بسمه تعالی

برنامه نویسی مبتنی بر وب

۱- طراحی صفحات وب (HTML)

۲- برنامه نویسی سمت سرویس گیرنده (JavaScript)

۳- برنامه نویسی سمت سرویس دهنده (PHP)

بخش اول

طراحی صفحات وب (HTML)

۱- مقدمه ای بر HTML

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

یک فایل HTML چیست؟

- HTML را "اچ تی ام ال" و یا اچتمل بخوانید.
- HTML برگرفته از حروف اول **L**anguage **M**arkup **T**ext **H**yper 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 ادیتور 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 (Mac کارها از SimplaeText استفاده کنند.) 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.html در درایو c:\ ذخیره کنید.

برای دیدن فایل اچتمل بالا در مرورگر یا Brower تان (معمولا اینترنت اکسپلورر و یا نتسکیپ نویگیتور) 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 c:\test.html)

یکی از مهمترین وظایف مرورگرها نمایش صفحات اچتمل 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!!! (به آدرس فایل در قسمت Address توجه کنید، مثلا http://www.khaterat.com/html/dars/test.html)

نکته ای در مورد ادیتورهای اچتمل (HTML Editors)

با وجودیکه با استفاده از ادیتورهای اختصاصی اچتمل مانند FrontPage و یا Home Page Claris امکان نوشتن و طراحی صفحات اچتمل بصورت WYSIWYG یا "**G**et **Y**ou **W**hat **I**s **S**ee **Y**ou **W**hat" وجود دارد اما توصیه میشود که از ادیتورهای 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-۲

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.
- اینترنت، وب، WWW، web یا **W**eb **W**ide **W**orld همگی یک چیزند.
- تمامی کامپیوترهای وب میتوانند با هم ارتباط داشته باشند.
- کامپیوترهای موجود در وب با کمک استاندارد ارتباطی یا پروتوکل **HTTP** با هم ارتباط برقرار میکنند .

نحوه کارکرد وب چگونه است؟

- اطلاعات وب داخل فایلهائی به نام Web Pages و یا صفحات وب قرار دارند.
- این فایلها یا صفحات روی Web Server یا کامپیوترهای سرویس 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 Netscape Navigator جزو معروفترین مرورگرها حساب میشوند.
- مرورگر Internet Explorer متعلق به شرکت مایکروسافت و مرورگر Navigator Netscape متعلق به شرکت نت اسکیپ 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 پروتوکل HTTP بوده و شامل آدرس صفحه مورد نظر میباشد.
- آدرس یک صفحه وب چیزی شبیه http://www.khaterat.com/faq.html است. بخش http:// نوع پروتوکل و یا استاندارد ارتباطی را تعیین میکند، www.khaterat.com نام دومین یا Domain است و 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 اچتمل (HTML tags) بیان چگونگی نمایش اطلاعات میباشد.
- یک تگ اچتمل چیزی شبیه <p> این تگ پاراگراف است! <p/> است.

چه کسانی استاندارد های وب را تعیین میکند؟

- تعیین استاندارد های وب ربطی به شرکتهای مایکروسافت و یا نت اسکیپ ندارد.

- World Wide Web Consortium یا **W3C** متولی تعیین استانداردهای وب است.
- **HTML, CSS and XML** از مهمترین استانداردهای تصویب شده وب میباشند.
- آخرین استاندارد HTML استاندارد **XHTML 1.0** میباشد .

۳- 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 و یا Elements ساخته میشوند.
- در زبان اچتمل حدود ۸۰ عنصر تعریف شده است.
- تگ های اچتمل بوسیله دو نویسه (char) > 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 یا element content نامیده میشود، مثلا "test"
- تگ های اچتمل را میتوانید بوسیله حروف لاتین کوچک (lower case) و یا بزرگ (upper case) بنویسید و case sensitive نیستند. برای مثال دو تگ و <B> 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 Attribute 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 اچتملتان 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 ۰ 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 یا Element 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.

۴- 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:

**<p> , <-- comment --!> , <h6> .... <h1> ,
 , <hr>**

آشنا خواهید شد . یکی از بهترین روش های یادگیری تگ های اچتمل دیدن مثالها و تغییر آنها میباشد و با کمک ادیتور اختصاصی ما و با استفاده از امکانات فارسی نویسی آن به صورت آنلاین میتوانید به مطالعه و یادگیری مثالها پرداخته، کدهای اچتمل را تغییر داده و با کلیک روی دکمه "نمایش نتایج" به مشاهده نتایج بپردازید.

۱-٤- 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>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

```
<h4>This is a heading</h4>
<h5>This is a heading</h5>
<h6>This is a heading</h6>
```

۲-٤- 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>
```

۳-٤- 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 <br> is a para<br>graph with line breaks</p>
```

٤-٤- توضیحات در اچتمل (Comments)

برای نوشتن شرح و توضیحات در مورد کدهای اچتمل باید از تگ خاصی استفاده کنید. برای اینکار باید متن و شرحتان را درون علائم <-- و --!> 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 اچتمل کار خواهند کرد مفید خواهد بود. (به محل نویسه "!" توجه کنید! فقط یکی و آنهم در ابتدا)

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

٥-٤- چند نکته کاربردی:

توجه داشته باشید که بدلیل وجود مرورگرهای متفاوت (اکسپلورر، نت اسکیپ، ..) و به دلیل تفاوت دقت نمایش صفحه نمایش کامپیوترها، صفحات اچتمل با اندکی تفاوت در حالتهای مختلف نمایش داده میشوند و همیشه سعی کنید که صفحاتتان را نه تنها با اکسپلورر که معروفترین مرورگر است بلکه با نت اسکیپ و حتی مرورگرهای کامپیوترهای مکینتاش چک کرده و همچنین در دقت های نمایش x600۸۰۰ و X768۱۰۲٤ آن را امتحان کنید.

از نظر فاصله و سطر بندی، متنی که در صفحه ادیتورتان 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 ادیتور توسط مرورگر در نظر گرفته نشده و نمایش داده نخواهد شد.

برای نمایش بیش از یک فاصله خالی باید از نویسها یا ترکیب کاراکتری خاصی (Breaking Space None) استفاده کنید. در بخش (Entities) نهادها به این مطلب بیشتر پرداخته خواهد شد.

فاصله های اضافی بین کلمات در یک متن اچتمل توسط مرورگرها دیده نخواهد شد و در نمایش همیشه تبدیل به یک فاصله (space) خواهد شد. در ضمن یک خط خالی در متن ادیتور بصورت یک فاصله یا space نمایش داده خواهد 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 عنصرها میتوان از پاراگراف (<p>) و سرتیترها (<..h>) نام برد.

تگ <hr> یا **R**oler **H**orizontal 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 اچتمل توجه داشته باشید و سعی کنید که از آنها استفاده نکنید.

Start Tag	Purpose	کاربرد
<html>	Defines a html document	نشان شروع متن اچتمل
<body>	Defines the document's body	تعیین بدنه و قسمت اصلی صفحه اچتمل
<h1>-<h6>	Defines heading 1 to heading 6	تعریف سر تیترهای h1 تا h6
<p>	Defines a paragraph	تعریف پاراگراف
 	Inserts a single line break	رفتن سر خط جدید
<hr>	Defines a horizontal rule	نمایش خط افقی
<!-->	Defines a comment in the HTML source code	نوشتن شرح و comment

۵- فارسی نویسی

در این درس با فارسی نویسی، فارسی سازی و استاندارد یونیکد و نحوه استفاده از ادیتور آنلاین و فارسی این سایت آشنا خواهید شد. همچنین متاتگ http-equiv و شناسه dir نیز معرفی خواهد شد.

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

- چگونه فارسی بنویسم ؟
- چگونه فارسی بخوانم؟
- چگونه صفحات اچتمل فارسی را نمایش دهم؟

در مورد فارسی نویسی و فارسی خوانی اگر از ادیتور آنلاین ما در بخش مثالها استفاده میکنید نیازی به فارسی سازی ندارید و با کمک ادیتور مزبور باید به راحتی قادر به تایپ فارسی و لاتین باشید.اگر میخواهید که در هر حال سیستمتان را فارسی کنید، مطالعه صفحات زیر راهنمای بسیار خوبی برای فارسی سازی میباشند:

۱-۵- دو نکته بسیار مهم در مورد فارسی نویسی:

□روش کدگذاری **UTF-8**

اکیدا توصیه میشود که برای تایپ و نمایش متون فارسی از استاندارد یونیکد و روش کدگزاری UTF-8 استفاده.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">

در اینصورت مرورگر یا Browser قبل از نمایش صفحه از روی متاتگ فوق نوع کدگذاری (Encoding) را تشخیص داده و دیگر مثلا نیازی به تعیین دستی (دراکسپلورور ۵ : View/Encoing/Unicode / UTF-8) نوع Encoing توسط بازدیدکننده سایت نخواهد بود.

□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 یا attribute ی 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

مثلا برای نمایش جمله فارسی "سلام بر دنیای وب!" با کمک عناصر p یا div کافی است که به یکی از شکلهای زیر عمل شود:

سلام بر دنیای وب

سلام بر دنیای وب

در بسیاری از عناصر و تگ های اچتمل امکان استفاده از شناسه dir میسر بوده و به عنوان نمونه میتوان از عناصر زیر نام برد:

 , , <html> , <body> , <table> , <tr> , <td> , <h1 ... h6> , <input> , <pre> , <select> , , ...

۲-۵- نحوه دیدن کدمنبع کدهای اچتمل (HTML Source)

- اگر میخواهید که سورس کدهای اچتمل صفحات وب را ببینید کافی است که در منوی View مرورگرتان حالت Source یا Page Source را انتخاب کرده و مرورگر کد اچتمل را درون ادیتوری باز کرده و قابل ذخیره توسط شما برای استفاده های بعدی میباشد.
- اگر صفحه مورد نظر از مجموعه فریمها (Frameset) استفاده کرده باشد برای دیدن سورس کد اصلی فریمها کافی است که ابتدا با کمک روش بالا به مشاهده سورس کدهای اچتمل پرداخته (لازم به توضیح است که در این حالت فقط تعاریف کلی فریمها قرار دارند.) و سپس برای دیدن کد اچتمل اختصاصی هر صفحه باید پس از "Right Click" روی هر صفحه "View Source" کرده و کد اچتمل هر فریم را جداگانه ببینید.
- اگر طراح سایت با کمک Java Script مانع دیدن سورس توسط شما میشود، ابتدا در منوی Tools/Internet Options.../Security/Custom Level مرورگرهای اکسپلورر امکانات جاواسکریپت مرورگر را غیر فعال کرده (disable) و سپس مراحل بالا را تکرار کنید!!!

٦- تگهای فرمت دهی متون

تگهای فرمت دهی متون

کاربرد	Purpose	Start Tag

نمایش توپر	Defines bold text	<b>
نمایش ایتالیک یا کج	Defines italic text	<i>
نمایش در اندازه کوچک	Defines small text	<small>
نمایش قوی ؟!	Defines strong text	
نمایش پایین تر از خط افقی	Defines subscripted text	<sub>
نمایش بالاتر از خط افقی	Defines superscripted text	<sup>
از رده خارج	Deprecated. Use <del> instead	<strike>
از رده خارج	Deprecated. Use styles instead	<u>
نمایش متون از قبل فرمت شده	Defines preformatted text	<pre>

چند نکته کاربردی:

I. در استفاده از Underline دقت کنید چون در اکثر مرورگرها، پیوندها با این خطوط نشان داده مي شوند و این مساله میتواند کاربران را دچار اشتباه کند .

II. به کارگیري تگهاي EM و STRONG که جایگزین تگهاي I و B هستند ، این اطمینان را مي دهد که صفحه شما توسط همه مرورگرها بصورت صحیح نمایش داده خواهد شد .

III. به ترتیب بسته شدن تگهاي پایاني در مثال زیر دقت کنید، همان گونه که مشاهده مي کنید عنصر STRONG بعد از عنصر U شروع شده است ، به همین خاطر تگ خاتمه دهنده آن قبل از تگ خاتمه دهنده U قرار مي گیرد.

```
<U>version <STRONG>2.0</STRONG></U>
```

۷- 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 نمیباشند. در این قسمت با فهرست این نویسه ها آشنا شده و نحوه درج آنها را با کمک Character Entities فرا خواهید گرفت.

Character Entities -۷-۱

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:

۱- نویسه ampersand (&)

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

۳ - 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 گذاریهای 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 بزرگي حروف حساس هستند و در واقع Case Sensitive میباشند.

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 ۱۰ نویسه فاصله، ۹ تای آنرا نادیده میگیرند و برای نمایش فاصله های اضافی باید از ** ** استفاده کنید.

۲-۷- فهرست پرکاربردترین Character Entities ها

Entity Number (شماره عددی)	Entity Name (نام)	Description (شرح)	Result (نمایش)
		non-breaking space	
<	<	less than	<
>	>	greater than	>
&	&	ampersand	&
"	"	quotation mark	"
'		apostrophe	'

۸-پیوندهای اچتمل (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 target،href و name تشریح خواهند شد.

در محیط وب، صفحات اچتمل 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.

۸-۱- 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، فایلهای صوتی یا حتی فیلم ها و ... اشاره کنند.

فرم کلی یک پیوند به قرار زیر است:

Some Text

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 **h**yerlink **ref**erence 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 اچتمل 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.

۲-۸- شناسه target در پیوندها

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>
```

۲-۸- شناسه Name

با کمک شناسه name میتوانید پیوندها را نام گذاری کنید.پیوندهای نامگذاری شده امکان حرکت میان قسمتهای مختلف یک صفحه یا page را فراهم میکنند. در اینصورت مثلا برای رفتن به آخر یک متن دیگر نیازی به Scroll down کردن تمامی صفحه نیست و کافی است که بازدیدکننده روی پیوندی که به آخر صفحه اشاره میکند کلیک کند.

استفاده از پیوندهای نامگذاری شده شامل دو مرحله است:

۱- ایجاد یک پیوند نامگذاری شده (این قسمت به عنوان لنگر کار خواهد کرد.) :

فرم کلی یک پیوند نام گذاری شده به قرار زیر است:

```
<a name="label">Text to be displayed</a>
```

وظیفه شناسه name تعیین نام برای پیوند است و مقدارآن همان نام پیوند میباشد. نام پیوند هم هر نام لاتینی میتواند باشد. در فرم کلی بالا، نام پیوند label و متنی که بعنوان پیوند نمایش داده میشود عبارت "Text to be displayed" خواهد بود. مثال زیر به ایجاد یک پیوند نامگذاری شده با نام top میپردازد:

```
<a name="top">Here is top of my page!</a>
```

۲- ایجاد یک پیوند به پیوند نامگذاری شده دیگر:

برای دادن لینک به پیوندی نام گذاری شده، کافی است که پیوندی معمولی ایجاد کرده و در قسمت href آن ابتدا url مقصد و سپس نویسه # و در نهایت نام پیوند نامگذاری شده را درج کنید. مثال زیر ایجاد پیوندی است که به لینک نامگذاری شده ای به نام top اشاره میکند:

