBS"]<=108){
9     echo "وضعیت قند خون نرمال است";
10  }
11  ?>
12 </body>
13 </html>

```

خروجی :

نتیجه آزمایش قند خون:

وضعیت قند خون نرمال است

←

میزان قند خون خود را وارد کنید::

دستور شرطی if ... else

شاید از خود بپرسیم اگر شرط if درست (true) نشد آنوقت تکلیف چیست؟! برنامه نویس خوب باید حساب همه چیز را کرده باشد و این چیزی است که شکل پیشرفته آن را هوش مصنوعی می نامند، پس باید شرایط بعد از برقرار نبودن رابطه if را هم تعریف کنیم، برای این کار از دستور شرطی else به معنی "در غیر اینصورت" استفاده می کنیم. else در واقع جایگزینی برای if است یعنی اگر رابطه if درست نباشد (false) باشد، آنگاه مقادیر بین {} در else اجرا می شود.

نحوه استفاده از این دستور مانند دستور if است با این تفاوت که بعد از بسته شدن آکولاد دستور if، دستور else اجرا می شود:

```

If (condition)
{
  code to execute;
}
else
{
  Another code to execute;
}

```


دستور شرطی if ... else if

این دستور این امکان را فراهم می کند تا در صورت عدم برقراری شرط دستور if، شرط های دیگری را نیز بررسی کنیم. ساختار کلی استفاده از این دستور به شرح ذیل است:

```
If (condition)
{
  code to execute;
}
else if (another condition)
{
  Another code to execute;
}
```

نکته مهم: از دستور else if می توانیم به هر تعداد که بخواهیم ما بین if و else استفاده کنیم، مفسر PHP به ترتیب از بالا به پایین مورد به مورد برقرار بودن رابطه آنها را بررسی کرده و اگر یکی از شرط ها درست (true) باشد مقادیر داخل {} اجرا شده و سلسله دستورات شرطی در همان نقطه خاتمه پیدا می کند.

برنامه ۴: در برنامه زیر با وارد کردن میزان قند خون با توجه به عدد وارد شده و شرط هایی که در برنامه قائل شده ایم وضعیت قند خون نمایش داده می شود. در اینجا کاربرد دستور if ... else if را مشاهده می کنید.

کد HTML (برنامه ۴)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<form action="program4.php" method="post">
  <fieldset>
    <br> : میزان قند خون خود را وارد کنید
    <input type="text" name="FBS"><br>
    <input type="submit" value="ارسال">
  </fieldset>
</form>
</body>
</html>
```

```

1 ▼ <html dir="rtl">
2 ▼ <head>
3     <meta charset="utf-8">
4 </head>
5 ▼ <body>
6 ▼ <form action="program4.php" method="post">
7 ▼     <fieldset>
8         <br> میزان قند خون خود را وارد کنید
9         <input type="text" name="FBS"><br>
10        <input type="submit" value="ارسال">
11    </fieldset>
12 </form>
13 </body>
14 </html>

```

کد PHP (برنامه ۴)

همان طور که مشاهده می کنید اگر عدد وارد شده بین مقادیر ۷۲ تا ۱۰۸ باشد وضعیت قند خون نرمال است، اگر بین ۱۰۹ تا ۱۲۵ باشد قند خون در وضعیت پیش دیابتی است و بیشتر از ۱۲۶ باشد قند خون در وضعیت دیابتی است. در انتهای برنامه دستور else قرار داده شده که با بررسی شروط بالا اگر عدد وارد شده در هیچ کدام از موارد فوق نبود پیغام "مقدار قند خون وارد شده خارج از محدوده است" را نمایش می دهد.

```

<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<h4>نتیجه آزمایش قند خون</h4>
<?php
  if ($_POST["FBS"]>=72 and $_POST["FBS"]<=108){
    echo "وضعیت قند خون نرمال است";
  }
  elseif ($_POST["FBS"]>=109 and $_POST["FBS"]<=125){
    echo "قند خون در وضعیت پیش دیابتی است";
  }
  elseif ($_POST["FBS"]>=126){
    echo "قندخون در وضعیت دیابتی است";
  }
  else{
    echo "مقدار قند خون وارد شده خارج از محدوده است";
  }
?>
<br><a href="index4.html">بازگشت</a>
</body>
</html>

```

```

1 <html dir="rtl">
2 <head>
3   <meta charset="utf-8">
4 </head>
5 <body>
6 <h4>نتیجه آزمایش قند خون</h4>
7 <?php
8   if ($_POST["FBS"]>=72 and $_POST["FBS"]<=108){
9     echo "وضعیت قند خون نرمال است";
10  }
11  elseif ($_POST["FBS"]>=109 and $_POST["FBS"]<=125){
12    echo "قند خون در وضعیت پیش دیابتی است";
13  }
14  elseif ($_POST["FBS"]>=126){
15    echo "قند خون در وضعیت دیابتی است";
16  }
17  else{
18    echo "مقدار قند خون وارد شدن خارج از محدوده است";
19  }
20  ?>
21  <br><a href="index4.html">بازگشت</a>
22 </body>
23 </html>

```

خروجی :

در برنامه بعدی نیز از دستور if ... else if استفاده شده است با این تفاوت که در این برنامه با توجه به اینکه از دو متغیر visit_type و insurance استفاده شده است برای هر یک دستورات if ... else if جداگانه ای را در نظر گرفته است.

برنامه ۵ : در برنامه زیر با وارد کردن نوع ویزیت و وضعیت بیمه، مبلغ ویزیت نمایش داده می شود.

کد HTML (برنامه ۵)

```

<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<h4>رسید نوبت </h4>
<form action="program5.php" method="post">
  <fieldset>
<br> : نوع ویزیت
  <select name="visit_type">
    <option value="ssd">پزشک فوق تخصص </option>
    <option value="sd">پزشک متخصص </option>
    <option value="gp">پزشک عمومی </option>
  </select> <br>
  <br> : وضعیت بیمه
  <input type="radio" name="insurance" value="yes"> دارد <br>
  <input type="radio" name="insurance" value="no"> ندارد <br>
  <input type="submit" value="صدور رسید نوبت">

```

```

</fieldset>
</form>
</body>
</html>

```

```

1 ▾ <html dir="rtl">
2 ▾ <head>
3     <meta charset="utf-8">
4 </head>
5 ▾ <body>
6 <h4>رسید نوبت</h4>
7 ▾ <form action="program5.php" method="post">
8     <fieldset>
9         نوع ویزیت: <br>
10 ▾ <select name="visit_type">
11     <option value="ssd">پزشک فوق تخصص</option>
12     <option value="sd">پزشک متخصص</option>
13     <option value="gp">پزشک عمومی</option>
14 </select> <br>
15     وضعیت بیمه: <br>
16     <input type="radio" name="insurance" value="yes"> دارد<br>
17     <input type="radio" name="insurance" value="no"> ندارد<br>
18     <input type="submit" value="صدور رسید نوبت">
19 </fieldset>
20 </form>
21 </body>
22 </html>

```

کد PHP (برنامه ۵)

نکته: عملگر == یعنی دو طرف رابطه برابر باشند و عملگر != یعنی دو طرف رابطه برابر نباشند

```

<html dir="rtl">
<head>
    <meta charset="utf-8">
</head>
<body>
<h4>رسید نوبت </h4>
<?php
    if ($_POST["visit_type"]=="ssd"){
        $cost=47900;
    }
    elseif ($_POST["visit_type"]=="sd"){
        $cost=38400;
    }
    elseif ($_POST["visit_type"]=="gp"){
        $cost=24500;
    }
    if ($_POST["insurance"]=="yes"){
        $insurance=$cost*0.3;
    }
    elseif ($_POST["insurance"]=="no"){
        $insurance=0;
    }
    $total=$cost-$insurance;

```

```

echo "مبلغ ویزیت: ".$cost . "<br>";
echo "مبلغ بیمه: ". $insurance. "<br>";
echo "مبلغ قابل پرداخت: ". $total. "<br>";
?>
<br><a href="index5.html">بازگشت</a>
</body>
</html>

```

```

1 <html dir="rtl">
2 <head>
3   <meta charset="utf-8">
4 </head>
5 <body>
6 <h4>رسید نوبت</h4>
7 <?php
8   if ($_POST["visit_type"]=="ssd"){
9     $cost=47900;
10  }
11  elseif ($_POST["visit_type"]=="sd"){
12    $cost=38400;
13  }
14  elseif ($_POST["visit_type"]=="gp"){
15    $cost=24500;
16  }
17
18  if ($_POST["insurance"]=="yes"){
19    $insurance=$cost*0.3;
20  }
21  elseif ($_POST["insurance"]=="no"){
22    $insurance=0;
23  }
24  $total=$cost-$insurance;
25  echo "مبلغ ویزیت: ". $cost . "<br>";
26  echo "مبلغ بیمه: ". $insurance. "<br>";
27  echo "مبلغ قابل پرداخت: ". $total;
28  ?>
29  <br><a href="index5.html">بازگشت</a>
30 </body>
31 </html>

```

خروجی:

رسید نوبت:	رسید نوبت:
مبلغ ویزیت: 38400 مبلغ بیمه: 0 مبلغ قابل پرداخت: 38400 <u>بازگشت</u>	نوع ویزیت: <input type="text" value="پزشک متخصص"/> وضعیت بیمه: <input type="radio"/> دارد <input checked="" type="radio"/> ندارد <input type="button" value="صدور رسید نوبت"/>

دستور شرطی Switch

دستور switch که یکی دیگر از دستورات شرطی می باشد را می توانید یک حالت دیگر از if-else در نظر بگیرید که با استفاده از این دستور می توان یک متغیر را با شرط ها بسیار زیادی مقایسه نمود، دستور switch اینگونه نوشته می شوند:

```
<?php
switch (n)
{
case label1:
code to be executed if n= label1;
break ;
case label2:
code to be executed if n= label2;
break ;
default:
code to be executed if n is different from both label1 and label2;
}
?>
```

همانطور که قطعه کد بالا را مشاهده نمودید ، ما یک متغیر را با دستور switch مورد شرط قرار می دهیم (n). سپس با استفاده از دستور case یک مقدار جایگذاری می کنیم که اگر n برابر با label1 بود دستورات ما اجرا شود و سپس با استفاده از دستور break از شرط خارج می شود و به همین صورت می توانید دستورات شرطی خود را با استفاده از case قرار دهید و در آخر یک بخش (Default) دارد که این دستور در صورتی اجرا می شود که مقدار متغیر با هیچ یک از مقادیر دستورات case برابر نباشد.

دستور Default اختیاری است و اگر از بدنه switch حذف شود هیچ اتفاقی نمی افتد. مکان این دستور هم مهم نیست اما بر طبق تعریف، آن را در پایان دستورات می نویسند.

نکته: حتی اگر تعداد کدهای داخل دستور case از یکی بیشتر باشد نباید از آکولاد استفاده کنیم.

- آخر هر دستور case با کلمه کلیدی break تشخیص داده می شود که باعث می شود برنامه از دستور switch خارج شده و دستورات بعد از آن اجرا شوند. اگر این کلمه کلیدی از قلم بیافتد برنامه با خطا مواجه می شود.

برنامه ۶: در برنامه زیر دستور switch را برای برنامه ای استفاده کرده ایم که با انتخاب نوع پزشک، تعرفه پزشک با در نظر گرفتن وضعیت بیمه بیمار، نمایش داده می شود.

کد HTML (برنامه ۶)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<h4> تعیین تعرفه پزشک </h4>
<form action="program6.php" method="post">
  <fieldset>
    <br> نوع پزشک را انتخاب کنید
    <select name="visit_type">
      <option value="ssd"> پزشک فوق متخصص </option>
      <option value="sd"> پزشک متخصص </option>
      <option value="gp"> پزشک عمومی </option>
    </select> <br>
    <input type="submit" value=" تعیین تعرفه ویزیت " >
  </fieldset>
</form>
</body>
</html>
```

```
1 ▼ <html dir="rtl">
2 ▼ <head>
3     <meta charset="utf-8">
4 </head>
5 ▼ <body>
6 <h4> تعیین تعرفه ویزیت </h4>
7 ▼ <form action="program6.php" method="post">
8 ▼     <fieldset>
9         <br> نوع پزشک را انتخاب کنید
10 ▼     <select name="visit_type">
11         <option value="ssd"> پزشک فوق تخصص </option>
12         <option value="sd"> پزشک متخصص </option>
13         <option value="gp"> پزشک عمومی </option>
14     </select> <br>
15     <input type="submit" value=" تعیین تعرفه ویزیت " >
16     </fieldset>
17 </form>
18 </body>
19 </html>
```

کد PHP (برنامه ۶)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<<h4> تعیین تعرفه پزشک </h4>
<?php
```

```

switch ($_POST["visit_type"]) {
case "ssd":
    echo " تعرفه پزشک فوق تخصص برابر است با: ۴۷۹۰۰ تومان ";
    break;
case "sd":
    echo " تعرفه پزشک متخصص برابر است با: ۳۸۴۰۰ تومان ";
    break;
case "gp":
    echo " تعرفه پزشک عمومی برابر است با: ۲۴۵۰۰ تومان ";
    break;
default:
    echo " شما گزینه ای را انتخاب نکردید ";
}
?>

```


بازگشت

</body>

</html>

```

1 <html dir="rtl">
2 <head>
3     <meta charset="utf-8">
4 </head>
5 <body>
6 <h4>تعیین تعرفه پزشک</h4>
7 <?php
8     switch ($_POST["visit_type"]) {
9         case "ssd":
10            echo "تعرفه پزشک فوق تخصص برابر است با: 47900 تومان ";
11            break;
12        case "sd":
13            echo "تعرفه پزشک متخصص برابر است با: 38400 تومان ";
14            break;
15        case "gp":
16            echo "تعرفه پزشک عمومی برابر است با: 24500 تومان ";
17            break;
18        default:
19            echo "شما گزینه ای را انتخاب نکردید ";
20    }
21 ?>
22
23 <br><a href="index6.html">بازگشت</a>
24 </body>
25 </html>

```

خروجی :

رسید نوبت:

نوع ویزیت:

پزشک متخصص

وضعیت بیمه:

دارد ○

ندارد ●

صدور رسید نوبت



رسید نوبت:

مبلغ ویزیت: 38400

مبلغ بیمه: 0

مبلغ قابل پرداخت: 38400

بازگشت

دستورات تکرار:

ساختارهای تکرار به شما اجازه می دهند که یک یا چند دستور کد را تا زمانی که یک شرط برقرار است تکرار کنید، بدون ساختارهای تکرار شما مجبورید همان تعداد کدها را بنویسید که بسیار خسته کننده است. ساختارهای تکرار در PHP عبارتند از :

- while
- do while
- for
- foreach

در این جزوه به توضیح ساختار و نحوه کاربرد **while** و **for** می پردازیم.

