

بسمه تعالی

جزوه درسی

برنامه نویسی مبتنی بر وب

آموزشگاه های فنی بیرجند

طراحی صفحات وب (HTML)

برنامه نویسی سمت سرور (PHP)

گردآورنده: سخی محمد محمدی

بخش اول : طراحی صفحات وب

1- مقدمه ای بر HTML

در این درس با ویژگیهای فایل‌های HTML یا اچتمل و نحوه ایجاد یک فایل ساده و نمایش آن در صفحه مرورگر آشنا خواهید شد.

یک فایل HTML چیست؟

- HTML را "اچ تی ام ال" و یا اچتمل بخوانید.
- HTML برگرفته از حروف اول **Hyper Text Markup Language** میباشد.
- یک فایل HTML فایلی از نوع **text** میباشد که متشکل از **markup tag** ها میباشد.
- مرورگر یا **Browser** از روی **markup tag** ها می فهمد که چگونه باید صفحه را نمایش بدهد.
- یک فایل HTML باید دارای انشعاب **htm** و یا **html** باشد.
- یک فایل HTML فایلی از نوع **text** میباشد که با هر ویرایشگر ساده ای قابل ایجاد است.
- وظیفه اصلی تگ های اچتمل (**markup tags**) بیان چگونگی نمایش اطلاعات میباشد.

میخواهید که یک فایل اچتمل بسازید؟

اگر از ویندوز مایکروسافت استفاده میکنید، **Notepad** را باز کرده و متن زیر را در آن تایپ کنید:

```
<html>
<head>
  <title>Title of page</title>
</head>
<body>
  This is my first html page. <b>This text is bold</b>
</body>
</html>
```

سپس فایل را با نام مثلا **test.htm** در درایو **d:** ذخیره کنید.

برای دیدن فایل اچتمل بالا در مرورگر یا **Brower** تان (معمولا اینترنت اکسپلورر و یا فایرفاکس) کافی است که از روی منوی **File|Open file** فایل بالا را از روی محل ذخیره شده بخوانید. (به آدرس فایل در قسمت **Address** توجه کنید، مثلا **d:\test.htm**)

یکی از مهمترین وظایف مرورگرها نمایش صفحات اجتمالی میباشد، چه این صفحات روی کامپیوتر شما ذخیره شده باشند و یا اینکه از اینترنت خوانده شوند. با کلیک روی این لینک فایلی شبیه فایل ذخیره شده توسط شما از روی وب سرور توسط مرورگرتان خوانده شده و سپس نمایش داده خواهد شد .

نکته ای در مورد ویرایشگرهای اجتمالی (HTML Editors)

با وجودیکه با استفاده از نرم افزارهای طراحی صفحات وب مانند FrontPage و یا Dreamweaver امکان نوشتن و طراحی صفحات اجتمالی بصورت WYSIWYG یا " What You See Is What You Get " وجود دارد اما توصیه میشود که از ویرایشگرهای معمولی متن برای تهیه صفحات خود استفاده کرده، عملکرد تگها را شناخته و هیچگاه بدون اینکه بدانید کاربرد یک تگ چیست آنرا بکار نبرید.

WWW & HTML-2

در این بخش با مفاهیم وب، اینترنت، مرورگرها و استانداردهای وب آشنا خواهید شد.

منظور از وب چیست؟

- وب شبکه ای است متشکل از تمامی کامپیوترهای دنیا، شبکه ای از شبکه ها.
- وب، WWW، web یا World Wide Web همگی یک چیزند.
- تمامی کامپیوترهای وب میتوانند با هم ارتباط داشته باشند.
- کامپیوترهای موجود در وب با کمک استاندارد ارتباطی یا پروتکل HTTP با هم ارتباط برقرار میکنند .
- HTTP یعنی Hyper Text Transfer Protocol (پروتکل انتقال ابرمتن)

نحوه کارکرد وب چگونه است؟

- اطلاعات وب داخل فایلهایی به نام Web Pages و یا صفحات وب قرار دارند.
- این فایلها یا صفحات روی Web Server یا کامپیوترهای سرورس دهنده وب ذخیره شده اند.
- برای دیدن صفحات وب از نرم افزاری به نام مرورگر و یا Web Browser استفاده میشود.
- دو مرورگر Internet Explorer و Firefox جزو معروفترین مرورگرها حساب میشوند.
- مرورگر Internet Explorer متعلق به شرکت مایکروسافت و مرورگر Firefox بصورت رایگان و اوپن سورس میباشد.

مرورگرها چگونه به خواندن صفحات وب میپردازند؟

- یک مرورگر با کمک یک Request درخواستی برای خواندن یک صفحه از وب سرور میکند.
- این Request یا درخواست بر اساس استاندارد ارتباطی یا پروتکل HTTP بوده و شامل آدرس صفحه مورد نظر میباشد.

▪ آدرس یک صفحه وب چیزی شبیه <http://www.astovan.com/faq.html> است. بخش <http://> نوع پروتکل و یا استاندارد ارتباطی را تعیین میکند، www.astovan.com نام دومین یا Domain است و [faq.html](http://www.astovan.com/faq.html) نام صفحه ای است که باید خوانده شود.

مرورگرها چگونه صفحات وب را نمایش میدهند؟

- چگونگی نمایش یک صفحه وب بصورت مستتر در آن وجود دارد.
- مرورگرها از روی دستورالعملهای داخل صفحات وب و با کمک تگ ها به نمایش صفحات میپردازند.
- وظیفه اصلی تگ های اجتمالی (HTML tags) بیان چگونگی نمایش اطلاعات میباشد.
- یک تگ اجتمالی چیزی شبیه `<p>` این تگ پاراگراف است! `<p/>` است.
- چه کسانی استانداردهای وب را تعیین میکنند؟
- تعیین استانداردهای وب ربطی به شرکتهای مایکروسافت و یا دیگر سازندگان مرورگرها ندارد.
- World Wide Web Consortium یا W3C متولی تعیین استانداردهای وب است.
- HTML, CSS و XML از مهمترین استانداردهای تصویب شده وب میباشد.

3- عناصر (Elements)

یک فایل اچتمل متنی از نوع text است که از عناصر یا element ها تشکیل میشود و برای ایجاد عناصر از تگ ها یا tags استفاده میشود. در این فصل با عناصر و تگ ها اچتمل آشنا خواهید شد.

تگ های اچتمل (HTML Tags)

- با کمک تگ های اچتمل عناصر ساخته میشوند.
- در زبان اچتمل حدود 80 عنصر تعریف شده است.
- تگ های اچتمل بوسیله دو کاراکتر < و > ساخته میشوند.
- تگ های اچتمل معمولا بصورت زوج ظاهر میشوند، مانند test
- تگ اول در یک زوج تگ مثلا تگ شروع و تگ دوم مثلا تگ پایانی نام دارد.
- متن بین تگ اول و تگ دوم در یک زوج تگ محتوای عنصر نامیده میشود.
- تگ های اچتمل را میتوان بوسیله حروف لاتین کوچک (lower case) و یا بزرگ (upper case) بنویسید و case sensitive (حساس به حالت) نیستند. برای مثال دو تگ و معادل هم هستند ولی شدیدا توصیه میشود که به خاطر سازگاری با XHTML از حروف کوچک استفاده شود.

عناصر اچتمل (HTML Elements)

مثال بخش مقدمه را در نظر بگیرید:

```
<html>
<head>
<title>Title of page</title>
</head>
<body>
This is my first html page. <b>This text is bold</b>
</body>
</html>
```

نمونه ای از یک عنصر اچتمل:

```
<b>This text is bold</b>
```

عنصر اچتمل بالا با تگ شروع شده و با تگ پایان می یابد. محتوای این عنصر عبارت " This text is bold" است. لازم به ذکر است که کاربرد تگ نمایش توپر یا bold متون است. *** مثال دوم از یک عنصر اچتمل :

```
<body>
This is my first homepage. <b>This text is bold</b>
</body>
```

عنصر اچتمل بالا با تگ <body> شروع شده و با تگ </body> پایان می یابد. همانطور که میبینید گاهی یک عنصر حاوی یک یا چند تگ دیگر میباشد. وظیفه تگ <body> تعیین بدنه اصلی یا body یک متن اچتمل است. لازم به یادآوری است که تنها اطلاعات بخش <body> یک فایل اچتمل در صفحه مرورگر نمایش داده خواهد شد.

شناسه های یک تگ (Tag Attributes)

تگ ها میتوانند حاوی اطلاعات اضافی دیگر باشند، به این اطلاعات شناسه یا خاصیت میگویند و وظیفه آنها بیان دیگر اطلاعات یک عنصر یا Element میباشد. مثلا در مورد تگ <body> شناسه ای به نام bgcolor وجود دارد که رنگ زمینه متن (background) را تعیین میکند برای نمونه اگر میخواهید که رنگ زمینه صفحه اچتمل تان سیاه باشد کافی است که به شکل زیر عمل کنید :

```
<body bgcolor="black">
```

در مثال زیر تگ <body> دارای چهار شناسه مختلف با نامهای width، height، align و border و مقادیر center، 60، 100 و 0 میباشد.

```
<table border="0" width="100" height="60" align="center" >
```

.....

```
</table>
```

شناسه ها به صورت کلی "مقدار" نام یا "name="value" نوشته میشوند و همیشه به تگ شروع یک عنصر یا Element اضافه میشوند و نهایتا اگر در یک عنصر شناسه ها قید نشوند از مقادیر قراردادی یا default آنها استفاده خواهد شد مثلا در تگ body اگر شناسه bgcolor نوشته نشود از رنگ سفید برای زمینه صفحه استفاده خواهد شد. مقدار یک شناسه را میتوانید داخل نویسه های " و یا ' بنویسید و اختیار دست شما است فقط در مواردی که مقدار یک شناسه شامل نویسه " هم میشود باید از نویسه ' استفاده شود.

4- تگ های اصلی (Basic Tags)

تگ ها (tags) اجزا تشکیل دهنده عناصر یا elements های اچتمل هستند و در این فصل با عناصر و تگ های پایه و مهم آشنا خواهید شد :

4-1- سر تیترها (Headings)

سر تیترها با کمک تگ های <h1> تا <h6> تعیین میشوند. <h1> معرف بزرگترین سر تیتر و <h6> معرف کوچکترین سر تیتر است. مرورگر به هنگام نمایش یک سر تیتر بصورت اتوماتیک یک سطر خالی قبل و بعد از هر سر تیتر اضافه خواهد کرد.

<h1>This is a heading</h1>	This is a heading
<h2>This is a heading</h2>	This is a heading
<h3>This is a heading</h3>	This is a heading
<h4>This is a heading</h4>	...
<h5>This is a heading</h5>	
<h6>This is a heading</h6>	

4-2- پاراگرافها (Paragraphs)

پاراگرافها با کمک تگ <p> معرفی میشوند. مرورگر به هنگام نمایش یک پاراگراف بصورت اتوماتیک یک سطر خالی قبل و بعد از آن اضافه خواهد کرد.

```
<p>This is a paragraph</p>
```

```
<p>This is another paragraph</p>
```

4-3- سطر جدید (Line Breaks)

برای رفتن سر سطر جدید از تگ `
` استفاده میشود. در اینحالت یک پاراگراف جدید ایجاد نمیشود. تگ `
` از نوع تگ های خالی بوده و دارای تگ انتهائی (مثلا `</br>`) نمیباشد.

`<p>This
 is a para
graph with line breaks</p>`

خروجی:

This
is a para
graph with line breaks

4-4- توضیحات در اجتمل (Comments)

برای نوشتن شرح و توضیحات در مورد کدهای اجتمل باید از تگ خاصی استفاده کنید. برای اینکار باید متن و شرح را درون علائم `<!-->` قرار دهید. مرور تگ های `comment` را در نظر نگرفته و محتوی آنها را نمایش نخواهد داد و فقط شرح و توضیحات برای برنامه نویس و دیگر افرادی که احتمالا در آینده با کد اجتمل کار خواهند کرد مفید خواهد بود. (به محل نویسه `!` توجه کنید! فقط یکی و آنهم در ابتدا)

`<!-- This is a comment -->`

4-5- چند نکته کاربردی:

- توجه داشته باشید که بدلیل وجود مرورگرهای متفاوت (اکسپلورر، فایرفاکس، ..) و به دلیل تفاوت دقت نمایش صفحه نمایش کامپیوترها، صفحات اجتمل با اندکی تفاوت در حالتهاى مختلف نمایش داده میشوند و همیشه سعی کنید که صفحاتتان را نه تنها با مرورگر IE که معروفترین مرورگر است بلکه با دیگر مرورگرها و در رزولوشن های مختلف امتحان کنید.
 - از نظر فاصله و سطر بندی، متنی که در صفحه ویرایشگرتان تایپ میکنید با چیزی که مرورگر نمایش خواهد داد متفاوت خواهد بود. همیشه به یاد داشته باشید که فاصله های اضافی (space) و خطهای خالی متن در صفحه ویرایشگر توسط مرورگر در نظر گرفته نشده و نمایش داده نخواهد شد.
 - برای ایجاد یک سطر جدید هیچگاه از یک تگ `<p>` خالی استفاده نکنید و به جای آن از تگ `
` استفاده کنید..
 - مرورگرها به هنگام نمایش بعضی عناصر بصورت اتوماتیک یک سطر خالی قبل و بعد از آن عنصر نمایش خواهند داد. برای نمونه این گروه از عناصر میتوان از پاراگراف (`<p>`) و سرتیترها (`<..h>`) نام برد.
 - تگ `<hr>` یا `Roler Horizontal` سبب نمایش یک خط افقی خواهد شد و در واقع بخش های مختلف مطالب این صفحات با کمک این تگ از هم جدا شده اند.
- تگهای اصلی**
در جدول زیر عناصر معرفی شده در این فصل به همراه لینکهای مربوطه جهت مطالعه بیشتر آورده شده است. توجه داشته باشید که برای هر عنصر فهرستی از شناسه ها یا `Attributes` موجود است و همچنین به شناسه های کنارگذاشته شده (`Deprecated`) در نسخه های آینده اجتمل توجه داشته باشید و سعی کنید که از آنها استفاده نکنید.