```
<a href="http://www.khaterat.com/www/index.cgi#top"> Goto Top! </a>
```

در اثر کلیک روی پیوند بالا مرورگر مستقیما به ابتدای بخش top صفحه http://www.khaterat.com/www/index.cgi خواهد رفت.

اگر مقصد پیوند در همان صفحه قرار دارد نیازی به قید url نیست و فقط نویسه # و سپس نام پیوند کافی است:

** Goto Top! **

٤-٨- چند نکته کاربردی در مورد پیوندها:

* یکی از کاربردهای رایج پیوندهای نام گذاری شده در صفحات و متونی میباشد که فهرست و یا لیستی از اقلام نمایش داده شده است، مانند بخش سرفصل مطالب، فهرست FAQ و یا ... .

* اگر مرورگر نتواند یک پیوند نامگذاری شده را بیابد ابتدای صفحه مقصد را نمایش خواهد داد.

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

۹- جداول (Tables)

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

۱-۹- 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 **t**able **r**ow و **d**ata **t**able میباشند.
محتوی یک سلول میتواند متن، تصویر، فهرستها، جداول دیگر، پاراگرافها و ... باشد.
مثال زیر جدولی است با دو سطر و سه ستون :

نمایش جدول روبرو توسط مرورگر	کد اچتمل جدولی با دو سطر و سه ستون
ردیف ۱، سلول ۱ / ردیف ۱، سلول ۲ / ردیف ۱، سلول ۳ ردیف ۲، سلول ۱ / ردیف ۲، سلول ۲ / ردیف ۲، سلول ۳	<table border="1" dir="rtl"> <tr> <td>ردیف ۱، سلول ۱</td> <td>ردیف ۱، سلول ۲</td> <td>ردیف ۱، سلول ۳</td>

نمایش جدول روبرو توسط مرورگر	کد اچتمل جدولی با دو سطر و سه ستون
	</tr> <tr> <td>۱ سلول ،۲ ردیف</td> <td>۲ سلول ،۲ ردیف</td> <td>۳ سلول ،۲ ردیف</td> </tr> </table>

۲-۹- جدولها و شناسه های border و dir

□در مثال بالا شناسه border مرز جدول را مشخص میکند، مقدار ۱ 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 چو ن در داخل تک table تعریف شده است به تمامی سلولهای جدول اعمال خواهد شد.

□ساده ترین جدول ممکن در اچتمل، جدولی است با یک سطر و یک ستون!

۳-۹- سرستون در جداول (Headings)

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

نمایش جدول روبرو توسط مرورگر	کد اچتمل جدولی با سه سطر و سه ستون
سرستون 1 / **سرستون ۲** / **سرستون ۳** ردیف ۱، سلول ۱ / ردیف ۱، سلول ۲ / ردیف ۱، سلول ۳ ردیف ۲، سلول ۱ / ردیف ۲، سلول ۲ / ردیف ۲، سلول ۳	<table border="1" dir="rtl">> <tr> <th>سرستون 1</th> <th>سرستون ۲</th> <th>سرستون ۳</th> </tr> <tr> <td>۱ سلول ،۱ ردیف</td> <td>۲ سلول ،۱ ردیف</td> <td>۳ سلول ،۱ ردیف</td> </tr> <tr> <td>۱ سلول ،۲ ردیف</td>

نمایش جدول روبرو توسط مرورگر	کد اچتمل جدولی با سه سطر و سه ستون
	<td>۲ سلول ،۲ ردیف</td> <td>۳ سلول ،۲ ردیف</td> </tr> </table>

همانطوریکه مشاهده میکنید سر ستونها مانند سلولهای معمولی جداول 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.

۴-۹- خانه های خالی در جداول (Empty Cells)

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 اچتمل جدولی با دو سطر و دو ستون
ردیف ۱، سلول ۱ / ردیف ۱، سلول ۲ / ردیف ۲، سلول ۱	<table border="1" dir="rtl">> <tr> <td>۱ سلول ،۱ ردیف</td> <td>۲ سلول ،۱ ردیف</td> </tr> <tr> <td>۱ سلول ،۲ ردیف</td> <td></td> </tr> </table>

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 اچتمل جدولی با دو سطر و دو ستون
ردیف ۱، سلول ۱ / ردیف ۱، سلول ۲ / ردیف ۲، سلول ۱	<table border="1" dir="rtl">> <tr> <td>۱ سلول ،۱ ردیف</td> <td>۲ سلول ،۱ ردیف</td> </tr> <tr> <td>۱ سلول ،۲ ردیف</td> <td> </td> </tr>

نمایش جدول روبرو توسط مرورگر	کد اچتمل جدولی با دو سطر و دو ستون
	</table>

توجه داشته باشید که اینبار مرزهای خانه خالی جدول بدرستی ترسیم گردیده اند.

۹-۵- چند نکته اساسی در مورد جداول:

□در جداول، ردیفها، خانه ها و سر ستونها از تعداد زیادی از شناسه ها میتوانید استفاده کنید که شرح همگی آنها از حوصله این دوره مقدماتی خارج است و توصیه میشود که از مثالهای زیر و جدول واقع در انتهای این صفحه برای مطالعه هر چه بیشتر در مورد جزئیات شناسه ها استفاده شود.

□شناسه dir قابل استفاده در بسیاری از تگها (مثلا table، tr، td، th) میباشد. با کمک شناسه dir و مقدار rtl برای آن، جهت نمایش متون فارسی از "راست به چپ" تعیین میگردد. توجه داشته باشید که مقدار شناسه dir به صورت موروثی از table به tr و th و از tr و th به td خواهد رسید.مثلا برای تعیین مقدار rtl برای تمامی خانه های یک جدول کافی است که فقط شناسه مزبور را در تگ table قید کنید و نیازی به قید آن در تمامی خانه های جدول نیست. البته میتوانید که در مورد یک ردیف خاص و یا یک خانه خاص شناسه dir را موردی تعیین کنید.

تگهای جداول

تگ	هدف	کاربرد
<table>	Defines a table	تعریف جدول
<th>	Defines a table header	تعریف سرستون در جداول
<tr>	Defines a table row	تعریف ردیف ها در جداول
<td>	Defines a table cell	تعریف سلول یا خانه های یک جدول
<caption>	Defines a table caption	تعریف عنوان جدول

۱۰- فهرستها (Lists)

این درس شما را با نحوه نمایش انواع فهرستها شامل فهرستهای مرتب(ordered) ، نامرتب(unordered) و فهرستهای تعریفی(definition lists) و تگ های مربوطه یعنی li ، ol، ul، dl، dt و dd آشنا خواهد کرد.

۱۰-۱- فهرستهای نامرتب(unordered list)

فهرستهائی هستند از یک یا چند قلم اطلاعات که معمولا با دایرهای کوچک و سیاه رنگ در شروع نمایش داده میشوند. یک فهرست از نوع نامرتب با تگ <ul> شروع شده و هر رقم اطلاعات با تگ <li> مشخص میگردد.

کدHTML	نمایش توسط مرورگر
  - Red - Blue	• Red • Blue • Green

Green 	
< "ul dir="rtl"> - قهوه - شیر - چای 	• قهوه • شیر • چای

داخل تگهای <li> میتوانید از پاراگرافها، تگهای br ، تصاویر و حتي 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.

۱۰-۲- فهرستهای مرتب(ordered list)

این نوع فهرست بسیار شبیه فهرستهای نامرتب بوده و فقط مرورگر به هنگام نمایش اقلام اطلاعاتی از اعداد به جای دایره های کوچک سیاه رنگ استفاده میکند. یک فهرست از نوع مرتب با تگ <ol> شروع شده و همانند فهرستهای نامرتب هر رقم اطلاعات با تگ <li> مشخص میگردد.

HTMLکد	نمایش توسط مرورگر
  - Red - Blue - Green 	1. Red 2. Blue 3. Green
< "ol dir="rtl"> - قهوه - شیر - چای 	۱. قهوه ۲. شیر ۳. چای

داخل تگهای <li> میتوانید از پاراگرافها، تگهای br ، تصاویر و حتي 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.

۱۰-۳- 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.

HTML code	نمایش توسط مرورگر
<dl>	HTTP Hypertext Transfer Protocol

<dt>HTTP</dt> <dd>Hypertext Transfer Protocol</dd> <dt>FTP</dt> <dd>File Transfer Protocol</dd> <dt>IP</dt> <dd>Internet Protocol</dd> </dl>	FTP File Transfer Protocol IP Internet Protocol
<dl dir="rtl" > < dt>مقدمه</dt> <dd>خلاصه ای از مقدمه</dd> <dt>فصل ۱</dt> <dd>خلاصه ای از فصل ۱</dd> <dt>فصل ۲</dt> <dd>خلاصه ای از فصل ۲</dd> </dl>	مقدمه خلاصه ای از مقدمه فصل ۱ خلاصه ای از فصل ۱ فصل ۲ خلاصه ای از فصل ۲

داخل تگهای تشریح یعنی <dd> میتوانید از پاراگرافها، تگهای br ، تصاویر و حتي 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.

تگهای فهرست

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

۱۱- فرمها (Forms)

از طریق استفاده از عنصر <form> و چند تگ مرتبط قادر به دریافت اطلاعات از بازدیدکنندگان صفحاتتان و یا 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. مثالهای cgi مدرسه وب از زبان اسکریپت Perl استفاده میکنند.

۱-۱۱- فرمها (Forms)

تمامی عناصر و تگهائی که تاکنون دیده اید فقط به نمایش اطلاعات پرداخته اند و هیچکدام به گرفتن و اخذ اطلاعات از کاربر نپرداخته اند. فرمها که با تگ <form> معرفی میشوند، طراح سایت را قادر به جمع آوری و اخذ اطلاعات از بازدیدکننده سایت خواهند کرد. عنصر فرم و گروهی از عناصر و تگهای درون آن به دریافت و ارسال اطلاعات به سمت وب سرور کمک خواهند کرد و لازم به ذکر است که 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 سطری(text fields)، عناصر ورود متنهای چند سطری (Textarea)، منوهای drop-down و radio buttons و ...

۲-۱۱- 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 , image , text , checkbox , radio , password

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

۱-۲-۱۱- 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
First name: [ ] Last name: [ ]	<form> First name: <input type="text" name="firstname">> Last name: <input type="text" name="lastname">> </form>
نـــام : [ ] نام خانوادگی : [ ]	<form dir="rtl" > نـــام: <input type="text" name="firstname">> نام خانوادگی: <input type="text" name="lastname">> </form>

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 ۲۰ 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.

۲-۲-۱۱- 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:

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

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

۳-۲-۱۱-ورودیهای 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:

کد اچتمل	نمایش توسط مرورگر
<form dir="rtl"> <input type="checkbox" name="bike">> دوچرخه دارم <input type="checkbox" name="car">> ماشین دارم </form>	دوچرخـــــــــه دارم ماشین دارم

۴-۲-۱۱-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 هائی 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 یا ارسال خواهد گردید.

کد اچتمل	نمایش توسط مرورگر
<form name="input" action="form_action.cgi"> Username: <input type="text" name="user"> <input type="submit" value="Submit"> </form>	Username: Submit
<form name="input" action="form_action.cgi" dir="rtl">> نام کاربر <input type="text" name="user">> <input type="submit" value="Submit">> </form>	نام کاربر ارسال

در مثال بالا در باکس 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 Perl با نام form_action.cgi در سمت سرور به ذخیره و سپس نمایش متنی خبری خواهد پرداخت.

۵-۲-۱۱-ورود متن فارسی در عناصری مانند Text field یا Text area

همانطور که در مثالهای بالا مشاهده کردید در مورد المانهای Text field و Textarea با کمک شناسه dir میتوان سمت و جهت ورود اطلاعات را "از راست به چپ" تعیین کنید. ولی اگر کاربری ویندوز فارسی نداشته باشد، از کامپیوترهای مکینتاش استفاده کند و یا سیستمش را فارسی نکرده باشد قادر به ورود اطلاعات به زبان فارسی نخواهد بود و نویسه های تایپ شده لاتین خواهند بود!، در اینگونه موارد چه باید کرد؟

معمولا برنامه نویسان وب برای فراهم ساختن امکان ورود متن فارسی از یکی از دو روش زیر استفاده میکنند:

- استفاده از اپلتهای جاوا با این مزیت که سورس و کد اصلی فارسی سازی قابل استفاده توسط دیگران نیست!!
- استفاده از زبان جاوا اسکریپت و با این عیب که سورس و کد اصلی JavaScript قابل مشاهده و استفاده توسط دیگران است!

تگهای فرم

Start Tag	کاربرد
<form>	تعریف فرم ورود اطلاعات
<input>	تعریف ورودی از نوع Input
<textarea>	تعریف ورودی متن چند سطری یا text-area
<label>	تعریف برچسب یا label
<fieldset>	تعریف fieldset

تعریف عنوان برای fieldset ها	<legend>
تعریف فهرستهای انتخابی یا drop-down box	<select>
تعریف option group ها	<optgroup>
تعریف گزینه ای از drop-down box ها	<option>
تعریف دکمه فشاری، متفاوت با button تگ input	<button>

۱۲- تصاویر (Images)

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

۱-۱۲- تگ img و شناسه src یا source

در زبان اچتمل تصاویر را با کمک تگ <img> تعریف میکنند. تگ <img> از نوع تگهای خالی است، بدین معنا که فقط دارای یک یا چند شناسه و attribute بوده و دارای تگ انتهائی یا <img/> نیست.

مهمترین شناسه برای درج و تعریف یک تصویر src یا source 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:

<img src="url">

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.

۱-۱-۱۲- شناسه های width و height در تگ img

کاربرد شناسه های width و height تعیین عرض و ارتفاع نمایش تصویر است. اگر از این شناسه ها استفاده نکنید، مرورگر ابتدا تصویر را لود کرده و سپس طول و عرض آنرا یافته و در نهایت نمایش میدهد. با نوشتن مقادیری غیر از مقادیر واقعی طول و عرض تصویر میتوانید به تغییر شکل و ابعاد آن بپردازید و آنرا مثلا کوچکتر، بزرگتر و یا کشیده تر نمایش دهید. مثال زیر نحوه درج تصویری را با قید ابعاد آن شرح میدهد:

<img src="http://www.google.com/images/logo.gif" width="276" height="110" >

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.(در واقع با قید ابعاد تصویر، مرورگر قبل از لود تصویر میداند که چه مقدار از صفحه نمایش باید به تصویر باید اختصاص داده شود و قبل از لوذ شدن تمامی تصاویر مرورگر شروع به نمایش قالب و چارچوب کلی صفحه خواهد کرد.)

۲-۱-۱۲- شناسه Alt یا "alternate text" یا متن جایگزین

اگر مرورگر به هر دلیلی نتواند که یک تصویر را نمایش دهد و یا مرورگر از نوع مرورگرهائی باشد که فقط متن را نمایش میدهند، متن و text ی که با کمک شناسه alt تعیین شده است نمایش داده خواهد شد. همچنین در اغلب مرورگرها با قرارگرفتن ماوس بروی یک تصویر، متن تعیین شده توسط شناسه alt نمایش داده خواهد شد. با کمک alt اطلاعات اضافی مربوط به تصویر قابل نمایش میباشد.مثال زیر نحوه درج تصویری را با قید متن جایگزین آن شرح میدهد:

<img src="images/madresehweb_88_31.jpg" alt="The web school!" >

< width=258 height=78 "img src="http://www.google.ca/images/hp0.gif>

۳-۱۲- چند نکته مهم

- اگر در یک صفحه اچتمل از ۱۰ تصویر استفاده شده باشد، مرورگر باید ۱۱ فایل را لود کند.(خود صفحه به علاوه ۱۰ 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 ۵۰ کیلوبایت برای کسی که از مودمی با سرعت kbps۲۸ استفاده میکند حداقل ۱۵ ثانیه طول خواهد کشید.
- برای دیدن مشخصات تصاویر در وب کافی است که روی تصویر Right click کرذه و سپس در بخش properities آدرس،ابعاد و ظرفیت تصویر را ببینید.
- برای ذخیره تصاویر در وب کافی است که روی تصویر Right click کرذه و سپس در بخش "Save picture as" آنرا ذخیره کنید.

۱۳- پس زمینه (Background)

در این درس با چگونگی تعیین تصاویر و رنگ پس زمینه (Background) آشنا خواهید شد. رنگ و یا تصویر پس زمینه فاکتور بسیار مهمی در طراحی صفحات اچتمل بوده و استفاده از رنگ و یا تصویر مناسب نقشی تعیین کننده در خوانا بودن صفحات دارد.

۱-۱۳- رنگ و تصویر زمینه در صفحات اچتمل

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.

۱-۱-۱۳- شناسه bgcolor برای رنگ پس زمینه (background color)

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">
```

در تمامی مثالهای بالا با کمک شناسه bgcolor رنگ زمینه صفحه "سیاه" تعیین میشود.

۲-۱-۱۳- شناسه Background برای تصویر پس زمینه

برای تعیین تصویری که به عنوان پس زمینه صفحه نمایش داده میشود از شناسه background استفاده میشود.مقدار این شناسه URL یا آدر س 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.khaterat.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.

۱۳-۲- نکات مهم

اگر از تصاویر پس زمینه در صفحات اچتمل استفاده میکنید به نکات زیر توجه کنید :

- هرتصویری 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 دیگرتصاویر 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 سایتهای وب از تصاویر پس زمینه استفاده نمی کنند!

۱٤- قلم ها (HTML Fonts)

با وجودی که هنوز بعضی از اچتمل کاران از تگ برای تعیین نام و نوع قلم، رنگ و اندازه متون استفاده میکنند ولی در استاندارد جدید اچتمل (نسخه ٤) و همچنین XHTML از تگ پشتیبانی نمیشود و استفاده از Style Sheet ها به عنوان جایگزین معرفی شده است.

**۱-۱٤- تگ **

در نسخه های ۳,۲ و قبلی تر زبان اچتمل 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:

کد اچتمل	نمایش کد روبرو توسط مرورگر
 This is a paragraph. 	This is a paragraph.
متنی با قلم Times و اندازه ۱ و رنگ FF9900	

کد اچتمل	نمایش کد روبرو توسط مرورگر
 FF9900 و اندازه ۱ و رنگ Times متنی با قلم 	
 و اندازه ۲ و رنگ قرمز Tahoma این متنی با قلم 	متنی با قلم Tahoma و اندازه ۲ و رنگ قرمز

۲-۱٤- شناسه های تگ <font>

کاربرد	Example	Attribute
تعیین اندازه قلم	size="2"	size="number"
افزایش اندازه قلم	size="+1"	size="+number"
کاهش اندازه قلم	size="-1"	size="-number"
تعیین نام قلم	face="Times"	face="face-name"
تعیین رنگ قلم	color="#eeff00"	color="color-value"
تعیین رنگ قلم	color="red"	color="color-name"

*** بیان مختصر عملکرد تگ <font> در این دروس فقط جهت اطلاع بوده و استفاده از آنها در طراحی صفحات اچتمل توصیه نمیشود.

**۳-۱٤- تگ **

تگ سبب تعریف محدوده و گستره ای مشخص میشود. تمامی المان هائی که درون تگهای و قرار خواهند گرفت خواص آنرا به ارث خواهند برد، خواصی از قبیل استیل (Style) و یا سمت و جهت نمایش متون(dir).

بر خلاف تگ مرورگرها یک سطر جدید به اول و آخر المان اضافه می کنند.

مثال:

کد اچتمل	نمایش کد روبرو توسط مرورگر
Some text <div style="color:#0000AF;">another text again text...	Some text another text again text...

شناسه های استاندارد المان div	id, class, title, style, dir, lang

کاربرد	تگ
تعریف بخش و ناحیه ای از صفحه اچتمل	
تعریف بخش و ناحیه ای از صفحه اچتمل	

۱۵- استایل شیتها (Style Sheets)

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

۱-۱۵- چگونگی استفاده از استایل شیتها

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

۱)استایل شیتهای خارجی (external style sheet)

در این روش تمامی استیلها و تعاریف نمایشی درون فایلی جداگانه قرار گرفته و بسیار مناسب حالتهائی است که قرار است که استایلی به بیش از یک صفحه اچتمل 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 معمولا 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 اچتمل 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:

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

۲)استایل شیتهای داخلی (Internal Style Sheet)

کاربرد این روش در حالتهائی است که صفحه اچتمل نیازمند تعریف استایلی منحصر به فرد و مجزا بوده و استایل طراحی شده فقط بدرد آن صفحه خواهد خورد. در این حالت باید کدهای حاوی تعریف استایل درون تگ های <style> و </style> قرار گرفته و معمولا نتیجه درون بخش head صفحه اچتمل گذاشته میشود.مثال:

```
<head>
<style type="text/css">
body {background-color: red}
p {margin-left: 20px}
</style>
</head>
```

۳)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 تقریبا در تمامی تگها قابل اعمال است استفاده شود. مثال:

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

تگ های استایل

تگ	کاربرد
<style>	تعریف استیلها درون یک سند
<link>	تعیین و تعریف فایل استایل شیت خارجی
<font>	کنار گذاشته شده.از استایل شیتها استفاده شود.
<basefont>	کنار گذاشته شده.از استایل شیتها استفاده شود.
	کنار گذاشته شده.از استایل شیتها استفاده شود.

۲-۱۵- تغییرات در نسخه جدید اچتمل (4/0)

هدف اولیه از طراحی زبان اچتمل ایجاد زبانی برای تعریف اطلاعات محض بدون در برداشتن جزئیات مربوط به فرمت و نمایش بود ولی با ایجاد نسخه 3/2 زبان اچتمل و اضافه شدن تگهای فرمت دهی مانند <font> و یا شناسه هائی مانند color و Align محتوا و نمایش در صفحات اچتمل ادغام گردیده و همین امر سبب بروز مشکلات عدیده ای برای طراحان و برنامه نویسان وب گردید. برای رفع این مشکل در نسخه 4/0 اچتمل توصیه بر جداسازی کامل این دو گروه از اطلاعات بوده و در نسخه های آینده اچتمل این جداسازی اجباری خواهد گردید و توصیه مدرسه وب نیز استفاده از استایل شیتها برای هر نوع تعریف فرمت و ترازبندی میباشد. برای مثال در کد اچتمل زیر که بر اساس نسخه 3/2 اچتمل میباشد، ساختار و محتوا (تگی از نوع پاراگراف با محتوای "This is a paragraph") با اطلاعات نمایش یعنی رنگ و ترازبندی (رنگ قرمز متن و ترازبندی در وسط صفحه) همزمان و در کنار هم آورده شده اند و این ادغام امکان تغییر و توسعه کد را بسیار محدود خواهد کرد:

```
<p align="center"><font color="red">This is a paragraph</font></p>
```

بخش دوم

برنامه نویسی سمت سرویس گیرنده (JavaScript)

۱- معرفي زبان جاوا اسكريپت

۱-۱- جاوااسكريپت چيست؟

براي يادگيري جاوااسكريپت، نياز به دانستن زبان HTML و درک کاملي از طراحي و ساخت صفحات وب دارد چون جاوااسكريپت يک زبان مجزا از اچ تي ام ال نيست و بايد همراه با کدهاي HTML در يک صفحه بکار رود.

همانطور که مي دانيد زبان اچ تي ام ال به تنهايي جوابگوي نياز بينندگان وب سايتها نمي باشد چون آنها علاقه خاصي به جذابيتها و ارتباط متقابل ميان خود و صفحات وب دارند که اچ تي ام ال نمي تواند جذابيت قابل توجهي به يک صفحه بدهد و يا با يک کاربر ارتباط برقرار کند. پس براي برطرف کردن اين مشکلات، صفحات وب نيازمند يک زبان قوي تر مي باشد که به همين منظور زبان جاوااسكريپت خلق شد.

جاوااسكريپت توسط شرکت Netscape اختراع شد و به عنوان اولين زبان اسكريپت نويسي در وب توسعه يافت که در حال حاضر پرکاربردترين و محبوب ترين زبان اسكريپت نويسي در دنيا مي باشد که يکي از دلايل آن، پشتيباني دو مرورگر معروف وب، Internet Explorer و Netscape Navigator از اين زبان مي باشد. نا گفته نماند که شرکت مايکروسافت هم براي اينکه از نت اسکيپ در اين زمينه عقب نماند، نسخه مربوط به خود را بنام Jscript توليد کرد.

۱-۲- تفاوت جاوااسكريپت با جاوا

جاوااسكريپت متعلق به شرکت نت اسکيپ مي باشد اما زبان جاوا متعلق به شرکت Sun Microsystems است. جاوااسكريپت در اصل يک زبان اسكريپت نويسي است نه يک زبان برنامه نويسي کامپيوتر، شايد بتوان اسكريپتها را يک نوع زبان برنامه نويسي خيلي سبک و ساده ناميد اما جاوا يک زبان برنامه نويسي کاملاً پيچيده مانند زبان C است. تفاوت ديگر آنها اينست که جاوااسكريپت توسط يک برنامه مفسر يعني interpreter در مرورگر تعريف و تفسير شده تا قابل اجرا باشد و حتماً هم اين کار در مرورگر انجام مي شود اما برنامه هاي جاوا بايد توسط يک کامپايلر Compiler ، کامپايل شده که کامپايل شدن يعني جمع آوري اطلاعات مورد نظر و ترجمه آن اطلاعات به زبان قابل فهم براي کامپيوتر و تبديل آن به يک برنامه کاربردي واحد و مستقل براي کاربر را گويند. به همين دليل برنامه هاي جاوا را که اپلت applet مي نامند، مستقل اجرا مي شوند. البته اپلتهاي جاوا را مي توان در صفحات وب هم گنجاند.

۱-۳- قابليتهاي جاوااسكريپت

جاوااسكريپت داراي تركيباتي ساده نسبت به زبانهاي برنامه نويسي است كه براحتي مي توان آنها را داخل كدهاي اچ تي ام ال يک صفحه قرار داد. يكي از قابليتهاي ويژه جاوااسكريپت رايگان بودن آن است كه اگر شما زبان HTML را بدانيد، مي توانيد با كمي دانستن جاوااسكريپت كدهاي آماده كه بيش از ۲۰۰۰ نوع از آنها در وب وجود دارد را در صفحات خود بكار گيريد.

با استفاده از اين زبان مي توانيد متنهاي پويا و متحرک در صفحات و در نقاط مختلف مرورگرها مانند نوار وضعيت آنها خلق كنيد. يكي ديگر از قابليتهاي اين زبان عكس العمل نشان دادن به وقايع و اتفاقات مي باشد بطور مثال واكنش نسبت به كليک يكي از دكمه هاي موس يا فشردن يک كليد بر روي كيبورد و يا انجام عملي هنگام بسته شدن صفحه مرورگر.

جاوااسكريپت حتي مي تواند كدهاي اچ تي ام ال را بخواند و يا توليد كند كه توسط اين خاصيت مي توانيد در هر زمان خاص يک متني را توليد يا تغيير دهيد و يا در زماني مشخص عكسي را جايگزين عكسي ديگر كنيد.

يكي از بهترين ويژگيهاي جاوااسكريپت، اعتبار بخشيدن به اطلاعات مي باشد مثلاً شما يک فرمي براي ارتباط با كاربر خود ساخته ايد و مي خواهيد قبل از اينكه آن اطلاعات به سرور فرستاده شود، بازبيني شده تا اگر مشكلي وجود داشته باشد به كاربر اطلاع داده شود تا در صدد رفع آن برآيد. چون جاوااسكريپت اصطلاحاًClient-Sideاست، يعني در سيستم كاربر اجرا شده و مستقل از وب سرور است مي تواند ترافيک روي سرور را كم كرده كه اين موضوع يک امتياز مثبت براي ميزبانان وب محسوب مي شود.

۲- آغازاسكريپت نويسي

۲-۱- تگ اسكريپت

جاوااسكريپت هم مانند زبان HTML از يكسري دستور و گرامر خاص خود پيروي مي كند. كدهاي جاوااسكريپت را بايد در ميان تگهاي HTML قرار دهيد. به همين دليل يک تگ براي آن در نظر گرفته شده است:

```
<script> ... </script>
```

كه كليه دستورات جاوااسكريپت بايد مابين آنها قرار گيرد. اين تگهاي اچ تي ام ال به مرورگر مي فهمانند كه در كجا اين زبان آغاز شده تا مفسر جاوااسكريپت مرورگر فعال شود و در كجا پايان مي پذيرد كه برنامه مفسر هم كار خود را خاتمه دهد.