حلقه **while**:

ابتدایی ترین ساختار تکرار در PHP حلقه **while** است. ابتدا یک شرط را مورد بررسی قرار می دهد و تا زمانی که شرط برقرار باشد کدهای درون بلوک اجرا می شوند. ساختار حلقه **while** به صورت زیر است:

```
<?php
while (condition)
{
Code to loop;
}
?>
```

می بینید که ساختار **while** مانند ساختار **if** بسیار ساده است. ابتدا یک شرط را که نتیجه آن یک مقدار بولی است می نویسیم، اگر نتیجه درست (**true**) باشد سپس کدهای داخل بلوک **while** اجرا می شوند. اگر شرط نادرست (**false**) باشد وقتی که برنامه به حلقه **while** برسد هیچکدام از کدها را اجرا نمی کند. برای متوقف شدن حلقه باید مقادیر داخل حلقه **while** اصلاح شوند.

به یک متغیر شمارنده در داخل بدنه حلقه نیاز داریم. این شمارنده برای آزمایش شرط مورد استفاده قرار می گیرد و ادامه یا توقف حلقه به نوعی به آن وابسته است. این شمارنده را در داخل بدنه باید کاهش یا افزایش دهیم. برای اینکه با کاربرد **while** آشنا شوید به مثال زیر توجه نمائید:

```
<html>
<body>

<?php
$i=1;
while($i<=5)
{
echo "The number is " . $i . "<br />";
$i++;
}
?>
</body>
</html>
```

در مثال فوق ، ما یک متغیر `a` با مقدار ۱ تعریف نموده ایم ، سپس در قسمت `while` شرط گذاشته ایم که تا زمانی ادامه پیدا کند که متغیر `a` (که برابر با یک می باشد) کوچکتر مساوی ۵ شود ، یعنی حلقه ما ۵ بار تکرار می شود ، سپس مقدار متغیر `a` چاپ شده و پس از چاپ یک مقدار به متغیر اضافه می شود مثلا بعد از اینکه ۱ چاپ شد ، متغیر با ++ یک مقدار اضافه می شود سپس مجددا شرط برقرار می شود.

همانطور که کد فوق را مشاهده نمودید ، ما بعد از چاپ هر مقدار متغیر ، یک مقدار به متغیر اضافه نموده ایم ، اگر این کار را انجام نمی دادیم ، شرط تا بی نهایت تکرار می شد.

برنامه ۷: در برنامه زیر با استفاده از حلقه تکرار `while` با توجه به عددی که وارد میکنیم به همان تعداد "نام و نام خانوادگی پزشک" چاپ و تکرار می شود.

کد HTML (برنامه ۷)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<h4> لیست پزشکان </h4>
<form action="program7.php" method="post">
  <fieldset>
    <br> تعداد پزشک را وارد کنید
    <input type="text" name="number_doctor">
    <input type="submit" value=" ایجاد لیست پزشکان " >
  </fieldset>
</form>
</body>
</html>
```

```
1 <html dir="rtl">
2 <head>
3   <meta charset="utf-8">
4 </head>
5 <body>
6 <h4> لیست پزشکان </h4>
7 <form action="program7.php" method="post">
8   <fieldset>
9     <br> تعداد پزشک را وارد کنید
10    <input type="text" name="number_doctor">
11    <input type="submit" value=" ایجاد لیست پزشکان " >
12  </fieldset>
13 </form>
14 </body>
15 </html>
```

کد PHP (برنامه ۷)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<h4>لیست پزشکان</h4>
<?php
  $x = 1;
  while($x<=$_POST["number_doctor"]) {
    echo $x."- نام و نام خانوادگی پزشک-";
    $x++;
  }
?>
<br><a href="index7.html">بازگشت</a>
</body>
</html>
```

```
1 ▾ <html dir="rtl">
2 ▾ <head>
3     <meta charset="utf-8">
4 </head>
5 ▾ <body>
6 <h4>لیست پزشکان</h4>
7 <?php
8     $x = 1;
9 ▾     while($x<=$_POST["number_doctor"]) {
10        echo $x."- نام و نام خانوادگی پزشک -";
11        $x++;
12    }
13    ?>
14 <br><a href="index7.html">بازگشت</a>
15 </body>
16 </html>
```

خروجی :

لیست پزشکان:

1- نام و نام خانوادگی پزشک:
2- نام و نام خانوادگی پزشک:

[بازگشت](#)

←

لیست پزشکان:

تعداد پزشک را وارد کنید:

ایجاد لیست پزشکان

حلقه for :

یکی دیگر از ساختارهای تکرار حلقه for است. این حلقه عملی شبیه به حلقه while انجام می دهد و فقط دارای چند خصوصیت اضافی است. ساختار حلقه for به صورت زیر است:

```
<?php
for (initialization; condition; operation) {
code to be executed;
}
?>
```

مقداردهی اولیه (initialization) اولین مقداری است که به شمارنده حلقه می دهیم. شمارنده فقط در داخل حلقه for قابل دسترسی است.

شرط (condition n) در اینجا مقدار شمارنده را با یک مقدار دیگر مقایسه می کند و تعیین می کند که حلقه ادامه یابد یا نه.

عملگر (operation) که مقدار اولیه متغیر را کاهش یا افزایش می دهد.

برای اینکه با کاربرد حلقه for آشنا شوید به مثال زیر توجه نمائید:

```
<?php
for ($i = 1; $i <= 10; $i++) {
    echo "Hello world" . <br>;
}
?>
```

در برنامه بالا ابتدا یک متغیر به عنوان شمارنده تعریف می کنیم و آن را با مقدار ۱ مقداردهی اولیه می کنیم. سپس با استفاده از شرط آن را با مقدار ۱۰ مقایسه می کنیم که آیا کمتر است یا مساوی؟

توجه کنید که قسمت سوم حلقه (i++) فوراً اجرا نمی شود. کد اجرا می شود و ابتدا رشته Hello world را چاپ می کند. آنگاه یک واحد به مقدار i اضافه شده و مقدار i برابر ۲ می شود و بار دیگر i با عدد ۱۰ مقایسه می شود و این حلقه تا زمانی که مقدار شرط true شود ادامه می یابد.

برنامه ۸: در برنامه زیر با استفاده از حلقه for با توجه به عددی که وارد می کنیم به همان تعداد "نام و نام خانوادگی پرستار" چاپ و تکرار می گردد.

کد HTML (برنامه ۸)

```
<html dir="rtl">
<head>
    <meta charset="utf-8">
</head>
<body>
<h4> لیست پرستاران </h4>
<form action="program8.php" method="post">
    <fieldset>
    <br>:تعداد پرستاران را وارد کنید
        <input type="text" name="number_nurse">
        <input type="submit" value="ایجاد لیست پرستاران" >
    </fieldset>
</form>
</body>
</html>
```

```

1 ▼ <html dir="rtl">
2 ▼ <head>
3     <meta charset="utf-8">
4 </head>
5 ▼ <body>
6 <h4>لیست پرستاران</h4>
7 ▼ <form action="program8.php" method="post">
8 ▼     <fieldset>
9         تعداد پرستاران را وارد کنید: <br>
10        <input type="text" name="number_nurse">
11        <input type="submit" value="ایجاد لیست پرستاران">
12    </fieldset>
13 </form>
14 </body>
15 </html>

```

کد PHP (برنامه ۸)

```

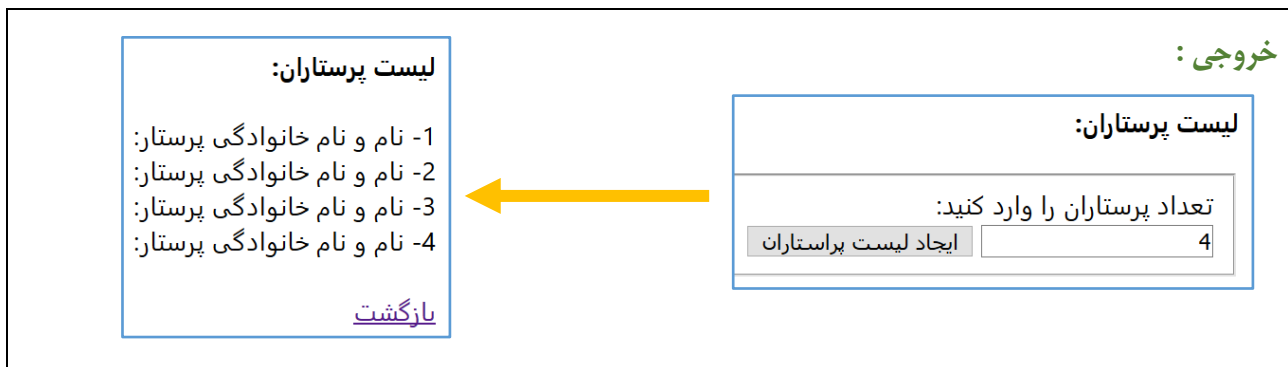
<html dir="rtl">
<head>
    <meta charset="utf-8">
</head>
<body>
<h4>لیست پرستاران</h4>
<?php
    for ($x = 1; $x <= $_POST["number_nurse"]; $x++) {
        echo $x." نام و نام خانوادگی پرستار - ";
        $x++;
    }
?>
<br><a href="index8.html">بازگشت</a>
</body>
</html>

```

```

1 ▼ <html dir="rtl">
2 ▼ <head>
3     <meta charset="utf-8">
4 </head>
5 ▼ <body>
6 <h4>لیست پرستاران</h4>
7 <?php
8 ▼     for ($x = 1; $x <= $_POST["number_nurse"]; $x++) {
9         echo $x." نام و نام خانوادگی پرستار - ";
10    }
11    ?>
12    <br><a href="index8.html">بازگشت</a>
13 </body>
14 </html>

```



برنامه نویسی شیء گرا:

مفهوم برنامه نویسی شیء گرای یا object-oriented programming

برنامه نویسی شیء گرا یا به اختصار OOP یک شیوه برنامه نویسی است که کدهای مشابه را در قالب یک کلاس جمع آوری و در قسمت های مختلف برنامه از آن استفاده میکند.

برای مثال فرض کنید یک وبسایت علمی طراحی کرده اید که برای قسمت های مختلف آن مانند صفحات اخبار، مقالات و ... امکان ارسال نظرات برای کاربر وجود دارد. خب در برنامه نویسی به شیوه معمولی شما میبایست برای هر گروه از صفحات سیستم نظردهی جداگانه ای بنویسید. اما در برنامه نویسی شیء گرای شما یک بار در قالب یک کلاس (class) سیستم نظردهی را می نویسید و برای صفحات مختلف یک نمونه شیء (object) از کلاس نظردهی ایجاد می کنید. به این ترتیب شما فقط یک بار سیستم نظردهی را برنامه نوشته اید و از آن در تمامی گروه های مطالب خود استفاده می کنید. در طبیعت نیز از مدل شیء گرای برای تمامی پدیده ها نیز استفاده شده است. مثلا انسان را در نظر بگیرید. یک کلاس کلی از انسان وجود دارد که دارای اجزای مشترک مانند قلب، چشم، گوش و ... می باشد. و نمونه های مختلف از این کلاس ساخته می شوند. مثلا یک نمونه انسان از کلاس انسان اصلی می سازیم که موارد کلی را از این کلاس به ارث می برد و موارد جز را نیز به آن اضافه می کنیم. مثلا دارای جنسیت مرد، نام علی، رنگ چشم مشکی و....