تگ	کاربرد
<code><html></code>	نشان شروع متن اجتمل
<code><body></code>	تعیین بدنه و قسمت اصلی صفحه اجتمل
<code><h1>--<h6></code>	تعریف سر تیترهای h1 تا h6
<code><p></code>	تعریف پاراگراف
<code>
</code>	رفتن سر خط جدید
<code><hr></code>	نمایش خط افقی

<!-->	نوشتن شرح و comment
-------	---------------------

5- فارسی نویسی

سؤالات مربوط به فارسی نویسی را میتوان به سه گروه عمده تقسیم کرد:

- چگونه فارسی بنویسیم؟
- چگونه فارسی بخوانیم؟
- چگونه صفحات اچتمل فارسی را نمایش دهیم؟

1-5 دو نکته بسیار مهم در مورد فارسی نویسی

- روش کدگذاری UTF-8

اکیدا توصیه میشود که برای تایپ و تمایش متون فارسی از استاندارد یونیکد و روش کدگذاری UTF-8 استفاده کنید. نکته مهم در نمایش صحیح متون فارسی تعیین نوع کدگذاری یا encoding صفحات فارسی است که برای اینکار باید از متاتگ خاصی به نام http-equiv در بخش head صفحات اچتمل به شکل زیر استفاده شود:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

- شناسه یا attribute ی به نام dir

جهت نمایش متون لاتین به صورت پیش فرض و default از چپ به راست (ltr) میباشد. در مورد متون و جملات فارسی باید با کمک روشی جهت نمایش پیش فرض را به "راست به چپ" تغییر دهید. شناسه ای به نام dir یا همان direction این کار را برای شما انجام میدهد. این شناسه دارای دو مقدار ممکن میباشد:

- dir="ltr" , Left-to-right text. چپ به راست

- dir="rtl" , Right-to-left text. راست به چپ

```
<p dir="rtl"> سلام بر دنیای وب </p>
```

در اکثر از عناصر و تگ های اچتمل امکان استفاده از شناسه dir وجود دارد.

2-5 نحوه دیدن کدمنبع کدهای اچتمل (HTML Source)

○ اگر میخواهید که سورس کدهای اچتمل صفحات وب را ببینید کافی است که در منوی View مرورگر گزینه Source یا Page Source را اجرا کنید.

○ اگر طراح سایت با کمک Java Script مانع دیدن سورس توسط شما میشود، ابتدا در کادر زیر:

Tools/Internet Options/Security/Custom Level

مرورگر امکانات جاوااسکریپت را غیر فعال کرده (disable) و سپس مراحل بالا را تکرار کنید.

6- تگهای فرمت دهی متون

تگهای فرمت دهی متون

کاربرد	Start Tag
نمایش توپر	
نمایش ایتالیک یا کج	<i>
نمایش در اندازه کوچک	<small>
نمایش قوی؟!	

<sub>	نمایش پایین تر از خط افقی
<sup>	نمایش بالاتر از خط افقی
<strike>	از رده خارج
<u>	زیرخط دار
<pre>	نمایش متون از قبل فرمت شده

7- نهادهای کاراکتری

در استاندارد اچتمل گروهی از نویسه ها (Character) دارای معنی خاصی بوده (مثلا نویسه < شروع یک تگ و نویسه > انتهای یک تگ را نشان میدهد.) و برای نمایش آنها در صفحاتتان مستقیما قابل مصرف نمی باشند. در این قسمت با فهرست این نویسه ها آشنا شده و نحوه درج آنها را با کمک Character Entities فرا خواهید گرفت.

7-1- Character Entities

در اچتمل بعضی از نویسه ها دارای معنی خاصی بوده و برای نمایش آنها در صفحاتتان باید از character entities ها استفاده کنید. یک character entities از سه قسمت تشکیل میشود:

1- نویسه ampersand (&)

2- نام entity یا نویسه # و سپس شماره عددی entity

3 - و نهایتا نویسه semicolon (;)

برای مثال برای نمایش نویسه < دو روش وجود دارد، باید از < یا از < استفاده کنید. در این مثال lt نام entity و 60 شماره عددی آن میباشد.

مزیت استفاده از نام entity در مقابل استفاده از شماره عددی آن سهولت یادآوری آن از روی نامش میباشد و عیب آن این است که تمامی مرورگرها نام گذارهای موجود برای Entity ها را قبول ندارند ولی در عوض همگی شماره های عددی entity ها را بخوبی میشناسند. توجه داشته باشید که entity ها به کوچک و بزرگی حروف حساس هستند و در واقع Case Sensitive میباشد.

پر کاربردترین character entity در اچتمل، نویسه فاصله یا Space یا Blank میباشد و نام رسمی آن non-breaking space میباشد. همانطور که میدانید مرورگرها فاصله های اضافی را در نظر نگرفته (truncate spaces) و مثلا از 10 نویسه فاصله، 9 تای آنها نادیده میگیرند و برای نمایش فاصله های اضافی باید از استفاده کنید.

7-2- فهرست پرکاربردترین کاراکترها

نمایش	نام	کد عددی
	 	
<	<	<
>	>	>
&	&	&
"	"	"

		#39;
--	--	------

8- پیوندهای اجتملی (HTML Links)

در این قسمت با پیوندها (Links) ، تگ های مربوطه مخصوصا تگ Anchor یا A و نحوه به کار گیری آنها آشنا خواهید شد. همچنین شناسه های بسیار مهم href و target و name تشریح خواهند شد. در محیط وب، صفحات اجتملی با کمک پیوندها به یکدیگر متصل (Link) میشوند. اصطلاح ابرمتن (Hyper Text) در مقابل متن خطی (Linear) قرار دارد. در یک متن معمولی خواندن به شکل خطی و از ابتدا به انتها میباشد و در مقابل در یک متن مختلط (Hyper) با کمک پیوندها میتوان از یک متن به هر صفحه دیگر در وب متصل شد. این کار با کمک عنصری معلوم الحال! به نام A یا Anchor میسر میگردد.

1-8- تگ Anchor و شناسه href

برای ایجاد پیوند به صفحات دیگر از تگ **<a>** استفاده میشود. پیوندها میتوانند به بخش دیگری از همان صفحه، صفحات دیگر وب، تصاویر، فایل های صوتی یا حتی فیلم ها و ... اشاره کنند. فرم کلی یک پیوند به قرار زیر است:

```
<a href="url">Some Text</a>
```

در مثال بالا تگ **<a>** برای ایجاد پیوندی به صفحه ای دیگر که آدرس اینترنتی آن url میباشد بکار رفته است. برای تعیین مقصد و یا آدرس صفحه جدید از شناسه ای به نام href که همان **hyperlink reference** میباشد، استفاده میشود و مقدار این شناسه در واقع همان آدرس اینترنتی صفحه مقصد است. بخش قابل رویت پیوند و در واقع متنی (و یا تصویری) که توسط مرورگر نمایش داده خواهد شد و بازدیدکننده روی آن کلیک خواهد کرد میان تگهای **<a>** و **** قرار داده میشود و هر چند که در مثال بالا این بخش متن "Some Text" است ولی میتواند حتی یک تصویر باشد. برای نمونه کد اجتملی زیر پیوندی به سایت google.com ایجاد خواهد کرد:

```
<a href="http://www.google.com/">Visit Google Site</a>
```

و مرورگر پیوند بالا را به شکل زیر نمایش داده و در اثر کلیک روی پیوند توسط بازدیدکننده، مرورگر به سایت Google خواهد رفت.

با کمک شناسه target امکان تعیین مقصد پیوند جدید فراهم میشود. در مثال بالا پس از کلیک روی پیوند، مرورگر سایت google.com را باز کرده و جایگزین سایت فعلی خواهد شد. اگر میخواهید که مرورگر پیوند را در صفحه ای جدید باز کند باید از شناسه target و مقدار "_blank" برای آن استفاده کنید. مثال زیر سبب باز شدن سایت google.com در پنجره جدیدی خواهد شد:

```
<a href="http://www.google.com/" target="_blank">Visit Google Site</a>
```

خواص تگ	کاربرد حالت های مختلف شناسه target
target="_blank"	مرورگر پیوند را در یک پنجره جدید باز میکند.
target="_self"	مرورگر پیوند را همان پنجره باز میکند، (حالت پیش فرض یا default)
target="_parent"	مرورگر پیوند را فریم parent باز میکند. (کاربرد در مبحث فریمها)
target="_top"	مرورگر پیوند را در فریم مادر و اصلی باز میکند.

9- جداول (Tables)

در این درس با عنصر جدول (Table) ، سلولها یا خانه ها، مرز و border در جداول و تگ های مربوطه مخصوصا تگهای TD , TR , TABLE آشنا خواهید شد. توجه داشته باشید که جداول یکی از مهمترین عناصر برای آرایش و طرح بندی صفحات میباشد.

9-1- جدولها

برای تعریف جداول از تگ <table> استفاده میشود. یک جدول از یک یا چند سطر که با کمک تگ <tr> تعریف میشوند، تشکیل میشود. هر ردیف یا row از یک یا چند سلول، خانه یا cell تشکیل گردیده که با کمک تگ <td> ایجاد میشوند. نام های td و tr به ترتیب خلاصه شده table row و table data میباشد. محتوی یک سلول میتواند متن، تصویر، فهرستها، جداول دیگر، پاراگرافها و ... باشد. مثال زیر جدولی است با دو سطر و سه ستون :

کد اجتملى جدولى با دو سطر و سه ستون	نمایش جدول روبرو توسط مرورگر						
<pre><table border="1" dir="rtl"> <tr> <td>ردیف 1، سلول 1</td> <td>ردیف 1، سلول 2</td> <td>ردیف 1، سلول 3</td> </tr> <tr> <td>ردیف 2، سلول 1</td> <td>ردیف 2، سلول 2</td> <td>ردیف 2، سلول 3</td> </tr> </table></pre>	<table border="1"> <tbody> <tr> <td>ردیف 1، سلول 1</td> <td>ردیف 1، سلول 2</td> <td>ردیف 1، سلول 3</td> </tr> <tr> <td>ردیف 2، سلول 1</td> <td>ردیف 2، سلول 2</td> <td>ردیف 2، سلول 3</td> </tr> </tbody> </table>	ردیف 1، سلول 1	ردیف 1، سلول 2	ردیف 1، سلول 3	ردیف 2، سلول 1	ردیف 2، سلول 2	ردیف 2، سلول 3
ردیف 1، سلول 1	ردیف 1، سلول 2	ردیف 1، سلول 3					
ردیف 2، سلول 1	ردیف 2، سلول 2	ردیف 2، سلول 3					

9-2- جدولها و شناسه های border و dir

در مثال بالا شناسه border مرز جدول را مشخص میکند، مقدار 1 مرزی با ضخامت یک پیکسل را نمایش خواهد داد و مقدار صفر جدول را بدون مرز نمایش خواهد داد. توجه داشته باشید که حالت پیش فرض یعنی جدولی بدون شناسه border ، جداول بدون مرز را نمایش خواهد داد.

در مثال بالا شناسه dir یا direction و مقدار rtl برای آن، سبب تعیین نمایش جهت متون از راست به چپ خواهد شد و چون در داخل تگ table تعریف شده است به تمامی سلولهای جدول اعمال خواهد شد. ساده ترین جدول ممکن در اجتملى، جدولی است با یک سطر و یک ستون!

9-3- سرستون در جداول (Headings)

سرستونها در جداول با کمک تگ <th> تعریف میشوند. مثال زیر نحوه تعریف سرستونها را در جداول نمایش میدهد:

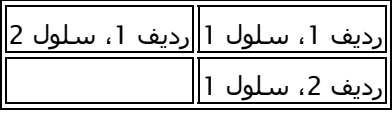
کد اجتملى جدولى با سه سطر و سه ستون	نمایش جدول روبرو توسط مرورگر									
<pre><table border="1" dir="rtl"> <tr> <th>سرستون 1</th> <th>سرستون 2</th> <th>سرستون 3</th> </tr></pre>	<table border="1"> <thead> <tr> <th>سرستون 1</th> <th>سرستون 2</th> <th>سرستون 3</th> </tr> </thead> <tbody> <tr> <td>ردیف 1، سلول 1</td> <td>ردیف 1، سلول 2</td> <td>ردیف 1، سلول 3</td> </tr> <tr> <td>ردیف 2، سلول 1</td> <td>ردیف 2، سلول 2</td> <td>ردیف 2، سلول 3</td> </tr> </tbody> </table>	سرستون 1	سرستون 2	سرستون 3	ردیف 1، سلول 1	ردیف 1، سلول 2	ردیف 1، سلول 3	ردیف 2، سلول 1	ردیف 2، سلول 2	ردیف 2، سلول 3
سرستون 1	سرستون 2	سرستون 3								
ردیف 1، سلول 1	ردیف 1، سلول 2	ردیف 1، سلول 3								
ردیف 2، سلول 1	ردیف 2، سلول 2	ردیف 2، سلول 3								

نمایش جدول روبرو توسط مرورگر	کد اجتملى جدولى با سه سطر و سه ستون
	<pre> <tr> <td>ردیف 1، سلول 1</td> <td>ردیف 1، سلول 2</td> <td>ردیف 1، سلول 3</td> </tr> <tr> <td>ردیف 2، سلول 1</td> <td>ردیف 2، سلول 2</td> <td>ردیف 2، سلول 3</td> </tr> </table> </pre>

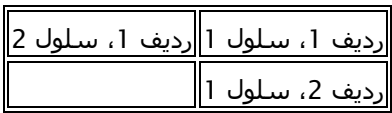
همانطوریکه مشاهده میکنید سر ستونها مانند سلولهای معمولی جدول تعریف میشوند و فقط به جای **td** از **th** استفاده شده و لی در نمایش محتوای سر ستونها **bold** یا توپر نمایش داده خواهند شد.

9-4- خانه های خالی در جداول

اگر محتوای خانه ای از یک جدول خالی باشد چه چیزی باید نمایش داده شود؟ بسته به مرورگری که استفاده میکنید نتیجه نمایش متفاوت است و مثلا در مثال زیر خانه خالی جدول بدون مرز نمایش داده خواهد شد:

نمایش جدول روبرو توسط مرورگر	کد اجتملى جدولى با دو سطر و دو ستون
	<pre> <table border="1" dir="rtl"> <tr> <td>ردیف 1، سلول 1</td> <td>ردیف 1، سلول 2</td> </tr> <tr> <td>ردیف 2، سلول 1</td> <td></td> </tr> </table> </pre>

در اینگونه موارد برای رفع مشکل کافی است که از **non-breaking space ()** یا همان نویسه و کاراکتر فاصله و بلانک استفاده شود و در مورد مثال ذکر شده مرورگر جدول را به شکل زیر نمایش خواهد داد:

نمایش جدول روبرو توسط مرورگر	کد اجتملى جدولى با دو سطر و دو ستون
	<pre> <table border="1" dir="rtl"> <tr> <td>ردیف 1، سلول 1</td> <td>ردیف 1، سلول 2</td> </tr> <tr> <td>ردیف 2، سلول 1</td> <td>&nbsp;</td> </tr> </table> </pre>

نمایش توسط مرورگر	کد HTML
1. Red 2. Blue	<pre> Red Blue </pre>
1. قهوه 2. شیر	<pre><ol dir="rtl" > قهوه شیر </pre>

داخل تگهای میتوانید از پاراگرافها، تگهای br ، تصاویر و حتی فهرستهای دیگر استفاده کنید. به شناسه dir در مثال بالا توجه کنید، امکان تعیین سمت و جهت نمایش متون در تمامی تگهای مربوط به فهرستها ممکن میباشد.