تگ script داراي خواص زير است:

- خاصيت language : زبان را تعيين مي كند و مقدار آنرا بايد برابر با javascript قرار دهيد.

```
<script language="javascript">
</script>
```

- خاصيت type : براي اين زبان بايد آنرا برابر با text/javascript قرار دهيد.

براي شروع كار مي توانيد از يک ويرايشگر متن مانند برنامه Notepad استفاده كرده و كد زير را در آن وارد كنيد:

```
<html>
<head>
<title> My JavaScript </title>
</head>
<body>
<script language="javascript1.5" type="text/javascript">
document.write(" Hello my friends ")
</script>
```

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

فایل را با پسوند htm ذخیره کنید و سپس آنرا در مرورگر اینترنت اکسپلورر اجرا کنید که جمله Hello my friends را باید در صفحه ببینید. نوشتن نسخه جاوااسکریپت اجباري نیست یعني مي توانید آنرا ننویسید.

```
document.write(" Hello my friends ")
```

در دستور فوق document یک شيء است یعني همان سند و فایل html که در آن این عملیات باید انجام شود عمل write یعني نوشتن را انجام دهد وهر چیزي که در بین پرانتز هست را مي نویسد.

- خاصیت src : کدهاي جاوااسکریپت را هم مانند استایل شیت مي توانید در یک فایل جداگانه نوشته و با پسوند js ذخیره کنید سپس با خصوصیت src آدرس آن فایل را مشخص کنید.

```
<html>
<head>
<script language="javascript" src="jsfiles/name.js" type="text/javascript">
</script>
</head>
```

۳- پیغامهاي خطا در جاوااسکریپت

۳-۱- چگونگي دریافت پیغام خطا

خطاهاي جاوااسکریپت را به دو صورت متوجه مي شوید یکي ظاهر شدن مثلثي زرد با یک علامت تعجب در سمت چپ نوار وضعیت مرورگر که در بعضي از مرورگرها این نوار مشاهده نمي شود و مي توانید براي مشاهده آن به منوي View مرورگر رفته و روي گزینه Bar Status کلیک کنید. حالت دیگر دیدن پیغامهاي خطا، باز شدن یک پنجره در وسط مرورگر است که در این حالت جزئیات خطاي ایجاد شده را هم مي توانید مشاهده کنید.

در نسخه هاي جدید مرورگر اینترنت اکسپلورر این پنجره مشاهده نمیشود و ابتدا آنرا باید فعال کنید. براي فعال کردن آن به منوي Tools مرورگر رفته و گزینه Internet Options را انتخاب کنید، در پنجره باز شده به قسمت Advanced رفته و گزینه notification about script error Display a را تیک بزنید، بهتر است که قبل از شروع کد نویسي جاوااسکریپت این عمل را انجام دهید تا هنگام بروز خطا در صدد رفع آن برآیید که به این کار اشکال زدایي یا Debugging مي گویند. اگر هم این گزینه را فعال نکردید، هنگام بروز خطا مي توانید بر روي آن مثلث زرد رنگ در نوار وضعیت رفته و دوبار کلیک کنید که همان پنجره اعلان خطا باز خواهد شد.

۳-۲- انواع خطاها

اساساً دو نوع خطا براي جاوااسکریپت وجود دارد:

۱- خطاهاي نحوي یا Syntax errors - مشکلي در ترکیبات کدها بوجود آمده

۲- خطاهاي زمان اجرا یا Run time errors - احتمالاً یک دستور بطور اشتباه جاي یک دستور دیگر نوشته شده است

۳-۳- ایجاد یک خطا

کد زیر را با نام error.htm ذخیره کنید:

```
<html>
<head>
<title> Error message </title>
</head>
<body>
```

```
<script language="javascript" type="text/javascript">
document.write("I'm ready for first error message.");
</script>
</body>
</html>
```

يكبار آنرا اجرا كنيد و نتيجه را مشاهده كنيد. حالا پرانتز دوم راحذف مي كنيم، بدين صورت:

```
document.write("I'm ready for first error message.";
```

سپس فايل را دوباره ذخيره كرده و اجرا مي كنيم. پنجره پيغام خطا باز مـي شـود، اگـر جزئيـات پيغـام را نداريـد روي دكمـه Details يا Show Details كليک كنيد تا ببينيم كه چه اطلاعاتي را مي توانيم بدست آوريم. در كادر جزئيات اولين گزينه Line مي باشد كه اگر مطابق كد بالا عمل كرده باشيد بايد درجلوي آن عدد ۷ نوشته شده باشد. اين خط به ما مي فهماند كـه ايراد در خط هفتم سورس كد است. دقت كنيد، تنها خطهاي مربوط به جاوااسـكريپت را نبايـد بخوانيـد بلكـه كليـه خطـوط از بالا به پايين خوانده مي شود، حتي خطهاي خالي هم بايد شمرده شوند.

دومين گزينه، كلمه Char است كه بيانگر عدد شمارش شده كاراكتري كه باعث بروز خطـا شـده اسـت و در اينجـا آن عـدد ۵۲ مي باشد. فراموش نكنيد كه هر فاصله بين كلمات هم يک كاراكتر به حساب ميايد. شـايد اينجـا سـؤالي برايتـان پـيش بيايد كه چرا عدد ۵۲ ؟ در صورتي كه در اينجا پرانتز كاراكتر۵۱ است. بايد يک نكتـه را خـدمتتان عـرض كـنم كـه علامـت ; در جاوااسكريپت يک كاراكتر اختياري است كه نشان دهنده پايان يک خط از كد مي باشد و برنامه نويسان بـر حـسب عـادتي كه دارند از اين علامت استفاده مي كنند ، در اينجا مفسر جاوااسكريپت آنرا به عنوان يک كاراكتر قبل از كاراكتر اصلي خطا شناسايي كرده است.

گزينه بعدي Code ميباشد كه ظاهراً بطور معمول عدد صفر جلوي آن ميباشد و هنوز نتوانستم كارآيي آنرا تـشخيص بـدهم و گزينه آخر هم URL است كه آدرس فايل اچ تي ام ال است كه در آن خطا توليد شده است.

در مجموع شما براي اشكال زدايي به سه گزينه اول نياز داريد كه با توجه به آنها بلافاصله مي توانيد محـل بـروز اشـكال را تشخيص دهيد. يک نكته را به خاطر بسپاريد كه در كدهاي طولاني ممكن است با تعداد خطاهاي زيـادي برخـورد كنيـد كـه در بيشتر مواقع برطرف كردن اولين خطا باعث رفع كليه اشكالات مي شود.

۴- در كجا بايد كدهاي جاوااسكريپت را نوشت

شـما ميدانيـد كـه بـراي اجـرا شـدن كـدهاي جاوااسـكريپت بايـد آنهـا را در ميـان تگهـاي اچ تـي ام ال بكـار بريـد. كـدهاي جاوااسكريپت مي توانند در هر كجاي سورس كد باشند اما بايد ابتدا تشخيص دهيد كه بكاربردن هـر قـسمت از اسـكريپت در كجا مناسب تر مي باشد. بطور كل كدهاي جاوااسكريپت به سه دسته تقسيم ميشوند، يـک سـري از آنهـا در قـسمت سر يعني head جاي مي گيرند و يک سري در قسمت بدنه كدهاي اچ تي ام ال و دسته سوم هم در يک فايـل جداگانـه و خارج از كدهاي html با پسوند js قرار مي گيرند.

۴-۱- نوشتن اسكريپت در قسمت head

كليه كدهاي جاوااسكريپت به محض باز شدن يک صفحه وب در مرورگر اجرا مي شوند اما گاهي اوقات ما نياز داريم تـا يـک اسكريپت زودتر از كدهاي ديگر اجرا شده و يا آماده باشـد تـا در صـورت لـزوم فراخـواني شـود. كـه اينگونـه كـدها را بايـد در قسمت سر و ما بين تگ head بنويسيم. بطور مثال شما نياز به اين داريد كه به محض ورود كاربر به صفحه در همـان ابتـدا پيغامي ظاهر شود و تا كاربر آنرا تأييد نكرده است صفحه تكميل نشود. اسكريپتهاي قسمت سر هميـشه آمـاده اجـرا و يـا آماده پاسخ به اتفاقي در صفحه وب هستند، كه در اين مورد شما مطمئن هستيد اسكريپت مورد نظر قبل از هر چيـزي در صفحه، اجرا خواهد شد. براي نوشتن كدهاي جاوااسكريپت بصورت زير عمل مي كنيد:

```
<html>
<head>
<title> ... </title>
<script language="javascript" type="text/javascript">
..... javascript codes .....
</script>
</head>
```

٤-٢- نوشتن اسکریپت در قسمت بدنه

هنگامیکه نیاز داشته باشید تا کدهاي جاوااسکریپت در بدنه فایل html اجرا شود آنها را در قسمت body صفحه وارد مي کنید،

```
<html>
<head>
</head>
<body>
<script language="javascript" type="text/javascript">
..... javascript codes .....
</script>
</body>
</html>
```

شما حتي مي توانید توسط کدهاي جاوااسکریپت قسمتي از یک صفحه را بسازید یعني با ترکیب جاوااسکریپت و تگهاي اچ تي ام ال در body یک عنصر در صفحه وارد کنید. کدهاي زیر را بنویسید و اجرا کنید:

```
<html>
<head>
</head>
<body>
<script language="javascript" type="text/javascript">
document.write("<h1 align='center'> Hello my friends! </h1>");
</script>
</body>
</html>
```

٤-٣- فایل خارجي جاوااسکریپت

براي حفظ نظم تگهاي اچ تي ام ال در یک صفحه وب و یا براي جلوگیري از تکرار یک اسکریپت در صفحات یک وب سایت، کدهاي جاوااسکریپت را در یک فایل جداگانه نوشته و با پسوند js آنها را ذخیره مي کنیم سپس در هر قسمت و هر صفحه اي که لازم بود، وارد مي کنیم.

مثال : خط زیر را در برنامه Notepad نوشته و آنرا به اسم test.js ذخیره کنید:

```
document.write(" JavaScript external file ")
```

حالا یک فایل اچ تي ام ال بسازید و تگهاي زیر را در آن وارد کنید سپس با یک اسم دلخواه آنرا در همان پوشه که فایل جاوااسکریپت را ذخیره کرده اید قرار دهید:

```
<html>
<head>
</head>
<body>
<script language="javascript" type="text/javascript" src="test.js">
</script>
</body>
</html>
```

اگر مراحل را بدرستي انجام داده باشید، جمله زیر را در مرورگرتان مشاهده خواهید کرد.

JavaScript external file

۵- متغیرها در جاوااسکریپت

۵-۱- متغیر چیست؟

متغیرها، کد نویسي را راحت تر کرده و نظم خاصي به اسکریپت ما مي بخشند.

مثال: تعریف و استفاده از متغیرها

```
<script language="javascript" type="text/javascript">
var strTest = "This is an example for variable."
document.write(txtTest)
</script>
```

اصطلاح var از کلمه variable مي آید که نوشتن آن اختیاري مي باشد، قسمت بعدي یعني strTest نامي است که براي آن متغیر انتخاب مي کنید و سپس علامت مساوي است که متغیر را برابر با یک مقدار قرار مي دهد و در آخر هم مقدار تعیین شده براي متغیر که در اینجا یک رشته متني است یا یک string text میباشد که به همین دلیل باید آنرا بین دو علامت " " قرار دهیم تا عیناً محتویات آن در مرورگر نوشته شود.

۵-۲- انتخاب نام متغیر

بهتر است نام یک متغیر را طوري انتخاب کنید که در عین سادگي، گویا بوده و یادآور نکاتي براي شما باشد.

- نام یک متغیر مي تواند از مجموع حروف، اعداد و علامت underline یا زیرین خط یعني _ تشکیل شود، که آغاز کننده نام نباید یک عدد باشد و همچنین هیچگونه فاصله اي هم در آن نباید باشد.
- در جاوااسکریپت حروف کوچک با بزرگ متفاوت مي باشند یعني strTest با strtest کاملاً تفاوت دارد که اصطلاحاً مي گویند جاوااسکریپت case sensitive است.

۵-۳- انواع متغیرها

دو نوع متغیر وجود دارد متغیرهاي کلي و سراسري که به آنها global گفته مي شود. این متغیرها در سراسر یک فایل اچ تي ام ال کاربرد دارند و متغیرهاي محلي یا local که فقط در یک قسمت از اسکریپت کارآیي دارند که ممکن است در قسمت دیگري همان متغیر با یک مقدار دیگري تعریف شود. معمولاً متغیر هاي محلي در توابع یا function ها کاربرد دارند.به قسمتي از اسکریپت که یک متغیر در آن کاربرد دارد، حوضه یا scope آن متغیر گفته مي شود.

۶- تاریخ و زمان در جاوااسکریپت

٦-١- Date Object

اشیاي در جاوااسکریپت وجود دارند که توسط متدهاي مختلف، اطلاعات مورد نیاز را از سیستم گرفته و در اختیار کاربران قرار مي دهند. یکي از آنها ، شيء Date مي باشد که به کمک آن مي توانیم تاریخ و زمان سیستم را هنگام اجراي کد دریافت کنیم، این object بصورت زیر تعریف مي شود:

```
new Date()
```

این شئ یک رشته متني شامل روز، ماه، سال، ساعت، دقیقه، ثانیه و اختلاف زماني با UTC را برمیگرداند. UTC مخفف Universal Time Coordinated مي باشد که به آن Zulu time هم مي گویند و مطابق با استاندارد جهاني زمان است.

```
<script language="javascript" type="text/javascript">
document.write ( new Date() )
</script>
```

٦-٢- متدهاي شئ Date

بعد از ساختن شئ جديد Date ، مي توانيد توسط متدهاي مربوط به آن جزئيات ديگري از اين شئ دريافت كنيد. اين متدها عبارتند از:

getDate(), getDay(), getMonth(), getYear(), getHours(), getMinutes(), getSeconds()

مثال : نمايش زمان ورود به صفحه

```
<script language="javascript" type="text/javascript">
var vorood = new Date()
document.write( vorood.getHours() + ":" + vorood.getMinutes() + ":" + vorood.getSeconds() )
</script>
```

نكته: هنگاميكه از متد getMonth() براي برگرداندن عدد ماه استفاده مي كنيد بايد هميشه آنرا بعلاوه عدد يک كنيد چون جاوااسكريپت اولين ماه ميلادي را برابر با عدد صفر قرار مي دهد.

```
document.write( new Date().getMonth()+1 )
```

در مورد متد getYear() هم اگر مي خواهيد نتيجه كامل برگردد يعني سال بصورت چهاررقمي نوشته شود بايد از متد getFullYear() استفاده كنيد.

٦-٣- تنظيم تاريخ و زمان

شئ Date به غير از متدهاي ذكر شده در بالا، داراي متدهاي ديگري است كه توسط آنها مي توانيد زمان و تاريخي را بصورت دلخواه تنظيم كنيد. در اينگونه متدها بجاي كلمه get بايد از set استفاده كنيد و مقدار معيني را در پرانتز وارد كنيد. بطور مثال مي خواهيم همين موقع را در سال ۲۰۰۸ داخل صفحه چاپ كنيم.

```
var zaman = new Date()
zaman.setFullYear(2008)
document.write ( zaman )
```

٧- عملگرها در جاوااسكريپت

عملگرها در جاوااسكريپت به چند دسته تقسيم مي شوند:

٧-١- عملگرهاي رياضي

عملگر	توصيف	مثال	نتيجه
+	جمع	x = 2, y = 2 x+y	4
-	تفريق	x = 5 , y = 3 x - y	2
*	ضرب	x = 5 , y = 4 x * y	20
/	تقسيم	x = 20 , y = 4 x / y	5
%	خارج قسمت	x = 19 , y = 4 x % y	3
++	افزودن	x = 5 x++	x = 6
--	كم كردن	x = 5	x = 4

		x--	

عملگرهاي زير به نام **Operators Assignment** معروف هستند كه يک مقداري را به يک متغير نسبت مي دهند.

عملگرها	مثال	برابر است با
=	x = y	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

۷-۲- عملگرهاي مقايسه اي

عملگرها	توصيف	مثال
==	مقادير اگر برابر باشند نتيجه درست خواهد بود.	x = 5 , y = 5 x == y
===	مقايسه بين مقادير و جنس متغير كه در مثال روبرو نتيجه غلط بازگردانده مي شود	x = 5 y = "5" x === y
!=	اگر مقادير برابر نباشند نتيجه درست است	5 != 8
>	بزرگتر	8 > 5
<	كوچكتر	8 < 10
>=	بزرگتر يا مساوي كه در اين مثال نتيجه غلط است	5 >= 8
<=	كوچكتر يا مساوي كه در اين مثال نتيجه درست است	5 <= 8

۷-۳-عملگرهاي منطقي

&& , || , !

عملگر && در حقيقت مانند حرف اضافه " **و** " در فارسي عمل مي كند:

```
x = 5
y = 3
( x < 10 && y > 1)
```

در اينجا اگر مقدار متغير x كوچكتر از عدد ۱۰ باشه **و** y بزرگتر از ۱ باشه، نتيجه true بازگردانده مي شود. چون حرف **و** آمده است پس بايد دو طرف عملگر جواب درست باشد تا در مجموع نتيجه درست در بيايد.

عملگر || مانند كلمه " **يا** " در فارسي است:

```
x = 6
y = 3
( x == 5 || y == 5 )
```

نتیجه غلط است یعني false بازگردانده مي شود. چون **یا** آمده است پس حداقل یکي از طرفین عملگر باید درست باشد تا نتیجه درست در بیاید اما اینجا هر دو طرف غلط است که در مجموع نتیجه غلط است. اما در حالت زیر نتیجه درست بازگردانده مي شود:

```
( x == 5 || y < 5 )
```

عملگر ! در حقیقت یک عملیات و معادله را بررسي مي کند اگر جواب آن معادله درست نباشد نتیجه true یا همان درست را برمیگرداند:

```
x = 5
y = 4
! ( x == y )
```

در اینجا در کل نتیجه درست است.

۷-٤- کاربرد عملگرها در متن

گاهي اوقات از بعضي از عملگرها مي توانیم براي رشته هاي متني هم استفاده کنیم:

```
strText01 = " Hello "
strText02 = "my friends!"
document.write ( strText01 + strText02 )
```

۸- دستورات شرطي

۸-۱- دستور if

این دستور دو قسمت دارد: یکي قسمت شرط و دیگري قسمت نتیجه شرط است. به کد زیر توجه کنید:

```
<script language="javascript" type="text/javascript">

var d = new Date()
var t = d.getHours()

if ( t < 10 )
{
document.write ( "<b> Good morning </b>")
}

</script>
```

۸-۲- دستور if...else

اگر ساعت قبل یا برابر ۱۰ بود **پس** جمله *صبح بخیر* نمایش داده شود **در غیر اینصورت** متن *سلام دوستان* ظاهر شود. در کد بجاي قسمت در غیر اینصورت مي توانیم کلمه else را وارد کنیم تا به مفهوم مورد نظر برسیم.

```
if ( t < 10 || t == 10 ) {
document.write ( "<b> Good morning </b>" )
}
else
{
document.write ( "<b> Hello friends </b>")
}
```

۸-۳- دستور if...else if...else

این دستور را زماني مي توانيد استفاده کنيد که بخواهيد يک شرط در شرايط مختلف بررسي شود تا نتيجه هاي متفاوت هم بدست بيايد:

```
if ( t <= 10  ) {
document.write ( "<b> Good morning </b>" )
}
else if ( t > 10 && t < 13 ) {
document.write ( "<b> Hello friends </b>" )
}
else if ( t >= 13 && t < 18 ) {
document.write ( "<b> Good afternoon </b>" )
}
else {
document.write ( "<b> Good evening </b>" )
}
```

۸-٤- دستور switch

دستور switch چندين شرط را براي متغيرها مقايسه مي کند تا نتايج مختلف بدست بيايد. در حقيقت دستور switch از تکرار if else جلوگيري مي کند. در مواردي که شرطها و نتايج آنها کم هستند مي توانيد از دستور if else استفاده کنيد اما براي شرطهاي زياد بهتر است switch را بکار بريد تا کد شما منظم تر باشد.

مي خواهيم يک صفحه درست کنيم که به محض ورود کاربر به آن، آنروز هفته را در صفحه نمايش دهد:

```
<script language="javascript" type="text/javascript">

var d = new Date ()
var weekday = d.getDay ()

switch ( weekday )
 {
case 0 :
        document.write ("Sunday")
        break
case 1:
       document.write ("Monday")
        break
case 2:
       document.write ("Tuesday")
       break
case 3:
       document.write ("Wednesday")
       break
case 4:
       document.write ("Thursday")
        break
case 5:
       document.write ("Friday")
       break
default:
       document.write ("Saturday")
}
</script>
```

۱۰- ساختارهاي تكرار

۱۰-۱- حلقه for

در اين حلقه، مجموعه كد مورد نظر به تعداد مشخص تكرار مي شود كه در ابتداي حلقه تعداد دفعات اجراي كد بايد اعلام شود. بطور مثال مي خواهيم اعداد از ۰ تا ۱۰ در صفحه چاپ شوند:

```
<script language="javascript" type="text/javascript">

for ( i = 0; i <= 10; i++ )
{
document.write ( " Number " + i + "<br />")
}
</script>
```

۱۰-۲- حلقه while

در مواردي كه مقدار انتهايي حلقه مشخص نيست مي توانيم از اين نوع استفاده كنيم چون كد مورد نظر تا جايي كه نتيجه دلخواه بدست بيايد تكرار خواهد شد. مثال بالا را در مورد اين حلقه مي توانيم به اين صورت ذكر كنيم:

```
var i = 0
while (i <= 10)
{
document.write(" Number " + i + "<br />")
i++
}
```

كلمه while را مي توانيم ترجمه كنيم به ***مادامیکه*** ، مادامیکه شرط كوچكتر يا مساوي ۱۰ بودن برقرار نيست حلقه را تكرار كن و يک واحد به آن اضافه كن تا بالاخره شرط ما درست دربيايد.