یکی از مهمترین عواملی که باعث شد برنامه نویسی شیء گرای پدید آید ابتدا پرهیز از تکرار کدنویسی و دوم توسعه آسان تر برنامه های نوشته شده بود. در برنامه های بزرگ اگر مشکل یا خطایی در برنامه ایجاد شود و یا در آینده قصد تغییر برنامه را داشته باشید، فقط کافیست کلاس کد مربوطه را تغییر دهید و مجبور به خواندن کلیه برنامه از ابتدا تا انتها نخواهید بود. این یعنی توسعه آسان تر.

کلاس:

کلاس به شما اجازه می دهد یک نوع داده ای که توسط کاربر تعریف می شود و شامل متغیرها و خواص (properties) و متدها است را ایجاد کنید. کلاس در حکم یک نقشه برای یک شیء می باشد. شیء یک چیز واقعی است که از ساختار، خواص و یا رفتارهای کلاس پیروی می کند. وقتی یک شیء می سازید یعنی اینکه یک نمونه از کلاس ساخته اید. نحوه ساخت یک کلاس در php بسیار ساده است. کلاس ها با کلمه کلیدی class و به دنبال آن نام کلاس و سپس مجموعه ای از کدها و توابع در بین دو علامت " {} " نیز نوشته می شوند.

```

class class name
{
Variable1;
Variable2;
...
Variable N;

method1;
method2;
...
method N;
}

```

در بدنه کلاس، متغیرها و متدهای آن قرار داده می شوند. متغیرها اعضای داده ای خصوصی هستند که کلاس از آنها برای رفتارها و ذخیره مقادیر ویژگی ها (property) استفاده می کند. متدها رفتارها یا کارهایی هستند که یک کلاس می تواند انجام دهد.

کلاس را با کلیدواژه ی class تعریف می کنیم. بعد از کلیدواژه ی class اسم کلاس را که حرف اول آن با capital letter آغاز می شود درج می کنیم.

اگر اسم کلاس از چندین کلمه تشکیل شده باشد، در آن صورت حرف اول تمامی کلمات تشکیل دهنده را با حروف بزرگ می نویسیم. از این سیستم نگارشی تحت عنوان camel case یاد می شود. به طور مثال می توان به JapaneseCars، AmericanIdol، EuropeTour اشاره کرد. پیاده سازی یا منطق کلاس را داخل { } قرار می دهیم که به آن در اصطلاح بدنه ی کلاس می گویند.

چگونگی تعریف property در کلاس:

به متغیرهایی که داخل بدنه ی کلاس تعریف می شوند، property گفته می شود. property ها می توانند مانند متغیرهای عادی مقادیری از نوع رشته (string)، عدد صحیح (integer) و بولی (Boolean) در خود نگهداری کنند. در زیر تعدادی property داخل کلاس Car تعریف می کنیم.

```

class Car {
public $comp;
public $color = "beige";
public $hasSunRoof = true;
}

```

- کلیدواژه ی public را قبل از اسم property می نویسیم.
- حرف اول اسم property را با lower case (حرف کوچک) آغاز می کنیم.
- اگر اسم property بیش از یک کلمه تشکیل شده باشد، در آن صورت تمامی کلمات به استثنای کلمه ی اول با حرف بزرگ آغاز می شوند. به عنوان مثال می توان به دو نمونه ی فوق، \$color و \$hasSunRoof اشاره کرد.
- یک Property می تواند مقدار پیش فرض داشته باشد که در همان خط تعریف و به آن اختصاص داده می شود. به عنوان مثال می توان به \$color = "beige" اشاره کرد.

- یک `property` می تواند مقدار پیش فرضی هم نداشته باشد. در مثال بالا می توان به متغیر `$comp` اشاره کرد.

نحوه ساخت `object` از روی یک کلاس:

می تواند از روی یک کلاس که به منزله ی یک نقشه عمل می کند، چندین `object` یا نمونه ایجاد کرد. هر شیء می تواند `property` های منحصر بفرد خود را داشته باشد. به منظور استفاده از یک کلاس می بایست ابتدا یک شیء یا نمونه ی جدید از آن ایجاد کرد. جهت ساخت شیء از روی کلاس مورد نظر، در برنامه نویسی شی گرا از کلیدواژه ی `new` استفاده می شود. مثال:

```
$bmw = new Car ();
```

در کد بالا، یک شیء به نام `$bmw` از کلاس `Car` با کلیدواژه ی `new` ایجاد می کنیم. از این پروسه تحت عنوان نمونه سازی از کلاس (ایجاد شیء از روی کلاس) یا `instantiation` یاد می شود. می توان چندین نمونه یا شیء از کلاس واحد ایجاد کرد. همان طور که در بالا نیز ذکر شد، کلاس یک نقشه برای ایجاد شیء (در صورت نیاز با پیاده سازی اختصاصی) می باشد.

```
$bmw = new Car;  
$mercedes = new Car;
```

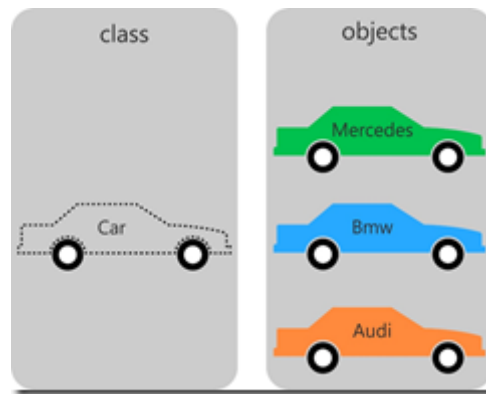
به عبارت ساده تر می توان از یک کلاس واحد هر تعداد شیء که لازم بود ایجاد کرد و سپس به هر شیء `property` های دلخواه و اختصاصی را اضافه نمود.

شیء چه کاربردی دارد؟

در برنامه نویسی مبتنی بر تابع، توابع و متغیرها در یک دامنه ی سراسری (`global scope`) قرار داشته و تمامی کدهای برنامه قابلیت دسترسی به این متغیرها و توابع را دارند. کافی است اسم تابع مورد نظر را جهت اجرای آن فراخوانی کرد. اما در برنامه نویسی شی گرا، می توان با استفاده از کلاس کد خود را سازمان دهی کرده و در واقع با تعریف توابع و متغیرها در سطح کلاس، آن ها را از دسترس بقیه برنامه خارج ساخت.

زمانی که توسعه دهنده کد را داخل بدنه ی کلاس تعریف می کند، این کد در واقع کپسوله سازی شده (`class scope`) و از دسترس کل برنامه یا به اصطلاح `global scope` خارج می شود. حال به منظور دسترسی به متغیرها یا توابعی که در حوزه ی اختصاصی کلاس قرار دارند، باید یک مکانیزم در نظر بگیریم که از طریق آن، کد خارج از کلاس بتواند در صورت نیاز به محتوای کپسوله سازی شده در کلاس دسترسی داشته باشد. برای این منظور در برنامه نویسی شی گرا از کلاس مورد نظر شیء ساخته می شود.

دقت داشته باشید که بر واژه ی جمع `objects` تاکید داریم، نه واژه ی مفرد `object` چرا که می توانیم هر تعداد شیء که لازم است از کلاس واحد ایجاد کنیم. شیء های ساخته شده از روی کلاس تمامی متدها و `property` های کلاسی که از آن مشتق شده اند را دارا می باشند. با مشاهده ی تصویر زیر این مفهوم را بهتر درک می کنید:



همان طور که مشاهده می کنید، از کلاسی واحد به نام Car، سه شیء مستقل به ترتیب به نام های Mercedes، Bmw و Audi ایجاد کردیم.

تمامی شیء ها از روی کلاسی واحد به نام Car ساخته شده اند و به همین جهت متدها و property های کلاس مزبور را به ارث برده اند (دارا می باشند)، با این وجود هر سه شیء از یکدیگر متفاوت و مستقل هستند. در واقع نه تنها دارای اسم متفاوت اند، بلکه به property های هر یک مقدار متفاوتی تخصیص داده شده است. در نمونه ی فوق، سه شیء از نظر رنگ با یکدیگر متفاوت هستند.

بنابراین یک کلاس، ظرف متغیرها و توابعی است که بین تمامی شیء های ساخته شده از روی آن مشترک است. اگرچه هر سه شیء حامل کدی یکسان می باشند، با این حال می توانند رفتار و قابلیت هایی متفاوتی داشته باشند چرا که مقادیر مختلفی به هر یک تخصیص یافته است.

چگونگی دسترسی به مقدار property های یک شیء

پس از ایجاد شیء از روی کلاس، می توان به property های آن دسترسی داشت. برای مثال:

```
echo $bmw -> color;
echo $mercedes -> color;
```

به منظور دسترسی به مقدار یک property از کلاس، ابتدا اسم شیء، عملگر -> و سپس اسم property را ذکر می کنیم. همان طور که مشاهده می کنید، اسم property با علامت \$ آغاز نمی شود، بلکه این علامت فقط در ابتدای اسم شیء درج می گردد.

**** از کاراکتر -> به معنی اشاره گر به منظور دسترسی به اجزای یک کلاس استفاده می شود.****

نحوه مقداردهی و تنظیم property های شیء:

به منظور مقداردهی property های یک شیء، از روشی یکسان استفاده می کنیم.

برای مثال، به منظور تنظیم مقدار property مزبور (color) بر روی true در شیء bmw، از کد زیر استفاده می کنیم:

```
$bmw -> color = "blue";
```

جهت مقداردهی متغیر \$comp در هر دو شیء، بدین صورت عمل می کنیم:

```
$bmw -> comp = "BMW";
```

```
$mercedes -> comp = "Mercedes Benz";
```

پس از تنظیم مقدار `property`، می توان به مقدار آن دسترسی داشت.

به منظور دسترسی به مقدار متغیر `color` از شیء `$bmw`، از دستور زیر استفاده می کنیم:

```
echo $bmw -> color;
Result:blue
```

همچنین می توان به اسم شرکت تولید کننده ی ماشین (ذخیره شده در متغیر `comp`) و رنگ شیء دوم (مقدار `color` در شیء `$mercedes`) دسترسی پیدا کرد.

```
echo $mercedes -> color;
echo $mercedes -> comp;
```

نحوه تعریف متد در کلاس:

کلاس ها معمولا علاوه بر متغیر، توابعی را دربر می گیرند که عملیاتی را انجام داده و رفتارهای آن کلاس محسوب می شوند. زمانی که تابع در محدوده ی یک کلاس تعریف می شود، در اصطلاح برنامه نویسی شی گرا به آن متد می گویند. در کد زیر می بینید که تابع `hello()` را با سطح دسترسی (کلیدواژه `public`) تعریف کردیم.

```
class Car {
    public $comp;
    public $color = 'beige';
    public $hasSunRoof = true;
    public function hello()
    {
        return "beep";
    }
}
```

در قطعه کد بالا می بینید که کلیدواژه ی `public` را قبل از متد قرار دادیم. در انتخاب اسم برای یک تابع، معمولا حرف اول اسم تابع را با حرف کوچک آغاز می کنیم.

چنانچه اسم تابع از چندین کلمه تشکیل شده باشد، در آن صورت تمامی کلمات به استثنای اولین کلمه با حرف بزرگ آغاز می شوند. به عنوان مثال می توان به `helloUser()` یا `flyPanAm()` اشاره کرد.

شیوه ی دسترسی به متد نیز مانند دسترسی به `property` های کلاس می باشد. اما قبل از هر چیز لازم است یک شیء از روی کلاس ایجاد کنیم.

```
$bmw = new Car ();
$mercedes = new Car ();
echo $bmw -> hello();
echo $mercedes -> hello();
Result:beep
beep
```

آموزش استفاده از کلیدواژه `$this` :

در مبحث قبلی با مفهوم سازمان دهی کدهای مرتبط که با همکاری وظیفه ی خاصی را به انجام می رسانند، داخل کلاس آشنا شدیم. همچنین یک کلاس به نام `Car` ایجاد کردیم که کدهای مربوط به ماشین از جمله ویژگی های ظاهری آن و رفتاری که از خود نشان می دهد را دربر می گرفت.

همچنین دو شیء از روی کلاس ایجاد کرده و به توابع و `property` های موجود در آن با دستور مناسب دسترسی پیدا کردیم.

```
$bmw = new Car;
$mercedes = new Car;
```

کلیدواژه ی `$this` در برنامه نویسی شی گرا با `php` :

- کلیدواژه ی `$this` به توسعه دهنده این امکان را می دهد تا به متدها و `property` های کلاس جاری از داخل خود کلاس (`class scope`) دسترسی داشته باشد.
- کلیدواژه ی `$this` برای این منظور تعبیه شده که برنامه نویس بتواند از `property` ها و متدهای کلاس جاری استفاده کند. نحوه ی دسترسی به توابع و `property` های داخل خود کلاس به شرح زیر می باشد:

نکته

تنها نکته ی قابل توجه این است که کلیدواژه ی مزبور با علامت `$` آغاز شده، در حالی که اسم `property` ها و متدهایی که با این کلیدواژه مورد دسترسی قرار می گیرند، دیگر این علامت را ندارند.

```
$this -> propertyName;
$this -> methodName();
```

کلمه ی کلیدی `$this` در واقع به متدها و `property` های کلاس فعلی اشاره دارد و این قابلیت را در اختیار برنامه نویس قرار می دهد که داخل کلاس مورد نظر، به اعضای آن دسترسی داشته باشد.