10-3- فهرستهای تعریفی (definition list)

فهرست تعریفی فهرستی است از اصطلاح ها و تشریح و توصیف آنها.

فهرستهای تعریفی با <dl> شروع شده و هر زوج اصطلاح و تعریف مربوط به آن با تگهای <dt> و <dd> تعریف میشوند.

نمایش توسط مرورگر	HTML code
HTTP Hypertext Transfer Protocol FTP File Transfer Protocol IP Internet Protocol	<pre><dl> <dt>HTTP</dt> <dd>Hypertext Transfer Protocol</dd> <dt>FTP</dt> <dd>File Transfer Protocol</dd> <dt>IP</dt> <dd>Internet Protocol</dd> </dl></pre>

تگهای فهرست

تگ	کاربرد
	تعریف فهرستهای مرتب
	تعریف فهرستهای نامرتب
	تعریف یک آیتم و قلم از یک فهرست
<dl>	تعریف فهرستهای تعریفی
<dt>	تعریف اصطلاح در فهرستهای تعریفی
<dd>	تعریف معنی و شرح اصطلاح در فهرستهای تعریفی

11- فرمها (Forms)

از طریق استفاده از عنصر <form> و چند تگ مرتبط قادر به دریافت اطلاعات از بازدیدکنندگان صفحات و با تبادل اطلاعات بین صفحات مختلف خواهید شد. با کمک این گروه از تگ ها قادر به نمایش باکسهای ورود اطلاعات متن (text fields) ، چک باکسها (check-boxes) ، رادیو باتونها (radio-buttons) و ... شده و همچنین امکان گذاشتن دکمه های

ارسال (submit button) و یا حذف (reset) را خواهید داشت. این فصل با شرح تگ های form و input و ... به چگونگی ایجاد ارتباط با بازدیدکنندگان و گرفتن اطلاعات از آنها بصورت لاتین و یا فارسی خواهد پرداخت. توجه داشته باشید که برای پردازش اطلاعات دریافتی از کاربر باید با یکی از زبانهای cgi از قبیل ASP، perl، PHP، CFM، JSP یا Java آشنائی داشته باشید.

11-1- فرمها (Forms)

تمامی تگهای مربوط به فرمها باید درون تگهای <form> و </form> قرار میگیرند. درون تگ فرم گروه زیادی از عناصر و تگهای مختلف قابل استفاده هستند از قبیل عناصر ورود متن یک سطر (text fields)، عناصر ورود متنهای چند سطر (Textarea)، منوهای drop-down و radio buttons و ...

11-2- تگ Input

به عنوان پر مصرف ترین تگ مربوط به ورود اطلاعات باید از تگ <input> نام برد. در این تگ شناسه ای به نام type به تعیین نوع اطلاعات ورودی اختصاص دارد. مقادیر ممکن برای این شناسه به قرار زیرند:

hidden , submit , reset , button , file , , text , checkbox , radio , password
image

در ادامه به شرح بعضی از type های کاربردی خواهیم پرداخت:

11-2-1- ورودیهای متن (Text Fields)

اگر میخواهید که بازدیدکننده اطلاعاتی از قبیل متن، اعداد و ... را وارد کند از شناسه ای با مقدار "text" استفاده میشود.

نمایش توسط مرورگر	کد اچتمل
USERNAME: <input type="text"/> PASSWORD: <input type="password"/>	<pre><form> USERNAME: <input type="text" name="user">
 PASSWORD: <input type="password" name="pass"> </form></pre>

توجه داشته باشید که تگ <form> چیزی را به نمایش نخواهد گذاشت بلکه تگهای درون آن توسط مرورگر نمایش داده خواهند شد. در مثال فارسی بالا به شناسه dir و مقدار rtl آن توجه داشته باشید. لازم به یادآوری است که اغلب مرورگرها در حالت پیش فرض برای ورودیهای متن اندازه 20 کاراکتر را در نظر میگیرند و اگر میخواهید که اندازه پیش فرض ورودیهای متن را تغییر دهید باید از شناسه ای به نام size استفاده کنید.

11-2-2- ورودیهای Radio Buttons

اگر میخواهید که بازدیدکننده گزینه ای را از بین چند گزینه محدود انتخاب کند، از مقدار "radio" برای شناسه type استفاده کنید:

نمایش توسط مرورگر	کد اچتمل
<input type="radio"/> مرد <input type="radio"/> زن	<pre><form dir="rtl"> <input type="radio" name="gender" value="male"> مرد
 <input type="radio" name="gender" value="female"> زن </form></pre>

همانطور که مشاهده میشود فقط امکان یکی از گزینه ها برای کاربر میسر است.

11-2-3- ورودیه‌های Checkboxes

اگر میخواهید که بازدیدکننده یک یا چند گزینه را از بین چند گزینه محدود انتخاب کند، از مقدار "checkbox" برای شناسه type استفاده کنید:

نمایش توسط مرورگر	کد اجتملی
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <input type="checkbox"/> دوچرخه دارم </div> <div style="text-align: center;"> <input type="checkbox"/> ماشین دارم </div> </div>	<pre><form dir="rtl"> <input type="checkbox" name="bike"> دوچرخه دارم
 <input type="checkbox" name="car"> ماشین دارم </form></pre>

11-2-4- ساخت لیست های انتخاب

```
<form >
<select name="state" >
<option value="1" > Tehran </option>
<option value="2" > Yazd </option>
<option value="3" > Fars </option>
</select>
</form>
```

11-2-5- شناسه Action و دکمه Submit در فرمها

در فرمها برای ارسال اطلاعات کسب شده از دکمه ای به نام ارسال یا Submit Button استفاده میشود و در اثر کلیک کاربر بروی این دکمه "Submit" ، اطلاعات درون فرم به فایلی دیگر ارسال خواهند شد. برای تعیین مقصد ارسال اطلاعات باید درون تگ form از شناسه ای به نام action استفاده کنید. مقدار شناسه action آدرس یا url فایلی است که به دریافت و سپس پردازش اطلاعات دریافتی خواهد پرداخت. معمولا فایل‌های بخش action برنامه ها و اسکریپت هایی نوشته شده با زبانهای cgi مانند ASP ، Perl ، PHP و ... بوده و وظیفه آنها دریافت اطلاعات فرمها و سپس پردازش آنها میباشد.

تعیین مقدار "submit" برای شناسه type سبب نمایش دکمه Submit با ارسال خواهد گردید.

نمایش توسط مرورگر	کد اجتملی
<div style="display: flex; align-items: center;"> Username: <input style="width: 100px;" type="text"/> <input style="margin-left: 10px;" type="submit" value="Submit"/> </div>	<pre><form name="input" action="form_action.php"> Username: <input type="text" name="user"> <input type="submit" value="Submit"> </form></pre>

در مثال بالا در جعبه ورودی متن، کلمه ای را وارد کرده و دکمه ارسال را کلیک کنید. مرورگر به محض کلیک دکمه ارسال، اطلاعات درون فرم را که در این مثال متنی ساده است به سمت فایلی که در شناسه action تعیین شده است ارسال کرده و در این مثال برنامه ای به زبان PHP با نام form_action.php در سمت سرور به ذخیره و سپس نمایش متنی خبری خواهد پرداخت.

کاربرد	Start Tag
تعریف فرم ورود اطلاعات	<form>
تعریف ورودی از نوع Input	<input>
تعریف ورودی متن چند سطری یا text-area	<textarea>
تعریف برچسب یا label	<label>
تعریف fieldset	<fieldset>
تعریف عنوان برای fieldset ها	<legend>
تعریف فهرستهای انتخابی یا drop-down box	<select>
تعریف گزینه ای از drop-down box ها	<option>
تعریف دکمه فشاری، متفاوت با button تگ input	<button>

12- تصاویر (Images)

در این درس با تگ `img` ، شناسه `src` و `alt` ، نگه‌های `Map` و `Area` و چگونگی نمایش تصاویر و همچنین تنظیم محل آنها در سندهای اچتمل آشنا خواهید شد.

12-1- تگ `img`

در زبان اچتمل تصاویر را با کمک تگ `` تعریف میکنند. تگ `` از نوع نگه‌های خالی است، بدین معنا که فقط دارای یک یا چند شناسه بوده و دارای تگ انتهایی یا `` نیست.

مهمترین شناسه برای درج و تعریف یک تصویر `src` نام دارد و مقدار این شناسه آدرس یا `URL` تصویر میباشد. تصاویری که مرورگر در یک صفحه وب نمایش میدهد میتوانند در همان دایرکتوری صفحه اچتمل، در دیگر دایرکتوریهای همان سایت و یا در فضای بیکران وب قرار داشته باشند.

شکل ساده درج یک تصویر:

```

```

مقدار `url` آدرس اینترنتی تصویر میباشد و مثلا اگر تصویری که قرار است که در صفحه درج شود در همان دایرکتوری قرار دارد و نامش `me.jpg` ، کافی است که به جای `url` فقط نام تصویر یعنی `me.jpg` را بنویسید و اگر تصویر در سایتی دیگر قرار دارد `url` کامل آنرا بنویسید.

مثلا تصویر لوگو سایت `google.com` دارای `url` ی برابر `http://www.google.com/images/logo.gif` است. در این `url` نام تصویر `logo.gif` بوده، در دایرکتوری `images` قرار داشته و روی وب سایت `google.com` قرار دارد.

شناسه های `width` و `height` تعیین عرض و ارتفاع نمایش تصویر است.

```

```

توصیه میشود که حتما ابعاد تصویر را با کمک شناسه های `width` و `height` قید کنید، اینکار سبب کمک به مرورگر در تسریع شروع نمایش صفحه خواهد شد.

اگر مرورگر به هر دلیلی نتواند که یک تصویر را نمایش دهد و یا مرورگر از نوع مرورگرهایی باشد که فقط متن را نمایش میدهند، متن و `text` ی که با کمک شناسه `alt` تعیین شده است نمایش داده خواهد شد.

12-3- چند نکته مهم

- اگر در یک صفحه اچتمل از 10 تصویر استفاده شده باشد، مرورگر باید 11 فایل را لود کند. (خود صفحه به علاوه 10 تصویر)
- استفاده از تصاویر، سرعت لود شدن صفحات را پایین می آورد. با احتیاط از تصاویر استفاده کنید و به اندازه و حجم تصویر توجه داشته باشید.
- لود تصویری با حجم 50 کیلوبایت برای کسی که از مودمی با سرعت 28kbps استفاده میکند حداقل 15 ثانیه طول خواهد کشید.
- برای دیدن مشخصات تصاویر در وب کافی است که روی تصویر **Right click** کرده و سپس در بخش **properties** آدرس، ابعاد و ظرفیت تصویر را ببینید.
- برای ذخیره تصاویر در وب کافی است که روی تصویر **Right click** کرده و سپس در بخش **Save picture as** آنرا ذخیره کنید.

13- پس زمینه (Background)

تگ **<body>** دارای دو شناسه مهم برای تعیین رنگ و تصویر پس زمینه میباشد. پس زمینه صفحات اچتمل رنگ و یا یک تصویر میتواند باشد. شناسه **bgcolor** رنگ پس زمینه را تعیین میکند. مقدار رنگ پس زمینه میتواند نام رنگ (مثل **red** یا **blue**) ، مقدار آن بر اساس استاندارد **RGB** و یا حتی مقدار هگزادسیمال (**Hexadecimal**) باشد.

<body bgcolor="#000000">

<body bgcolor="rgb(0,0,0)">

<body bgcolor="black">

برای تعیین تصویری که به عنوان پس زمینه صفحه نمایش داده میشود از شناسه **background** استفاده میشود. مقدار این شناسه **URL** یا آدرس تصویر (**image**) میباشد. اگر ابعاد تصویر کوچکتر از ابعاد صفحه مرورگر باشد، تصویر زمینه تکثیر خواهد شد تا حدی که تمام صفحه نمایش از تصویر زمینه پوشانده شود.

<body background="bg.gif">

<body background="http://www.astovan.com/images/bg.gif">

در مثال اول از آدرس دهی نسبی (**relative**) برای تعیین **URL** یا آدرس تصویر استفاده شده و در واقع در این مثال فایل اچتمل و تصویر **bg.gif** باید در یک دایرکتوری قرار گیرند. در مقابل مثال دوم از آدرس دهی مطلق (**absolute**) برای تعیین آدرس تصویر استفاده شده است و همانطور که میدانید آدرس تصویر در هر نقطه از فضای وب میتواند باشد.

13-2- نکات مهم

- اگر از تصاویر پس زمینه در صفحات اچتمل استفاده میکنید به نکات زیر توجه کنید :
- هر تصویری با فرمت **gif** و یا **jpg** قابل استفاده به عنوان تصویر زمینه میباشد.
- از تصاویر بزرگتر از ده کیلوبایت ظرفیت به هیچ وجه استفاده نکنید، در غیر این صورت سرعت لود صفحه بشدت پایین خواهد آمد.
- از تصاویر زمینه هماهنگ با دیگر تصاویر صفحه استفاده کنید.
- از تصاویر زمینه هماهنگ با متون صفحه استفاده کنید.
- در رزولوشنهای مختلف به دیدن و آزمایش تصاویر پس زمینه بپردازید.
- پر مصرف ترین رنگهای زمینه وب سایتها، رنگهای سفید، سیاه و خاکستری میباشند.
- اغلب سایتها و از تصاویر پس زمینه استفاده نمی کنند!

14- قلم ها (HTML Fonts)

با وجودی که هنوز بعضی از اجتملی کاران از تگ `` برای تعیین نام و نوع قلم، رنگ و اندازه متون استفاده میکنند ولی در استاندارد جدید اجتملی (نسخه 4) و همچنین XHTML از تگ `` پشتیبانی نمیشود و استفاده از `Style Sheet` ها به عنوان جایگزین معرفی شده است.

در نسخه های 3.2 و قبلی تر زبان اجتملی از تگ `` برای تعیین نام قلم، رنگ و اندازه متون استفاده میشود. در این تگ شناسه `face` نام قلم، شناسه `size` اندازه قلم و شناسه `color` معرف رنگ متن خواهد بود. مثالهای زیر کاربرد تگ `` را نشان میدهند:

```
<font face="نام قلم" size="اندازه" color="رنگ" >
```

محتوی

```
</font>
```

15- استایل شیتها (Style Sheets)

در نسخه جدید (4/0) اجتملی امکان جداسازی کامل فرمت دهی از ساختار صفحات اجتملی فراهم شده است. در این نسخه توصیه میشود که تمامی اطلاعات مربوط به نمایش و ترازبندی به فایل‌های مجزا از صفحه اجتملی منتقل شوند. بخشی از صفحات اجتملی که مربوط به نمایش، ترازبندی و فرمت دهی است، استایل شیت یا "Style Sheet" نامیده شده و معمولاً به صورت فایلی مجزا درون بخش `<head>` صفحات اجتملی معرفی میگردند. در این درس به صورت بسیار خلاصه با روشهای مختلف استفاده از استایل شیتها در صفحات اجتملی آشنا خواهید شد.