۱۰-۳- حلقه do...while

اين نوع حلقه برعكس حلقه while مي باشد. در اين حلقه، شرط در آخر بدنه حلقه بررسي مي شود. فرق اين حلقه با حلقه while در اينست كه حداقل يكبار دستورات اجرا خواهند شد ولي در حلقه while ممكن بود در صورت برقرار بودن شرط در همان ابتدا ديگر دستورات اجرا نشوند. مثال:

```
 var i=0
do
{
document.write(" Number " + i + "<br />")
i=i+1
}
while (i <= 10);
```

۱۰-٤- break

هنگاميكه شما نياز به نوشتن حلقه هاي پيچيده تري داريد، بايد دقت كنيد كه حلقه شما دچار حلقه بي نهايت يا همان حلقه نامتناهي نشود. به اين منظور مي توانيد از دستور break استفاده كنيد و يک شرطي هم براي آن در نظر بگيريد تا از حلقه هاي بي نهايت جلوگيري شود.

```
var i = 11
while (i >= 10)
{
```

```
document.write(" Number " + i + "<br />")
i++
if ( i == 21 ) break
}
```

۱۱- آرایه

۱۱-۱- آرایه چیست؟

آرایه یکي دیگر از امکانات جاوااسکریپت مي باشد، بهترین خاصیت این دستور، کوتاه کردن اسکریپت ما است. شئ آرایه یا همان Array براي ذخیره کردن مقادیر متفاوت در یک متغیر بکار مي رود. توسط یک آرایه شما مي توانید چندین مقدار را به یک متغیر نسبت دهید. بطور مثال مي توانید اسامي روزهاي هفته را در یک متغیر ذخیره کنید. دستور آرایه داراي قسمتهایي است که با توجه به مثال زیر به توضیح آنها مي پردازیم:

```
var weekdays = new Array(7)
weekdays[0] = "Shanbeh"
weekdays[1] = "Yekshanbeh"
weekdays[2] = "Doshanbeh"
weekdays[3] = "Seshanbeh"
weekdays[4] = "Chaharshanbeh"
weekdays[5] = "Panjshanbeh"
weekdays[6] = "Jomeh"
```

مثال بالا را مي توان بصورت زیر هم نوشت:

```
var weekdays = new Array ("Shanbeh","Yekshanbeh",...)
```

یعني بجاي نوشتن اندیس کلي در پرانتز، مقادیر هر اندیس را بنویسیم و توسط کاما آنها را از هم جدا کنیم. مي خواهیم هنگام باز شدن صفحه وب، روزهاي هفته به ترتیب نمایش داده شوند.

```
 for ( i=0; i<7; i++ ) {
document.write ( weekdays[ i ] + "<br />" )
}
```

۱۱-۲- خصوصیت آرایه

خصوصیتي که بیشترین کاربرد را براي آرایه دارد، length مي باشد. وظیفه این خصوصیت نمایش اندیس کلي یک آرایه است. اگر بخواهیم در مثال بالا این خصوصیت را بکار بریم، باید خط زیر را بعد یا قبل از حلقه for بکار بریم:

```
document.write ( weekdays.length + "<br />" )
```

بعد از وارد کردن این کد و اجراي اسکریپت، عدد هفت در صفحه نوشته خواهد شد چون اندیس ما در اینجا هفت بود.

۱۱-۳- متدهاي آرایه

آرایه داراي متدهایي هم هست که نحوه تعریف کردن آنها همانند خصوصیت است متدهاي آرایه عبارتند از:

concat(), join(), pop(), shift(), push(), unshift(), reverse(), slice(), sort(), splice(), toString(), valueOf()

متد concat براي ترکیب مقادیر دو یا سه آرایه بکار مي رود. این متد هیچ چیزي را در آرایه ها تغییر نمي دهد و فقط مقادیر تعریف شده را در صفحه چاپ مي کند.

متد join تقريباً مانند متد قبلي عمل مي كند و كليه مقادير را بصورت يک رشته متني در صفحه نشان مي دهد با اين تفاوت كه در اين متد مي توانيد يک علامت مانند نقطه، تعيين كنيد تا بين مقادير چاپ شده در صفحه قرار گيرد. بصورت پيش فرض اين علامت، كاما مي باشد.

متد pop آخرين مقدار آرايه را جدا كرده و در صفحه نمايش مي دهد. دقت كنيد كه اين متد در خصوصيت length تأثير مي گذارد يعني اگر بعد از اين متد خصوصيت length را بكار بريد، يک انديس كمتر نشان مي دهد.

متد shift برعكس متد قبلي است يعني اين متد اولين مقدار آرايه را جدا مي كند و نشان مي دهد.

متد push يک يا چند مقدار را به آخر يک آرايه اضافه مي كند و انديس جديد برمي گرداند.

متد unshift يک يا چند مقدار به ابتداي يک آرايه اضافه مي كند و انديس جديد بر مي گرداند. دقت كنيد كه اين متد در مرورگر اينترنت اكسپلورر بدرستي كار نمي كند.

متد reverse ترتيب نمايش مقادير يک آرايه را برعكس كرده و در صفحه وب نمايش مي دهد.

متد slice يكسري از مقادير موجود در يک آرايه را جدا مي كند كه در اين متد شماره انديس مقداري كه مي خواهيد از آنجا جداسازي شروع شود را بايد بنويسيد و همچنين مي توانيد آخرين مقدار هم مشخص كنيد كه اگر آنرا معين نكنيد تا آخرين مقدار انتخاب خواهد شد.

متد sort براي منظم كردن مقادير يک آرايه به ترتيب حروف الفباي انگليسي است. اگر هم مقادير يک آرايه، عدد باشند به ترتيب اعداد تنظيم خواهند شد كه مفسر جاوااسكريپت به اولين عدد از سمت چپ نگاه مي كند و بعد از رديف كردن آنها سپس به اعداد بعدي نگاه مي كند.

متد splice براي اضافه يا حذف يک مقدار در آرايه مي باشد. اولين پارامتر شماره انديس كه در آنجا يک مقدار حذف يا اضافه خواهد شد و بايد عدد باشد. دومين پارامتر مقدار عددي است كه معين مي كند چند تا مقدار از آن انديسي كه در پارامتر قبل انتخاب كرديم، بايد حذف شود كه اگر آنرا صفر در نظر بگيريد هيچ مقداري حذف نخواهد شد. اين دو پارامتر حتماً بايد نوشته شوند. در ضمن اگر مي خواهيد مقاديري هم اضافه شود بعد از اين پارامتر ها به عنوان پارامترهاي بعدي ذكر مي كنيد.

متد toString مقادير يک آرايه را تبديل به يک رشته متني مي كند.

۱۲- توابع

۱۲-۱- تابع چيست؟

توابع يا همان functions در جاوااسكريپت مي توانند كمک بزرگي به سازمان دهي اسكريپت ها كنند. مي تواند يک مجموعه كد را كه بايد در جاهاي مختلف تكرار شود را به صورت يک تابع تعريف كرد و در هر جاي اسكريپت به آن نياز بود آنرا فراخواني كرد و از نوشتن مجدد پرهيز كرد.

```
function message ( )
{
alert ( " Hello friends! " )
}
```

۱۲-۲- فراخواني يک تابع

توابعي كه در يک سند اچ تي ام ال تعريف مي شوند به خودي خود اجرا نخواهند شد. توابع پس از تعريف بايد در جاي مورد نظر فراخواني شوند . در حقيقت آنها آماده هستند تا پس از فراخواني اجرا شوند. هر جا كه نياز به دستورات آن تابع داشتيد، نامش را بنويسيد:

```
<html>
<head>
```

```
<title> Call function </title>

<script type="text/javascript" type="text/javascript">
function message ( )
{
alert ( " Hello friends! " )
}
</script>

</head>
<body>

<form>
<input type="button" onclick="message( )" value="Click me">
</form>

</body>
</html>
```

۱۲-۳- آرگومان یک تابع

گاهي اوقات هنگام تعريف يک تابع، آرگومان و يا پارامتري را براي آن داخل پرانتز تعيين مي کنيد تا هنگام صدا زدن تابع آن آرگومان مقداردهي شود. در کل مي توان گفت که يک آرگومان، متغيري است که هنگام صدا زدن تابع مقدار آن ارسال مي شود تا تابع از آن استفاده کند. مي توانيم مثال بالا را با يک آرگومان تعريف کنيم:

```
function message ( msg )
{
alert ( msg )
}
</script>

<input type="button" onclick="message( 'Hello friends!' )" value="Click me">
```

در اينجا يک آرگومان بنام msg تعيين کرديم تا هنگام فراخواني تابع، مقدارش به تابع فرستاده شود و نتيجه آن اجرا شود. در حقيقت عبارت Hello friends مقدار اين آرگومان مي باشد. تعيين آرگومان هيچ محدوديتي ندارد و فقط بايد آنها را با علامت کاما از هم جدا کنيد.

دقت کنيد که در دستورات جاوااسکريپت، بين دو علامت " " بايد از ' ' استفاده شود چون اگر مجدداً از خود علامت نقل قول يعني " " استفاده کنيد با پيغام خطا مواجه مي شويد.

۱۲-٤- بازگرداندن يک مقدار توسط دستور return

توابع جاوااسکريپت داراي يک دستوري بنام return هستند که وظيفه اين دستور برگرداندن مقادير تعريف شده در آن تابع است.

```
function greeting ( ) {
return ( " Hello world! " )
}
.......

document.write ( greeting( ) )
```

اين مجموعه کد را اگر بدون استفاده از return اجرا کنيد، نتيجه درست نخواهيد گرفت. به يک مثال ديگر توجه کنيد:

```
<html>
<head>
<title> return statement </title>

<script type="text/javascript" type="text/javascript">
function total (a,b )
{
x = a * b
return x
}
</script>

</head>
<body>
<script type="text/javascript" type="text/javascript">
price = total ( 4, 15 )
document.write ( price )
</script>
</body>
</html>
```

همانطور که مشاهده مي کنيد دو آرگومان براي تابع total تعريف شده است که در متغير x ذخيره مي شوند و توسط دستور return مقدارشان به تابع ارسال خواهد شد.

در نظر داشته باشيد که توابع در جاوااسکريپت بسيار کاربرد دارند و شما هنگام نوشتن اسکريپت خود پي به اهميت آنها خواهيد برد مخصوصاً زماني که يک سري کد را بخواهيد در کليه صفحات يک وب سايت بکار بريد که مي توانيد آنها را بصورت يک تابع در يک فايل خارجي جاوااسکريپت ذخيره کنيد و سپس در هر صفحه آن تابع را فراخواني کنيد.

۱۳- پنجره هاي ارتباط با کاربر

سه نوع پنجره در جاوااسکريپت وجود دارند که مي توانيد توسط آنها پيغامي را در صفحه براي کاربر نمايش دهيد و يا اطلاعاتي براي تأييد آنها بفرستيد و يا اينکه اطلاعاتي را از آنها بخواهيد. آنها پنجره هاي کوچکي هستند که چون خيلي سريع در صفحه بالا آمده و در مقابل کاربر ظاهر مي شوند، به آنها popup گفته مي شود.

اين پنجره ها عبارتند از: پنجره alert ، پنجره confirm و پنجره prompt .

۱۳-۱- پنجره Alert

پنجره alert ، حاوي يک اخطاريه براي کاربر است که در آن هشداري داده مي شود تا کاربر متوجه کاري که مي کند باشد. دستور اين پنجره به صورت زير است:

```
alert (" text ")
```

اين پنجره فقط داراي يک دکمه OK است که کاربر بعد از خواندن آن اخطاريه با زدن دکمه ok ، پنجره را مي بندد.

```
<html>
<head>
<script type="text/javascript" type="text/javascript">
function alertbox ( ) {
alert ( " This is an Alert box! " )
}
</script>
</head>
```

```
<body>
<form>
<input type="button" onclick="alertbox ( )" value="Click here">
</form>
</body>

</html>
```

دقت کنید که در این نوع پنجره یک علامت مثلث زرد که نشان دهنده علامت اخطار هم هست، ظاهر مي شود.

۱۳-۲- پنجره Confirm

این مدل پنجره ، همانطور که از اسمش پیداست براي گرفتن تأیید از کاربر اجرا مي شود. این پنجره داراي دو دکمه OK و Cancel مي باشد که با زدن دکمه OK توسط کاربر مقدار true بازگردانده مي شود و با زدن Cancel مقدار false برگردانده مي شود. دستور این پنجره به این صورت است:

```
confirm ( " text " )
```

این پنجره براي مواقعي خوب است که شما نیاز به جواب کاربر براي ادامه مطلبي یا چیزي دارید، در صورتیکه کاربر تأیید کند آن مطلب ادامه پیدا مي کند و در غیر اینصورت همانجا متوقف خواهد شد.

۱۳-۳- پنجره prompt

از این پنجره زماني استفاده مي شود که شما نیاز داشته باشید تا اطلاعاتي را از کاربر دریافت کنید و آنرا در جایي مناسب بازگردانید. معمولاً اجراي این پنجره هنگام باز شدن صفحه وب است و داراي دو قسمت یا دو پارامتر مي باشد:

```
prompt ( "text or question", " default value " )
```

در قسمت اول یک متن که معمولاً سؤال است را وارد مي کنید تا در پنجره ظاهر شود و در قسمت دوم یک مقدار پیش فرض براي راهنمایي کاربر وارد مي کنید که معمولاً آن قسمت را خالي مي گذارند تا کاربر آن مقدار را وارد نماید. البته اگر هم شما مقدار پیش فرض را وارد نمایید، باز کاربر قادر به وارد کردن مقدار میباشد.

۱۴- رویدادها و حوادث

رویدادها و یا حوادث در جاوااسکریپت، کدهاي کوتاهي هستند که بین اقدامات انجام شده توسط کاربر مانند کلیک کردن موس و اسکریپت، ارتباط برقرار مي کنند. یعني هنگامیکه کاربر بر روي دکمه هاي موس فشار مي آورد، رویدادها به اسکریپت این موضوع را مي فهمانند تا اسکریپت براي آن حرکت عکس العملي در نظر بگیرد.

این اقدامات شامل حرکاتي است مانند کلیک کردن دکمه هاي موس، قرار گرفتن نشانگر موس بر روي یک نقطه از صفحه، فشردن کلیدهاي کیبورد، انتخاب یا تغییر در اجزاي یک فرم، فشردن دکمه هاي فرم و یا باز و بستن یک صفحه وب.

در کل رویدادها باعث مي شوند تا جاوااسکریپت اینگونه حرکات را شناسایي و درک کند.

رویدادها مانند بقیه خصوصیات تگهاي اچ تي ام ال هستند که باید در میان تگها قرار گیرند و آنها را برابر با یک کد کوتاه جاوااسکریپت و یا اسم یک تابع تعریف شده باید قرار داد.

در صفحات قبل با بعضي از رویدادها آشنا شدید و حالا در اینجا کلیه آنها را یاد خواهید گرفت. لازم بذکر است که، همه رویدادها در تمام مرورگرها اجرا نمي شوند.

١٤-١-رويدادهاي موس

زمانيكه مي خواهيد جاوااسكريپت نسبت به حركات موس، عكس العمل نشان دهد از اين رويدادها استفاده مي كنيد. به سورس كد مثالها دقت كنيد تا با كاربرد آنها بيشتر آشنا شويد.

onClick براي كليک كردن دكمه چپ موس توسط كاربر.

onDblClick رويداد براي دوبار كليک دكمه چپ موس.

onMouseDown فشار آوردن بر روي دكمه چپ موس.

onMouseUp رها كردن دكمه موس پس از فشردن آن.

onMouseOver قرار گرفتن نشانگر موس بر روي يک چيزي.

onMouseOut خارج شدن موس از روي يک چيزي.

١٤-٢- رويدادهاي كيبورد

در اين رويدادها، جاوااسكريپت نسبت به فشردن يک كليد بر روي كيبورد توسط كاربر، عكس العمل نشان خواهد داد.

onKeyPress فشردن يک كليد كيبورد.

onKeyDown مانند رويداد بالا براي فشردن يک كليد.

onKeyUp رها كردن يک كليد فشرده شده.

١٤-٣- رويدادهاي فرم

اين رويدادها مربوط به عملياتي است كه كاربر بر روي اجزاي فرم انجام مي دهد. البته ممكن است بعضي از اين رويدادها بر روي متنهاي عادي در صفحه هم عكس العمل نشان دهند اما بيشترين كاربرد را در فرمها دارند.

onFocus هنگاميكه توسط موس و يا دكمه Tab كيبورد بر روي يكي از اجزاي فرم متمركز شويد.

onBlur هنگاميكه تمركز از روي يكي از اجزاي فرم خارج شود.

onSelect انتخاب كردن يک متن در صفحه يا در يک فرم.

onChange تغيير در اجزاي فرم، مانند تغيير متن در كادر متني.

onSubmit فشردن دكمه تأييد يک فرم.

onReset فشردن دكمه reset يا همان پاک كردن اطلاعات فرم.

١٤-٤- رويدادهاي يک صفحه

يکسري از رويدادها هم مربوط به صفحات وب مي باشد، مانند لود شدن صفحه يا بستن آن.

onLoad تکميل شدن يک صفحه وب يا يک تصوير در مرورگر.

onUnload بستن مرورگر يا خروج از يک صفحه وب.

onResize تغيير اندازه پنجره مرورگر.

يک رويداد ديگر هست که مي توان هم براي يک صفحه و هم براي يک عکس بکار برد:

onError اين رويداد زماني اجرا مي شود که يک عکس به هر دليلي لود نشود يعني در جاي خود قرار نگيرد.

معمولاً براي رويدادها، ابتدا يک تابع تعريف مي شود تا هنگام نوشتن رويداد در يک تگ اچ تي ام ال، نظم برقرار باشد و از شلوغي تگ جلوگيري شود. بيشترين کاربرد اين رويدادها در تأييد اعتبار فرمها و پرسشنامه ها مي باشند که در يک بخش مجزا آنرا توضيح خواهم داد.

١٥- سلسله مراتب اشياء DOM

جاوااسکريپت، شئ هايي دارد که مي توانند بطور مستقيم با خود مرورگر ارتباط برقرار کنند و همچنين شئ هايي که با صفحه وب در ارتباط هستند. در کل مي توان اين اشياء را به دو دسته اصلي تقسيم کرد، دسته اول که BOM گفته مي شود يعني **Browser Object Model** شئ هايي که با پنجره مرورگر مي توانند ارتباط برقرار کنند و دسته دوم DOM که با Document يعني همان سند اچ تي ام ال. تمامي اين اشياء بر اساس يک سلسله مراتبي اين ارتباط را برقرار مي کنند که در رأس آنها شئ window مي باشد. براي ديدن اين سلسله مراتب اينجا را کليک کنيد.

همانطور که در شکل مشاهده کرديد از شئ window تا شئ Document جزو دسته BOM مي باشند و اشياء بعد از Document در دسته DOM قرار مي گيرند. ناگفته نماند که در حال حاضر کمتر صحبت اصطلاح BOM مي شود و کلاً اين مجموعه را DOM مي گويند.

شما بايد اين سلسله مراتب را براي استفاده اشياء آن، رعايت کنيد. اين سلسله مراتب را مي توان به شجره نامه تشبيه کرد و شما مجبور هستيد براي صدا کردن هر شخص آنرا با نام پدرش بياوريد و در بعضي جاها حتي نام پدر بزرگ هم لازم است، بطور مثال: علي فرزند محمد يا علي فرزند محمد فرزند حسين. در اينجا هم پدر اصلي Window مي باشد وپدر بعدي Document است که فرزندهاي آن همان عناصر اچ تي ام ال ميباشند و سپس Form و در آخر هم Select که تنها يک فرزند دارد. براي استفاده از آنها بايد ابتدا اسم پدر و سپس اسم فرزند را نوشت و در آخر هم خصوصيت و يا متد آن فرزند که همه اينها با يک نقطه از هم بايد جدا شوند. بطور مثال اگر ميخواهيد يک فرم را بنام form1 صدا بزنيد تا با آن ارتباط برقرار کنيد بايد به شکل زير عمل شود:

```
window.document.form1
```

مانند همه اشياء در جاوااسکريپت ، اين شئ ها هم داراي خصوصيات و متدهايي هستند که بعد از رعايت کردن سلسله مراتب با گذاشتن يک نقطه مي توانيد آنها را بنويسيد. مانند:

```
window.history.go (-1)
```

که اگر اين شئ را در يک لينک قرار دهيد با کليک کردن بر روي آن، کاربر به صفحه قبل بازگشت داده مي شود.

در جدول زير شئ هاي اصلي اين سلسله مراتب ليست شده است.

Window	ايـن شـئ مـستقيماً بـا پنجـره مرورگـر در ارتبـاط اسـت كـه معمـولاً خـصوصيات و متدهاي اين شئ در تگهاي body و frameset بكار مي روند.
Navigator	اين شئ شامل اطلاعاتي در رابطه با مرورگر كاربر است كـه توسـط خـصوصيات و متدهاي آن اين اطلاعات ذخيره تا مورد استفاده قرار گيرند.
Frame	براي كار كردن بر روي فريمهاي اچ تي ام ال.
Document	دسترسي و ارتباط با كليه عناصر اچ تي ام ال مانند، متنها فرمها عكسها و غيره.
Location	شـامل اطلاعـاتي در مـورد URL يـا همـان آدرس اينترنتـي صـفحه اي كـه در حـال نمايش است.
History	ايـن شـئ داراي اطلاعـاتي بـراي رجـوع كـاربر بـه صـفحاتي كـه قبـل از آن صـفحه مشاهده شده و بر عكس.
Screen	اطلاعات صفحه مرورگر كاربر را بر مي گرداند مانند اندازه طول و عرض.