حال از کلیدواژه ی نام برده برای دسترسی به اعضای کلاس `Car` استفاده می کنیم. با استفاده از این امکان می توان به `property` های کلاس جاری داخل تابع `hello()` از همین کلاس دسترسی داشت.

به منظور دسترسی به متغیر `$comp` از کلاس جاری، می توان از دستور زیر استفاده کرد:

```
$this -> comp
```

به منظور دسترسی به متغیر `$color` از کلاس مورد نظر از دستور زیر استفاده می کنیم.

```
$this -> color
```

حال دو شیء از روی کلاس جاری می سازیم:

```
$bmw = new Car();
$mercedes = new Car ();
```

با دستور زیر مقادیر `property` های دو شیء را تنظیم می کنیم:

```
$bmw -> comp = "BMW";
$bmw -> color = "blue";
```

```
$mercedes -> comp = "Mercedes Benz";
```

```
$mercedes -> color = "green";
```

اکنون متد `hello()` را برای اولین شیء `car` فراخوانی می کنیم:

```
echo $bmw -> hello();
```

حال آن را برای شیء دوم فراخوانی می کنیم:

```
echo $mercedes -> hello();
```

➤ خوب حالا که با نحوه تعریف و کاربرد کلاس ها آشنا شدیم در ادامه برنامه هایی با استفاده از کلاس ها می نویسیم که و کاربرد هریک از موارد فوق را فرا می گیریم.

برنامه ۹: در برنامه زیر با استفاده از کلاس ها برنامه ای نوشتیم که با وارد کردن وزن و قد، وزن و قد را نمایش داده می شود.

کد HTML (برنامه ۹)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<h4>محاسبه BMI:</h4>
<form action="program9.php" method="post">
  <fieldset>
وزن: <br>
    <input type="text" name="weight"><br>
قد: <br>
    <input type="text" name="height"><br>
    <input type="submit" value="محاسبه BMI">
  </fieldset>
</form>
</body>
</html>
```

```
1 <html dir="rtl">
2 <head>
3   <meta charset="utf-8">
4 </head>
5 <body>
6 <h4>محاسبه BMI:</h4>
7 <form action="program9.php" method="post">
8   <fieldset>
9     وزن: <br>
10    <input type="text" name="weight"><br>
11    قد: <br>
12    <input type="text" name="height"><br>
13    <input type="submit" value="محاسبه BMI">
14  </fieldset>
15 </form>
16 </body>
17 </html> |
```

کد PHP (برنامه ۹)

```

<html dir="rtl">
<head>
  <meta charset="utf-8">
  <title>Define class & object</title>
</head>
<body>
<h4>محاسبه BMI:</h4>
<?php
  class bmi{
    public $weight=76;
    public $height=174;
  }
  $bmi1=new bmi;
  echo 'وزن برابر است با: ' . $bmi1->weight . '<br>';
  echo 'قد برابر است با: ' . $bmi1->height . '<br>';
?>
</body>
</html>

```

```

1 ▾ <html dir="rtl">
2 ▾ <head>
3   <meta charset="utf-8">
4   <title>Define class & object</title>
5 </head>
6 ▾ <body>
7 <h4>محاسبه BMI:</h4>
8 <?php
9 ▾   class bmi{
10      public $weight=76;
11      public $height=174;
12   }
13
14   $bmi1=new bmi;
15
16   echo 'وزن برابر است با: ' . $bmi1->weight . '<br>';
17   echo 'قد برابر است با: ' . $bmi1->height . '<br>';
18   ?>
19 </body>
20 </html>

```

خروجی:

محاسبه BMI:

وزن:
76

قد:
174

محاسبه BMI



محاسبه BMI:

وزن برابر است با: 76
قد برابر است با: 174

برنامه ۱۰: در این برنامه با استفاده از ساختار کلاس ها با وارد کردن قد و وزن شاخص توده بدنی (BMI) محاسبه می شود.

کد PHP (برنامه ۱۰)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
  <title>Initialize to objects</title>
</head>
<body>
<h4>محاسبه BMI:</h4>
<?php
  class bmi{
    public $weight=87;
    public $height=1.35;
    public $bmi_calc;
    public function bmi_calc (){
      return 87/(1.35*1.35);
    }
  }

  $bmi1=new bmi;
  echo 'وزن برابر است با' . $bmi1->weight . '<br>';
  echo 'قد برابر است با' . $bmi1->height . '<br>';
  echo 'شاخص توده بدنی برابر است با' . $bmi1->bmi_calc() . '<br>';
  ?>
</body>
</html>
```

```
1 <html dir="rtl">
2 <head>
3   <meta charset="utf-8">
4   <title>Initialize to objects</title>
5
6 </head>
7 <body>
8 <h4>محاسبه BMI:</h4>
9 <?php
10 class bmi{
11   public $weight=87;
12   public $height=1.35;
13   public $bmi_calc;
14   public function bmi_calc (){
15     return 87/(1.35*1.35);
16   }
17 }
18
19 $bmi1=new bmi;
20
21 echo 'وزن برابر است با' . $bmi1->weight . '<br>';
22 echo 'قد برابر است با' . $bmi1->height . '<br>';
23 echo 'شاخص توده بدنی برابر است با' . $bmi1->bmi_calc() . '<br>';
24 ?>
25 </body>
26 </html>
```

خروجی :

محاسبه BMI:

وزن برابر است با: 87
 قد برابر است با: 1.35
 شاخص توده بدنی برابر است با: 47.736625514403

برنامه ۱۱ : برنامه زیر دقیقا همانند برنامه ۱۰ است با این تفاوت که این در مثال از کلیدواژه **this** استفاده کردیم تا با نحوه کاربرد آن آشنا شوید.