15-1- چگونگی استفاده از استایل شیتها (Cascade Style Sheet)

مرورگرها از روی استایل‌های تعریف شده توسط استایل شیتها به فرمت دهی و نمایش اطلاعات درون صفحه اجتملی میپردازند. تعریف و استفاده از استایل‌ها در صفحات اجتملی به سه روش مختلف ممکن میباشد:

1) استایل شیت‌های خارجی (external style sheet)

در این روش تمامی استایل‌ها و تعاریف نمایشی درون فایلی جداگانه قرار گرفته و بسیار مناسب حالت‌هایی است که قرار است که استایلی به بیش از یک صفحه اجتملی اعمال شود. مزیت این روش سهولت در تغییر ظاهر صفحات یک وب سایت میباشد و برای یک تغییر کلی در سایت کافی است که فایل استایل شیت کل سایت که معمولاً یک فایل واحد است تغییر داده شود. (باید توجه داشت که در این حالت تمامی صفحات سایت باید به یک فایل استایل شیت واحد لینک شده باشند.)

استایل شیت‌های خارجی توسط تگ `<link>` که درون بخش `head` صفحات اجتملی قرار میگیرد، معرفی میشوند.

در مثال زیر با کمک شناسه `href` از تگ `link` نام و در حالت کلی `url` فایل در برگیرنده استایل‌های تعریف شده تعیین میگردد. فایل‌های استایل شیت دارای پسوند `CSS` هستند.

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

```
</head>
```

2) استایل شیت‌های داخلی (Internal Style Sheet)

کاربرد این روش در حالت‌هایی است که صفحه اجتملی نیازمند تعریف استایلی منحصر به فرد و مجزا بوده و استایل طراحی شده فقط بدر آن صفحه خواهد خورد. در این حالت باید کدهای حاوی تعریف استایل درون تگ‌های `<style>` و `</style>` قرار گرفته و معمولاً نتیجه درون بخش `head` صفحه اجتملی گذاشته میشود. مثال:

```
<head>
```

```
<style type="text/css">
```

```
body {background-color: red}
```

```
p {margin-left: 20px}
```

```
</style>
</head>
```

3) استایل شیت‌های درجا (Inline Styles)

کاربرد این روش در مواردی است که باید برای یک المان (یا گروهی) خاص استایلی اعمال شود. در این حالت باید از شناسه ای به نام `style` که تقریباً در تمامی تگ‌ها قابل اعمال است استفاده شود. مثال:

```
<p style="color: red; margin-left: 20px">
This is a paragraph
</p>
```

2-15- انواع انتخاب کننده های استایل شیت

هر مجموعه از تنظیمات استایل شیت های داخلی و خارجی در داخل علامت های { و } قرار می گیرند. قبل از علامت { یک کلمه نوشته می شود.

- اگر این کلمه نام یک تگ باشد، تنظیمات تعیین شده به تمام تگ های همنام در کل صفحه بصورت خودکار اعمال خواهد شد.

TD{

Color:red;

Border : 1px dashed #545454;

}

- اگر این کلمه نامی غیر از تگ ها با علامت نقطه باشد، یک کلاس استایل شیت ایجاد کرده ایم و تنظیمات تعیین شده تنها به تگ هایی اعمال خواهد شد که شناسه `class` آنها به این کلاس مقاردهی شود:

.myclass{

Color:red;

Border : 1px dashed #545454;

}

<td class="myclass"> .. </td>

<p class="myclass"> .. </p>

- اگر این کلمه نامی غیر از تگ ها با علامت # باشد، یک شناسه استایل شیت ایجاد کرده ایم و تنظیمات تعیین شده تنها به تگ هایی اعمال خواهد شد که شناسه `id` آنها به این شناسه مقاردهی شود:

#ali{

Color:red;

Border : 1px dashed #545454;

}

<td id="ali"> .. </td>

<p id="ali"> .. </p>

کلاس و شناسه تعریف شده فوق می توانند به تمام تگ ها داده شوند، برای محدود کردن آنها به تگ خاصی باید قبل از علامت نقطه یا # باید نام تگ نوشته شود.

```
td#ali{
  Color:red;
  Border : 1px dashed #545454;
}
<td id="ali"> .. </td>
<p id="ali">.. </p>
```

بخش دوم : برنامه نویسی سمت سرور (PHP)

1- معرفی PHP

PHP یک زبان برنامه نویسی سمت سرور دهنده است

آنچه شما باید از قبل بدانید

قبل از اینکه یادگیری PHP را شروع کنید باید با موارد زیر آشنایی داشته باشید

سرور دهنده وب (Web Server): نرم افزاری است که بصورت دائمی بر روی کامپیوتر سرور دهنده در حال اجرا بوده به درخواست های HTTP گوش داده و با دریافت درخواست از سرور گیرنده (مثلا مرورگر وب) به آن پاسخ می دهد. وب سرورهای مهم عبارتند از

- IIS یا Internet Information Services: یکی از سرور های ویندوز است و می توان در برنامه Control panel آنرا نصب و فعال کرد.

- Apache: یک وب سرور قدرتمند ، اوپن سورس و رایگان و دارای نسخه های مختلفی برای استفاده در محیط ویندوز، لینوکس و غیره می باشد.

سرور دهنده کاربردی (Application Server): در صورت استفاده از یک زبان اسکریپت نویسی مثل PHP نیاز به مفسر یا امکانات نرم افزاری دیگری برای تفسیر و اجرای دستورات می باشد. زبان های برنامه نویسی مهم عبارتند از:

ASP یا (Active Server Pages): یک زبان با گرامر شبیه VB

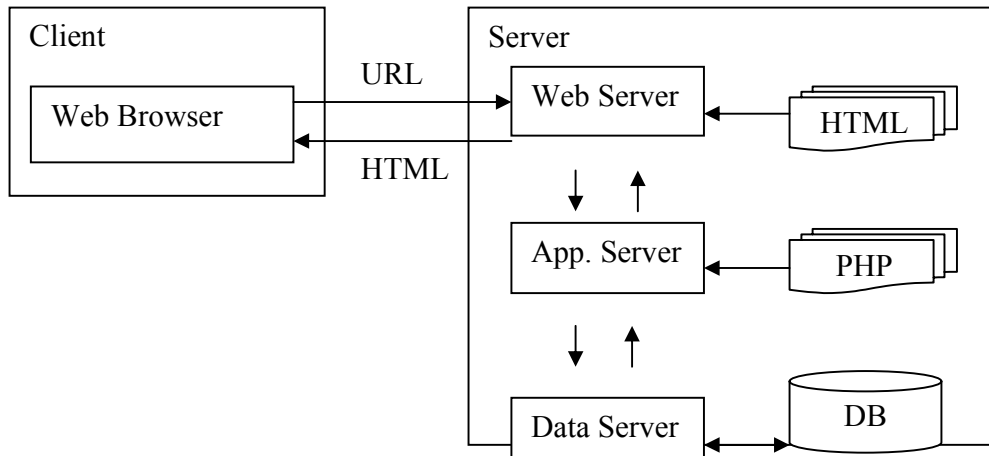
JSP یا (Java Server Pages): یک زبان مبتنی بر تکنولوژی جاوا

ASP.NET: یک زبان مبتنی بر تکنولوژی دات نت

PHP و ...

سرور دهنده داده (Data Server): در صورت نیاز به بانک اطلاعاتی در برنامه ها نیاز به یک نرم افزار DBMS یا بخشی از آن است که امکان مدیریت بانک اطلاعاتی و دسترسی برنامه ها به آن را فراهم می کند. این سرور دهنده

درخواست ها را به زبانی مثل SQL گرفته و پاسخ آنرا به برنامه بر می گرداند، این نرم افزار می تواند Access ، SQL Server ، Oracle ، MySQL و غیره باشد.
رابطه بین اجزای مختلف:



همیشه سرویس گیرنده، شروع کننده ارتباط است ، وقتی درخواست (Request) به وب سرور رسید، نوع فایل درخواستی بررسی می شود:

اگر مربوط به یک فایل اچتمل بود یا فایل های دیگری مثل txt و ... اگر فایل وجود داشته باشد، محتوای آن به درخواست کننده ارسال می شود.

اگر مربوط به اسکریپت برنامه نویسی بود از نرم افزار مخصوص آن برای اجرا کمک گرفته می شود. اگر دستورات برنامه برای دسترسی به بانک اطلاعاتی بود از نرم افزار فراهم کننده داده ها استفاده می شود .

نتیجه هر چه باشد عنوان پاسخ (Response) به سرویس گیرنده فرستاده می شود. برای شروع یک پروژه برنامه نویسی مبتنی بر وب باید تکنولوژی مورد نظر و مناسب را انتخاب کرد. ما در این درس از تکنولوژی WAMP استفاده می کنیم. یعنی Windows–Apache–MySQL–PHP . تکنولوژی AMP کاملاً اوپن سورس است و می تواند در سیستم عامل های ویندوز ، لینوکس و ... استفاده شود.

PHP چیست؟

- PHP اشاره دارد به Preprocessor Hypertext Performance :PHP
- PHP یک زبان برنامه نویسی سمت سرور مشابه ASP است
- اسکریپت های PHP بر روی سرور اجرا می شوند
- PHP از سیستم های بانک اطلاعاتی مختلفی پشتیبانی می کند نظیر (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc).
- PHP یک نرم افزار با کد منبع باز است
- PHP برای دانلود و استفاده رایگان است

فایل PHP چیست؟

- فایل های PHP می توانند حاوی متن ساده ، HTML و اسکریپت باشند.
- فایل های PHP به مرورگرها شبیه فایل های HTML ساده برگردانده می شوند.

- فایل های PHP می توانند پسوندهای php، php3 و phtml داشته باشند.

MySQL چیست؟

- MySQL یک سرویس دهنده بانک اطلاعاتی است
- MySQL برای استفاده در برنامه های کوچک و بزرگ کاربرد دارد.
- MySQL استاندارد SQL را پشتیبانی می کند.
- MySQL بر روی چندین پلتفرم قابل استفاده است.
- MySQL برای دانلود و استفاده رایگان است.

PHP + MySQL

- PHP و MySQL مستقل از پلتفرم هستند (شما می توانید برنامه هایتان را در محیط ویندوز توسعه داده و بر روی سیستم های UNIX اجرا کنید)

چرا PHP؟

- PHP بر روی پلتفرم های مختلف اجرا می شود (Windows, Linux, Unix, etc).
- PHP با اغلب سرورهایی که امروزه معمول هستند قابل استفاده است (Apache, IIS, etc).
- PHP رایگان است
- PHP برای یادگیری راحت است و بصورت کارا روی سرور اجرا می شود

چگونه شروع کنیم؟

برای اجرای برنامه های PHP نیاز به راه اندازی سرویس دهنده IIS و یا Apache و نصب مفسر PHP دارید، برای این کار می توانید یکی از بسته های زیر را دانلود و نصب کنید:

- EasyPHP
- WAMP

2- قواعد برنامه نویسی

کدهای PHP بر روی سرویس دهنده اجرا شده و نتیجه آنها بصورت HTML به مرورگر ارسال می شود.

یک بلوک برنامه نویسی PHP همیشه با `<?php` شروع شده و با `>?` پایان می یابد. یک بلوک برنامه نویسی PHP می تواند در هر جای صفحه قرارگیرد.

بر روی بعضی سرور ها می توانید به جای `<?php` کد PHP را با `<?>` شروع کنید. ولی توصیه می شود برای سازگاری بیشتر همیشه از همان `<?php` استفاده نمائید.

```
<?php
```

```
?>
```

یک فایل PHP حاوی تگ های HTML است مثل یک فایل HTML و در بعضی جاهای آن کدهای برنامه نویسی قرار گرفته اند.

در زیر مثالی از یک فایل PHP ساده نشان داده شده است که متن Hello World را به مرورگر می فرستد.

```
<html>
```

```
<body>
```

```
<?php
echo "Hello World";
?>
</body>
</html>
```

هر دستور PHP با علامت سمیکالن خاتمه پیدا می کند. سمیکالن جداکننده است و برای تشخیص دستورات از یکدیگر استفاده می شود.

دو دستور برای خروجی در PHP وجود دارد: `echo` و `print` در مثال فوق از دستور `echo` برای خروجی متن `Hello World` استفاده شد.

توجه: فایل PHP باید حتماً دارای پسوند `php` باشد. اگر پسوند آنرا `html` تعیین کنید اجرا نخواهد شد.

توضیحات در PHP

در PHP از `//` برای ایجاد یک خط توضیح و از `/* */` برای ساخت یک بلوک بزرگ از توضیحات استفاده می کنیم.

```
<html>
<body>
<?php
//This is a comment
/*
This is a comment
block
*/
?>
</body>
</html>
```

3- متغیرها در PHP

متغیر برای نگهداری اطلاعات استفاده می شود.

متغیرها برای ذخیره کردن مقادیری مثل رشته ها ، اعداد یا آرایه ها استفاده می شوند. همین که یک متغیر تعریف شد بارها و بارها می تواند در برنامه استفاده و مقداردهی شود.

نام همه متغیرها در PHP با علامت `$` شروع می شود.

روش صحیح تعریف متغیر در PHP:

```
$var_name = value;
```

برنامه نویسان مبتدی PHP اغلب علامت `$` را در ابتدای متغیرها فراموش می کنند و این باعث می شود که برنامه کار نکند. اجازه دهید یک متغیر برای نگهداری رشته و یک متغیر برای نگهدار اعداد تعریف کنیم:

```
<?php
$txt="Hello World!";
$x=16;
?>
```

PHP یک زبان برنامه نویسی با نوع داده ضعیف است

در PHP لازم نیست در زمان ایجاد متغیر و قبل از مقداردهی ، نوع داده آنها صریحا" تعریف شود. در مثال فوق می بینید که به PHP گفته نشده چه نوع داده هایی را نگهداری کند. بلکه PHP بطور خودکار نوع داده متغیر را بسته به مقدار تخصیص داده شده به آن بطور مناسب تغییر می دهد. در زبان های برنامه نویسی با تعیین نوع داده قوی مثل C باید صریحا" نوع داده متغیر را مشخص کنید. ولی در PHP متغیر در صورت مقداردهی خود به خود ایجاد شده و نوع داده آن تعیین می گردد.