شيء history

ليستي از صفحات نمايش داده شده توسط مرورگر را نگه داشته و امكان حركت بين آنها را فراهم مي كند.

خواص:

خاصيت	كاربرد
Length	تعداد عناصر موجود در ليست را دارد

متدها :

نام متد	كاربرد
back()	نمايش صفحه قبلي
forward()	نمايش صفحه بعدي
go(r)	نمايش r صفحه بعدي (قبلي)

شيء location

اين شيء اطلاعاتي در مورد آدرس صفحه نمايش داده شده و نيز تغييراتي در اين زمينه را فراهم مي كند.

خواص:

خاصيت	كاربرد
host	
href	
protocol	

متدها :

نام متد	كاربرد
reload()	بارگذاري مجدد صفحه جاري
replace(newURL)	جايگزيني صفحه فعلي با صفحه اي جديد

شيء screen

اين شيء اطلاعاتي در مورد صفحه نمايش ارائه مي دهد.

خواص:

خاصيت	كاربرد
availHeight	ارتفاع قابل استفاده صفحه نمايش (بجز نوار وظيفه)
availWidth	عرض "
height	ارتفاع صفحه نمايش
width	عرض صفحه نمايش

شيء navigator

اطلاعاتي در مورد برنامه مرورگر، سيستم عامل و غيره در اختيار قرار مي دهد

خواص:

خاصيت	كاربرد
appName	نام مرورگر وب
appVersion	شماره نسخه مرورگر

platform	پلت فرم مورد استفاده (نام سیستم عامل)

مدل اشياي سند(DOM)

این مدل تمام اشياي صفحه (تگ ها) را بصورت سلسله مراتبي شامل مي شود. هر کدام از اشيا داراي خواص و متدهاي مربوط به خود مي باشند. برنامه نویس جاوا اسکریپت به این اشیاء دسترسی پیدا کرده، خواص آنها را استفاده کرده و گاها" تغییر دهد، متدهای آنها را اجرا کند و نیز اشیای مشابهی به آن اضافه نماید.

شیء Anchor

این شیء به تگ های پیوند (A) در صفحه اشاره دارد

خواص:

خاصیت	کاربرد
href	آدرس (URL)
innerHTML	کد HTML داخل تگ A
target	نام پنجره میزبان

* هرکدام از تگ های دارای تگ شروع و پایان دارای خاصیت InnerHTML هستند که کد HTML داخل آنها را تعیین می کند.

شیء Image

این شیء به تگ های تصویر (IMG) در صفحه اشاره دارد

خواص:

خاصیت	کاربرد
align	ترازبندی (مقدار left,right,center)
alt	متن جایگزین
height	ارتفاع تصویر
width	عرض تصویر
src	آدرس و نام فایل تصویری
border	ضخامت کادر تصویر
complete	وضعیت کامل شدن دریافت تصویر (true,false)

شیء Form

این شیء به تگ های فرم (Form) در صفحه اشاره دارد

خواص:

خاصیت	کاربرد
action	آدرس و نام برنامه پردازنده فرم
method	شیوه ارسال
elements[]	آرایه ای شامل اجزای فرم

متدها:

نام متد	کاربرد
reset()	پاک کردن فرم
submit()	ارسال فرم

شیء Input

این شیء به تگ های INPUT در صفحه اشاره دارد

خواص:

خاصیت	کاربرد
checked	وضعیت انتخاب جعبه انتخابی مخصوص دکمه رادیویی و جعبه انتخاب
Value	مقدار کنترل ورودی
disabled	وضعیت فعال یا غیرفعال بودن کنترل

متدها:

نام متد	کاربرد
focus()	انتقال کنترل به دکمه
blur()	خروج کنترل از دکمه
click()	شبیه سازی عمل کلیک روی دکمه

شیء Select

این شیء به تگ های SELECT در صفحه اشاره دارد

خواص:

خاصیت	کاربرد
length	تعداد گزینه های لیست
size	اندازه (تعداد گزینه های قابل نمایش)
selectedIndex	اندیس گزینه انتخاب شده
options[]	آرایه ای حاوی گزینه های لیست

متدها:

نام متد	کاربرد
focus ()	انتقال کنترل به لیست
blur()	خروج کنترل از لیست
add()	اضافه کردن گزینه به لیست
remove()	حذف کردن گزینه از لیست remove(index)

شیء Table

این شیء به تگ های TABLE در صفحه اشاره دارد

خواص:

خاصیت	کاربرد
width	عرض جدول
border	ضخامت کادر جدول
rows[]	آرایه ای شامل اشیای برای سطرهای جدول
cells []	آرایه ای حاوی اشیایی برای سلول های جدول

خاصیت style

هر کدام از اشیای قابل نمایش در صفحه علاوه و خواص اصلی شان دارای یک خاصیت بنام style هستند که آن هام شامل مجموعه ای از صفات برای تنظیم خصوصیات ظاهری اشیا است. این خواص بصورت زیر استفاده می شوند:

Object.style.property = value

خاصیت (property)	کاربرد
backgroundColor	رنگ پس زمینه
backgroundImage	تصویر پس زمینه
border	نوع و ضخامت و رنگ کادر
borderBottom	نوع و ضخامت و رنگ کادر پائین
borderTop	نوع و ضخامت و رنگ کادر بالا
borderLeft	نوع و ضخامت و رنگ کادر چپ
borderRight	نوع و ضخامت و رنگ کادر راست
height	ارتفاع
width	عرض
color	رنگ متن برای اشیای متنی
fontFamily	نام قلم برای اشیای متنی

توابع کاربردی جاوا اسکریپت

جاوا اسکریپت دارای مجموعه ای کامل از توابع در زمینه محاسباتی، رشته ها، تاریخ و... می باشد مثل:

abs(x) : محاسبه قدرمطلق x
ceil(x) : محاسبه کف
floor(x) : محاسبه سقف
sqrt(x) : محاسبه جذر
eval(s) : ارزشیابی رشته بعنوان یک عبارت محاسباتی
isFinite(s) : بررسی اینکه رشته یک عدد متناهی است یا خیر
parsInt(s) : تبدیل رشته به عدد صحیح
parsFloat(s): تبدیل رشته به عدد اعشاری

بخش سوم

برنامه نویسی سمت سرویس دهنده (PHP)

۱- مفاهیم اولیه

۱-۱- سرویس دهنده وب

برنامه اي است که بر روي یک کامپیوتر سرویس دهنده در اینترنت و یا کامپیوتر محلي در حال اجرا بوده و به درخواست هاي HTTP پاسخ مي دهد. چند سرویس دهنده وب معروف عبارتند از :

PWS : سرویس دهنده وب قابل اجرا بر روي سیستم عامل ویندوز ۹۸

IIS : یکي از اجزاي سیستم عامل ویندوز ۲۰۰۰ و بالاتر که مي توان آنرا فعال نمود.

Apache : یک سرویس دهنده وب کوچک و قدرتمند براي کار در محیط هاي ویندوز و لینوکس

۱-۲- سرویس گیرنده وب

هر سیستمي که بتواند از سرویس دهنده وب درخواست سرویس کند سرویس گیرنده وب گفته مي شود. معمولا وظیفه مرورگر است که با درخواست یک صفحه وب با سرویس دهنده ارتباط برقرار کرده و نتیجه دریافتي را نمایش مي دهد.

۱-۳- پروتکل HTTP

یک پروتکل ارتباطي connection-less براي تبادل اطلاعات ابر متن بین سیستم هاي شبکه است. در این پروتکل ابتدا سرویس گیرنده به عنوان شروع کننده ارتباط درخواستي (Request) را به سرویس دهنده ارسال کرده و سرویس دهنده وب با گرفتن درخواست آنرا بررسي نموده و اگر درخواست براي یک صفحه html بود محتواي فایل درخواستي را بعنوان پاسخ (Response) به سرویس دهنده برمي گرداند و چنانچه درخواست مربوط به یک اسکریپت اجرایي (برنامه تحت وب) بود با کمک سرویس دهنده کاربردي خاصي اسکریپت را اجرا کرده و خروجي ان که با قالب HTML است را به مرورگر برگشت مي دهد. لازم به ذکر است بعضي اوقات سرویس گیرنده به همراه درخواست (request)مجموعه اي از اطلاعات دریافتي از یک فرم اطلاعاتي را هم به سرویس دهنده ارسال مي نماید.

۱-۴- سیستم مدیریت بانک اطلاعاتي (DBMS)

وظیفه دریافت دستورات و اجراي آن بر روي بانک اطلاعاتي و برگرداندن نتایج به برنامه درخواست کننده را برعهده دارند. معمولا این سیستم ها در خواست ها را با زبان SQL دریافت مي کنند و قادرند آنها را تفسیر و اجرا کنند. نتیجه دستور SQL ممکن است یک مقدار منطقي مبني بر اجراي موفقیت آمیز دستور یا وقوع خطا بوده و یا یک مجموعه سطر از یک یا چند جدول اطلاعاتي باشد.

یکي از این سیستم ها MySQL است که سازگاري بسیار زیادي با سرویس دهنده apache و زبان PHP دارد.

۲- شروع کار با PHP

براي شروع یک ویرایشگر متني را باز کنید. Php مانند html مبتني بر متن ساده است بنابراین از هر ویرایشگري مي شود براي ساخت فایل php استفاده کرد، مثلا Notepad را باز کنید.

خطوط زیر را وارد کنید و فایل را به هر اسمي با پسوند php. ذخیره کنید. مثلاfirst.php

```
1: <?
2: print "Hello Web!";
3: ?>
```

در صورتیکه php روي کامپیوتر خود نصب کرده اید ، فایل را در وب سرور خود کپي کنید و آن را از طریق browser صدا کنید. و یا اینکه فایل را در یک سایت با دسترسي PHP بارگذاري نمایید و آدرس آن را در مرورگر تایپ کنید . پس از اجراي فایل جمله Hello Web! را بر روي صفحه خواهید دید.

در صورتیکه کدها خود را یعني همان چیزي که در ادیتور تایپ کردید را روي صفحه دیدید. این بدان معني است که وب سرور شما فایل php را اجرا نکرده (یا php بر روي آن نصب نیست و یا پسوند فایل را به درستي انتخاب نکرده اید.)

حالا که صفحه اول php خود را upload کرده اید کمي در آن دقت کنید. در شروع و در اولین خط کد ?> را داریم. همیشه شروع یك کد php باید با همین تگ ها باشد. در غیر اینصورت سیستم با کدها مثل html رفتار میکند و بدون هیچگونه عملیاتي همان متن کد را نمایش خواهد داد و در انتهاي کد PHP نیز حتما باید <? را وارد کنید که به server مي فهماند که کد php اینجا تمام شده است و از این به بعد با کدهاي html روبوو است.
در انتهاي هر خط از کدها باید از ; استفاده شود. در غیر اینصورت سرور کدها را در یک خط و پشت سر هم تشخیص مي دهد. پس شما مي توانستید کد بالا را در یک خط و به صورت <? Print "Hello Word!" ; ?>نیز بنویسید.
دستور Print : این دستور در واقع نمایش دهنده است و هر چیزي که به آن بدهید را بر روي صفحه نمایش مي دهد . شما مي توانید مقدار یک متغیر را نیز با استفاده از Print نمایش دهید.

۲-۱- ترکیب PHP و HTML در یک صفحه

کدهاي زیر را در notepad وارد نمایید و ذخیره و upload کنید.

```
1: <html>
2: <head>
3: <title>Listing 3.2 A PHP script including HTML</title>
4: </head>
5: <body>
6: <b>
7: <?
8: print "hello world";
9: ?>
10: </b>
11: </body>
12: </html>
```

همانطور که مي بینید کدهاي HTML به راحتي مي توانند در کنار کدهاي PHP کار کنند. در واقع سرور قبل از رسیدن به کد ?> همه کدها را htmlفرض کرده و کدهاي در داخل ?> و <? را به عنوان کد php مي شناسد و بر روي آنها عملیات انجام مي دهد.

۲-۲- گذاشتن Comment در میان کدها

commentدر واقع جملاتي است که نویسنده برنامه در میان کدها مي نویسد تا توضیحي باشد بر کد. به این صورت که اگه دفعه بعد خود نویسنده و یا کس دیگري کدها را دید ، متوجه بشود که هر دستور براي چه چیزي نوشته شده است. نحوه قراردادنcomment هم به این صورت است که در اول خط از تگ // و یا # استفاده کنید.

مثلا

```
// This is a comment
```

یا

```
# in yek comment ast
```

۳ – متغیرها

متغیر نگهدارنده ویژه ایست براي مقادیر. هر متغیر داراي یک نام است که با علامت $ در اول آن مشخص مي شود. نام یک متغیر مي تواند شامل حروف ، اعداد و _ باشد. نام یک متغیر نمي تواند شامل space و یا کارکاترهاي غیر حرفي باشد .
کدهاي زیر چند متغیر را تعریف مي کنید:

```
$a
$a_longish_variable_name
$2453
$sleepyZZZZ
```

براي مقدار دادن به متغیر کافي است که آن را مساوي با مقدارش قرار دهید. به طور معمول شما در یک دستور php متغیر را تعریف مي کنید و به آن مقدار مي دهید. مانند کدهاي زیر :

```
$num1 = 8;
$num2 = 23;
```

وقتي كه شما به يک متغير مقدار داديد مي توانيد دقيقا مانند يک كاراكتر با آن رفتار كنيد. به طور مثال :

```
print $num1;
```

دقيقا برابر با دستور print 8; مي باشد .

۳-۱- نوع داده متغير DATA TYPE

انواع مختلف اطلاعات در يک متغير مي تواند ذخيره شود كه در طول برنامه مي توانيد رفتارهاي متفاوتي با آن داشته باشيد.
برخي زبانهاي برنامه نويسي شما را وادار مي كنند كه در ابتدا و در موقع تعريف متغير نوع آن را نيز مشخص نماييد . ولي در PHP لزومي به اين كار نيست و نوع اولين مقداري كه وارد متغير شود ، به عنوان نوع متغير شناخته مي شود.

نوع داده	مثال	شرح
Integer	5	همه اعداد صيح
Double	3.234	عدد اعشاري
String	"hello"	يک مجموعه از كاراكترها
Boolean	true	يكي از مقادير true يا false

۳-۲- آرايه وOBJECT

جدول فوق انواع داده ها را نشان مي دهد.شما مي توانيد از دستور gettype() براي مشاهده نوع يک متغير استفاده كنيد. به عنوان مثال :

```
1: <html>
2: <head>
3: <title>Listing 4.3 Testing the type of a variable</title>
4: </head>
5: <body>
6: <?php
7: $testing = 5;
8: print gettype( $testing ); // integer
9: print "<br>";
10: $testing = "five";
11: print gettype( $testing ); // string
12: print("<br>");
13: $testing = 5.0;
14: print gettype( $testing ); // double
15: print("<br>");
16: $testing = true;
17: print gettype( $testing ); // boolean
18: print "<br>";
19: ?>
20: </body>
21: </html>
```

كدهاي بالا در خروجي جملات زير را نشان خواهد داد :

```
integer
string
double
Boolean
```

INTEGERيک عدد صحيح مي باشد. به كلام ساده يک عدد بدون مميز مي باشد STRING .يک سري كاراكتر مي باشد. وقتي در PHP با STRING كار مي كنيد بايد حتما اطراف آن از " و يا ' استفاده شود DOUBLE .يک عددي است كه مميز نيز دارد BOOLEAN .يا TRUE است و يا FALSE.

۳-۳- تغيير با استفاده از دستور settype()

در PHP با استفاده از دستور settype() شما مي توانيد نوع يک متغير را تغيير دهيد. براي اين کار بايد نام متغير و نوع متغير که مي خواهيد به آن تغيير يابد را در بين پرانتز و با فاصله يک کاما در بين شان مشخص نماييد . به عنوان مثال:

```
1: <html>
2: <head>
3: <title>Listing 4.5 Changing the type of a variable with settype()</title>
4: </head>
5: <body>
6: <?php
7: $undecided = 3.14;
8: print gettype( $undecided ); // double
9: print " -- $undecided<br>"; // 3.14
10: settype( $undecided, string );
11: print gettype( $undecided ); // string
12: print " -- $undecided<br>"; // 3.14
13: settype( $undecided, integer );
14: print gettype( $undecided ); // integer
15: print " -- $undecided<br>"; // 3
16: settype( $undecided, double );
17: print gettype( $undecided ); // double
18: print " -- $undecided<br>"; // 3.0
19: settype( $undecided, boolean );
20: print gettype( $undecided ); // boolean
21: print " -- $undecided<br>"; // 1
22: ?>
23: </body>
24: </html>
```

در هر دفعه تغيير متغير ما با استفاده از دستور GETTYPE() نوع متغير را چاپ مي کنيم که از تغيير آن مطمئن شويم. همانطور که مي بينيد در خط ۷ مقدار متغير ۳,۱٤ است و به صورت DOUBLE و در خط ۱۰ به STRING تبديل مي شود و در خط ۱۳ به INTEGER تغيير مي کند و به ۳ تبديل مي شود. (يعني رند مي شود) و به همين صورت .....

تغيير نوع داده بدون اينکه اصل متغير تغيير کند

با قرار دادن نام نوع داده Data Type در داخل پرانتز و قبل از نام متغير يک کپي از متغير با نوع داده جديد بدون تغيير دادن متغير اصلي ايجاد مي کند. به عنوان مثال :

```
1: <html>
2: <head>
3: <title>Listing 4.6 Casting a variable</title>
4: </head>
5: <body>
6: <?php
7: $undecided = 3.14;
8: $holder = ( double ) $undecided;
9: print gettype( $holder ) ; // double
10: print " -- $holder<br>"; // 3.14
11: $holder = ( string ) $undecided;
12: print gettype( $holder ); // string
13: print " -- $holder<br>"; // 3.14
14: $holder = ( integer ) $undecided;
15: print gettype( $holder ); // integer
16: print " -- $holder<br>"; // 3
17: $holder = ( double ) $undecided;
18: print gettype( $holder ); // double
19: print " -- $holder<br>"; // 3.14
20: $holder = ( boolean ) $undecided;
21: print gettype( $holder ); // boolean
22: print " -- $holder<br>"; // 1
```

```
23: ?>
18: print gettype( $holder ); // double
19: print " -- $holder<br>"; // 3.14
20: $holder = ( boolean ) $undecided;
21: print gettype( $holder ); // boolean
22: print " -- $holder<br>"; // 1
23: ?>
```

در کد بالا هيچ وقت نوع متغير اصلي را تغيير نداديم بلكه مثلا در خط ١٤ در متغير $holder مقدار Integer شده متغير اصلي يعني مقدار ٣ را قرار داديم در خط ١٦ آن را چاپ كرديم.

۴ - عملگرها

قبلا" ياد گرفتيم كه مقدار به متغير بدهيم و data type متغير ها را تغيير بدهيم. يک زبان برنامه نويسي تا وقتي كه نتوانيم به وسيله آن بر روي متغيرها عمليات رياضي انجام دهيم كاربردي ندارد .
عملگرها سمبول هايي هستند كه به وسيله انها مي توانيم با استفاده از چند مقدار ، مقدار جديدي را توليد كنيم. يک عملگر به عنوان مثال همان + است .

4 + 5 = 9

در اينجا ما از عملگر + استفاده كرديم تا با استفاده از دو مقدار ٤ و ٥ مقدار جديد ٩ را توليد كنيم. عملگر مقدار ده يا همان (=) كارش اين است كه مقدار سمت راست خودش را در متغير سمت چپ مي ريزد.

print ($name = "matt");

دستور بالا كلمه matt را چاپ مي كند و همچنين متغير name را مساوي matt قرار مي دهد.

۴-۱- عملگرهاي رياضي

عملگر	نام	مثال	جواب
+	اضافه	۳+۵	۸
-	تفريق	۳-۱۰	7
/	تقسيم	۲/۱۰	۵
*	ضرب	۱۰*۲	۲۰
%	قدرمطلق	۳%۱۰	۱

۴-۲-عملگر پيوند دهنده يا همان (.)

اين عملگر وظيفه پيوند دادن متغير هاي متني را دارد .
به عنوان مثال :

"hello"." world"
returns
"hello world"

در php يک سري عملگرهاي ديگر براي مقدار دهي وجود دارد. =+ عملگري است كه با استفاده از ان متغير با خودش جمع مي شود. مثلا :

```
$x = 4;
$x += 4; // $x now equals 8
```

همچنين -= و /= نيز مي توان استفاده نمود .

```
$x = 4;
$x -= 4; // $x now equals 1
```

۴-۳- عملگرهاي مقايسه اي

عملگرهاي مقايسه اي بر روي متغيرها اعمال مي شود و مقايسه مي كند و در صورت درست بودن True و در صورت غلط بودن False بر مي گرداند.
مثلا

$x < 5

اگر مقدار x مثلا ۳ باشد این عملگر True را برمي گرداند .
مثلا == مقدار سمت راست و سمت چپ را چک مي کند. اگه مثلا ما x را ٤ قرار داده باشيم.

```
$x == 5
```

مقدار False بر مي گرداند.
=!چک مي کند که مقدار سمت راست و چپ برابر نباشند و اگر x همان ٤ باشد:

```
$x =! 5
```

Trueرا بر مي گرداند.
===چک مي کند که مقدار چپ و راست برابر باشند و همچنين نوع ديتا آنها Data type آنها يکي باشد .
علامت بزرگتر و کوچکتر و بزرگتر مساوي و کوچکتر مساوي نيز به همين صورت.