کد PHP (برنامه ۱۱)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
  <title>This</title>
</head>
<body>
<h4>محاسبه BMI:</h4>
<?php
class bmi{
  public $weight=87;
  public $height=1.35;
  public $bmi_calc;
  public function bmi_calc (){
    return $this->weight/($this->height*$this->height);
  }
}
$bmi1=new bmi;
echo '$bmi1->weight . ' . $bmi1->weight . '<br>';
echo '$bmi1->height . ' . $bmi1->height . '<br>';
echo '$bmi1->bmi_calc() . ' . $bmi1->bmi_calc() . '<br>';
```

<pre>?> </body> </html></pre>	<pre> 1 <html dir="rtl"> 2 <head> 3 <meta charset="utf-8"> 4 <title>This</title> 5 6 </head> 7 <body> 8 <h4>محاسبه BMI:</h4> 9 <?php 10 class bmi{ 11 public \$weight=87; 12 public \$height=1.35; 13 public \$bmi_calc; 14 public function bmi_calc (){ 15 return \$this->weight/(\$this->height*\$this->height); 16 } 17 } 18 19 \$bmi1=new bmi; 20 21 echo 'وزن برابر است با: ', \$bmi1->weight . '
'; 22 echo 'قد برابر است با: ', \$bmi1->height . '
'; 23 echo 'شاخص توده بدنی برابر است با: ', \$bmi1->bmi_calc() . '
'; 24 ?> 25 </body> 26 </html></pre>	
	<p style="text-align: center;">محاسبه BMI:</p> <p style="text-align: center;">وزن برابر است با: 87 قد برابر است با: 1.35 شاخص توده بدنی برابر است با: 47.736625514403</p>	<p style="text-align: center;">خروجی :</p>

آموزش کار با پایگاه داده MySQL در PHP

اگر شما هم تجربه کمی کار در دنیای برنامه نویسی وب و کار با PHP را داشته باشید، حتما واژه‌ی MySQL به گوشتان خورده است و این سوال برایتان پیش آمده است که MySQL چیست؟ در اینجا به این سوال پاسخ می‌دهیم!!
برای پاسخ به سوال MySQL چیست؟ خوب است که در ابتدا، بصورت خیلی کلی با مفهوم دیتابیس یا پایگاه داده آشنا شویم.

بازی اسم، فامیل و دفترچه‌ی تلفن!

اگر دوران دبستان‌تان را یادتان باشد، حتما به خاطر دارید که قبل از آمدن معلم به سر کلاس، شخصی روی تخته نام اسامی «خوب‌ها» و «بدها» را می‌نوشت. یا در مثال دیگری، هنگامی که به بازی «اسم، فامیل» می‌پرداختیم، جمعی از اسم‌ها، فامیلی‌ها، شهرها، کشورها و... را روی کاغذ مینوشتیم.

در مثال اول اسم افراد و در مثال دوم اسم شهرها، کشورها، میوه‌ها و... همگی یک نوع از اطلاعات هستند که هرکدام در جایی ذخیره شده‌اند، یکی روی تخته سیاه و دیگری روی کاغذ.

کار (Database) یا پایگاه داده این است که مجموعه‌ای از اطلاعات را برای ما در جدول‌ها و ستون‌های مختلف ذخیره کند. همانطور که از اسمش نیز پیداست، دیتابیس یک پایگاه و محلی برای داده‌ها و اطلاعات ماست!

اگر بخواهیم یک مثال خیلی ساده‌ی دیگری بزنیم، دفترچه‌ای تلفنی که در اکثر خانه‌های ما وجود دارد که حاوی شماره تلفن دوستان و اقوام است، یک دیتابیس و پایگاه داده‌ای از نام‌های افراد و شماره‌های آن‌هاست.

در دنیای کامپیوتر و اینترنت هم ما در بسیاری از جاها با اطلاعات زیادی سر و کار داریم. برای مثال:

- ممکن است سایت‌مان کاربران زیادی داشته باشد و یک لیست از کاربران سایت‌مان بخواهیم داشته باشیم.
- یا ممکن است یک لیستی از مطالب، نظرات، اخبار و... را برای سایت‌مان نیاز داشته باشیم.

برای ذخیره سازی این اطلاعات در دنیای وب، به یک دیتابیس یا پایگاه داده نیاز داریم که این اطلاعات را برای ما ذخیره کند تا در آینده بتوانیم به آن‌ها دسترسی داشته باشیم. (همانطور که هر چند وقت یکبار، به سراغ دفترچه تلفن می‌رفتیم و از شماره‌های داخل آن استفاده می‌کردیم.)

پایگاه داده یا دیتابیس (Database) چیست؟

بعد از این مقدمه، حالا میتوانیم خیلی ساده بگوییم که:

دیتابیس یک جایی است که میتوانیم اطلاعات مختلف‌مان را در آن ذخیره سازی کنیم تا در آینده از آن استفاده کنیم.

در همین راستا میخواهیم به سوال MySQL چیست؟ نیز پاسخ دهیم:

MySQL یک سیستم مدیریت دیتابیس است که با استفاده از آن، می‌توانیم یک دیتابیس بسازیم و آن را مدیریت کنیم. که در انگلیسی به آن **Database Management System** یا به اختصار **DBMS** نیز می‌گویند.

در واقع کار MySQL این است که استفاده از دیتابیس را برای ما راحت کند و آن را مدیریت کند. در ادامه می‌گوییم که چطور MySQL میتواند داده‌های ما را مدیریت، ویرایش یا حذف کند.

MySQL چه کاربردی دارد؟

ممکن است این سوال برایتان پیش بیاید که MySQL چه کاربردی دارد. سوال بسیار خوبی است. در دنیای کامپیوتر، نرم افزارها، اپلیکیشن ها، سایت ها و... یک چیز بسیار مهم است: داده ها!

منظورمان از داده ها چه چیزهایی است؟ تمام آن چیزی که در نرم افزارها، اپلیکیشن ها، سایت ها و... مشاهده می کنیم. مثل:

- محصولات سایت دیجی کالا
- مقالات آموزشی سایت ی
- لیست رانندگان اطراف شما در اسنپ یا تپسی
- عکس ها، فیلم ها، استوری ها، کپشن ها، کامنت ها و... در اینستاگرام
- و ...

در پاسخ به سوال MySQL چیست؟ می توانیم بگوییم که MySQL یک نرم افزار و سیستمی است که می تواند همه ی این اطلاعات را برای ما ذخیره و مدیریت کند!

ویژگی های MySQL چیست؟

در ادامه ی پاسخ به سوال MySQL چیست؟ می خواهیم درباره ی ویژگی های MySQL صحبت کنیم. MySQL به عنوان یک نرم افزار و سیستم مدیریت دیتابیس، دارای ویژگی های زیادی است که در ادامه برخی از آن ها را ذکر کرده ایم:

- کارایی بالا
- هزینه ی پایین در راه اندازی و نگه داری
- سهولت در تنظیم و یادگیری
- پشتیبانی بسیار خوب
- کاربرد بالا در وب
- رابطه ی بسیار خوب با زبان برنامه نویسی PHP

Query یا کوئری چیست؟

همانطور که گفتیم، MySQL یک سیستم برای مدیریت اطلاعات و داده های ماست. و ما می توانیم انواع داده ها را در آن ذخیره کنیم. اما زمانی را تصور کنید که تعداد این داده ها زیاد می شود و شما نیاز به انجام یک کار خاص روی این داده ها دارید. مثلا می خواهید:

- یک داده خاص را پاک کنید
- بخشی از یک داده را ویرایش کنید
- یک داده ی جدید را اضافه کنید
- و یا به دنبال یک داده ی جدید بگردید.
- مثلا در دفترچه تلفن، می خواهید شماره یک نفر را پاک کنید، شماره یا نام او را ویرایش کنید، یک شماره ی جدید به دفترچه ی تلفن تان اضافه کنید و یا یک شماره ی خاص را جستجو کنید.

اینکارها را با استفاده از کوئری ها (Query) انجام می دهید. در واقع کوئری یک دستوری است که شما به پایگاه داده خود می دهید و از آن می خواهید که یکی از کارهای بالا را برای شما انجام دهد. در واقع کوئری ها، درخواست های شما از پایگاه داده هستند.

با استفاده از زبان PHP ، می توانید به یک پایگاه داده متصل شده و اطلاعات را ذخیره و ویرایش کنید. پایگاه داده MySQL محبوب ترین پایگاه داده ای است که در سیستم برنامه نویسی PHP استفاده می شود.

پایگاه داده MySQL چیست؟

- پایگاه داده MySQL یک سیستم پایگاه داده است که بر روی وب استفاده می شود.
- MySQL یک سیستم پایگاه داده می باشد که بر روی سرور اجرا می شود.
- MySQL یک database ایده آل برای پروژه های کوچک و بزرگ است.
- پایگاه MySQL بسیار سریع، قابل انعطاف و ساده برای کار کردن است.
- پایگاه داده MySQL از زبان استاندارد SQL برای کدنویسی استفاده می کند.
- پایگاه داده SQL را می توان بر روی طیف وسیعی از پلتفرم ها اجرا نمود.
- پایگاه داده SQL یک دیتابیس Open Source بوده و دانلود و استفاده از آن رایگان می باشد.
- اطلاعات پایگاه داده MySQL درون جدول (Table) ذخیره می شود. یک جدول یا Table مجموعه ای از داده های مرتبط به هم می باشد که از تعدادی سطر (row) و ستون (column) تشکیل شده است.
- معمولاً پایگاه داده ها (Database) برای نگهداری اطلاعات دسته بندی شده مناسب هستند. برای مثال یک شرکت ممکن است شامل یک پایگاه داده با جدول های مثل لیست زیر باشد:
 - Employees
 - Products
 - Customers
 - Order

سیستم کارکرد پایگاه داده MySQL با PHP

- کارکرد سیستم PHP و پایگاه داده MySQL به صورت cross-platform است، به این معنی که PHP را می توان بر روی یک سرور مثل ویندوز اجرا کرد، در حالی که پایگاه داده MySQL بر روی یک سرور با سیستم عامل دیگری مثل Unix قرار داشته باشد. به عبارت دیگر می توان سرور PHP و MySQL را با سیستم عامل های متفاوتی اجرا کرده و مشکلی رخ نمی دهد.

ایجاد جدول و دیتابیس در phpMyAdmin

در مرحله اول می بایست نرم افزار xampp را اجرا کنیم.

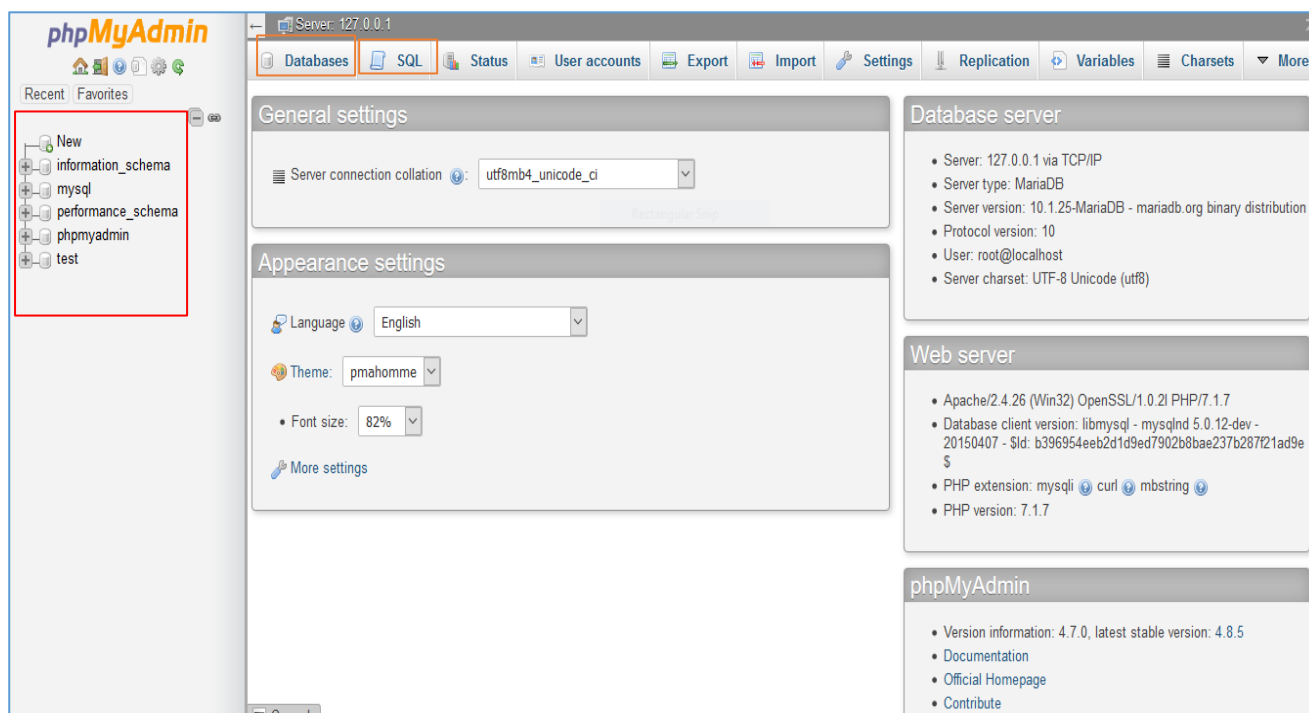
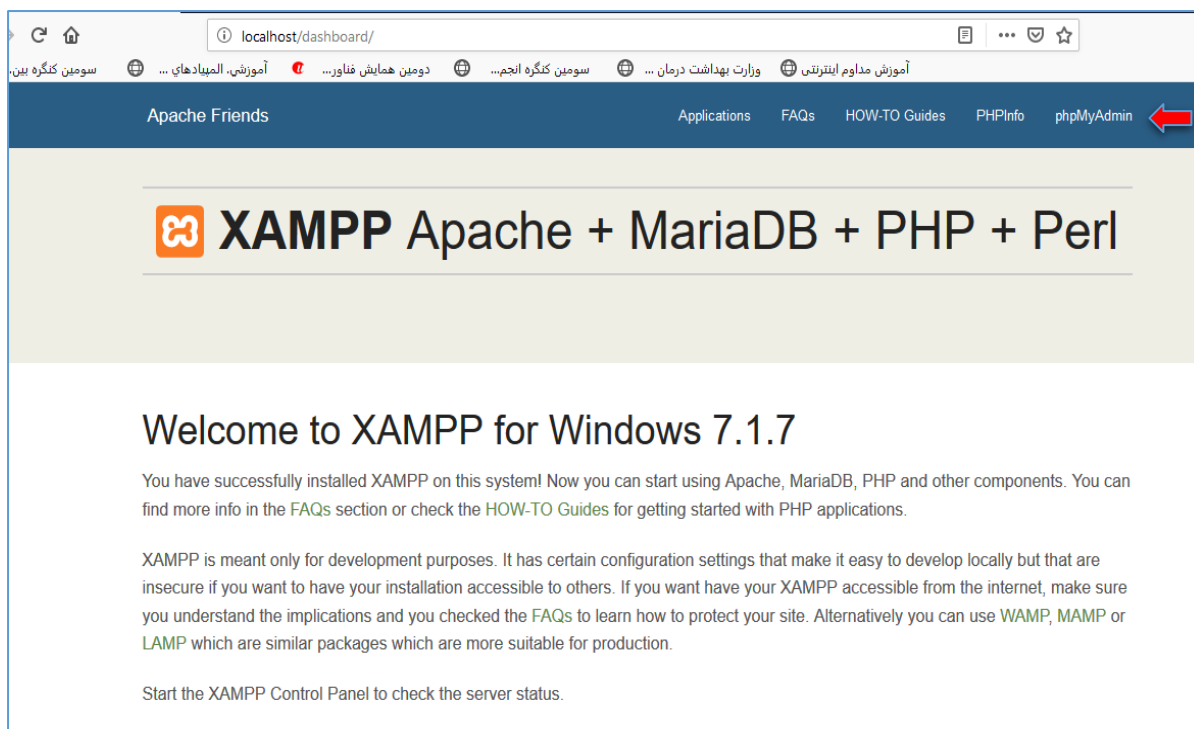
بعد از اجرای برنامه xampp لازم است که سرویس های Apache و MySQL را start نماییم .

خب حالا برای رفتن به MySQL در مرورگر لینک زیر را تایپ می کنیم. تا phpMyAdmin برای شما باز گردد

<http://127.0.0.1/phpmyadmin> **phpMyAdmin رابط کاربری ساده و کاربردی برای استفاده و کار با Mysql می باشد.**

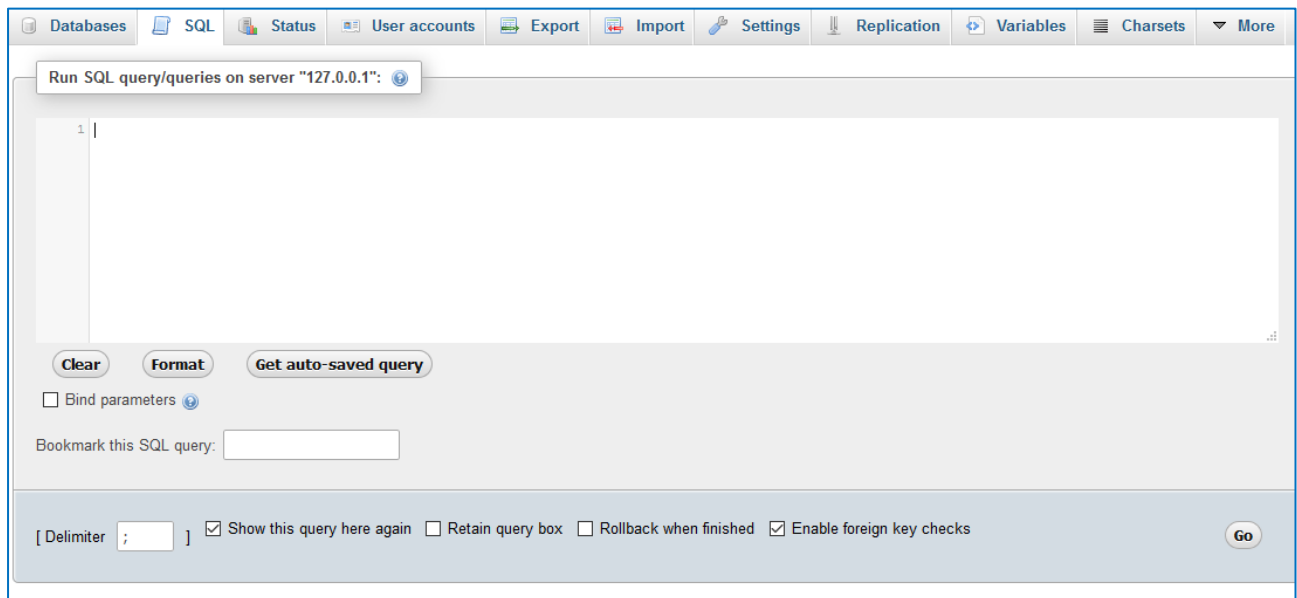
<http://127.0.0.1/phpmyadmin>

روش دیگری هم وجود دارد. میتوانید در مرورگر آدرس localhost// را تایپ کنید و بعد از باز شدن صفحه زیر گزینه phpMyAdmin را کلیک نمایید.

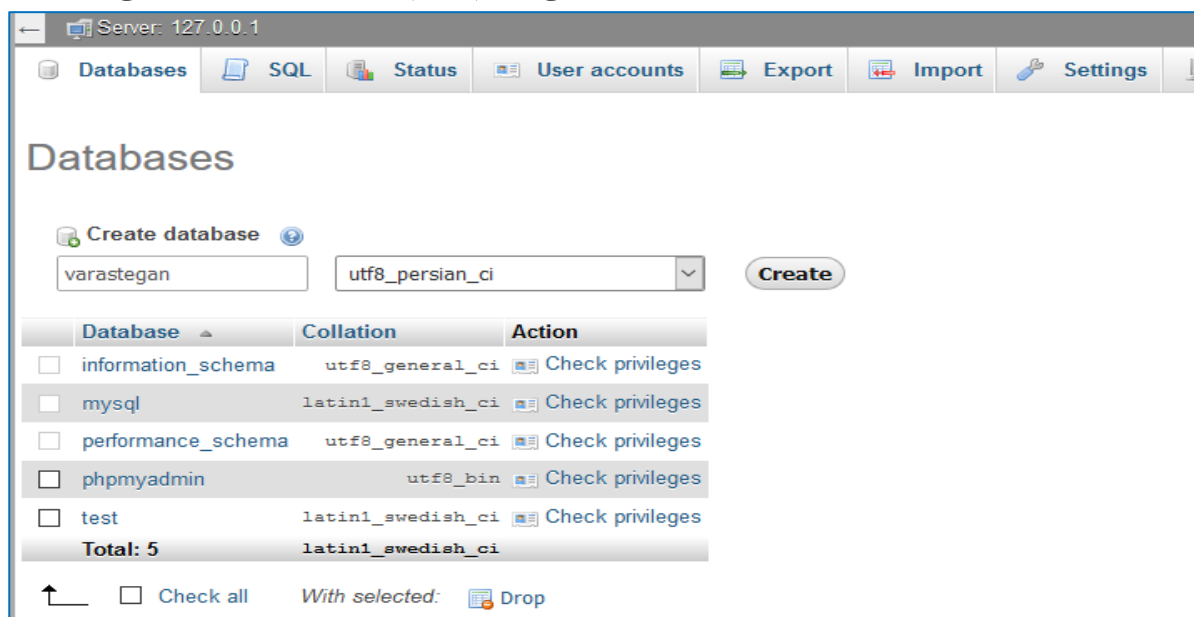


تمام دیتابیس های موجود در سمت چپ صفحه (همانطور که در شکل مشخص شده اند) قرار دارند که می توان به اطلاعات موجود در هریک به راحتی دسترسی داشت. این کار از قسمت Databases هم ممکن است.

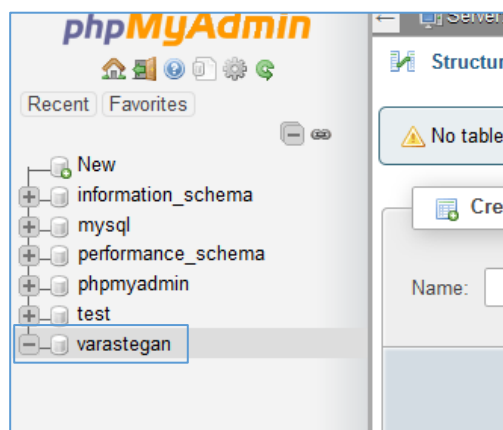
همچنین کدهایی را که می توان به صورت کدهای php در صفحات اجرا کرد را می توان مستقیماً در قسمت SQL وارد کرد و خروجی را مشاهده کرد. صفحه ای که با انتخاب قسمت SQL نمایش داده می شود به شکل زیر است:



برای افزودن یک دیتابیس جدید نیز به قسمت Databases می توانیم برویم. این صفحه به شکل زیر می باشد:



در فیلد متنی خالی باید نام دیتابیس مورد نظر را نوشت و در قسمت بعد باید نوع collation را انتخاب کرد. برای زبان فارسی باید مقدار utf8_persian_ci را انتخاب کرد تا بتوان مقادیر فارسی را درون جدول های دیتابیس قرار داد. و سپس Create را کلیک نماید. همان طور که مشاهده می کنید دیتابسی با نام varastegan در قسمت دیتابیس ها افزوده می شود.



با انتخاب دیتابیس به صفحه ای به شکل زیر وارد می شوید:

No tables found in database.

Create table

Name: HIT Number of columns: 3

Go

اگر جدولی موجود باشد در این قسمت نمایش داده می شود. فرم بالا برای افزودن جدول می باشد. تنها لازم است که نام جدول را نوشته و تعداد فیلد (ستون) های موردنظر را وارد کنیم که در اینجا ما جدولی به نام HIT با سه ستون ایجاد کنیم و بعد از زدن دکمه go در صفحه باز شده مقادیر را به شکل زیر وارد می کنیم:

جدول ما دارای سه ستون می باشد به صورت زیر:

id: این ستون در اصل شناسه هر ردیف را تعیین می کند و یک شماره است که نمی تواند تکراری باشد (شناسه دو ردیف نمی تواند یکی باشد). تعداد ارقام این ستون برابر با ۵ حرف می باشد.

name: نام هر فرد را مشخص می کند که نوع آن یک رشته می باشد به تعداد ۲۰ حرف.

age: این ستون سن افراد را مشخص می کند که یک مقدار عددی به تعداد ۳ رقم می باشد.

همچنین نوع این جدول را می توانیم برابر با `utf8_persian_ci` قرار دهیم.

پس می توانیم فیلدهای موردنظر را به شکل زیر وارد کنیم:

۱. در قسمت Name نام ستون های مورد نظر را مشخص می کند.

۲. در قسمت Type نوع مقادیری که قرار است در فیلد مورد نظر وارد شوند را مشخص می کند.

۳. در قسمت Length/Values. تعداد حروف و حداکثر طول مقدار وارد شده را مشخص می کند.

۴. در قسمت collation نوع جدول مشخص می شود که برای مقادیر فارسی برابر با `utf8_persian_ci` خواهد بود.

در آخر گزینه Save را انتخاب کنید تا مقادیر ذخیره شوند.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments
id	INT	5	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
name	VARCHAR	20	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
age	INT	3	None			<input type="checkbox"/>	---	<input type="checkbox"/>	

Table comments:

Collation: utf8_persian_ci Storage Engine: InnoDB

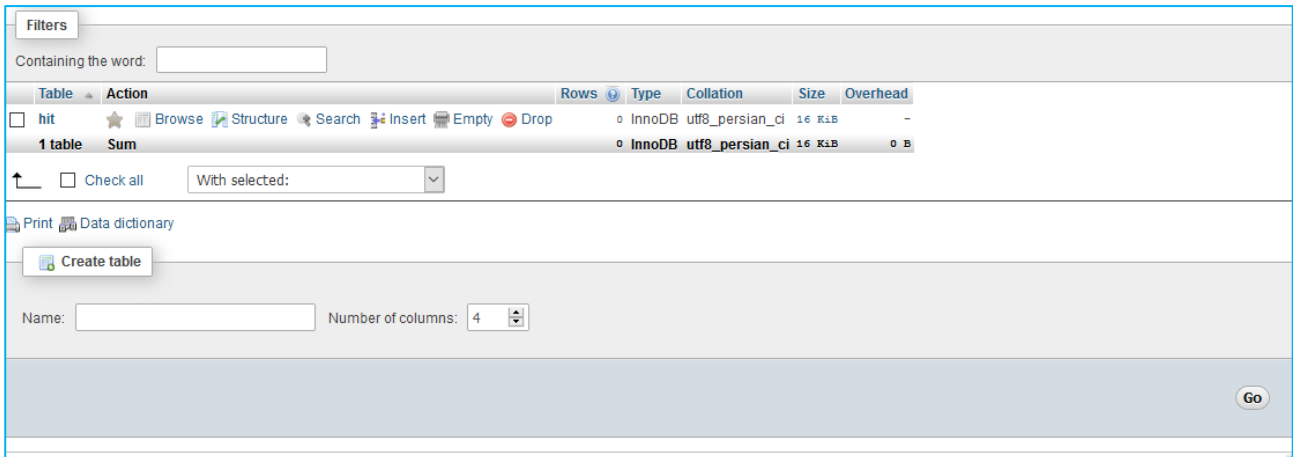
PARTITION definition:

Partition by: (Expression or column list)

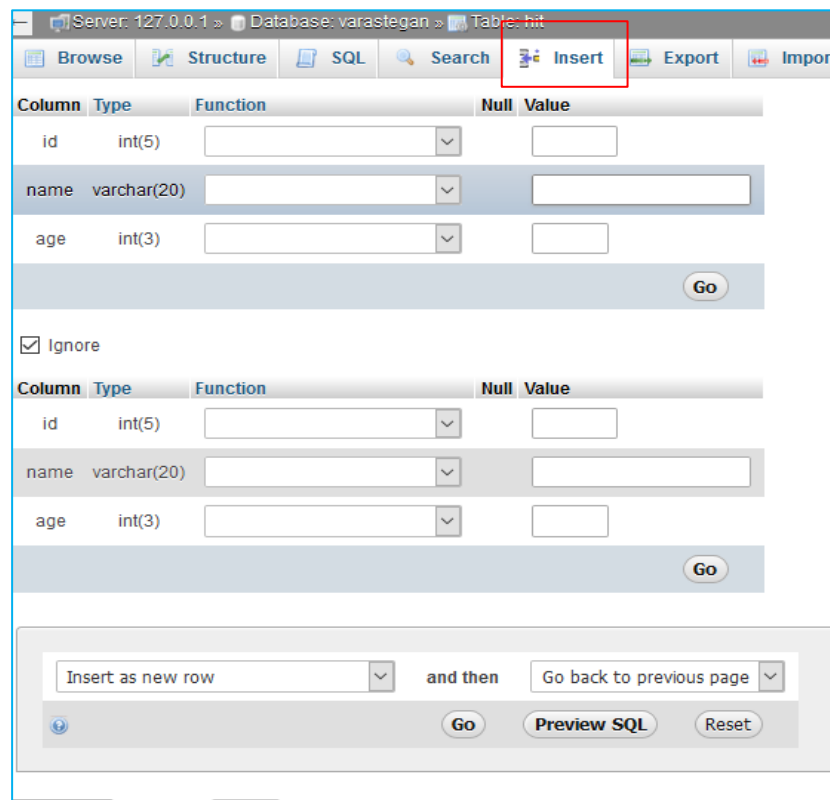
Partitions:

Preview SQL Save

حالا جدول ما آماده شده است و می توانیم مقادیر را به جدول خود اضافه کنیم. بعد از فشردن کلید Save صفحه ای به شکل زیر خواهیم داشت:



می بینیم که جدول HIT آماده است و همچنین می توان جدول های جدید را هم در قسمت پایین اضافه کرد. برای افزودن سطرهایی به جدول می بایست قسمت Insert را انتخاب کنیم. پس از انتخاب این قسمت به صفحه ای مانند زیر منتقل می شویم:



در این قسمت باید فیلدهای خالی را پر کرده و سپس کلید GO را بفشاریم تا سطرهای جدید به جدول افزوده شوند. برای افزودن یک سطر تنها فیلدهای قسمت اول و برای افزودن دو سطر فیلدهای قسمت دوم را هم باید پر کنیم. فیلدها را به شکل زیر مقداردهی می کنیم:

Server: 127.0.0.1 » Database: varastegan » Table: hit

Column	Type	Function	Null	Value
id	int(5)			971
name	varchar(20)			علی محمدی
age	int(3)			19

Ignore

Column	Type	Function	Null	Value
id	int(5)			951
name	varchar(20)			مهدی حسینی
age	int(3)			21

Insert as new row and then Go back to previous page

Go Preview SQL Reset

اگر دوسطر مقداره‌ی شوند باید کلید GO که در دومین فرم قرار گرفته را بفشاریم. پس از فشردن این دکمه به صفحه‌ی زیر منتقل می‌شویم:

Server: 127.0.0.1 » Database: varastegan » Table: hit

2 rows inserted.