قواعد نامگذاری برای متغیرها

- نام یک متغیر همیشه با یک حرف الفبا و یا علامت _ (underline) شروع می شود.
- نام یک متغیر می تواند شامل حروف الفبا، ارقام و علامت _ باشد.
- نام متغیر نمی تواند شامل space باشد. اگر نام متغیر شامل چند کلمه باشد برای جداکردن کلمات باید از _ استفاده شود(\$my_string) و یا کلمات به هم بچسبند(\$MyString).

4- متغیرهای رشته ای در PHP

متغیرهای رشته ای برای ذخیره کردن مقادیری که شامل کاراکترها باشند استفاده می شوند.

در این بخش نگاهی خواهیم داشت به توابع مهم و عملگرهای مورد استفاده بر روی رشته ها در PHP بعدا" رشته ایجاد کرده و با آن کار خواهیم کرد. یک رشته می واند مستقیما" با توابع رشته ای استفاده شده و یا در متغیر رشته ای ذخیره شود.

قطرنامه زیر متغیر رشته ای \$txt را نشان می دهد که حاوی رشته "Hello World" است.

```
<?php
$txt="Hello World";
echo $txt;
?>
```

خروجی کد بالا به صورت زیر است:

Hello World

حال اجازه دهید به عملکرد چند تابع و عملگر رشته ای پردازیم.

عملگر الحاق رشته ها

تنه یک عملگر رشته ای در PHP وجود دارد و آنهم عملگر الحاق رشته ها یعنی . (نقطه) است که دو رشته را به هم می چسباند. برای الحاق دو رشته با یکدیگر از عملگر الحاق بصورت زیر استفاده کنید.

```
<?php
$txt1="Hello World!";
```



```
$txt2="What a nice day!";
echo $txt1 . " " . $txt2;
?>
```

خروجی کد بالا به صورت زیر خواهد بود.

Hello World! What a nice day!

در کد بالا دو بار از عملگر الحاق استفاده شده و سه رشته به هم چسبیده اند و برای جداکردن دو کلمه از یکدیگر با فاصله استفاده کرده ایم.

تابع strlen()

تابع strlen() برای برگرداندن طول رشته استفاده می شود. بصورت زیر:

```
<?php
echo strlen("Hello world!");
?>
```

خروجی کد فوق به شکل زیر خواهد بود.

12

طول یک رشته اغلب برای ایجاد حلقه های تکرار یا توابع رشته ای دیگر اگر دانشتان پایان رشته مهم باشد کاربرد دارد.

تابع strpos()

تابع strpos() برای جستجوی یک رشته در یک متن استفاده می شود.

اگر کاراکتر در رشته پیدا شد، این تابع موقعیت آن در اولین انطباق را برمی گرداند و اگر پیدا نشد مقدار false برگشت داده می شود.

اجازه دهید رشته "World" را یک رشته دیگر جستجو کنیم.

```
<?php
echo strpos("Hello world!","world");
?>
```

خروجی کد فوق بصورت زیر است:

6

موقعیت رشته "World" در رشته ما مکان 6 است. دلیل اینکه 6 است (نه 7) این است که موقعیت اولین کاراکتر در رشته ها 0 است نه 1.

5- عملگرها در PHP عملگرهای محاسباتی

عملگر	شرح	مثال	نتیجه
+	جمع	$x=2$ $x+2$	4
-	تفریق	$x=2$ $5-x$	3
*	ضرب	$x=4$	20

	$x * 5$		
3	15/5	تقسیم	/
2.5	5/2		
1	5%2	باقیمانده	%
2	10%8		
0	10%2		
$x=6$	$x=5$ $x++$	افزایش	++
$x=4$	$x=5$ $x--$	کاهش	--

عملگرهای تخصیص

عبارت معادل	مثال	عملگر
$x=y$	$x=y$	=
$x=x+y$	$x+=y$	+=
$x=x-y$	$x-=y$	-=
$x=x*y$	$x*=y$	*=
$x=x/y$	$x/=y$	/=
$x=x.y$	$x.=y$.=
$x=x\%y$	$x\%=y$	\%=

عملگرهای مقایسه

مثال	شرح	عملگر
$5==8$ returns false	مساوی	==
$5!=8$ returns true	نا مساوی	!=
$5<>8$ returns true	نا مساوی	<>
$5>8$ returns false	بزرگتر	>
$5<8$ returns true	کوچکتر	<
$5>=8$ returns false	بزرگتر یا مساوی	>=
$5<=8$ returns true	کوچکتر یا مساوی	<=

عملگرهای منطقی

مثال	شرح	عملگر
x=6 y=3 (x < 10 && y > 1) returns true	and	&&
x=6 y=3 (x==5 y==5) returns false	or	
x=6 y=3 !(x==y) returns true	not	!

6- ساختارهای شرطی

دستورات شرطی برای انجام عملیات مختلف با توجه به شرایط مختلف استفاده می شوند.

دستور switch برای انتخاب یکی از چند بلوک کد برای اجرا استفاده می شود.
گرامر

```
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;
}
```

کد فوق چگونه عمل می کند: ابتدا یک عبارت n expression داریم (اغلب یک متغیر) ، که یکبار ارزیابی می شود. مقدار عبارت سپس با مقادیر هر کدام از ساختارهای case مقایسه می شود. اگر با آن مساوی بود ، بلوک کد قرارگرفته در آن

اجرا می شود. از **break** برای جلوگیری از ادامه اجرای کدها در بلوک **case** بعدی استفاده می گردد. دستورالعمل **default** در صورتی اجرا می شود که مقدار عبارت با هیچکدام از **case** ها مطابقت نداشته باشد.

مثال

```
<html>
<body>

<?php
switch ($x)
{
case 1:
    echo "Number 1";
    break;
case 2:
    echo "Number 2";
    break;
case 3:
    echo "Number 3";
    break;
default:
    echo "No number between 1 and 3";
}
?>

</body>
</html>
```

7- حلقه های تکرار

حلقه های تکرار یک بلوک از دستورات را به تعداد دفعات معین یا تا زمان برقرار بودن شرط خاصی تکرار می کنند.

اغلب وقتی کدنویسی می کنیم، مایل هستیم بلوک هایی از کد بیشتر از یک بار اجرا شوند. بجای اضافه کردن چندین کپی از دستورات بصورت پشت سرهم از حلقه های تکرار برای اتمام این کار استفاده می کنیم.

در PHP چندین دستورالعمل برای تکرار داریم:

- **while** - تکرار یک بلوک از کد تا زمان برقرار بودن یک شرط
- **do...while** - اجرای بلوک کد برای یکبار و سپس تکرار آن در صورت برقرار بودن یک شرط
- **for** - تکرار یک بلوک کد برای تعداد دفعات مشخصی
- **foreach** - تکرار یک مجموعه کد برای تمام عناصر آرایه

حلقه تکرار while

حلقه تکرار **while** تا زمان درست بودن نتیجه یک شرط دستورات را تکرار می کند.
گرامر

while (*condition*)

```
{
  code to be executed;
}
```

مثال

مثال زیر حلقه ای را نشان می دهد که با $i=1$ شروع شده و تا زمانی که i کوچکتر یا مساوی 5 باشد تکرار می کند. در هر بار تکرار به i یک واحد اضافه می شود.

```
<html>
<body>

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

</body>
</html>
```

خروجی:

```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
```

دستورالعمل do ... while

دستورالعمل **do...while** همیشه بلوک دستورات را یکبار اجرا می کند و سپس شرطی را بررسی می کند و دستورات را در صورتی که شرط درست بود تکرار می کند.

گرامر

```
do {
    code to be executed;
}while (condition);
```

مثال

در مثال زیر یک حلقه تکرار تعریف شده که با $i=1$ شروع می کند. سپس i را یک واحد افزایش می دهد و خروجی را چاپ می کند. سپس شرط را بررسی می کند و تا زمانی که i کوچکتر یا مساوی 5 باشد حلقه ادامه پیدا می کند.

```
<html>
<body>

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

خروجی:

```
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
```

حلقه های `for` و `foreach` در بخش بعدی بررسی خواهد شد.

حلقه های تکرار یک بلوک از دستورات را به تعداد دفعات معین یا تا زمان برقرار بودن شرط خاصی تکرار می کنند.

حلقه تکرار for

حلقه تکرار `for` زمانی استفاده می شود که تعداد دفعات تکرار اسکریپت را بدانیم.
گرامر

```
for (init; condition; increment)
{
    code to be executed;
}
```

پارامترها

- ***init***: اغلب برای مقدار اولیه دادن به شمارنده استفاده می شود (ولی می تواند شامل هر تعداد دستور باشد که در شروع اجرای حلقه اجرا خواهند شد)
- ***condition***: برای هر تکرار حلقه بررسی می شود.. اگر حاصل آن TRUE باشد حلقه ادامه پیدا می کند . اگر مقدار آن FALSE باشد حلقه پایان می یابد.
- ***increment***: اغلب برای افزایش شمارنده استفاده می شود. (ولی می تواند شامل هر مجموعه دستوری باشد که در انتهای حلقه تکرار می شوند)
نکته: هر کدام از پارامترهای فوق می تواند خالی باشد، یا می تواند بیش از یک دستور باشد (جدا شده با ویرگول).
مثال
مثال زیر حلقه ای را نشان می دهد که با $i=1$ شروع شده و تا زمانی که i کوچکتر یا مساوی 5 باشد تکرار می کند. در هر بار تکرار به i یک واحد اضافه می شود.

```
<html>
<body>

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

```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
```

8- توابع

قدرت واقعی PHP از توابع آن می آید. در PHP بیش از 700 تابع پیش ساخته (built-in) وجود دارد.

در این فصل نشان شما را با نحوه ایجاد توابع آشنا خواهیم کرد. برای جلوگیری از اجرای اسکریپت هنگام بارشدن صفحه، باید آنرا در داخل یک تابع قرار دهید. یک تابع در صورتی اجرا می شود که فراخوانی گردد. شما می توانید یک تابع را از هر جای صفحه فراخوانی کنید.

ایجاد یک تابع در PHP

یک تابع توسط یک فراخوانی تابع اجرا خواهد شد.

گرامر

```
function functionName()
{
code to be executed;
}
```

دونکته در مورد توابع:

برای تابع نامی را انتخاب کنید که نمایش دهنده کار آن باشد. نام تابع می تواند با یک حرف الفبا و یا خط زیر شروع شود

مثال

یک تابع ساده که وقتی فراخوانی شود نام من را چاپ می کند:

```
<html>
<body>

<?php
function writeName()
{
echo "S.M. Mohammad";
}
echo "My name is ";
writeName();
?>
</body>
</html>
```

خروجی:

My name is S.M. Mohammadi

توابع PHP - اضافه کردن پارامترها

برای افزایش کارایی و کاربرد یک تابع، می توانیم پارامترهایی به آن اضافه کنیم. پارامتر به سادگی یک متغیر است. پارامترها بعد از نام تابع و داخل پرانتز نوشته می شوند.

مثال 1

مثال زیر نام های مختلفی و فامیل های یکسانی را چاپ خواهد کرد:

```
<html>
<body>
<?php
function writeName($fname)
{
echo $fname . "Akbari.<br />";
}
```



```

echo "My name is ";
writeName("Ali");
echo "My sister's name is ";
writeName("Sara");
echo "My brother's name is ";
writeName("Saeed");
?>
</body>
</html>

```

خروجی:

My name is Ali Akbari.
 My sister's name is Sara Akbari.
 My brother's name is Saeed Akbari.

مثال 2

تابع زیر دو پارامتر دارد:

```

<html>
<body>

<?php
function writeName($fname,$punctuation)
{
echo $fname . "Akbari" . $punctuation . "<br />";
}

echo "My name is ";
writeName("Ali",".");
echo "My sister's name is ";
writeName("Sara","!");
echo "My brother's name is ";
writeName("Saeed","?");
?>

</body>
</html>

```

خروجی:

My name is Ali Akbari.
 My sister's name is Sara Akbari!
 My brother's name is Saeed Akbar?

توابع PHP - برگرداندن مقادیر

برای اینکه تابع مقداری را برگرداند، از دستور `return` استفاده می شود.
مثال

```
<html>
<body>
<?php
function add($x,$y)
{
$total=$x+$y;
return $total;
}
echo "1 + 16 = " . add(1,16);
```

6-7- استفاده از متغیر به صورت Global

بطور پیش فرض متغیرهای تعریف شده بیرون یک تابع، داخل تابع در دسترس نیست. مثلاً در مثال زیر

```
<?php
$life = 42;
function meaningOfLife()
{
    print "The meaning of life is $life<br>";
}
meaningOfLife();
?>
```

خروجی خالی چاپ می شود و مقدار `$life` را چاپ نخواهد کرد. در برخی موارد ما نیاز به استفاده از متغیرهای بیرون تابع داخل یک تابع داریم. برای این کار کافیست که از دستور `Global` استفاده کنیم. به طور مثال می توانیم برای این منظور کد بالا را به صورت زیر بازنویسی کنیم :

```
<?php
$life=42;
```

```
function meaningOfLife()
{
    global $life;
    print "The meaning of life is $life<br>";
}
meaningOfLife();?>
```

در خط 10 از دستور `global $life;` استفاده کردیم. در این حالت مقدار `$life` که بیرون تابع و در خط 7 تعریف شده در داخل تابع قابل دسترس می شود و خروجی این کد `The meaning of life is 42` چاپ خواهد شد. شما باید برای هر متغیری که می خواهید ارزش در تابع استفاده کنید از این دستور استفاده کنید. و همچنین در هر تابعی که می خواهید از متغیری خارج از آن تابع استفاده کند باید از این دستور استفاده شود. مهم: دقت کنید که اگر `$life` داخل تابع تغییر دهید مقدار `$life` در کل برنامه عوض می شود.

9- ضمیمه کردن فایل ها

ضمیمه کردن فایل ها

شما می توانید محتوای یک فایل `PHP` را در فایل دیگری درج کنید قبل از اینکه سرویس دهنده آنرا اجرا کند. برای این کار از توابع `include()` یا `require()` استفاده می کنید. دو تابع از هر نظر بجز دستکاری خطا با هم یکسان هستند.

- `include()` تولید `warning` کرده ولی اجرای اسکریپت ادامه می یابد.
- `require()` خطای غیرقابل چشم پوشی تولید می کند، و اجرای اسکریپت متوقف می شود.

هر دوی این توابع برای ایجاد توابع، هدرها، فوترها یا عناصری که در چندین صفحه استفاده مجدد دارند استفاده می شوند. این کار باعث راحت شدن طراحی سایت و تغییرات بعدی و صرفه جویی در وقت می شوند. بدین معنی که می توانیم یک فایل هدر، فوتر یا منوی استاندارد ایجاد کرده و آنرا در همه صفحات وب استفاده کنیم. هر وقت نیاز به تغییر هدر بود کافی است فایل هدر را اصلاح کنیم یا هرگاه صفحه جدیدی به سایت اضافه شد کافی است فایل حاوی منو تغییر کند (به جای تغییر گزینه ها در تک تک صفحات وب سایت)

تابع `include()`

تابع `include()` تمام محتوای فایل تعیین شده را در فایل جاری درج می کند. اگر خطایی اتفاق افتد، تابع `include()` یک `warning` تولید کرده و اجرای اسکریپت قطع نخواهد شد.