۴-۴-عملگرهاي منطقي

اولين عملگر منطقي همان عملگر) or يا (مي باشد. ديگر نشانه اين عملگر || مي باشد.
مثلا true || false
مقدار True را بر مي گرداند.
&&فقط وقتي True برمي گرداند که هر دو طرف True باشند.
مثلا

```
( $x > 2 ) && ( $x < 15 )
```

هنگامي مقدار True برمي گرداند که x بزرگتر از ۲ و کوچکتر از ۱۵ باشد.

۴-۵-عملگر ++ و --

عملگر ++ يک عدد به متغير اضافه مي کند و -- يک متغير از آن کم مي کند . مثلا ;$x++ يکي به $x اضافه مي کند.
به عنوان مثال :

```
$x = 3;
$x++ < 4; // true
```

مقدار فوق True است. در مثال بالا همونطور که مي بينيد ابتدا عمل مقايسه انجام شده و بعد جمع صورت گرفته.
حال :

```
$x = 3;
++$x < 4; // false
```

مقدار فوق False مي باشد يعني ابتدا جمع صورت گرفته و بعد مقايسه شده است.

۴-۶-ترتيب اجراي عملگرها

شايد همه شما اين را بدانيد ولي در php ترتيب اجراي عملگرها به صورت زير مي باشند:

```
++ − − (cast)
% / *
+ −
< <= => >
== === !=
&&
||
= += − = /= *=%= .=
and
xor
or
```

يعني مثلا در ٤ + ٥ * ۲ ابتدا ۲ در ٥ ضرب مي شود و بعد با ٤ جمع مي شود. يعني جواب مقدار فوق ١٤ مي باشد. البته شما با گذاشتن پرانتز مي توانيد php را مجبور کنيد که به صورت دلخواه شما عمل کند. مثلا 2 * (4+5) مقدار ۱۸ را مي دهد.

۵- ساختارهاي شرطي و تکرار

بدون استفاده از ساختارهاي کنترلي، همواره کدها در يک جهت حرکت مي کردند. در واقع خط به خط کدهاي ما اجرا مي شد و جلو مي رفت. اين روش جايي براي کدنويسي منعطف نمي گذارد. اغلب script ها نياز به تغيير خروجي در شرايط مختلف دارند. با استفاده از IF شما مي توانيد خروجي کدهاي خودتان را در php با توجه به يک سري شرايط تغيير دهيد.

۵-۱- ساختار if

ifعبارت داخل پرانتز جلوي خود را کنترل مي کند و درصورتي کهTrue باشد آن قسمت از کد را اجرا مي نمايد.

```
if ( expression )
{
// code to execute if the expression evaluates to true
}
```

فرمت كلي if به صورت بالا است expression .عبارتي است كه بايد كنترل شود. و عبارات بين { و } كدي است كه در صورت true بودن عبارت If بايد اجرا شود.

```
1: <html>
2: <head>
3: <title>Listing 5.1</title>
4: </head>
5: <body>
6: <?php
7: $mood = "happy";
8: if ( $mood == "happy" )
9: {
10: print "Hooray, I'm in a good mood";
11: }
12: ?>
13: </body>
14: </html>
```

در خط ۸ كد بالا ما از == استفاده كرديم تا كنترل كنيم كه mood مقدار happy را در خود دارد يا خبر . از } و } فقط وقتي استفاده مي كنيم كه كدهاي ما يشتر از يک خط باشند . كد بالا را به صورت زير نيز مي توان نوشت:

```
if ( $mood == "happy" )
print "Hooray, I'm in a good mood";
```

در اين حالت Hooray, I'm in a good mood چاپ مي شود .اگر ما متغير mood را "sad"قرار دهيم ديگر چيزي چاپ نخواهد شد.

۵-۲- ساختار if..else

فرمت كلي آن به صورت زير است:

```
if ( expression )
{
// code to execute if the expression evaluates to true
}
else
{
// code to execute in all other cases
}
```

قسمت بعد از else فقط وقتي اجرا مي شود كه عبارت داخل if ، برابر False باشد.

```
1: <html>
2: <head>
3: <title>Listing 5.2</title>
4: </head>
5: <body>
6: <?php
7: $mood = "sad";
8: if ( $mood == "happy" )
9: {
10: print "Hooray, I'm in a good mood";
11: }
```

```
12: else
13: {
14: print "Not happy but $mood";
15: }
16: ?>
17: </body>
18: </html>
```

در مثال بالا متغير mood مقدار "sad" دارد و در واقع با "happy" برابر نيست پس قسمت داخل IF اجرا نمي شود و فقط قسمتي كه داخل else مي باشد اجرا خواهد شد. خروجي دستور بالا به صورت
Not happy but sad
خواهد بود.

۵-۳- استفاده از ElseIf

ElseIfمجددا يک عبارت ديگر را اجرا مي نمايد و در صورت True بودن دستورات داخلش اجرا خواهد شد. فرمت كلي به صورت زير است :

```
if ( expression )
{
// code to execute if the expression evaluates to true
}
elseif ( another expression )
{
// code to execute if the previous expression failed
// and this one evaluates to true
else
{
// code to execute in all other cases
}
```

اگر عبارت IF درست نباشد تكه اول كد ناديده گرفته مي شود و نوبت عبارت Esleif مي رسد اگر اين عبارت درست باشد قسمت دوم كدها اجرا مي شود. دستور else در نهايت وقتي اجرا مي شود كه هيچ كدام از عبارات IF و Elseif درست نباشند. شما به هر تعداد كه مي خواهيد مي توانيد esleif در كدتان بگذاريد. و در نهايت Else دلخواه است و مي تواند در كد وجود نداشته باشد.

```
1: <html>
2: <head>
3: <title>Listing 5.3</title>
4: </head>
5: <body>
6: <?php
7: $mood = "sad";
8: if ( $mood == "happy" )
9: {
10: print "Hooray, I'm in a good mood";
11: }
12: elseif ( $mood == "sad" )
13: {
14: print "Awww. Don't be down!";
15: }
16: else
17: {
18: print "Neither happy nor sad but $mood";
19: }
20: ?>
21: </body>
```

```
22: </html>
```

در مثال بالا mood مقدار sad دارد. اين مقدار با Happy برابر نيست پس قسمت اول كدها ناديده گرفته مي شود Elseif . متغير mood را با مقدار sad مقايسه مي كند كه True است و كدهاي قسمت دوم اجرا مي شوند.

۵-۴- دستور Switch

اين دستور روش ديگري براي تغيير مسير حركت اجرا شدن كدهاست. دستور switch فقط يک عبارت را چک مي كند و مي تواند اين عبارت را با مقادير متفاوتي مقايسه كند و فقط كدي را اجرا كند كه مقدار مورد نظر در آن True شود. فرمت كلي به صورت زير است:

```
switch ( expression )
{
case result1:
// execute this if expression results in result1
break;
case result2:
// execute this if expression results in result2
break;
default:
// execute this if no break statement
// has been encountered hitherto
}
```

عبارت داخل دستور) switch منظور همان expression داخل پرانتز جلوي آن است) معمولا يک متغير است. در كد هاي داخل) switch منظور قسمت بين { و } مي باشد) شما case هاي مختلفي را مي بينيد كه مقدار متغير switch با همه اين case ها مقايسه مي شود و وقتي كه مقدار ها با هم برابر بود كد مربوطه اجرا مي شود.
گذاشتن قسمت default اختياري است. در صورتي كه متغير با هيچ كدام از مقاديرcase ها برابر نباشد ، دستورات داخل defaultاجرا خواهد شد.
احتياط : دقت كنيد كه كلمه break در انتهاي هر case را حتما بگذاريد. در غير اينصورت كد بعد از اجرا كردن case مورد نظر به راه خود ادامه مي دهد و به مقدار default مي رسد و آن را نيز اجرا مي كند و اين در اكثر مواقع چيزي نيست كه ما دنبالش باشيم. دستور break در واقع از كل كدهاي case خارج شده و به انتهاي دستور switch مي رود.
مثال:

```
1: <html>
2: <head>
3: <title>Listing 5.4</title>
4: </head>
5: <body>
6: <?php
7: $mood = "sad";
8: switch ( $mood )
9: {
10: case "happy":
11: print "Hooray, I'm in a good mood";
12: break;
13: case "sad":
14: print "Awww. Don't be down!";
15: break;
16: default:
17: print "Neither happy nor sad but $mood";
18: }
19: ?>
20: </body>
21: </html>
```

در مثال بالا ، در ابتدا mood مقدار sad را دارد و وقتي كه در خط ۸ Php وارد switch مي شود ابتدا mood را با happy مقايسه مي كند (خط ۱۰) و به دليل اينكه True نمي شود به خط ۱۳ مي رود در آنجا چون mood=sadاست كد خط ۱۱ و سپس خط ۱۲ را اجرا مي كند. در خط ۱۲ با ديدن دستور break به خط ۱۸ ميرود.

۵-۵-عملگر شرطي علامت سوال(؟)

در اين روش شما بدون نياز به نوشتن دستورات IF مي توانيد تابع شرطي ايجاد كنيد كه در آن عبارتي چک شود و در صورت درست بودن يک سري کد اجرا شود و در صورت اشتباه بودن کد ديگر.

```
( expression )?returned_if_expression_is_true:returned_if_expre ssion_is
_false;
```

اگر عبارت داخل پرانتز ما True بود کد بعد از علامت سوال اجرا مي شود و در غير اينصورت کد بعد از : اجرا خواهد شد.

```
1: <html>
2: <head>
3: <title>Listing 5.5</title>
4: </head>
5: <body>
6: <?php
7: $mood = "sad";
8: $text = ( $mood=="happy" )?"Hooray, I'm in a good mood":"Not happy but
$mood";
9: print "$text";
10: ?>
11: </body>
12: </html>
```

در مثال بالا در خط ۸ از اين روش استفاده شده است. در اينجا متغيرmood با مقدار happy مقايسه مي شود اگر درست بود خواهد نوشت Hooray, Im in good moodو در غير اينصورت (كه اينجا اينطور نيست) مي نويسد Not Happy But $moodكه چون اينجا $mood مقدار Sad دارد خروجي مي شود.Not Happy But Sad

نوشتن و خواندن کد به اين صورت کمي سخت است ولي اگر شما فقط يک شرايط را بخواهيد چک کنيد و علاقه به نوشتن کدهاي فشرده داريد اين روش بسيار خوبي است.

۶- حلقه هاي تکرار

۶-۱- حلقه while

تا به حال روشهايي را ديديد که به وسيله آن کد مي تواند بين اينکه کدام کد را اجرا کند انتخاب داشته باشد. همچنين کد مي تواند تصميم بگيرد که چند دفعه يک قسمت را اجرا کند. حلقه هاي براي اين ايجاد شده اند که به شما اجازه دهند يک عمليات را چند مرتبه اجرا نماييد. تقريبا بدون استثناء ، همه حلقه ها آنقدر اجرا مي شوند تا اينکه يک شرايطي (که از توسط شما مشخص مي شود) اتفاق بيافتد و يا اينکه شما شخصا دستور قطع و خروج از loop را بدهيد.

```
while
while ( expression )
{
// do something
}
```

تا وقتي که عبارت while (منظور همان expression داخل پرانتز جلوي آن است) True باشد کد داخل while پشت سر هم اجرا مي شود. معمولا شما در داخل حلقه کاري مي کنيد که عبارت مرتبا تغيير نمايد و يک جا False شود در غير اينصورت حلقه شما تا بينهايت اجرا مي شود . مثال زير حلقه اي ايجاد مي کند و مضارب ۲ را نمايش مي دهد.

```
1: <html>
2: <head>
3: <title>Listing 5.6</title>
4: </head>
5: <body>
6: <?php
7: $counter = 1;
8: while ( $counter <= 12 )
9: {
10: print "$counter times 2 is ".($counter*2)."<br>";
11: $counter++;
12: }
13: ?>
```

```
14: </body>
15: </html>
```

در خط ۷ ما مقدار counter را ۱ گذاشتيم. در خط ۸ حقله اي تشكيل دادم كه تا وقتي اجرا مي شود كه counter از ۱۲ كوچكتر و يا مساوي آن باشد. در خط ۱۰ خروجي برنامه را مي نويسيم و در خط ۱۱ به counter دو عدد اضافه مي كنيم. روش ++ را در درس قبلي خوانديم. پس خروجي برنامه مي شود.

۲ - ٤ - ۸ - ۱۰ – ۱۲

اگر شما در خط ۱۱ فراموش مي كرديد كه counter را زياد كنيد اين حلقه تا بينهايت اجرا مي شد چون هيچگاه counter زياد نمي شد و هيچوقت از ۱۲ بيشتر نمي شد.

۶-۲- حلقه do و while

اين حلقه شبيه حلقه while است فقط سر و ته شده است ! بزرگترين فرق آن اين است كه ابتدا كدها اجرا مي شوند و بعد درستي يا نادرستي عبارت چک مي شود.

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

توجه : آخر عبارت while در خط آخر حتما بايد ; گذاشته شود. اين متد وقتي خيلي به درد مي خورد كه شما بخواهيد كد شما حداقل يكبار اجرا شود حتي اگر شرايط عبارت while اتفاق نيافتاده باشد.

```
1: <html>
2: <head>
3: <title>Listing 5.7</title>
4: </head>
5: <body>
6: <?php
7: $num = 1;
8: do
9: {
10: print "Execution number: $num<br>\n";
11: $num++;
12: }
13: while ( $num > 200 && $num < 400 );
14: ?>
15: </body>
16: </html>
```

در مثال بالا حقله do....while كنترل مي كند كه num بزرگتر از 200 و كوچتر از ٤۰۰ باشد. چون ما num را در خط ۷ مساوي يک قراردايم پس عبارت while درست نيست و False مي باشد ولي در هر حال خروجي كد حداقل يک خط است كه نوشته مي شود

Execution numnber : 1 .

۶-۳- حلقه For

شما هيچ چيزي از استفاده از For به دست نمي آوريد كه نتوانيد با While آن را ايجاد كنيد. ولي در هر حال در اغلب مواقع استفاده از For كدهاي زيباتر و بهتري نسبت به while ايجاد مي كند .

فرمت كلي به صورت زير است :

```
for ( variable assignment; test expression; variable increment )
{
// code to be executed
}
```

هر عبارت داخل for بايد حتما با ; از هم جدا شوند. معمولا ، عبارت اول يک متغير شمارنده ايجاد مي كند ، و در عبارت دوم عبارت كنترلي براي loop است ، و قسمت سوم اضافه كردن و كم نمودن متغير را ايجاد مي كند.

مثال :

```
1: <html>
2: <head>
3: <title>Listing 5.8</title>
4: </head>
5: <body>
```

```
6: <?php
7: for ( $counter=1; $counter<=12; $counter++ )
8: {
9: print "$counter times 2 is ".($counter*2)."<br>";
10: }
11: ?>
12: </body>
13: </html>
```

در خط ۷ بالا مي توانيد ببينيد ، در قسمت اول counter را مساوي ۱ نموديم و در عبارت وسط محدود كرديم كه حلقه تا وقتي كار كند كه counter از ۱۲ كوچكتر و يا مساوي آن باشد. در قسمت سوم هم عبارتي است كه هر بار كه كد اجرا شود يک عدد به counter اضافه نمايد.
مثال بالا و مثال قبلي هر دو يک خروجي را مي دهند فقط كد for كمي جمع و جور تر از while است. به علت اينكه شروع ، پايان و شرايط حلقه در همان خط اول در For مشخص است ، با يک نگاه به اين نوع حلقه كل حقله دستمان مي آيد .

۶-۴- خروج از حلقه با استفاده از دستور break

در هر دو روش ايجاد حلقه دستوري براي پايان دادن حلقه وجود دارد. گاهي اوقات نياز است كه وقتي كد به شرايط خاصي رسيد حلقه پايان يابد . مثلا:

```
1: <html>
2: <head>
3: <title>Listing 5.9</title>
4: </head>
5: <body>
6: <?php
7: for ( $counter=1; $counter <= 10, $counter++ )
8: {
9: $temp = 4000/$counter;
10: print "4000 divided by $counter is... $temp<br>";
11: }
12: ?>
13: </body>
14: </html>
```

در كد بالا ما عدد ٤٠٠٠ را بر counter كه از ۱ تا ۱۰ متغير است تقسيم مي كنيم و خروجي را چاپ مي كنيم. تا ايجا كد بدون نقص به نظر مي رسد. ولي اگه مثلا counter از ورودي كاربر گرفته شود و كاربر مثلا عددي منفي بزند يا صفر وارد نمايد و يا اينكه يک كلمه به جاي عدد وارد كند. در اين صورت ما بايد حلقه را قطع كنيم چون مي دونيم كه تقسيم كردن يک عدد بر صفر ايجاد خطا در php مي كند.
مثلا فرض كنيد از -٤ شروع كنيم تا ۱۰ اين وسط صفر هم جزو مقادير counter خواهد شد .

```
1: <html>
2: <head>
3: <title>Listing 5.10</title>
4: </head>
5: <body>
6: <?php
7: $counter = – 4;
8: for ( ; $counter <= 10; $counter++ )
9: {
10: if ( $counter == 0 )
11: break;
12: $temp = 4000/$counter;
13: print "4000 divided by $counter is... $temp<br>";
14: }
15: ?>
16: </body>
17: </html>
```

در كد بالا همونطور كه مي بينيد در خط ۱۰ و ۱۱ تعريف كرديم كه اگه counter صفر شد حلقه قطع شود. بنابراين كد بعد از رسيدن به صفر و به break مي رسد حلقه را قطع مي نمايد و به خط ١٤ مي رود.

نکته جالب در کد بالا در خط ۷ است که ما counter را خارج از حلقه تعریف نمودیم. بنابراین در خط ۸ دیگه counter را تعریف نکردیم و جاي آنرا خالي گذاشتیم .شما هر کدام از قسمتهاي for را مي تونید خالي بگذارید ولي ; ها را حتما باید بگذارید.

۶-۵- استفاده از دستور continue

حالا فرض کنید که ما در کد بالا نمي خواهیم که وقتي به صفر رسید کد قطع شود و فقط مي خواهیم که حلقه براي مقدار صفر اجرا نشود ولي باقي مقادیر اجرا شود. در این حالت از دستور continue استفاده مي کنیم.

```
1: <html>
2: <head>
3: <title>Listing 5.11</title>
4: </head>
5: <body>
6: <?php
7: $counter = − 4;
8: for ( ; $counter <= 10; $counter++ )
9: {
10: if ( $counter == 0 )
11: continue;
12: $temp = 4000/$counter;
13: print "4000 divided by $counter is... $temp<br>";
14: }
15: ?>
16: </body>
17: </html>
```

در کد بالا در خط ۱۰ و ۱۱ وقتي مقدار counter صفر شود ، حلقه قطع مي شود و دوباره کد بر مي گردد به خط ۸ ولي فقط مقدار صفر را اجرا نخواهد کرد.

۶-۶- حلقه هاي تو در تو

در php این قابلیت را دارید که در داخل یک حقله یک یا چند حلقه دیگر را نیز بیارید. ولي باید توجه داشته باشید که هر حلقه اي که در یک حلقه دیگر استفاده مي کنید باید در همان حلقه تمام شود . مثلا:

```
1: <html>
2: <head>
3: <title>Listing 5.12</title>
4: </head>
5: <body>
6: <?php
7: print "<table border="1">\n";
8: for ( $y=1; $y<=12; $y++ )
9: {
10: print "<tr>\n";
11: for ( $x=1; $x<=12; $x++ )
12: {
13: print "\t<td>";
14: print ($x*$y);
15: print "</td>\n";
16: }
17: print "</tr>\n";
18: }
19: print "</table>";
20: ?>
21: </body>
22: </html>
```

۷ - توابع(Functions)

۷-۱- تابع چیست ؟

شما مي توانيد تابع را يک ماشين در نظر بگيريد. يک ماشين مواد اوليه را از شما مي گيرد و بر روي آنها عمليات از پيش تعيين شده را انجام مي دهد و در نهايت به شما محصولي را مي دهد. تابع مقادير را از شما دريافت مي كند ، بر روي آنها عمليات انجام مي دهد و در نهايت كاري كه مي خواهيد با آن انجام مي دهد و نتيجه را براي شما بر مي گرداند.
اگر شما نياز به درست كردن يک كيک داشته باشيد مسلما خودتان آن را درست مي كنيد. ولي اگه ۱۰۰۰ كيک بخواهيد درست كنيد مطمئنا ماشيني طراحي مي كنيد كه براي شما كيک درست كند. در موقع نوشتن تابع هم هميشه بايد اين مورد مدنظرتان باشد كه طوري تابع را بنويسيد كه بتوان از آن بارها استفاده كرد.
تابع در خود كدهايي را جاي مي دهد كه شما هر وقت به آن نياز داريد آن تابع را صدا مي كنيد مقادير اوليه را به آن مي دهيد و تابع جواب را براي شما برمي گرداند .