```
INSERT INTO `hit` (`id`, `name`, `age`) VALUES ('971', '19', 'علی محمدی'), ('21', 'مهدی حسینی', '951');
```

Run SQL query/queries on table varastegan.hit:

```
1 INSERT INTO `hit` (`id`, `name`, `age`) VALUES ('971', '19', 'علی محمدی'), ('21', 'مهدی حسینی', '951');
```

Columns: id, name, age

SELECT * SELECT INSERT UPDATE DELETE Clear Format

Get auto-saved query

Bind parameters

Bookmark this SQL query:

هر قسمت از تصویر بالا مختصرا توضیح داده شده است:

B: این قسمت گزارش کار انجام شده را نشان می دهد که شامل یک پیغام است و به معنی اضافه شدن دو سطر به صورت موفق را گزارش می دهد.

C: این قسمت حاوی کدی است که با اجرای این کد اعمال انجام شده، انجام می شود. یعنی این کد، کد حاصل از افزودن دو سطر با مقادیر مشخص شده و... می باشد. می توان از این کد در کدهای php برای آسان تر شدن کار استفاده کرد.

D: این قسمت برای اجرای یک کد می باشد که می توان از ابزارهای قسمت های E و F هم استفاده کرد تا کدی ساخته و اجرا کنیم.

حالا برای مشاهده تمام سطرهای یک جدول که شامل مقادیری می باشند باید به قسمت A یعنی Browse برویم. این صفحه، صفحه ای به شکل زیر است:

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy

Showing rows 0 - 3 (4 total, Query took 0.0016 seconds.)