مثال 1

یک فایل `header` استاندارد به نام `header.php` در نظر بگیرید. برای ضمیمه کردن فایل `header` در صفحه تابع `include()` را بصورت زیر استفاده می کنیم:

```
<html>
<body>

<?php include("header.php"); ?>
<h1>Welcome to my home page!</h1>
```

```
<p>Some text.</p>
```

```
</body>
```

```
</html>
```

مثال 2

فرض کنید یک فایل منوی استاندارد به نام `menu.php` داریم، که باید در همه صفحات استفاده شود:

```
<a href="/default.php">Home</a>
```

```
<a href="/tutorials.php">Tutorials</a>
```

```
<a href="/references.php">References</a>
```

```
<a href="/examples.php">Examples</a>
```

```
<a href="/about.php">About Us</a>
```

```
<a href="/contact.php">Contact Us</a>
```

همه صفحات موجود در وب سایت باید فایل منو را ضمیمه کنند بصورت زیر:

```
<html>
```

```
<body>
```

```
<div class="leftmenu">
```

```
<?php include("menu.php"); ?>
```

```
</div>
```

```
<h1>Welcome to my home page.</h1 >
```

```
<p>Some text.</p>
```

```
</body>
```

```
</html>
```

اگر نگاهی به کد منبع صفحه نشان داده در مرورگر بیندازید تگ های زیر را خواهید دید.

```
<html>
```

```
<body>
```

```
<div class="leftmenu">
```

```
<a href="/default.php">Home</a>
```

```
<a href="/tutorials.php">Tutorials</a>
```

```
<a href="/references.php">References</a>
```

```
<a href="/examples.php">Examples</a>
```

```
<a href="/about.php">About Us</a>
<a href="/contact.php">Contact Us</a>
</div>
```

```
<h1>Welcome to my home page!</h1>
<p>Some text.</p>
```

```
</body>
</html>
```

تابع require()

تابع require() شبیه include() است بجز در روش برخورد با خطاها

اگر خطایی اتفاق بیفتد تابع include() تولید warning کرده و اجرا ادامه پیدا می کند ولی تابع require تولید خطای غیرقابل چشم پوشی کرده و اجرای اسکریپت متوقف می شود.

مثالی از بروز خطا در تابع include()

```
<html>
<body>

<?php
include("wrongFile.php");
echo "Hello World!";
?>

</body>
</html>
```

پیغام خطا:

```
Warning: include(wrongFile.php) [function.include]:
failed to open stream:
No such file or directory in C:\home\website\test.php on line 5
```

```
Warning: ()include [function.include]:
Failed opening 'wrongFile.php' for inclusion
(include_path='.;C:\php5\pear')
in C:\home\website\test.php on line 5
```

Hello World!

توجه داشته باشید که دستور `echo` اجرا شده است! یعنی تولید `warning` اجرای اسکریپت را متوقف نکرده است.

مثالی از بروز خطا در تابع `require()`

اکنون اجازه دهید مثال مشابهی را با تابع `require()` اجرا کنیم.

```
<html>
<body>

<?php
require("wrongFile.php");
echo "Hello World!";
?>

</body>
</html>
```

پیغام خطا:

Warning: require(wrongFile.php) [function.require]:

failed to open stream:

No such file or directory in C:\home\website\test.php on line 5

Fatal error: require() [function.require]:

Failed opening required 'wrongFile.php'

(include_path='.;C:\php5\pear')

in C:\home\website\test.php on line 5

دستور `echo` اجرا نشده است زیرا اجرای اسکریپت پس از وقوع خطا متوقف شده است.

توصیه می شود از همیشه از تابع `require()` به جای `include()` استفاده کنید تا اجرای اسکریپت پس از وقوع خطا ادامه پیدا نکند.

10- آرایه ها

یک آرایه چندین مقدار را در یک متغیر ذخیره می کند.

آرایه چیست؟

یک متغیر یک ناحیه ذخیره سازی است که یک مقدار عددی یا متنی را نگه می دارد. مسئله اصلی این است که یک متغیر تنها یک مقدار را نگهداری می کند.

یک آرایه متغیر مخصوصی است که می تواند چندین مقدار را در یک متغیر ذخیره کند.

اگر لیستی از اقلام داشته باشید (مثلاً یک لیست از اسامی ماشین ها) ، ذخیره کردن ماشین ها در متغیرها بصورت تکی شبیه زیر خواهد بود:

```
$cars1="Saab";
$cars2="Volvo";
```

```
$cars3="BMW";
```

با وجود این نوشتن حلقه های تکرار و پیدا کردن یک ماشین از بین ماشین ها چه خواهید کرد؟ حالا اگر به جای 3 ماشین 300 ماشین داشته باشید چه؟

بهترین راه حل استفاده از یک آرایه است!

یک آرایه می تواند همه مقادیر را تحت یک نام ذخیره کرده و برای مراجعه به هر مقدار از همان نام متغیر استفاده می کنید. هر عنصر در آرایه اندیس خودش را دارد و براحتی می تواند مورد دسترسی قرار گیرد.

در PHP، سه نوع آرایه وجود دارد:

- **آرایه شمارشی** - آرایه ای با اندیس های عددی
- **آرایه انجمنی** - آرایه ای که یک کلید شناسایی (ID key) به هر مقدار تخصیص داده می شود.
- **آرایه چند بعدی** - آرایه ای که شامل یک یا چند آرایه است.

آرایه های شمارشی

یک آرایه شمارشی هر عنصر آرایه را با یک اندیس عددی ذخیره می کند. در روش برای ایجاد ژرلیه شمارشی وجود دارد.

1. در مثال زیر اندیس ها بصورت خودکار به عناصر داده خواهد شده (اندیس ها از صف شروع خواهند شد):

```
$cars=array("Saab","Volvo","BMW","Toyota");
```

2. در مثال زیر اندیس ها را خود ما بصورت دستی تعیین کرده ایم:

```
$cars[0]="Saab";
```

```
$cars[1]="Volvo";
```

```
$cars[2]="BMW";
```

```
$cars[3]="Toyota";
```

مثال

در مثال زیر به مقادیر متغیر با استفاده از نام آرایه و اندیس دسترسی پیدا می کنیم:

```
<?php
```

```
$cars[0]="Saab";
```

```
$cars[1]="Volvo";
```

```
$cars[2]="BMW";
```

```
$cars[3]="Toyota";
```

```
echo $cars[0] . " and " . $cars[1] . " are Swedish cars.";
```

```
?>
```

خروجی کد فوق بصورت زیر است:

Saab and Volvo are Swedish cars.

آرایه های انجمنی

در یک آرایه انجمنی (associative)، یک کلید یکتا به هر مقدار تخصیص داده شده است. وقتی داده هایی در مورد اسامی خاصی ذخیره می شود، آرایه شمارشی روشی خوبی نیست. با استفاده از آرایه های انجمنی می توانیم از اسامی به عنوان کلید استفاده کرده و مقادیر را به آنها نسبت دهیم.

مثال 1:

در این مثال از یک آرایه برای نگهداری سن افراد مختلف استفاده شده است:

```
$ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);
```

مثال 2:

این مثال مشابه مثال 1 است ، ولی روش دیگری برای ایجاد آرایه نشان می دهد:

```
$ages['Peter'] = "32";
$ages['Quagmire'] = "30";
$ages['Joe'] = "34";
```

کلیدهای یکتا می توانند در اسکرپت استفاده شوند:

```
<?php
$ages['Peter'] = "32";
$ages['Quagmire'] = "30";
$ages['Joe'] = "34";

echo "Peter is " . $ages['Peter'] . " years old.";
?>
```

خروجی کد فوق عبارت است از:

Peter is 32 years old.

آرایه های چندبعدی

در آرایه چندبعدی، هر عنصر در آرایه اصلی می تواند خودش یک آرایه باشد. و هر عنصر در زیر-آرایه نیز می تواند یک آرایه باشد و به همین ترتیب.

مثال 1

در این مثال یک آرایه چندبعدی ایجاد کرده ایم:

```
$families = array
("Griffin" =>array ( "Peter", "Lois", "Megan" ),
 "Quagmire"=>array ( "Glenn" ),
 "Brown" =>array ( "Cleveland", "Loretta", "Junior" ) );
```

آرایه فوق را می توان بصورت زیر در نظر گرفت:

```
Array
(
 [Griffin] => Array
 (
 [0] => Peter
 [1] => Lois
 [2] => Megan
 )
 [Quagmire] => Array
 (
```



```
[0] => Glenn
)
[Brown] => Array
(
[0] => Cleveland
[1] => Loretta
[2] => Junior
)
)
```

مثال 2

اجازه دهید یکی از مقادیر آرایه فوق را نمایش دهیم:

```
echo "Is " . $families['Griffin'][2] .
" a part of the Griffin family?";
```

:The code above will output

Is Megan a part of the Griffin family?

حلقه تکرار foreach

حلقه تکرار foreach برای تکرار بر اساس آرایه ها استفاده می شود.
گرامر

```
foreach ($array as $value)
{
    code to be executed;
}
```

در هر بار اجرای حلقه مقدار جاری از آرایه به **\$value** نسبت داده می شود (و اشاره گر آرایه یک خانه به جلو حرکت می کند). بنابراین در تکرار بعدی عنصر بعدی آرایه بدست خواهد آمد.

مثال

مثال زیر نشان می دهد که چگونه از حلقه تکرار برای چاپ عناصر آرایه داده شده استفاده شده است.

```
<html>
<body>

<?php
$x=array("one","two","three");
foreach ($x as $value)
{
    echo $value . "<br />";
```

```

}
?>

</body>
</html>

```

خروجی:

```

one
two
three

```

11- کار با فرم ها

متغیرهای `$_GET` و `$_POST` برای بازیابی اطلاعات از فرم ها استفاده می شوند، مشابه ورودی کاربر

چیزی که خیلی مهم است این است که وقتی با فرم های `HTML` و `PHP` سرو کار داریم ، هر عنصر در صفحه `HTML` بطور خودکار در اسکریپت `PHP` در دسترس خواهد بود. (به اسکریپت ارسال خواهد شد)

مثال

مثال زیر یک فرم `HTML` با دو فیلد ورودی و یک دکمه `submit` برای ارسال فرم را نشان می دهد:

```

<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="fname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>

</body>
</html>

```

هر وقت کاربر فرم بالا را پر کرده و روی دکمه `submit` کلیک کرد، اطلاعات فرم به فایل `PHP` با نام `welcome.php` ارسال خواهد شد:

فایل `welcome.php` بصورت زیر خواهد بود.

```

<html>
<body>

Welcome <?php echo $_POST["fname"]; ?>!<br />
You are <?php echo $_POST["age"]; ?> years old.

```

</body>

</html>

خروجی می تواند بصورت زیر نمایش داده شود:

Welcome

John!

You are 28 years old.

عملکرد متغیرهای \$_GET و \$_POST در ادامه توضیح داده شده است.

اعتبارسنجی فرم

ورودی های کاربر باید از نظر درستی بررسی شوند، این کار می تواند توسط مرورگر انجام شود (با استفاده از برنامه نویسی سمت مشتری) که در این صورت سریعتر بوده و بار سرور را کاهش خواهد داد. اگر داده های وارد شده توسط کاربر قرار است در بانک اطلاعاتی درج شود باید در طرف سرور نیز اعتبار ورودی ها بررسی شود. بهترین کار برای اعتبارسنجی ، ارسال فرم به خود همان صفحه است تا صفحه دیگری. در این صورت در همان صفحه فرم می توان پیغام های خطا را نمایش داد.

12- ارسال فرم به روش GET

آرایه \$_GET برای جمع آوری اطلاعات فرم ارسال شده با متد GET استفاده می شود.

اطلاعات ارسال شده از فرم با متد GET برای هر کسی قابل رویت هستند. (این اطلاعات در نوار آدرس مرورگر نشان داده می شوند) ، و طول اطلاعات ارسالی محدود است (حداکثر 100 کاراکتر)

مثال

<form action="welcome.php" method="get">

Name: <input type="text" name="fname" />

Age: <input type="text" name="age" />

<input type="submit" />

</form>

وقتی روی دکمه submit کلیک شود URL بصورت زیر به سرور دهنده فرستاده خواهد شد(متغیرهای فرم در ادامه آدرس صفحه قرار خواهند گرفت):

http://www.w3schools.com/welcome.php?fname=Peter&age=37

فایل welcome.php اکنون می تواند از آرایه \$_GET برای جمع آوری اطلاعات فرم استفاده کند . (فیلدهای فرم بصورت خودکار تبدیل به کلیدهای آرایه \$_GET خواهند شد):

Welcome <?php echo \$_GET["fname"]; ?>.

You are <?php echo \$_GET["age"]; ?> years old!

چه موقع از متد GET استفاده می کنیم؟

وقتی از متد GET در فرم های HTML استفاده می کنیم همه متغیرها و مقادیر آنها در URL نمایش داده می شود. **نکته:** این متد نمی تواند برای اطلاعات حساسی مثل کلمه عبور استفاده شود. با وجود این چون اطلاعات در نوار آدرس نشان داده می شود می تواند در علامت گذاری صفحات و مرور آنها استفاده شود. **نکته:** این متد نمی تواند برای ارسال اطلاعات بلند استفاده شود و طول اطلاعات نباید از 100 کاراکتر تجاوز کند

13- ارسال فرم با روش POST

آرایه پیش ساخته \$_POST برای جمع آوری مقادیر فرم ارسال شده با شیوه post استفاده می شود.

اطلاعات ارسال شده از فرم با متد post بصورت مخفی ارسال شده و می توانند دارای طول نامحدود باشند.
نکته: طول داده های ارسال شده با متد post حداکثر 8 مگابایت است ولی با تنظیم متغیر post_max_size در فایل پیکربندی php.ini قابل تغییر است.
 مثال

```
<form action="welcome.php" method="post">
Name: <input type="text" name="fname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
```

وقتی کاربر روی دکمه submit کلیک کند URL زیر درخواست می شود:

http://www.w3schools.com/welcome.php

فایل welcome.php می تواند از آرایه \$_POST برای جمع آوری اطلاعات فرم استفاده کند (نام فیلدهای فرم بصورت خودکار تبدیل به کلیدهای آرایه \$_POST خواهند شد):

```
Welcome <?php echo $_POST["fname"]; ?>!<br />
You are <?php echo $_POST["age"]; ?> years old.
```

چه موقع از method=post استفاده می کنیم:

اطلاعاتی که نباید به دیگران نشان داده شده و یا دارای طول زیادی هستند با این شیوه ارسال می شوند.
 با وجود این چون متغیرها در URL نشان داده نمی شوند برای تشخیص و علامتگذاری صفحات مناسب نخواهد بود.