۷-۲-فراخواني تابع

دو مدل تابع وجود دارد. اولي توابعي هستند كه درون خود php هستند و ديگري توابعي است كه شما مي نويسيد . يكي از ابتدايي ترين توابعي كه در خود php هستند تابع print است .

```
print("Hello Web");
```

در جلو تمامي توابع حتما بايد () پرانتزها باشند ، البته print يک استثنا است كه بدون پرانتز هم كار مي كند

```
print(("Hello Web");
and
print "Hello Web";
```

هر دو دستور بالا يک خروجي را مي دهند ولي اين مورد فقط در دستور print عملي است. در مثال بالا ما تابع print را صدا كرديم و مقدار "hello word" رابراي اون فرستاديم .حالا تابع وارد عمل مي شود و اين جمله را چاپ مي كند. تابع شامل دو بخش است. اولي نام تابع Print در اينجا و ديگري مقاديري كه براي تابع مي فرستيم argument همان كه در داخل پرانتز جلوي تابع آمده است. برخي توابع نياز به چند Argument دارند كه آنها را با كاما ، جدا مي كنيم. مثلا:

```
some_function( $an_argument, $another_argument );
```

بسياري از توابع اطلاعاتي براي شما بر مي گرداند در راستاي عملي كه انجام مي دهند. مثلا در صورت درست بودن يا نبودن True يا False بر مي گردانند .
ABS()مثلا ، يک عدد را مي گيرد و قدر مطلق آن را بر مي گرداند .

```
1: <html>
2: <head>
3: <title>Listing 6.1</title>
4: </head>
5: <body>
6: <?php
7: $num = − 321;
8: $newnum = abs( $num );
9: print $newnum;
10: // prints "321"
11: ?>
12: </body>
13: </html>
```

در اين مثال ما عدد -۳۲۱ را به $num داديم. اين مقدار را به تابع abs فرستاديم در آنجا محاسبات لازم انجام شد و جواب برگردانده شد كه ما آنرا در داخل $newnum ريختيم و آن را چاپ كرديم. البته ما مي توانستيم كد را كمي جمع و جور تر بنويسيم و مستقيما عدد را به abs بدهيم و همانجا چاپ كنيم.

```
print( abs( − 321 ) );
```

اين يک خط كد همان كاري را مي كند كه در مثال قبل انجام داديم. قوانين استفاده از توابعي كه خودمان مي نويسيم هم به همين شكل است .

۷-۳- تعريف يک تابع

شما مي توانيد تابع را با استفاده از دستور Function تعريف نماييد.

```
function some_function( $argument1, $argument2 )
{
// function code here
}
```

نام تابع درست بعد از دستور Function مي آيد و بلافاصله بعد از آن پرانتزها قرار مي گيرند. اگر تابع شما Argument احتياج دارد ، شما بايد متغيرهاي مورد نياز را (كه به وسيله كاما از هم جدا شده اند) را داخل پرانتز بنويسيد. اگر تابع شما به Argument احتياجي ندارد داخل پرانتز چيزي ننويسيد.

```
1: <html>
2: <head>
3: <title>Listing 6.2</title>
4: </head>
5: <body>
6: <?php
7: function bighello()
8: {
9: print "<h1>HELLO!</h1>";
10: }
11: bighello();
12: ?>
13: </body>
14: </html>
```

در خط ۷ کد بالا ما تابع bighello را تعريف كرديم. مشخص است عملياتي كه اين تابع انجام مي دهد اين است كه كلمه Hello!را بين كدهاي H1 چاپ خواهد نمود. ما تابع bighello را بدون Argument تعريف كرديم ، به همين دليل داخل پرانتز چيزي ننوشتيم .

```
1: <html>
2: <head>
3: <title>Listing 6.3</title>
4: </head>
5: <body>
6: <?php
7: function printBR( $txt )
8: {
9: print ("$txt<br>\n");
10: }
11: printBR("This is a line");
12: printBR("This is a new line");
13: printBR("This is yet another line");
14: ?>
15: </body>
16: </html>
```

در مثال بالا ما تابع printBR را با Argument تعريف مي كنيم. حالا در خطوط ۱۱ و۱۲ و ۱۳ سه مقدار متفاوت را به تابع مي فرستيم و مثلا سه خط چاپ شده در خروجي خواهيم داشت $txt .همانطور كه مي بينيد در خط ۷ تعريف شده است. موقعي كه در خطوط ۱۱ و ۱۲ و۱۳ ما تابع را صدا مي كنيم $txt هر دفعه مقداري كه برايش فرستاده شده است را به خود مي گيرد و در خط ۹ آن را چاپ مي كند.

هر مقدار كه بخواهيم مي توانيم اين تابع را اجرا كنيم و خروجي بگيريم.

توجه داشته باشيد كه اگر تابعي Argument نياز داشته باشد ، موقع صدا كردن تابع بايد حتما مقدار براي آن بفرستيم .

نكته در صورتيكه تابع را به اين صورت تعريف كنيد :

```
function printBR( $txt = "nothing")
```

در اين حالت $txt به صورت default مقدار "nothing" را دارد. يعني اگر ما موقع صدا كردن تابع مقداري براي Argument نفرستيم تابع خودش مقدار Default را به $txt مي دهد ولي اگر ما مقدار بفرستيم $txt برابر با مقدار فرستاده شده مي باشد.

مثلا :

```
PrintBR();
PrintBr("Hello");
[php]
```

در خط اول كد بالا چاپ خواهد شد Nothing و در خط بعد چاپ مي شود . Hello مقدار Default در برخي توابع بسيار كارا هستند.

۴-۷- بازگرداندن مقادير از توابع تعريف شده توسط كاربر

شما مي توانيد از داخل تابع با استفاده از دستور Return مقداري را برگردانيد. دستور return عمليات تابع را متوقف مي نمايد و مقدار گفته شده را بر مي گرداند.

```
1: <html>
2: <head>
3: <title>Listing 6.4</title>
4: </head>
5: <body>
6: <?php
7: function addNums( $firstnum, $secondnum;
8: {
9: $result = $firstnum + $secondnum )
10: return $result;
11: }
12: print addNums(3,5);
13: // will print "8"
14: ?>
15: </body>
16: </html>
```

کد بالا عدد ۸ را در خروجي چاپ مي کند. عددهاي ۳ و ۵ در $firstnum and $secondnum ذخيره شدهند و بعد با هم جمع شدند. و جواب آنها در $result ذخيره شد و سپس در خط ۱۰ آن مقدار برگردانده شده است. با دستور Return شما مي توانيد هر چيزي را برگردانيد مثلا :

```
function addNums( $firstnum, $secondnum )
{
return ( $firstnum + $secondnum );
}
```

تابع بالا نيز دقيقا همان کاري را مي کند که تابع قبلي مي نمود . حتي مي توانيد به اين صورت نيز از return استفاده کنيد.

```
return 4;
```

مي توانيد نتيجه يک عمليات را برگردانيد:

```
return ( $a/$b );
```

يا حتي مقداري از يک تابع ديگر را برگردانيد:

```
return ( another_function( $an_argument ) );
```

۷-۴- صدا کردن يک Function به صورت داينامیک

اين امکان وجود دارد که شما اسم تابع يک String يا يک متغير بگذاريد. و براي صدا کردنش ازآن استفاده کنيد. مثلا:

```
1: <html>
2: <head>
3: <title>Listing 6.5</title>
4: </head>
5: <body>
6: <?php
7: function sayHello()
8: {
9: print "hello<br>";
10: }
11: $function_holder = "sayHello";
12: $function_holder();
13: ?>
14: </body>
15: </html>
```

در مثال بالا در خط ۷ تابع با اسم Sayhello تعريف شده و در خط function_holder 11 يک متغيري تعريف شده با مقدار sayHelloحالا مي توان از function_holder با اضافه پرانتزها براي صدا کردن تابع استفاده کرد.

شايد اين سوال پيش آيد که چرا ما بايد همچين چيزي را لازم داشته باشيم. در مثال فوق عملا ما فقط کار خودمان را زيادتر کرديم. ولي در واقع در برخي مواقع لازم داريم که جريان کد را با توجه به مولفه هاي داخل url يا شرايط برنامه عوض کنيم. يعني مثلا در شرايطي يک function اجرا شود و در شرايط ديگر function ديگري.

۷-۵- متغير ها در داخل تابع

مهم : متغيرهايي كه داخل يک تابع تعريف مي شوند ، فقط در داخل همان تابع قابل دسترسي هستند. يعني ان متغير ها بيرون تابع يا در تابع هاي ديگر در دسترس نيستند. در پروژهاي بزرگ اين امكان خيلي به شما كمک مي كند، چون شما مي توانيد از اسم هاي تكراري براي متغيرهايتان در تابع هاي مختلف استفاده كنيد بدون اينكه دخالتي در يكديگر داشته باشند.

```
1: <html>
2: <head>
3: <title>Listing 6.6</title>
4: </head>
5: <body>
6: <?php
7: function test()
8: {
9: $testvariable = "this is a test variable";
10: }
11: print "test variable: $testvariable<br>";
12: ?>
13: </body>
14: </html>
```

در مثال بالا خروجي چيزي را چاپ نخواهد نمود. چون $testvariable كه در خط ۱۱ براي چاپ خوانده مي شود قبلا تعريف نشده است و $testvariable خط ۹ فقط در داخل function قابل دسترسي هستند.

۶-۷- استفاده از متغير به صورتGlobal

بطور پيش فرض متغيرهاي تعريف شده بيرون يک تابع ،داخل تابع در دسترس نيست. مثلا در مثال زير

```
1: <html>
2: <head>
3: <title>Listing 6.7</title>
4: </head>
5: <body>
6: <?php
7: $life = 42;
8: function meaningOfLife()
9: {
10: print "The meaning of life is $life<br>";
11: }
12: meaningOfLife();
13: ?>
14: </body>
15: </html>
```

خروجي خالي چاپ مي شود و مقدار $life را چاپ نخواهد كرد. در برخي موارد ما نياز به استفاده از متغيرهاي بيرون تابع داخل يک تابع داريم. براي اين كار كافيه كه از دستور Global استفاده كنيم. به طور مثال مي توانيد براي اين منظور كد بالا را به صورت زير بازنويسي كنيد :

[php]

```
1: <html>
2: <head>
3: <title>Listing 6.8</title>
4: </head>
5: <body>
6: <?php
7: $life=42;
8: function meaningOfLife()
9: {
10: global $life;
11: print "The meaning of life is $life<br>";
12: }
13: meaningOfLife();
14: ?>
```

```
15: </body>
16: </html>
```

در خط 10 از دستور global $life; استفاده کردیم. در این حالت مقدار $life که بیرون تابع و در خط ۷ تعریف شده در داخل تابع قابل دسترس مي شود و خروجي این کد The meaning of life is 42 چاپ خواهد شد.
شما باید براي هر متغيري که مي خواهید ازش در تابع استفاده کنید از این دستور استفاده کنید. و همچنین در هر تابعي که مي خواهید از متغيري خارج از آن تابع استفاده کند باید از این دستور استفاده شود.
مهم : دقت کنید که اگه $life داخل تابع تغییر دهید مقدار $life در کل برنامه عوض مي شود.

۸– آرایه ها

۸-۱- تعریف آرایه

آرایه ها در واقع مانند یك ظرف هستند که میتوانیم چندین مقدار را در آن قرار دهیم و سپس از توي ظرف مقادیر را یا مقدار مورد نظرمان را برداریم ویکي از پرکاربرد ترین امکانات در همه زبانها بشمار مي آید. در زبان PHP اندیس آرایه ها از صفر شروع مي شود.
براي مثال:

```
Array index value
0 Mehdi
1 Asef
2 tsotodeh
3 knowhow
4 carp
```

بگذارید چندین مقدار را در یك مقدار دیگر ذخیره کنیم:

```
$users=array("Mehdi","Asef","tsotodeh","knowhow","carp");
```

در اینجا اگر بخواهیم مقدار Asef را چاپ کنیم کافي است بنویسیم:

```
Print "$users[1]";
```

و براي اضافه کردن آرایه جدید میتوانیم از دستور زیر استفاده کنیم

```
$users[]="azemati";
```

وبراي صدا کردن فقط اندیس یا ایندکسش را صدا بزنیم.
در ضمن میتوانیم خودمان هم ایندکس گذاري کنیم یعني بجاي ۰ – ۱- ۲ – ۳ .. .. - خودمان حروف تعیین کنیم براي مثال به کد زیر توجه کنید:

```
$user = array (name=>"Asef",job=>"Programming",age=>24,
"skill"=>"everyThing");
```

خوب حالا براحتي هرکدام را که بخواهیم میتوانیم صدا بزنیم:

```
Print"$user[name]";
Or
Print"$user[job]";
```

همانطور که دیدید در قسمت تعریف کردن ارایه ها حروف را در داخل گیومه قرار داده و قرار دادن اعداد اجباري نیست و هم میتوانند داخل گیومه قرار بگیرند یا خیر.
همچنین میتوانیم متغیر هاي داخل آرایه را مقدار دهي کنیم براي مثال:

```
$user[name]="azemati";
$user[job]="webmaster";
```

و براي تعریف کردن چندین ارایه با چندین مقدار به این صورت عمل میکنیم:

```
$user = array (array(name=>"Asef",job=>"Programming",age=>24,
"skill"=>"everyThing"),array(name=>"mehdi",job=>"Programming",age=>18,
"skill"=>"noThing"),array(name=>"daftarkhatereh",job=>"Programming",age=>24,
"skill"=>"everyThing") );
```

و براي صدا کردن مقداري خاص به این شکل عمل میکنیم

```
print $user[0][job];
//Print "Programming"
```

۸-۲- توابع کاربردي آرایه

براي پي بردن به تعداد مقادیر یك ارایه میتوانید از دستور

```
Print count($user);
```

استفاده کنید . ولي دقت کنید که در دستور کانت مقدار ایندکس ما از صفر شروع نمیشود بلکه از يك شروع میشود براي مثال براي دسترسي به مقداري با استفاده از کانت به این صورت عمل میکنیم:

```
<?php
$users=array("a","b","c","d","e");
print $users[count($users)– 1];
//Print 5
?>
```

که در این کد اخرین مقدار یعني حرف اي را چاپ میکند! ارایه ها را به شکل هاي گوناگوني میتوان استفاده کرد و کاربرد بسیار زیادي دارند

دستور دیگري که کاربرد زیادي دارد، دستور Array_merge() است که با مثالي اشنا میشویم:

```
<?php
$first = array("a", "b", "c");
$second = array(1,2,3);
$third = array_merge( $first, $second );
foreach ( $third as $val )
{
print "$val<BR>";
}
?>
```

در این مثال متغیرهاي دوم و سوم با هم ترکیب شده و در متغییر سوم قرار مي گیرند. دستور foreach مثل دستور for عمل میکند با این تفاوت که در اینجا متغییر سوم را در متغییر جدیدي قرارداده و آن را چاپ میکند شکل کلي این دستور به این صورت است:

```
foreach( $array as $temp )
{
//...
}
```

براي مرتب کردن يك ارایه از دستور sort استفاده میکنیم مانند مثال

```
<?php
$an_array = array("x","a","f","c");
sort( $an_array);
foreach ( $an_array as $var )
{
print "$var<BR>";
}
?>
```

بدیهي است که با اعداد هم میتوانیم همچین کاري را بکنیم همانطور که دیدید با دستور sort(); میتوانیم مقادیر يك ارایه را مرتب کنیم. در مثال بالا در خروجي حروف به صورت مرتب شده نمایش داده میشوند.

دستور asort(); هم داریم که بر اساس مقادیر ارایه ها انها را مرتب میکند باز به مثالي دیگر توجه کنید:

```
<?php
$first = array("first"=>5,"second"=>2,"third"=>1);
asort( $first );
foreach ( $first as $key => $val )
{
print "$key = $val<BR>";
}
?>
```

که در خروجي این چنین چاپ میشود:

```
third = 1
second = 2
first = 5
```

در دستور بالا همانطور که دیدید بر حسب مقادیر مرتب شد ولي اگر بخواهیم بر حسب ایندکسي که میدهیم مرتب شود میتوانیم از دستور ksort(); استفاده کنیم

```
<?php
$first = array("x"=>5,"a"=>2,"f"=>1);
ksort( $first );
```

```
foreach ( $first as $key => $val )
{
print "$key = $val<BR>";
}
?>
```

که در خروجي چاپ ميكند:

```
a = 2
f = 1
x = 5
```

۹- آبجكت ها

۹-۱- ابجكت چيست ؟

آبجت (شيء)مجموعه اي از متغييرها و توابع است كه از يك الگوي خاص به نام كلاس ساخته شده اند. اما كلاس ها چه هستند ؟ فرض كنيد ما يك شركت داريم كه اين شركت از بخش هاي مختلفي تشكيل شده است حال ما در هربخش احتياج داريم كه هر ماه يك گزارش مالي بگيريم ! ما اينجا دو كار ميتوانيم بكنيم هم ميتوانيم براي هربخش چند نفر تعيين كرده و انها گزارش مالي را تهيه كنند و هر ماه تحويل دهند در اين صورت در هربخش شلوغي و همچنين كارمند بيشتري نيازمنديم ! راه ديگر اينست كه يك قسمت به عنوان اتاق گزارش مالي درست كنيم و هر بخش داده هاي خود را به اين قسمت بدهيم و گزارش مالي خود را دريافت كنيم در اين روش هم بخش ها منظم تر خواهند بود و هم ديگر احتياج به كارمند اضافي نداريم. در اينجا ابجكتها نقش كارمند در بخش گزارش مالي را بر عهده دارند .

۹-۲- شكل كلي يك كلاس

```
Class First_class
{
//كلاس هستش اين شكل كلي از يك
}
```

۹-۳- خصوصيات

آبجكتها به متغيير هاي خاصي دسترسي دارند كه به آنها خاصيت (property) ميگويند اين خواص ميتوانند در هرجاي بدنه كلاس باشند اما براي اينكه كد برنامه مرتب باشد بهتراست در بالاي كلاس تعريف شوند. بگذاريد با مثالي ديگر بيشتر توضيح دهيم:

```
Class f_class {
var $name="mehdi";
}
$obj1=new f_class();
$obj2=new f_class();
$obj1->name="Ali";
print "$obj1->name<br>";
print"$obj2->name<br>";
```

ديديم كه براي اختصاص دادن يك كلاس به يك متغيير اينگونه عمل كرديم

```
$obj1=new f_class();
```

علامت -> به شما اجازه ميدهد به متغيير هاي درون يك كلاس دسترسي داشته باشيد و انها را تغيير دهيد همانطور كه در كد مي بينيد در خط ششم متغيير name در ابجكت يك را مساوي علي قرار داديم كه باعث عوض شدن متغيير ميشود همچنين براي چاپ خروجي نيز به همين صورت عمل كرديم ولي بدون علامت مساوي

```
Print " $obj1->name ";
```

۹-۴- متد ها

متدها در واقع توابعي هستند كه داخل يك كلاس وجود دارند . مثال:

```
class f_class
{
var $name;
function sayHello()
{
Print "Hello World";
}
}
```

```
$obj1=new f_class();
$obj1->sayHello();
// Hello World چاپ میشود
```

همانطور که مي بينيد يك متد خيلي شبيه به تابع معمولي هست با اين تفاوت که متد هميشه داخل کلاس تعريف ميشود در ضمن ميتوانيد با علامت -> يك متد ابجکت را صدا بزنيد. مهمتر اينکه متدها به اعضاي متغيير هاي يك کلاس دسترسي دارند.

ديديد که چگونه به يك property از خارج يك آبجکت دسترسي پيدا مي کنيم اما چگونه يك آبجکت ميتواند خودش را به اصطلاح Return کند:

```
class f_class
{
var $name="mehdi";
function sayHello()
{
Print "Hello My names $this->name<br>";
}
}
$obj1=new f_class();
$obj1->sayHello();
// Hello My names mehdi میکنه چاپ
```

يك عبارت مخصوص را بکار برديم به اسم $this تا کلاس به آبجکت فعلي Return شود. شما با ترکيب اين عبارت با علامت ->ميتوانيد داخل يك کلاس به هر پروپرتي و متدي که بخواهيد دسترسي داشته باشيد حالا اگر بخواهيم به پروپرتي nameدر همه آبجکت هاي کلاس مقدار خاصي بدهيم ميتوانيم به اين صورت عمل کنيم:

```
class f_class {
var $name="mehdi";
function setName($n){
$this->name=$n; }
function sayHello()
{
Print "Hello My names $this->name<br>";
}
}
$obj1=new f_class();
$obj1->setName("Ali");
$obj1->sayHello();
// Hello My names Ali چاپ میکنه
```

همانطور که ديديد يک تابع تعريف کرديم که اسم آنرا بتوانيم همه جا تغيير دهيم واز دستور thisبراي عوض کردن آن داخل کلاس استفاده کرديم. در ابتدا اسم ما مهدي بود ولي بعد از اينکه تابع عوض کردن اسم را بکار برديم متد صدا زده شد و اسم تغيير کرد کد بالا را ميتوانستيم بصورت ساده تر وکمي پيچيده تر هم بنويسيم:

```
class first_class {
var $name;
function first_class( $n="mehdi" ) {
$this->name = $n;
}
function sayHello() {
print "hello my name is $this->name<BR>";
}
}
$obj1 = new first_class("Ali");
$obj2 = new first_class("Asef");
$obj1->sayHello();
// hello my name is Ali میکنه چاپ
$obj2->sayHello();
// hello my name is Asef میکنه چاپ
```

همانطور که ديديد يک متد يا تابع با همان اسم کلاس خودمان ساختيم و مقدار پيش فرض آنرا روي مهدي گذاشتيم که اگر هيچي وارد نشد اين عبارت چاپ شود .حالا در پايين در خط دهم و يازدهم هنگامي که ابجکت يك و ابجکت دو را به

کلاس ها اختصاص مي دهيم همان موقع هم مقدار اسم را عوض ميكنيم و مشكلي پيش نميايد كلاس ما هنگامي كه متد صدا زده شود خود به خود صدا زده مي شود.
حال اجازه دهيد با عبارت ديگري به اسم extends اشنا شويم اين عبارت به اين معني ارث بردن است به اين مثال توجه كنيد:

```
class first_class{
var $name = "mehdi";
function first_class( $n ) {
$this->name = $n;
}
function sayHello(){
print "Hello my name is $this->name<br>";
}
}
class second_class extends first_class {
}
$test = new second_class("son of mehdi");
$test->sayHello();
// outputs "Hello my name is son of mehdi"
```

همانطور كه مي بينيد ما كلاس دو را تنظيم كرديم تا از كلاس يك ارث بري كند وهمه متد هاي كلاس يك را خواهد داشت ما ميتوانيم اينجا داخل كلاس دو يك تابع با اسم sayHello بسازيم و بگوييم عبارت I dont know my name را چاپ كند در اينصورت اگر حتي در خط ۱۲ ام ما يك عبارت براي تعويض اسم بدهيم تاثيري ندارد و همان عبارت I dont know my name را چاپ ميكند راه ديگري هم هست و ان چاپ هردو كلاس باهم هست:

```
class first_class {
var $name = "harry";
function first_class( $n ) {
$this->name = $n;
}
function sayHello() {
print "Hello my name is $this->name<br>";
}
}
class second_class extends first_class {
function sayHello() {
print "I'm not going to tell you my name -- ";
first_class::sayHello();
}
}
$test = new second_class("son of harry");
$test->sayHello();
// "I'm not going to tell you my name -- Hello my name is son of harry" چاپ ميكند
```

همانطور كه مي بينيد دستور (متد:: وارث) ما ميتوانيم هر متدي را كه ما تغييرش داديم دوباره صدا بزنيم چون در كلاس دو ما متد sayHello را تغيير داديم با اين دستور آنها را دوباره برگرداندیم.