```
SELECT * FROM `hit`
```

Show all | Number of rows: 25 | Filter rows: Search this table

id	name	age
971	مریم محمدی	18
961	نرگس ابراهیمی	22
971	علی محمدی	19
951	مهدی حسینی	21

Show all | Number of rows: 25 | Filter rows: Search this table

در MySQL دستوراتی برای کار با دیتابیس وجود دارد که در زیر به آنها اشاره شده است:

- CREATE برای ایجاد یک دیتابیس، جدول و یا اندیس در MySQL استفاده می شود.
- DROP برای حذف یک دیتابیس و یا یک جدول به کار می رود.
- SELECT برای استخراج اطلاعات مورد نظر از یک یا چند جدول استفاده می شود.
- INSERT برای درج اطلاعات در جدول به کار می رود.

- REPLACE برای جایگزین کردن داده ها به کار می رود. اگر داده ای از قبل وجود داشته باشد آن را حذف و مقدار جدید را جایگزین می کند.
- UPDATE برای به روز رسانی داده های جدول به کار می رود.
- DELETE برای حذف داده ها از جدول به کار می رود.

نحوه اتصال به پایگاه داده (دیتابیس) در mysql

قبل از این که بتوانیم به اطلاعات درون پایگاه داده MySQL دسترسی داشته باشیم، بایستی یک اتصال یا Connection را به سرور برقرار کنیم. کد زیر نحوه اتصال به MySQL را به صورت عملی نشان داده است :

```
$conn = mysqli_connect($servername, $username, $password,dbname);
```

برای اتصال به هر سرور نیاز به یک نام کاربری (username) و رمز عبور (password) می باشد.

پارامترها:

servername : نام هاست را مشخص می کند. برای localhost این مقدار برابر با localhost می باشد.

username: نام کاربری اتصال را مشخص می کند. برای localhost این مقدار معمولاً برابر با root می باشد.

password : رمز عبور را مشخص می کند. برای localhost این مقدار معمولاً تنظیم نمی شود. (منظور از این که تنظیم نمی شود یعنی برابر با مقدار خالی null یا "" قرار می گیرد).

dbname : این پارامتر اختیاری است و نام دیتابیس موردنظر برای اتصال را مشخص می کند.

آموزش بستن یا Close اتصال در MySQL

اتصال (Connection) به پایگاه داده، به صورت اتوماتیک پس از اجرای کدهای اسکریپت بسته می شود، اما برای بستن زودتر آن می توان از کد زیر استفاده کرد:

```
$conn->close();
```

کد زیر نحوه برقراری اتصال به MySQL را نمایش می دهد.

کد PHP (برنامه ۱۲)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
  <title>Connection to DB</title>
</head>
<body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$conn = new mysqli($servername, $username, $password);
```

```

if ($conn->connect_error) {
    die ( " : عدم برقراری اتصال به پایگاه داده");
}
else{
    echo"اتصال به پایگاه داده برقرار شد";
}
$conn->close();
?>
</body>
</html>

```

```

1 ▼ <html dir="rtl">
2 ▼ <head>
3     <meta charset="utf-8">
4     <title>Connection to DB</title>
5
6 </head>
7 ▼ <body>
8 <?php
9 $servername = "localhost";
10 $username = "root";
11 $password = "";
12
13 $conn = new mysqli($servername, $username, $password);
14
15 ▼ if ($conn->connect_error) {
16     die("عدم برقراری اتصال به پایگاه داده");
17 }
18 ▼ else{
19     echo "اتصال به پایگاه داده برقرار شد";
20 }
21
22 $conn->close();
23 ?>
24 </body>
25 </html>

```

اتصال به پایگاه داده برقرار شد

خروجی :

آموزش Export کردن (خروجی گرفتن) در database در phpMyAdmin

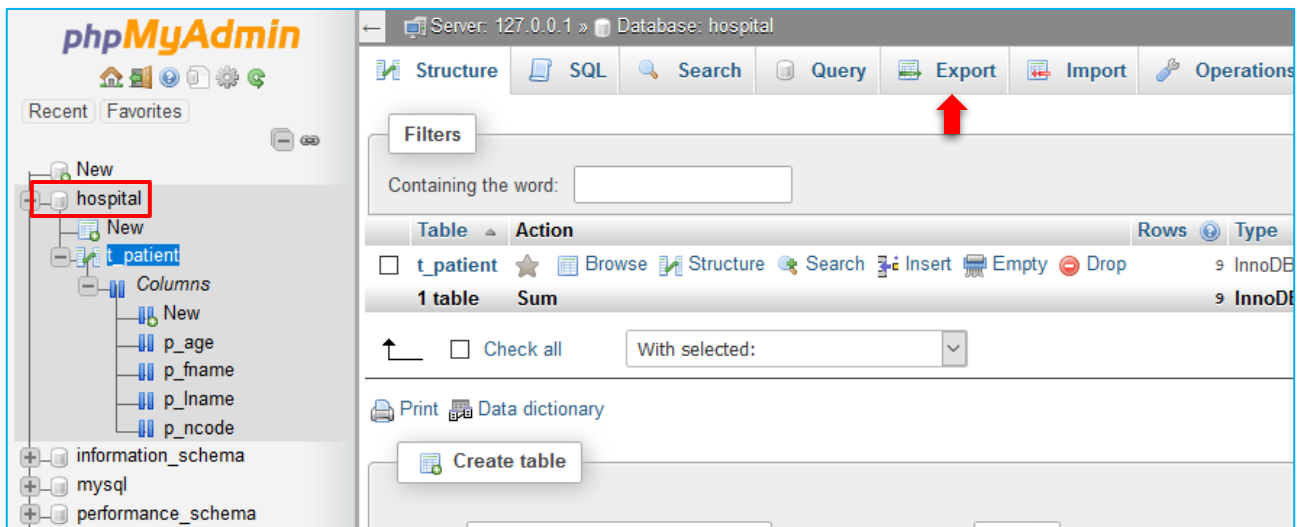
در این قسمت می خواهیم درباره‌ی این موضوع صحبت کنیم که چطور می توانیم از پایگاه داده‌ای که داریم خروجی بگیریم. اینکار زمانی کاربرد دارد که شما پروژه‌ای را نوشتید و میخواهید آن پروژه را روی سیستم دیگری نیز اجرا کنید و یا اینکه میخواهید آن پروژه را به یک host واقعی منتقل کنید.

برای انجام اینکار، علاوه بر انتقال فایل ها، باید پایگاه داده خود را نیز منتقل کنید و برای انتقال پایگاه داده باید از آن خروجی یا Export بگیرید. برای انجام اینکار باید از phpmyadmin که استفاده کنید.

از طریق منوی سمت چپ تصویر، دیتابیس موردنظر خود را انتخاب کنید و از طریق گزینه های بالای این قسمت گزینه Export را انتخاب کنید.

برای Export کردن شما می توانید از طریق دور روش Quick و Custom اقدام کنید که خروجی در روش Quick از تمام قسمت های دیتابیس است ولی در روش Custom شما خروجی ها را مشخص می کنید.

همچنین فرمت خروجی را نیز شما باید SQL انتخاب کنید.



در قسمت New templates نام مورد نظر خود را وارد کرده و در قسمت Export method یکی از دو گزینه را با توجه به توضیحات بالا انتخاب کرده و فرمت انتخاب شده نیز باید Sql باشد و در انتها گزینه GO را کلیک نمایید.

Exporting tables from "hospital" database

Export templates:

New template:

Existing templates: Template:

Export method:

Quick - display only the minimal options
 Custom - display all possible options

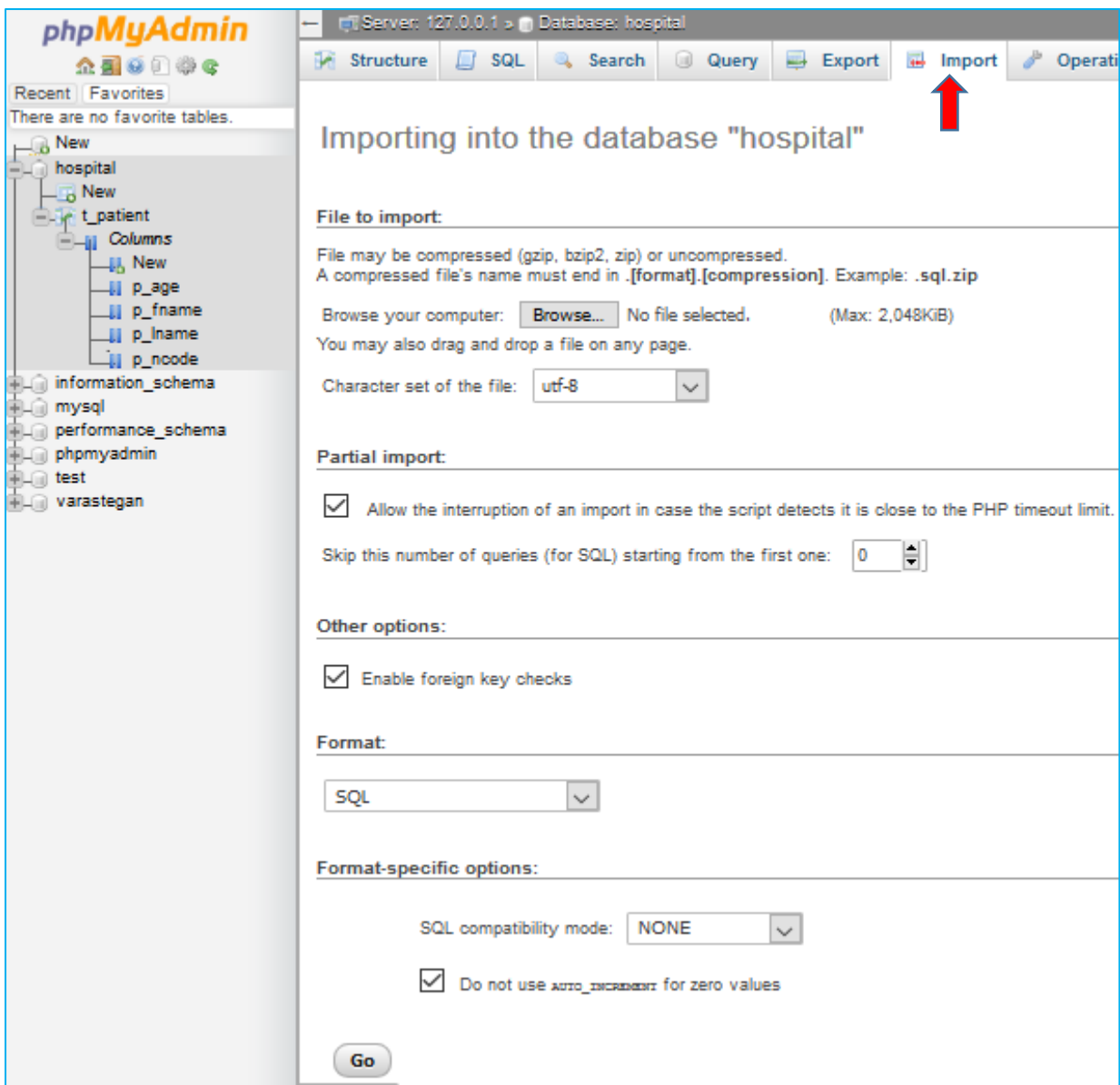
Format:

آموزش import کردن (وارد کردن) در database در phpMyAdmin

مانند export کردن باید وارد قسمت phpmyadmin شوید و از طریق منوی سمت چپ دیتابیس مورد نظر خود را انتخاب کنید. (و یا ابتدا از طریق سربرگ databases دیتابیس خود را ایجاد کنید.) سپس از طریق گزینه های بالا گزینه import را انتخاب کنید.

در قسمت Browse your computer فایل مربوط به دیتابیس را انتخاب کنید.

در قسمت Character set of the file گزینه utf-8 را انتخاب کنید و روی گزینه Go کلیک کنید.



آموزش وارد کردن اطلاعات (Insert) در MySQL

پس از این که یک پایگاه داده (database) و جدول (table) درون آن را ایجاد کردید، می توانید اقدام به وارد کردن اطلاعات (Insert Data) نمایید.

در لیست زیر، برخی از قوانین مهم در هنگام وارد کردن اطلاعات در MySQL را بیان کرده ایم:

- کد SQL یا (SQL query) را بایستی درون کدهای PHP قرار دهید.
- مقادیر متنی یا String در دستورات SQL query بایستی داخل دو " " قرار بگیرند.
- مقادیر عددی را بایستی بدون " " قرار دهید (به صورت ساده).
- کلمه کلیدی Null نیز نباید بین دو " " قرار گیرد.