آرایه \$_REQUEST

آرایه پیش ساخته \$_REQUEST شامل محتوای هر \$_POST ، \$_GET و \$_COOKIE می باشد.
 آرایه \$_REQUEST نیز می تواند برای جمع آوری اطلاعات ارسال شده با متدهای GET و POST استفاده شود.
 مثال

```
Welcome <?php echo $_REQUEST["fname"]; ?>!<br />
You are <?php echo $_REQUEST["age"]; ?> years old.
```

14- آپلود فایل ها

با استفاده از PHP امکان آپلود فایل ها به سرورس دهنده وجود دارد.

ایجاد فرم برای آپلود فایل

برای آپلود فایل ها از فرم ها استفاده می شود.
 به فرم HTML زیر برای آپلود فایل دقت کنید:

```
<html>
```

```

<body>

<form action="upload_file.php" method="post"
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file" />
<br />
<input type="submit" name="submit" value="Submit" />
</form>

</body>
</html>

```

در مورد فرم فوق به نکات زیر توجه کنید.

- خاصیت **enctype** فرم تعیین می کند که چه نوع اطلاعاتی باید ارسال شود. "multipart/form-data" وقتی استفاده می شود که فرم برای آپلود فایل استفاده شود.
 - خاصیت **type="file"** از تگ **<input>** تعیین می کند که ورودی باید بصورت فایل باشد. بعنوان مثال وقتی فرم فوق در مرورگر نمایش داده شود یک دکمه **"browse"** برای انتخاب فایل نشان داده خواهد شد.
- توجه :** اجازه دادن به کاربر برای ارسال فایل یک ریسک امنیتی بزرگی است. تنها به کاربران شناخته شده و مطمئن اجازه این کار را بدهید.

ایجاد اسکریپت آپلود

فایل "upload_file.php" دستوراتی برای آپلود فایل دارا می باشد :

```

<?php
if ($_FILES["file"]["error"] > 0)
{
    echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
else
{
    echo "Upload: " . $_FILES["file"]["name"] . "<br />";
    echo "Type: " . $_FILES["file"]["type"] . "<br />";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
?>

```

با استفاده از آرایه سراسری **\$_FILES** می توان فایل های ارسال شده به سرور را آپلود کرد.

اولین پارامتر در این آرایه نام ورودی فرم است و دومین اندیس می تواند یکی از اندیس های "name" ، "type" ، "size" ، "tmp_name" یا "error" باشد. مشابه زیر:

متغیر	توصیف
\$_FILES["file"]["name"]	نام فایل آپلود شده
\$_FILES["file"]["size"]	اندازه فایل آپلود شده بر حسب بایت
\$_FILES["file"]["type"]	نوع فایل آپلود شده
\$_FILES["file"]["tmp_name"]	نام کپی موقتی فایل که بر روی سرورس دهنده ذخیره شده است
\$_FILES["file"]["error"]	کد خطای تولید شده در حین آپلود فایل

مثال فوق یک روش ساده و ابتدایی برای آپلود فایل را نشان می دهد. به دلایل امنیتی باید محدودیت هایی تعیین شود و کاربر را برای آپلود فایل های خاصی محدود کنیم.

محدودیت ها برای آپلود

در اسکریپت زیر محدودیت هایی برای آپلود فایل تعیین شده است. کاربر تنها می تواند فایل های gif. تا jpeg. و با اندازه کمتر از 20 کیلوبایت را آپلود کند:

```
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
if ($_FILES["file"]["error"] > 0)
{
echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
}
else
{
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
}
else
{
```

```
echo "Invalid file";
}
?>
```

توجه: در مرورگر IE برای تشخیص فایل های .jpg نوع آن باید pjpeg نوشته شود ، برای مرورگر Firefox باید jpeg.

ذخیره کردن فایل آپلود شده

مثال های فوق یک کپی موقت از فایل آپلود شده در پوشه موقت PHP روی سرور ایجاد می کنند. کپی موقت بلافاصله پس از پایان یافت اجرای اسکریپت از بین خواهد رفت. برای ذخیره کردن فایل آپلود شده ، باید آن را به مکان دیگری کپی کنیم:

```
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg")))
&& ($_FILES["file"]["size"] < 20000))
{
if ($_FILES["file"]["error"] > 0)
{
echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
}
else
{
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

if (file_exists("upload/" . $_FILES["file"]["name"]))
{
echo $_FILES["file"]["name"] . " already exists. ";
}
else
{
move_uploaded_file($_FILES["file"]["tmp_name"],
"upload/" . $_FILES["file"]["name"]);
echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
}
```

```

}
}
}
else
{
echo "Invalid file";
}
?>

```

اسکرپت فوق بررسی می کند که فایل وجود دارد یا خیر، اگر موجود نبود آنرا را در پوشه مشخص شده کپی می کند.
توجه: این مثال فایل را در پوشه جدیدی بنام "upload" کپی می کند.

15- اتصال به بانک اطلاعاتی

استفاده از بانک اطلاعاتی MySQL در PHP خیلی معمول است

یک اتصال برای ارتباط با بانک اطلاعاتی MySQL ایجاد کنید
قبل از اینکه بتوانید به داده های بانک اطلاعاتی دسترسی پیدا کنید، باید یک اتصال به بانک اطلاعاتی ایجاد کنید.
در PHP، این کار با تابع `mysql_connect()` انجام می شود. گرامر

`mysql_connect(servername,username,password);`

پارامتر	کاربرد
servername	اختیاری است نام سرویس دهنده بانک اطلاعاتی است و پیشفرض آن "localhost:3306" است
username	اختیاری است، نام کاربر بانک اطلاعاتی است و بطور پیش فرض نام کاربر جاری استفاده کننده از سرور است
password	اختیاری است، کلمه عبور کاربر است، پیش فرض آن "" است .

توجه: پارامترهای بیشتری برای آن وجود دارد، ولی موارد فوق اهمیت بیشتری دارد.
مثال

در زیر مثالی ارائه شده است که اتصال را برای استفاده های بعدی در متغیر `$con` ذخیره می کند. بخش "die" در صورتی اجرا می شود که ایجاد ارتباط با مشکل مواجه گردد:

```

<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}

```



```
// some code
?>
```

با نصب MySQL بطور پیشفرض کاربر root با کلمه عبور "" برای آن در ایجاد می شود.

```
$con = mysql_connect("localhost","root","");
```

بستن یک اتصال

هنگام پایان یافتن اجرای اسکریپت بصورت خودکار، اتصال بسته خواهد شد. برای بستن اتصال قبل از پایان یافتن اسکریپت، از تابع `mysql_close()` استفاده می کنیم:

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
```

```
// some code
```

```
mysql_close($con);
?>
```

16- دسترسی به رکوردهای بانک اطلاعاتی

دستور `SELECT` برای انتخاب داده ها از بانک اطلاعاتی استفاده می شود.

انتخاب داده ها از جدول بانک اطلاعاتی

دستور `SELECT` برای انتخاب داده های جدول از بانک اطلاعاتی استفاده می شود.
گرامر

```
SELECT column_name(s)
FROM table_name
```

برای وادار کردن PHP به اجرای دستور فوق باید از تابع `mysql_query()` استفاده کنیم. این تابع برای ارسال پرس و جو یا دستور به یک اتصال MySQL استفاده می گردد.

مثال

مثال زیر همه داده های ذخیره شده در جدول "Persons" را انتخاب می کند:

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
```

```

{
die('Could not connect: ' . mysql_error());
}

mysql_select_db("my_db", $con);

$result = mysql_query("SELECT * FROM Persons");

while($row = mysql_fetch_assoc ($result))
{
echo $row['FirstName'] . " " . $row['LastName'];
echo "<br />";
}

mysql_close($con);
?>

```

در مثال فوق داده های برگشت داده شده از تابع `mysql_query()` در متغیر `$result` ذخیره شده است. سپس، از تابع `mysql_fetch_assoc` استفاده می کنیم تا اولین سطر از مجموعه رکورد بازیابی شده را بصورت یک آرایه بدست آوریم. هربار فراخوانی `mysql_fetch_assoc()` ، سطر بعدی از رکورد ست را به ما می دهد. یک حلقه `while` می تواند همه رکوردهای رکورد ست را استخراج کند. برای چاپ مقادیر هر سطر، از متغیر آرایه `row$` بصورت `$row['LastName']` ، `$row['FirstName']` و ... استفاده می کنیم. خروجی کد فوق بصورت زیر خواهد بود:

```

Peter                               Griffin
Glenn Quagmire

```

نمایش نتیجه در یک جدول HTML

مثال زیر داده هایی مشابه مثال فوق انتخاب می کند، ولی داده ها را در یک جدول HTML نشان خواهد داد:

```

<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}

```

```
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Persons");
echo "<table border='1' align='center' >
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>";

while($row = mysql_fetch_assoc ($result))
{
echo "<tr>";
echo "<td>" . $row['FirstName'] . "</td>";
echo "<td>" . $row['LastName'] . "</td>";
echo "</tr>";
}
echo "</table>";

mysql_close($con);
?>
```

خروجی اسکرپت فوق بصورت زیر خواهد بود:

Lastname	Firstname
Quagmire	Glenn
Griffin	Peter

بخش Where در دستور Select

بخش WHERE زمانی استفاده می شود که قصر داریم تنها رکورد های دارای شرط یا شرایط مورد نظر را بازیابی کنیم. گرامر

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator value
```

مثال

در مثال زیر تمام سطرهایی از جدول "Persons" انتخاب شده اند که برای آنها "FirstName='Peter'" است.

```
<?php
```

```

$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

mysql_select_db("my_db", $con);

$result = mysql_query("SELECT * FROM Persons
WHERE FirstName='Peter'");

while($row = mysql_fetch_assoc($result))
{
    echo $row['FirstName'] . " " . $row['LastName'];
    echo "<br />";
}
?>

```

17- درج رکورد در بانک اطلاعاتی

دستور **INSERT INTO** برای درج رکورد های جدید در جدول استفاده می شود.

داده های جدیدی در جدول بانک اطلاعاتی درج کنید

دستور **INSERT INTO** برای درج رکورد های جدید در جدول بانک اطلاعاتی استفاده می شود.
گرامر

برای نوشتن دستور **INSERT INTO** دو روش وجود دارد.

در اولین فرم، نام ستون هایی که برای آنها داده درج می کنیم را ذکر نمی کنیم، و تنها مقادیر را می نویسیم:

```

INSERT INTO table_name
VALUES (value1, value2, value3,...)

```

در فرم دوم هم نام ستونها و هم مقادیر آنها نوشته می شود:

```

INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)

```

برای وادار کردن PHP به اجرای دستور فوق باید از تابع `mysql_query()` استفاده کنیم. این تابع برای ارسال پرس و جو یا دستور به یک اتصال MySQL استفاده می گردد.

فرض می کنیم ، جدولی به نام "Persons" با سه ستون "Age", "FirstName", "LastName" داشته باشیم.
در مثال زیر دو رکورد جدید به جدول "Persons" اضافه می کنیم:

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

mysql_select_db("my_db", $con);

mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('Peter', 'Griffin', '35')");

mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('Glenn', 'Quagmire', '33')");

mysql_close($con);
?>
```

درج اطلاعات از یک فرم در بانک اطلاعاتی

اکنون یک فرم HTML ایجاد می کنیم که برای درج رکورد جدید در جدول "Persons" مورد استفاده قرار گیرد.
در زیر فرم HTML را داریم:

```
<html>
<body>

<form action="insert.php" method="post">
Firstname: <input type="text" name="firstname" />
Lastname: <input type="text" name="lastname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
```

```
</body>
```

```
</html>
```

وقتی کاربر روی دکمه "submit" در فرم HTML فوق کلیک کند، فرم به برنامه "insert.php" ارسال خواهد شد. فایل "insert.php" به بانک اطلاعاتی اتصال پیدا می کند، و مقادیر فرم را با استفاده از متغیرهای \$_POST بدست می آورد.

سپس، تابع mysql_query() دستور INSERT INTO را اجرا می کند، و یک رکورد جدید به جدول "Persons" اضافه خواهد شد.

در زیر صفحه "insert.php" را می بینید:

```
<?php
```

```
$con = mysql_connect("localhost","root","");
```

```
if (!$con)
```

```
{
```

```
die('Could not connect: ' . mysql_error());
```

```
}
```

```
mysql_select_db("my_db", $con);
```

```
$sql="INSERT INTO Persons (FirstName, LastName, Age)
```

```
VALUES
```

```
('$_POST[firstname]','$_POST[lastname]','$_POST[age]');
```

```
if (!mysql_query($sql,$con))
```

```
{
```

```
die('Error: ' . mysql_error());
```

```
}
```

```
echo "1 record added";
```

```
mysql_close($con)
```

```
?>
```

18- اصلاح رکوردهای بانک اطلاعاتی

دستور UPDATE برای اصلاح داده های يك جدول استفاده مي شود.

اصلاح داده های يك بانک اطلاعاتی

دستور UPDATE برای بروزرسانی رکوردهای موجود در يك جدول استفاده مي شود.

گرامر

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

توجه: به بخش WHERE در گرامر دستور UPDATE توجه کنید. بخش WHERE تعیین می کند که چه رکورد یا رکوردهایی باید اصلاح شوند. اگر این بخش نوشته نشود کلیه رکوردها بروز خواهند شد! برای وادار کردن PHP به اجرای دستور فوق باید از تابع mysql_query() استفاده کنیم. این تابع برای ارسال پرس و جو یا دستور به یک اتصال MySQL استفاده می گردد.

مثال

در قسمتهای قبلی با جدول "Persons" آشنا شدید. فرض کنید این جدول دارای رکوردهای زیر باشد:

Age	LastName	FirstName
35	Griffin	Peter
33	Quagmire	Glenn

مثال زیر بعضی از رکوردهای جدول "Persons" را بروز می کند.