۱۰- كار كردن با فرم ها

۱۰-۱- تعريف فرم

در اين قسمت ميخواهيم ببينيم چگونه با استفاده از فرمها اطلاعات را بين صفحات ارسال كنيم.
يک مثال ساده ولي پركاربرد:

```
#this page name is a.php
<form action="b.php" method="get">
<input type="text" name="user">
<input type="submit" name="btn">
</form>
```

قصد داريم اطلاعات يك جعبه متن را كه در اين صفحه قرار دارد توسط صفحه دوم به اسم b.php بخوانيم پس اسم اين صفحه را كه هيچ كد پي اچ پي در آن استفاده نكرديم ميگذاريم a.php و ذخيره ميكنيم. سپس اين كد را در صفحه دوم قرار بدهيد:

```
<?php
print "$user"
?>
```

خيلـي سـاده و بـه ايـن صـورت اطلاعـات را ميگيـريم user اسـم جعبـه متنـي اسـت كـه در فـرم اول قـرار داشـت. بقيه كد ها هم اچ تي ام هست و نيازي به توضيح نيست فقط در مورد get بايد گفته شود كه اين دستور متغيير ها را موقع ارسال در ادرس نشان داده و شايد بعضي جاها زياد راه مطمئني نباشد ولي اگر بجاي اين عبارت از post استفاده كنيم ديگر در ادرس نشان داده نميشود.

۱۰-۲- تركيب كردن PHP با HTML

به اين كد نگاه كنيد :

```
<form action="<?php print $PHP_SELF?>" method="POST">
name: <input type="text" name="user">
</form>
```

وقتي اين فرم را اجرا كنيم فرم همواره خودش را صدا ميزند چون ما از عبارت $PHP_SELF استفاده كرده ايم توجه كنيد كه ما هيچ دكمه اي را صفحه قرار نداديم و در بيشتر مرورگرها با زدن كليد اينتر ميتوانيم فرم را اجرا كنيم. البته در كد بالا هيچ اطلاعاتي را بيرون نداده است. در كد زير ميخواهيم يه بازي ساده بنويسيم :

```
<?php
$num_guess=42;
$msg=" ";
if (! Isset($guess))
{
$msg="welcome To This Little Game";
}
elseif ($guess>$num_guess)
{
$msg="number $guess is Big! Try Smaller number.";
}
elseif($guess<$num_$guess)
{
$msg="Number $guess is Small ! Try Big Number";
}
else // vagti mosavi bashe
{
$msg="Well Done You Win";
}
?>
```

توضيحات : با دستور ! اشنا هستيد (دستور نقيض)،دستور isset چك ميكند ببيند اصلا" اطلاعاتي وارد شده يا نه ما در خط اول چك ميكنيم ببينيم (اگه اطلاعاتي وارد شده) انگاه پيغام خوش آمد گويي را پخش كند اين كد موقعي ميتواند اتفاق بيفتد كه براي اولين بار صفحه را باز كرده يا هيچ اطلاعاتي وارد نكرده باشيم .

اين كد را زير كد بالايي قرار دهيد:

```
<h1>
<?php print $msg ?>
</h1>
<form action="<?php print $PHP_SELF?>" method="POST">
Guess Number : <input type="text" name="guess">
</form>
```

اين كد نيز اطلاعات را از كاربر گرفته و صفحه را دوباره اجرا ميكند و پيغام مناسب را پخش ميكند.

ميخواهيم با يک دستور جديد و كاربردي اشنا شويم. فرض كنيم بعد از اينكه كاربر در بازي بالا برنده شد ميخواهيم او را به يک صفحه جديد بفرستيم يعني Redirectكنيم ميتوانيم از دستور زير استفاده كنيم اين دستور را بجاي پيغامي بگذاريد كه وقتي كاربر برنده شد انتخاب ميشود، يعني بعد از else قرار دهيد.

```
header("Location: page.html");
exit;
```

با دستور exit هم از داخل اين كد بيرون آمده و به صفحه بعد منتقل ميشويم.

۱۱- كار با فايلها

۱۱-۱- پیوست دادن کدهاي دیگر

شما میتوانید یك کد را در یك فایل دیگر بنویسد و هرجا لازم شد در یك کد دیگر که در فایل دیگري هم قرار دارد آنرا صدا بزنید تابع ;()Include این امکان را به شما داده و باعث سهولت و کم حجم شدن صفحات میشود. این تابع فقط به یك آرگومان نیاز داشته و آن مسیر فایل پي اچ پي است که میخواهیم به صفحه پیوند بزنیم.

فرض کنید ما در فایل a.php یك کد داریم که یك پیغام را چاپ میکند، حالا ما میخواهیم همین دستور در فایل اول را در فایل b.phpبدون نوشتن دوباره دستور و با دستورInclude بنویسیم کافي است بنویسیم

```
// file name is b.php
<?php
include("a.php");
?>
//will Print message in a.php
```

البته چون در اینجا ما در فایل اول یك خط دستور داریم فرقي نمیکند ولي اگر چندین خط و چندین دستور باشد کار خیلي راحتتر مي شود چون فقط با یک خط کد میتوانیم همان دستورات را صدا بزنیم. حتي میتوانیم یك مقدار را با دستور includeصدا بزنیم یا دستورinclude را با توجه به یک شرط صدا بزنیم. یك مثال :

```
<?php
//this file name is a.php
$ret=(4+4);
return $ret;
?>
_____
<?php
//this file name is b.php
$flag=true;
if($flag) {
$resualt=include("a.php");
print " The Sum Of (4+4) Is $resualt";
}
?>
```

همانطور که دیدید میتوانیم طوري تعریف کنیم که اگر شرط درست نبود اصلا دستور include اجرا نشده و در خط بعدي مقدار فایل a.php را در یک متغییر دیگر نوشته و چاپ میکنیم.

۱۱-۲- تابع چك کردن وجود یك فایل

به این مثال توجه کنید :

```
<?php
if (file_exists("a.php"))
print "The File Exists";
?>
```

حتي میتوانید با تابع دیگري بفهمید مسیر داده شده یك فایل هست یا یك دایرکتوري

```
//Check if it's a file
<?php
if(is_file("a.php")
print"yes this is file";
?>
/*-----------------------------*/
//Check if Current Path is a dir
<?php
if(is_dir("/tmp"))
print"/tmp is valid";
?>
```

توابع دیگر is_executeable , is_writeable , is_readable هستند که چك میکنند: فایل مورد نظر قابل خواندن و یا نوشتن ویا اجرا شدن هست یا مسیر داده شده معتبر هست یا نه و یك مقدار از نوع بولین برمیگردانند . تابع دیگري نیز وجود دارد که سایز یك فایل را برمیگرداند:

```
Print filesize("a.php");
```

این تابع اندازه فایل را برحسب بایت نمایش میدهد.

تابع ;(fileatime() آخرين باري كه يك فايل دسترسي پيدا كرد را به ما بر ميگرداند. در مثال زير ميخواهيم بدانيم فايل a.php در چه تاريخ و زماني براي اخرين بار دسترسي پيدا كرده است :

```
<?php
$lasttime=fileatime("a.php");
print "The File last time accessed in ".date("D d M Y g:i A",$lasttime).".";
// Will Print Sat 14 jan 2006 10:30 Pm
?>
```

تابع ;(filemtime نيز مشابه ;(fileatime است با اين تفاوت كه تاريخ و زمان اخرين باري كه فايل ويرايش شده را برميگرداند .

تابع ;(filectime نيز وجود دارد كه تاريخ تغيير يا ويرايش فايل را برميگرداند ويا در بعضي پلت فرم هاي تاريخ بوجود آمدن فايل را برميگرداند.

تابع ;("touch("file-path.txt در صورتي كه فايلي با اين نام وجود نداشته باشد اين فايل را ايجاد كرده ولي اگر وجود داشته باشد كاري نميكند و فقط تاريخ ويرايش فايل تغيير پيدا ميكند و فايل از بين نميرود .

با تابع ;("unlink("file-path.txt ميتوانيد يك فايل را پاك كنيد.

نكته : در سيستم هاي يونيكس براي اينكه يك فايل را پاك يا ويرايش يا دست يابي پيدا كنيم لازم است كه دسترسي به فايل را داده باشيد.

۱۱-۳- بازكردن فايل قبل از خواندن و نوشتن

قبل از اينكه بتوانيد يك فايل را بخوانيد يا محتوايش را عوض كنيد به اين احتياج داريد كه ان فايل را باز كنيد شما با اين دستور ميتوانيد يك فايل را براي خواندن اماده كنيد

```
$f=fopen("file.txt",'r');
```

و با اين دستور ميتوانيد فايل را براي نوشتن اماده كنيد

```
$f=fopen("file.txt",'w');
```

و براي اضافه كردن اطلاعات به يك فايل بايد از اين دستور استفاده كنيد(Append)

```
$f=fopen("file.txt",'a');
```

بهتراست قبل از اينكه اقدام به ويرايش يا باز كردن يك فايل كنيد ان را امتحان كنيد ببينيد اجازه باز شدن يا ويرايش شدن را دارد ؟

```
If ($fp=fopen("file.txt",'w'))
{
// codehaie marboot be viraiesh file
}
```

يا ميتوانيد بجاي كد بالا اينگونه عمل كنيد:

```
($fp=fopen("file.txt",'w')) or die("Could Not open file");
```

اگر دستور فوق مقدار درست را برگرداند پيغام Could Not open file نشان داده نشده در غير اينصورت نشان داده ميشود . هر عملياتي كه بخواهيم روي فايل اجرا كنيم بايد داخل دستورات زير قراردهيم:

```
Fopen();
//                                                    Code
Fclose();
```

پي اچ پي امكانات زيادي را براي خواندن يك فايل در اختيار ما ميگذارد، بعنوان مثال مي توان يك فايل را برحسب بايت يا برحسب خط يا برحسب كاراكتر خواند .

```
<?php
$filename = "test.txt";
$fp = fopen( $filename, "r" ) or die("Couldn't open $filename");
while ( ! feof( $fp ) )
{
$line = fgets( $fp, 1024 );
print "$line<br>";
}
?>
```

با استفاده از دستور ;(feof چك ميكنيم ببينم به اخر فايل رسيديم يا نه و اگر نرسيده، خط به خط با دستور whileخط هاي فايل را داخل يك متغيير ريخته و انها را چاپ ميكنيم . دستور (fgets($fp,1024 اين كار را كرده وطول هر خط ميتواند تا ۱۰۲٤ بايت باشد.

ميتوانيم مقدار خاصي از فايل را بخوانيم مثلا ۱٦ بايت يا ۱٦ كاراكتر از فايل را بخوانيم

```
<?php
$filename = "test.txt";
$fp = fopen( $filename, "r" ) or die("Couldn't open $filename");
```

```
while ( ! feof( $fp ) )
{
$chunk = fread( $fp, 16 );
print "$chunk<br>";
}
?>
```

با دستور ;($fp, 16)fread ما ١٦ كاراكتر از فايل را ميخوانيم.
ميتوانيد با تابع ()fseek جاي مشخصي از فايل را بخوانيد به مثال زير نگاه كنيد

```
<?php
$filename = "test.txt";
$fp = fopen( $filename, "r" ) or die("Couldn't open $filename");
$fsize = filesize($filename);
$halfway = (int)( $fsize / 2 );
fseek( $fp, $halfway );
$chunk = fread( $fp, ($fsize – $halfway) );
print $chunk;
?>
```

در كد بالا نيمه دوم يك فايل را چاپ ميكنيم .دستور ;(fgetc مثل دستور ()fgets ميباشد كه اگر در كد بالا كه خط به خط يك فايل را اجرا ميكرد قراردهيد كاراكتر به كاراكتر فايل را نشون ميدهد . براي نوشتن يا اضافه كردن مقدار به يك فايل بايد ابتدا فايل را بصورت

```
Fopen("file.txt",'w');
Or
Fopen("file.txt",'a');
```

ميتوانيد با تابع ;(fwrite داخل يك فايل مقداري را قرار دهيد ، دقت كنيد كه در اينصورت محتواي فايل قبلي پاك شده و ميتوانيد با تابع ;(fputs يك مقدار را به فايل مورد نظر اضافه كنيد .

```
<?php
$filename = "test.txt";
$fp = fopen( $filename, "w" ) or die("Couldn't open $filename");
fwrite( $fp, "Hello world\n" );
fclose( $fp );
print "Appending to $filename<br>";
$fp = fopen( $filename, "a" ) or die("Couldn't open $filename");
fputs( $fp, "And Hello To You\n" );
fclose( $fp );
?>
```

۱۱-۴- كار با پوشه ها

شما ميتوانيد با دستور ()rmdir , ()mkdir پوشه اي ايجاد يا پاك كنيد

```
<?php
$dirname = "testdir";
$dh = opendir( $dirname );
while ( gettype( $file = readdir( $dh )) != boolean )
{
if ( is_dir( "$dirname/$file" ) )
print "(D)";
print "$file<br>";
}
closedir( $dh );
?>
```

ما با دستور ()opendir پوشه مورد نظرمان را در ابتدا باز ميكنيم سپس با دستور while مسير فايلها را ميگيريم و چك ميكنيم اگر متعلق به اين دايركتوري بودند عبارت (D) و سپس اسم فايل و مسيرش را چاپ ميكنيم ودرنهايت با دستور ()closedirمي بنديم .

۱۲- كار با بانک هاي اطلاعاتي **MySQL**

Sqlیک زبان ساختار یافته براي ارتباط با بانک اطلاعاتي است که بیشتر بانک اطلاعاتي هاي sql روش هاي خاص خودشان را دارند , مثل مرورگرها که براي خواندن HTML روش خودشان را دارند .
Mysqlبه عنواي یک سرویس اجرا شده که کاربر میتواند بصورت local یا remote به آن وصل شده و بعد از اتصال پایگاه داده مورد نظر را در صورتي که مجوز دسترسي به آنرا دارد، انتخاب نماید . هر بانک اطلاعاتي میتواند چندین جدول داشته باشد که هرجدول شامل سطر و ستون است . هر ستون (Column)فقط مقادیري را مي پذیرد که براي آن تعریف شده است . مثلا Int براي مقادیر Integer یا Text براي مقادیر متني

۱۲-۱- معرفي چند دستورپرکاربرد Sql

```
Create Table mytable(firstname VARCHAR(30),lastname VARCHAR(30),age INT);
```

دستور بالا یک جدول با سه ستون ایجاد میکند که firstname و lastname به عنوان متغیر رشته اي استفاده شده که حاوي ۳۰ کاراکتر هستند، یعني بیشتر از ۳۰ کاراکتر قبول نمیکنند و age به عنوان متغیر عددي میتواند عددي را قبول کنند.
براي اضافه کردن اطلاعات به این جدول به شکل زیرعمل میکنیم :

```
INSERT INTO mytable(firstname,lastname,age)values('Mehdi', 'Safary',19) ;
```

عدد مربوط به سن را داخل ' نگذاشتیم چون متغیرهاي عددي احتیاج ندارند که داخل ' قرار گیرند . براي انتخاب این اطلاعات وارد شده از دستور :

```
SELECT * From mytable;
```

علامت * یعني همه ستون ها انتخاب شود ... ما میتوانیم تکي هم اینکار را انجام دهیم به این صورت :

```
SELECT age From mytable;
```

که تنها متغیر age انتخاب خواهد شد .
براي ذخیره یا تغییر اطلاعات یک سطر (رکورد) از این دستور استفاده میکنیم :

```
UPDATE mytable SET firstname='Mahdi' ;
```

دستور بالا در تمامي سطر ها ستون firstname را برابر Mahdi قرار میدهد , مسلما این همیشه چیزي نیست که میخواهیم. براي تغییر سطر خاصي از یک جدول از دستور WHERE استفاده مي کنیم .

```
UPDATE mytable SET firstname='Mahdi' WHERE lastname='Safary' ;
```

در دستور بالا سطري تغییر پیدا میکند که lastname برابر با Safary باشد , در غیر اینصورت تغییري صورت نمي گیرد.
Mysqlیکي از قدرتمند ترین DBMS ها و یکي از سریعترین DBMS هاي موجود است و سادگي استفاده از آن توسط PHP باعث محبوبیت زیاد ان در بین برنامه نویسان PHP شده است.

۱۲-۲- نحوه برقراري ارتباط با دیتابیس Mysql توسط PHP

```
$link = mysql_connect("localhost","username","pa ssword");
if (! $link) die("Could not Connect To Database");
```

در دستور PHP بالا ما به دیتابیس وصل مي شویم دقت کنید که ما در مثال ها فرض بر این مي گیریم که بر روي کامپیوتر شخصي خودمان برنامه ها را اجرا مي کنیم و معمولا نام کاربري برابر root بوده و کلمه عبوري وجود ندارد دستور بالا به این صورت است :

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

براي انتخاب دیتابیس هم از دستور mysql_select_db(); استفاده میکنیم
در این مثال اطلا عاتي را وارد دیتابیس مي کنیم که قبلا باید بانک اطلاعاتي db1 در mySQL ایجاد شده باشد.

```
$dbname="db1";
//ertebat ra ijad mikonim
$link=mysql_connect("localhost","root"," ");
// agr ertebat movafag nabood error zir ra midahim
if(! $link) die("Could not connect to database");
// database ra entekhab mikonim
mysql_select_db($dbname,$link) or die ("could not open $dbname : ".mysql_error() );
```

```
//query ra tariff mikonim
$query="INSERT INTO mytable(firstname,lastname,age)values('Mehdi', 'Safary',19)";
//query ro ejra mikonim
mysql_query($query,$link) or die("Could not add data".mysql_error());
// ertebat ro mibandim
mysql_close($link);
```

براي بدست آوردن تعداد نتيجه به اين صورت عمل ميكنيم :

```
$result=mysql_query("select * from mytable");
$numrows=mysql_num_rows( $result );
print "There are currently $numrows rows in the table<P>";
```

بيرون كشيدن داده ها از ديتابيس

در روش اول ما تمام داده ها را داخل يک حلقه قرار مي دهيم:

```
<?php
$db = "db1";
$link = mysql_connect( "localhost", "root", "" );
if ( ! $link ) die( "Couldn't connect to MySQL" );
mysql_select_db( $db, $link ) or die ( "Couldn't open $db: ".mysql_error() );
$result = mysql_query( "SELECT * FROM mytable" );
$num_rows = mysql_num_rows( $result );
print "There are currently $num_rows rows in the table<P>";
print "<table border=1>\n";
while ( $a_row = mysql_fetch_row( $result ) ) {
print "<tr>\n";
foreach ( $a_row as $field )
print "\t<td>$field</td>\n";
print "</tr>\n";
}
print "</table>\n";
mysql_close( $link );
?>
```

در مثال بالا تمامي اطلاعات داخل جدول mytable را داخل يک جدول HTML چاپ ميكنيم .

در روش دوم ميخواهيم فقط اطلاعات خاصي را چاپ كنيم براي مثال ستون هاي خاصي را چاپ كنيم نه همه ستون ها را :

```
<?php
$db = "db1";
$link = mysql_connect( "localhost", "root", "" );
if ( ! $link ) die( "Couldn't connect to MySQL" );
mysql_select_db( $db, $link ) or die ( "Couldn't open $db: ".mysql_error() );
$result = mysql_query( "SELECT * FROM mytable" );
$num_rows = mysql_num_rows( $result );
print "There are currently $num_rows rows in the table<P>";
print "<table border=1>\n";
while ( $arow = mysql_fetch_array( $result ) )
{
print "<TR>\n";
print "<TD>$arow[firstname]</TD><TD>$arow[lastname]</TD>\n";
print "</TR>\n";
}
print "</table>\n";
mysql_close( $link );
?>
```

براي آبديت كردن بانک اطلاعاتي شامل درج سطر جديد، ويرايش يک سطر و يا حذف كافي است از همان دستورات Sql بالا استفاده كرده و به هيم صورت با بانک اطلاعاتي ارتباط برقرار كرده و دستور را اجرا كنيم.