از دستور INSERT INTO همانند کد زیر برای وارد کردن تعدادی رکورد جدید درون جدول MySQL Table استفاده می شود:

```
$sql="INSERT INTO table_name (column1, column2, column3,...)
```

```
VALUES ('value1', 'value2', 'value3',...)
```

در برنامه زیر ابتدا فرم ثبت اطلاعات بیماران را با استفاده از HTML می نویسیم سپس با توجه به کدهای PHP برنامه، در صورتی که برقراری اتصال به MySQL با موفقیت انجام شود با توجه به دیتابیس با نام hospital که در MySQL ایجاد کرده ایم و دارای جدولی با نام t_patient و ستون های p_ncode ، p_fname ، p_lname و p_age می باشد دسترسی پیدا خواهد کرد و با وارد کردن اطلاعات بیماران در فرم، این اطلاعات جدید در پایگاه داده نیز ذخیره می گردد. در کد برنامه زیر نحوه وارد کردن اطلاعات جدید به جدول MySQL را با دستور INSERT INTO نشان داده ایم:

کد HTML (برنامه ۱۳)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<form action="program۱۳.php" method="post">
  <fieldset>
<h4> فرم ثبت اطلاعات بیماران </h4>
  <br> : کد ملی
  <input type="text" name="ncode"><br>
  نام : <br>
  <input type="text" name="fname"><br>
  نام خانوادگی : <br>
  <input type="text" name="lname"><br>
  سن : <br>
  <input type="text" name="age"><br>
  <input type="submit" value="ثبت">
  </fieldset>
</form>
</body>
</html>
```

```

1 ▼ <html dir="rtl">
2 ▼ <head>
3     <meta charset="utf-8">
4 </head>
5 ▼ <body>
6 ▼ <form action="program13.php" method="post">
7 ▼     <fieldset>
8         <h4>فرم ثبت اطلاعات بیمار</h4>
9         کد ملی: <br>
10        <input type="text" name="ncode"><br>
11        نام: <br>
12        <input type="text" name="fname"><br>
13        نام خانوادگی: <br>
14        <input type="text" name="lname"><br>
15        سن: <br>
16        <input type="text" name="age"><br>
17
18        <input type="submit" value="ثبت">
19    </fieldset>
20 </form>
21 </body>
22 </html>

```

کد PHP (برنامه ۱۳)

```

<html dir="rtl">
<head>
    <meta charset="utf-8">
    <title>Insert data to DB</title>
</head>
<body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname="hospital";

$ncode=$_POST["ncode"];
$fname=$_POST["fname"];
$lname=$_POST["lname"];
$age=$_POST["age"];

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die ( "عدم برقراری اتصال به پایگاه داده". $conn->connect_error);
}
$sql = "INSERT INTO t_patient (p_ncode, p_fname, p_lname, p_age)
VALUES ('$ncode', N'$fname', N'$lname', '$age)";
if ($conn->query($sql) === TRUE {
    echo "اطلاعات بیمار با موفقیت ثبت شد";
}

```

```

}
else {
    echo "خطا: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
</body>
</html>

```

```

1 ▾ <html dir="rtl">
2 ▾ <head>
3     <meta charset="utf-8">
4     <title>Insert data to DB</title>
5
6 </head>
7 ▾ <body>
8 <?php
9 $servername = "localhost";
10 $username = "root";
11 $password = "";
12 $dbname="hospital";
13
14 $ncode=$_POST["ncode"];
15 $fname=$_POST["fname"];
16 $lname=$_POST["lname"];
17 $age=$_POST["age"];
18
19 $conn = new mysqli($servername, $username, $password, $dbname);
20
21 ▾ if ($conn->connect_error) {
22     die("عدم برقراری اتصال به پایگاه داده: " . $conn->connect_error);
23 }
24
25 $sql = "INSERT INTO t_patient (p_ncode, p_fname, p_lname, p_age)
26     VALUES ('$ncode', N'$fname', N'$lname', '$age)";
27
28 ▾ if ($conn->query($sql) === TRUE) {
29     echo "اطلاعات بیمار با موفقیت ثبت شد";
30 }
31 ▾ else {
32     echo "خطا: " . $sql . "<br>" . $conn->error;
33 }
34
35 $conn->close();
36 ?>
37 </body>
38 </html>

```


خروجی:

اطلاعات بیمار با موفقیت ثبت شد

فرم ثبت اطلاعات بیماران

کد ملی:

نام:

نام خانوادگی:

سن:

آموزش استخراج اطلاعات (Select Data) از پایگاه داده MySQL

از دستور SELECT برای استخراج اطلاعات مورد نظر از یک یا چند جدول استفاده می شود. شکل کلی استفاده از دستور SELECT در MySQL به صورت زیر است:

`SELECT column_name(s) FROM table_name`

۱- اگر از کاراکتر * به جای نام ستون استفاده کنید، دستور SELECT اطلاعات کلیه ستون های جدول را می خواند:

`SELECT * FROM table_name`

۲- به جای علامت * که معادل تمامی ستون های جدول است می توانیم نام برخی از ستون ها را مقابل SELECT بنویسیم به عنوان مثال: (مانند برنامه شماره ۱۴)

`SELECT p_ncode, p_fname, p_lname, p_age FROM t_patient`

۳- اگر بخواهیم برخی ردیف های یک جدول را با وجود یک شرط بازخوانی کنیم، از کلمه WHERE در انتهای کوئری استفاده می کنیم: (مانند برنامه شماره ۱۵)

`SELECT p_ncode, p_fname, p_lname, p_age FROM t_patient WHERE p_ncode=$ncode`

در برنامه زیر که از پایگاه داده hospital که در برنامه شماره ۱۳ جزئیات آن بیان شد استفاده کردیم و در اینجا با استفاده از دستور SELECT تمام اطلاعاتی که در پایگاه داده مان وارد کردیم در خروجی نمایش داده می شود.

➤ با استفاده از تابع (`fetch_assoc()` در PHP رکوردها را یک به یک خوانده و در متغیر `row` می ریزیم و آن جدول را چاپ می کنیم.

➤ برای چاپ همه رکوردها از دستور `while` در PHP استفاده می کنیم.

➤ پس از چاپ همه رکوردها با دستور `close` اتصال را قطع می کنیم.

در ادامه به تشریح نکات کد برنامه زیر می پردازیم:

در ابتدا یک SQL query را با استفاده از دستور SELECT ایجاد کرده و اطلاعات فیلدهای های p_fname ، p_ncode ، p_age و p_lname را از جدول t_patient خوانده ایم.

خط بعدی SQL query فوق را اجرا کرده و اطلاعات دریافت شده را در یک متغیر به \$result ذخیره می کند. در مرحله بعدی تابع num_rows() چک می کند آیا اطلاعات یک یا تعداد بیشتری سطر (از صفر بیشتر) برگردانده شده است یا خیر. اگر بیشتر از صفر سطر یا رکورد برگردانده شده باشد، تابع fetch_assoc() اطلاعات را در یک متغیر آرایه رابطه ای (associative array) قرار داده که می توانیم با استفاده از دستوراتی مثل حلقه یا loop به جستجو درون آن پردازیم.

در مرحله آخر هم، حلقه while ، به جستجو درون آرایه پرداخته و اطلاعات خروجی شامل فیلدهای p_fname ، p_ncode ، p_age و p_lname را نشان می دهد.

کد PHP (برنامه ۱۴)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
  <title> Select data from DB </title>
</head>
<body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname="hospital";

$conn = new mysqli($servername, $username, $password, $dbname);
$conn->set_charset("utf8");

if ($conn->connect_error) {
  die ( " : عدم برقراری اتصال به پایگاه داده");
}

$sql = "SELECT p_ncode, p_fname, p_lname, p_age FROM t_patient";
$result = $conn->query($sql);
if ($result->num_rows > 0){
  echo "<table><tr><th>سن</th><th>نام خانوادگی</th><th>نام</th><th>کد ملی</th></tr>";
  while($row = $result->fetch_assoc())
  {
  echo"<tr><td>".$row["p_ncode"]."</td><td>".$row["p_fname"]."</td><td>".$row["p_lname"].
  "</td><td>".$row["p_age"]."</td></tr>";
  }
  echo "</table>";
}
```

```

} else{
    echo "0 results";
}
$conn->close();
?>
</body>
</html>

```

```

1 <html dir="rtl">
2 <head>
3     <meta charset="utf-8">
4     <title>Select data from DB</title>
5
6 </head>
7 <body>
8 <?php
9     $servername = "localhost";
10    $username = "root";
11    $password = "";
12    $dbname="hospital";
13
14    $conn = new mysqli($servername, $username, $password, $dbname);
15    $conn->set_charset("utf8");
16
17    if ($conn->connect_error) {
18        die("عدم برقراری اتصال به پایگاه داده" . $conn->connect_error);
19    }
20
21    $sql = "SELECT p_ncode, p_fname, p_lname, p_age FROM t_patient";
22    $result = $conn->query($sql);
23
24    if ($result->num_rows > 0) {
25        echo "<table><tr><th>کد ملی</th><th>نام</th><th>نام خانوادگی</th><th>سن</th></tr>";
26        while($row = $result->fetch_assoc()) {
27            echo "<tr><td>".$row["p_ncode"]."</td><td>".$row["p_fname"]."</td><td>".$row["p_lname"]."</td><td>".$row["p_age"]."</td>
</tr>";
28        }
29        echo "</table>";
30    } else {
31        echo "0 results";
32    }
33    $conn->close();
34    ?>
35 </body>
36 </html>

```

خروجی

کد ملی	نام	نام خانوادگی	سن
9425638741	محمد	علی پور	25
2589641478	بهار	یعقوبی	18
5874126954	حسین	امینی	24
3658741258	مریم	ایزدی	22

در برنامه زیر با استفاده از فرم طراحی شده در HTML و با دستورات MySQL، مشخصات یک شخص با رکورد خاص که در اینجا بر اساس کد ملی است، در خروجی نمایش می دهد. برای این منظور از کلمه کلیدی WHERE (به معنای: به شرطی که) به صورت زیر استفاده می شود:

```
$sql = "SELECT p_ncode, p_fname, p_lname, p_age FROM t_patient WHERE p_ncode=$ncode";
```

همچنین با توجه به اینکه این جستجو بر اساس کد ملی بیماران انجام می شود می بایست دستور زیر لحاظ گردد:

```
$ncode=$_POST["ncode"];
```

به همین ترتیب اگر جستجو بر اساس نام انجام شود:

```
$fname=$_POST["fname"];
```

و اگر جستجو بر اساس نام خانوادگی انجام شود:

```
$lname=$_POST["lname"];
```

کد HTML (برنامه ۱۵)

```
<html dir="rtl">
<head>
  <meta charset="utf-8">
</head>
<body>
<form action="program۱۵.php" method="post">
  <fieldset>
<h4>جستجوی اطلاعات بیماران بر اساس کد ملی</h4>
  <br>: کد ملی
  <input type="text" name="ncode"><br>
  <input type="submit" value="جستجو بر اساس کد ملی"><br>
  </fieldset>
</form>
</body>
</html>
```

```
1 ▼ <html dir="rtl">
2 ▼ <head>
3     <meta charset="utf-8">
4 </head>
5 ▼ <body>
6 ▼ <form action="program15.php" method="post">
7 ▼     <fieldset>
8         <h4>جستجوی اطلاعات بیماران بر اساس کد ملی</h4>
9         <br>: کد ملی
10        <input type="text" name="ncode">
11        <input type="submit" value="جستجو بر اساس کد ملی"><br>
12        </fieldset>
13    </form>
```

کد PHP (برنامه ۱۵)

```
<html dir="rtl">
```

```

<head>
  <meta charset="utf-8">
  <title> Select data from DB </title>
</head>
<body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname="hospital";
$ncode=$_POST["ncode"];
$conn = new mysqli($servername, $username, $password, $dbname);
$conn->set_charset("utf8");

if ($conn->connect_error) {
  die ( " عدم برقراری اتصال به پایگاه داده ". $conn->connect_error);
}
$sql = "SELECT p_ncode, p_fname, p_lname, p_age FROM t_patient WHERE p_ncode=$ncode";
$result = $conn->query($sql);
if ($result->num_rows > 0){
  echo "<table><tr><th> کد ملی</th><th>نام</th><th> نام خانوادگی</th><th>سن</th></tr>";
  while($row = $result->fetch_assoc())
  {
    echo"<tr><td>".$row["p_ncode"]."</td><td>".$row["p_fname"]."</td><td>".$row["p_lname"].
    "</td><td>".$row["p_age"]."</td></tr>";
  }
  echo "</table>";
} else{
  echo "اطلاعاتی یافت نشد";
}
$conn->close();
?>
</body>
</html>

```

```

1 <html dir="rtl">
2 <head>
3   <meta charset="utf-8">
4   <title>Select data from DB</title>
5
6 </head>
7 <body>
8 <?php
9 $servername = "localhost";
10 $username = "root";
11 $password = "";
12 $dbname="hospital";
13
14 $ncode=$_POST["ncode"];
15
16 $conn = new mysqli($servername, $username, $password, $dbname);
17 $conn->set_charset("utf8");
18
19 if ($conn->connect_error) {
20     die("عدم برقراری اتصال به پلگه داده: " . $conn->connect_error);
21 }
22
23 $sql = "SELECT p_ncode, p_fname, p_lname, p_age FROM t_patient WHERE p_ncode=$ncode";
24 $result = $conn->query($sql);
25
26 if ($result->num_rows > 0) {
27     echo "<table><tr><th>کد ملی</th><th>نام</th><th>نام خانوادگی</th><th>سن</th></tr>";
28     while($row = $result->fetch_assoc()) {
29         echo "<tr><td>". $row["p_ncode"]. "</td><td>". $row["p_fname"]. "</td><td>". $row["p_lname"]. "</td><td>". $row["p_age"]. "</td>";
30     }
31     echo "</table>";
32 } else {
33     echo "اطلاعاتی یافت نشد";
34 }
35 $conn->close();
36 ?>
37 </body>
38 </html>

```

خروجی

<p>جستجوی اطلاعات بیماران بر اساس کد ملی</p> <p>کد ملی: 3658741258</p> <p>جستجو بر اساس کد ملی</p>	←	<p>کد ملی نام نام خانوادگی سن</p> <p>22 مریم ایزدی</p>
--	---	--