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

mysql_select_db("my_db", $con);

mysql_query("UPDATE Persons SET Age = '36'
WHERE FirstName = 'Peter' AND LastName = 'Griffin'");

mysql_close($con);
?>
```

بعد از اجرای برنامه فوق رکوردهای جدول "Persons" بصورت زیر خواهد شد.

Age	LastName	FirstName
36	Griffin	Peter
33	Quagmire	Glenn

یک متغیر `session` در `PHP` برای نگهداری اطلاعات درباره کاربر و یا تغییر تنظیمات جلسه کاربر استفاده می شود. متغیرهای `session` اطلاعاتی در مورد یک کاربر نگهداری کرده و در همه صفحات یک برنامه در دسترس هستند.

متغیرهای `session` در `PHP`

وقتی شما با یک برنامه کاربردی کار می کنید، آنرا را باز می کنید، یکسری تغییرات انجام داده و سپس آنرا می بندید. به این فرآیند یک `session` یا جلسه کاری می گویند. کامپیوتر می داند که شما کی هستید. همچنین می داند که چه وقت برنامه را شروع کرده و چه وقت پایان داده اید. ولی در اینترنت یک مشکل وجود دارد؛ وب سرور نمی داند شما کی هستید و چه می خواهید انجام دهید. چون آدرس های `HTTP` حال را نگه نمی دارند ، در واقع پروتکل `HTTP` بصورت `state-less` عمل می کند.

`session` ها در `PHP` مشکل را حل می کنند و به شما اجازه می دهند تا اطلاعاتی را برای استفاده های بعدی ذخیره کنید (اطلاعاتی مثل نام کاربری، اقلام خریداری شده و غیره) . با این حال اطلاعات `session` موقتی هستند و خارج شدن شما از وب سایت از بین خواهند رفت. اگر می خواهید اطلاعات به صورت دائمی باقی بمانن باید آنها را در بانک اطلاعاتی ذخیره کنید.

`session` ها با ایجاد یک شناسه یکتا (`UID`) برای هر مراجعه کننده و ذخیره متغیرها بر اساس `UID` کار می کند. `UID` خودش در یک کوکی ذخیره شده و یا با `URL` جابجا می شود.

شروع بکار `Session` در `PHP`

قبل از اینکه بتوانید در `session` تان اطلاعاتی ذخیره کنید، باید آنرا شروع کنید.

توجه : تابع `session_start()` باید قبل از تگ `<html>` قرار گیرد:

```
<?php session_start(); ?>
```

```
<html>
```

```
<body>
```

```
</body>
```

```
</html>
```

کد فوق `session` مربوط به کاربر را در سرور ثبت نام خواهد کرد، به شما اجازه خواهد داد تا اطلاعات کاربر را ذخیره کنید و یک `UID` به `session` کاربر اختصاص خواهد داد.

ذخیره سازی متغیرهای `session`

روش صحیح برای ذخیره و بازیابی متغیرهای `session` استفاده از متغیر `$_SESSION` است:

```
<?php
```

```
session_start();
```

```
// store session data
```



```
$_SESSION['views']=1;
?>
```

```
<html>
<body>
```

```
<?php
//retrieve session data
echo "Pageviews=". $_SESSION['views'];
?>
```

```
</body>
</html>
```

خروجی:

```
Pageviews=1
```

در مثال زیر یک شمانده تعداد مراجعات به صفحه ساده ایجاد کرده ایم. تابع `isset()` بررسی می کند که متغیر "views" از قبل وجود دارد و ست شده است. اگر "views" از قبل وجود داشته باشد، آنرا یک واحد اضافه کرده و گرنه یک متغیر "views" ایجاد کرده و آنرا مقدار 1 می دهد.

```
<?php
session_start();

if(isset($_SESSION['views']))
$_SESSION['views']=$_SESSION['views']+1;
else
$_SESSION['views']=1;
echo "Views=". $_SESSION['views'];
?>
```

از بین بردن session

برای از بین بردن بعضی از داده های session می توان از تابع `unset()` یا تابع `session_destroy()` استفاده کرد.

تابع `unset()` برای از بین بردن متغیرهای مشخصی از session استفاده می شود:

```
<?php
unset($_SESSION['views']);
```

```
?>
```

می توانید برای از بین بردن کامل `session` تابع `session_destroy()` را فراخوانی کنید:

```
<?php
session_destroy();
?>
```

توجه: `session_destroy()` کل `session` شما را ریست کرده و همه داده های ذخیره شده در `session` را از دست خواهید داد.

20- کوکی ها

کوکی معمولاً برای شناسایی کاربر استفاده می شود.

کوکی چیست؟

کوکی اغلب برای شناسایی کاربر استفاده می شود. یک کوکی فایل کوچکی است که سرور آنرا بر روی کامپیوتر کاربر قرار می دهد. هر وقت همان کامپیوتر از طریق مرورگر درخواستی را به سرور فرستاد، کوکی را نیز همراه آن خواهد فرستاد. با استفاده از PHP می توانید مقادیر کوکی ها را ذخیره و یا بازیابی کنید.

چطور کوکی را ایجاد کنیم؟

تابع `setcookie()` برای تنظیم یک کوکی استفاده می شود.

توجه: تابع `setcookie()` باید همیشه قبل از تگ `<html>` قرار داشته باشد.
گرامر

```
setcookie(name, value, expire, path, domain);
```

مثال 1

در مثال زیر، می خواهیم یک کوکی با نام "user" ایجاد کنیم و مقدار "Porter Alex" را به آن نسبت دهیم. ما همچنین مشخص کرده ایم که کوکی پس از گذشت یک ساعت از بین برود:

```
<?php
setcookie("user", "Alex Porter", time()+3600);
?>
```

```
<html>
```

```
.....
```

توجه: مقدار کوکی هنگام ارسال آن بصورت خودکار کدگذاری شده (URLencoded)، و هنگام دریافت بصورت خودکار کدگشایی می گردد.

مثال 2

شما همچنین می توانید یک زمان انقضای کوکی را به صورت دیگری محاسبه کرده و برحسب ثانیه بدهید.

```
<?php
$expire=time()+60*60*24*30;
setcookie("user", "Alex Porter", $expire);
?>
```

```
<html>
```

```
.....
```

در مثال فوق زمان انقضای کوکی به يك ماه تنظیم شده است (يك ماه مساوي است با 30*24*60*60 ثانیه)

چطور مقدار يك کوکی را بازیابی کنیم؟

در PHP از متغیر `$_COOKIE` برای بازیابی مقدار یک کوکی استفاده می شود.

در مثال زیر، یک مقدار از کوکی به نام "user" را بازیابی کرده و آنرا در یک صفحه نمایش می دهد.

```
<?php
```

```
// Print a cookie
```

```
echo $_COOKIE["user"];
```

```
// A way to view all cookies
```

```
print_r($_COOKIE);
```

```
?>
```

در مثال زیر از تابع `isset()` برای تشخیص اینکه یک کوکی تنظیم شده یا خیر استفاده کرده ایم.

```
<html>
```

```
<body>
```

```
<?php
```

```
if (isset($_COOKIE["user"]))
```

```
    echo "Welcome " . $_COOKIE["user"] . "!<br />";
```

```
else
```

```
    echo "Welcome guest!<br />";
```

```
?>
```

```
</body>
```

```
</html>
```

چگونه یک کوکی را حذف کنیم؟

برای حذف یک کوکی، باید تاریخ انقضای آنرا به گذشته تغییر دهید.

مثال حذف:

```
<?php
```

```
// set the expiration date to one hour ago
```

```
setcookie("user", "", time()-3600);
```

```
?>
```

اگر مرورگر کوکی ها را پشتیبانی نکند چه باید کرد؟

اگر برنامه کاربردی شما با مرورگرهایی سروکار دارد که کوکی ها را پشتیبانی نمی کنند، باید از روشی غیر از کوکی برای انتقال اطلاعات عبور از یک صفحه به صفحه دیگر کمک بگیرید. یک روش برای این کار استفاده از فرم ها است.

در زیر ما ورودی کاربر را هر وقت روی دکمه "submit" کلیک شود، به صفحه "welcome.php" می فرستیم.

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>

</body>
</html>
```

مقادیر در صفحه "welcome.php" بصورت زیر بدست می آیند:

```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?>.<br />
You are <?php echo $_POST["age"]; ?> years old.

</body>
</html>
```

21- ارسال نامه الکترونیکی

PHP به شما اجازه می دهد تا e-mail هایتان را مستقیماً از يك اسکریپت ارسال کنید.

تابع mail() در PHP

تابع mail() در PHP برای ارسال نامه های الکترونیکی از داخل يك اسکریپت استفاده می شود.
گرامر

mail(to,subject,message,headers,parameters)

پارامتر	شرح
to	لازم. گیرنده /گیرندگان نامه را مشخص میکند.

subject	لازم. موضوع ایمیل را مشخص می کند. توجه: این کاراکتر نباید حاوی خط جدید (new line) باشد.
message	لازم. پیامی را که باید ارسال شود، مشخص می کند. خطوط می توانند با LF(\n) ر از هم جدا شوند و تعداد کاراکترهای هر خط نباید از 70 تجاوز کند.
headers	اختیاری. هدرهای اضافی را مشخص می کند مثل from, Cc, Bcc. هدرهای اضافی می توانند با CRLF (\r\n) از هم جدا شوند.
parameters	اختیاری. پارامترهای اضافی مثل نام برنامه ارسال email را مشخص می کند.

توجه: برای اینکه توابع mail در دسترس قرار بگیرند، PHP به نصب و بکارگیری یک سیستم email نیاز دارد. برنامه استفاده شده برای این کار باید در فایل تنظیمات php.ini تعیین شده باشد.

یک E-mail ساده

ساده ترین روش برای ارسال یک email با PHP ارسال یک نامه متنی است. در مثال زیر، ابتدا متغیرهایی تعریف می کنیم (\$to, \$subject, \$message, \$from, \$headers)، سپس از متغیرها در تابع mail() برای ارسال ایمیل استفاده می کنیم:

```
<?php
$to = "someone@example.com";
$subject = "Test mail";
$message = "Hello! This is a simple email message.";
$from = "someone@example.com";
$headers = "From: $from";
mail($to,$subject,$message,$headers);
echo "Mail Sent.";
?>
```

فرم ارسال Email

با استفاده از PHP، می توانید یک فرم دارای فیدبک برای سایت ایجاد کنید. مثال زیر یک پیام متنی به آدرس پست الکترونیکی تعیین شده ارسال می کند:

```
<html>
<body>

<?php
```

```

if (isset($_REQUEST['email']))
//if "email" is filled out, send email
{
//send email
$email = $_REQUEST['email'] ;
$subject = $_REQUEST['subject'] ;
$message = $_REQUEST['message'] ;
mail( "someone@example.com", "Subject: $subject",
$message, "From: $email" );
echo "Thank you for using our mail form";
}
else
//if "email" is not filled out, display the form
{
echo "<form method='post' action='mailform.php'>
Email: <input name='email' type='text' /><br />
Subject: <input name='subject' type='text' /><br />
Message:<br />
<textarea name='message' rows='15' cols='40'>
</textarea><br />
<input type='submit' />
</form>";
}
?>

</body>
</html>

```

مثال فوق بصورت زیر عمل می کند:

- ابتدا، بررسی می کند که فیلد ورودی email پرشده باشد.
 - اگر مقداری برای email تعیین نشده بود، فرم HTML دوباره نشان داده خواهد شد.
 - اگر فیلد مقدار داشت ، ایمیلی ارسال خواهد شد.
 - وقتی که بعداز پرشدن فرم، دکمه submit فشرده شود، صفحه دوباره فراخوانی شده و نامه ارسال خواهد شد.
- توجه :** مثال فوق ساده ترین روش برای ارسال ایمیل را نشان می دهد، ولی امنیت خوبی ندارد

22- کار با فایل ها

تابع `fopen()` برای باز کردن فایل در PHP استفاده می شود.

بازکردن فایل

تابع `fopen()` برای باز کردن فایل در PHP استفاده می شود.

اولین پارامتر از این تابع شامل نام فایلی است که باید باز شود و دومین پارامتر تعیین می کند که فایل با چه مدی باز شود:

```
<html>
<body>

<?php
$file=fopen("welcome.txt","r");
?>

</body>
</html>
```

فایل می تواند به یکی از مدهای زیر باز شود:

مد	توصیف
r	فقط خواندنی .
r+	خواندنی/نوشتنی.
w	فقط نوشتنی. باز شده و تمام محتویات قبلی فایل پاک می شود، اگر فایل از قبل وجود نداشته باشد ، ایجاد می گردد.
w+	فقط نوشتنی. باز شده و تمام محتویات قبلی فایل پاک می شود، اگر فایل از قبل وجود نداشته باشد ، ایجاد می گردد.
a	اضافه شدنی. باز شده و اطلاعات جدید به انتهای فایل اضافه می شود، اگر فایل از قبل وجود نداشته باشد ، ایجاد می گردد.
a+	خواندنی/اضافه شدنی.محتوای فایل خواندهنی شود و اطلاعات جدید می توان به انتهای آن اضافه کرد
x	فقط نوشتنی . فایل جدیدی ایجاد می کند.اگر فایل از قبل موجود باشد مقدار FALSE برمی گرداند.
x+	خواندنی/نوشتنی. فایل جدیدی ایجاد می کند.اگر فایل از قبل موجود باشد مقدار FALSE برمی گرداند.

توجه: اگر تابع `fopen()` قادر به بازکردن فایل تعیین شده نباشد، صفر (`false`) را برمی گرداند.
مثال

مثال زیر اگر تابع `fopen()` نتوانست فایل تعیین شده را باز کند، یک پیغام نمایش می دهد.

```
<html>
<body>
```

```
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
?>

</body>
</html>
```

بستن فایل

تابع `fclose()` برای بستن یک فایل باز استفاده می شود:

```
<?php
$file = fopen("test.txt","r");

//some code to be executed

fclose($file);
?>
```

بررسی انتهای فایل

تابع `feof()` بررسی می کند که اشاره گر به انتهای فایل رسیده است یا خیر. تابع `feof()` برای حلقه های تکراری که داده های به طول نامعلوم را می خوانند، مناسب است.

توجه: باید از فایل باز شده با شیوه "W" خوانده شود، نه شیوه "X"!

```
if (feof($file)) echo "End of file";
```

خواندن فایل بصورت خط به خط

تابع `fgets()` برای خواندن تنها یک خط از فایل استفاده می شود. **توجه:** بعد از هر بار فراخوانی این تابع، اشاره گر فایل به خط بعدی منتقل خواهد شد.

مثال

مثال زیر یک فایل را بصورت خط به خط می خواند، تا زمانی که به انتهای فایل برسد.

```
<?php
$file = fopen("welcome.txt", "r") or exit("Unable to open file!");
while(!feof($file))
{
    echo fgets($file). "<br />";
}
fclose($file);
?>
```


خواندن فایل بصورت کاراکتر به کاراکتر

تابع `fgetc()` برای خواندن تنها یک کاراکتر از یک فایل استفاده می شود.
توجه: بعد از فراخوانی این تابع اشاره گر فایل به کاراکتر بعدی حرکت خواهد کرد.

مثال

مثال زیر یک فایل را بصورت کاراکتر به کاراکتر می خواند ، تا زمانی که به انتهای فایل برسد:

```
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
while (!feof($file))
{
    echo fgetc($file);
}
fclose($file);
?>
```