



دانشگاه جامع علمی کاربردی

آموزش کاربردی

HTML – CSS – JAVA SCRIPT

گردآورندگان:

فرهاد ایمانی – فرزاد ایمانی

استاد راهنما:

آقای مهندس فرهاد پوررضا

فهرست مطالب

| صفحه | عنوان |
|------|---|
| ۵ | فصل اول - شروع کار با HTML |
| ۸ | فصل دوم - متن‌ها در HTML |
| ۱۴ | فصل سوم - لینکها و تصاویر |
| ۱۷ | فصل چهارم - کار با فرمها |
| ۲۵ | فصل پنجم - قابها و جداول |
| ۳۲ | فصل ششم - نکات تکمیلی |
| ۳۷ | فصل هفتم - شروع کار با CSS |
| ۴۳ | فصل هشتم - ویژگیهای متن در CSS |
| ۴۹ | فصل نهم - شروع کار با Java Script |
| ۵۳ | فصل دهم - متغیرها و دستورات شرطی در Java Script |
| ۶۲ | فصل یازدهم - توابع پویا - حلقه‌ها و آرایه‌ها در Java Script |
| ۷۰ | فصل دوازدهم - آشنایی با توابع رشته‌ای |

مقدمه

با سلام

بسیار خوشحالیم با مبحثی دیگر این بار در مورد طراحی صفحات وب در خدمت شما عزیزان هستیم. کتاب آموزشی که در اختیار شما عزیزان قرار گرفته است حاصل کار ما دو برادر می‌باشد. از استاد گرانقدر و مهربانمان آقای مهندس فرهاد پوررضا بسیار ممنونیم که با ارایه این مطالب دانش ما را در زمینه طراحی صفحات وب فزونی بخشیدند. امید است که مقبول افتد. لطفا در صورت داشتن هر گونه پیشنهاد یا انتقاد در مورد این کتاب ما را راهنمایی بفرمائید.

به امید موفقیت شما عزیزان

Farhad.Imany@Gmail.Com

Far_1779@Yahoo.com

imany-brothers.blogfa.com

بخش اول

HTML

فصل اول: شروع کار با HTML

تعریف HTML:

برنامه‌های تحت Web بصورت کلی به دو دسته تقسیم می‌شوند:

⌘ برنامه‌های استاتیک (Static)

⌘ برنامه‌های دینامیک (Dynamic)

برنامه‌های استاتیک برنامه‌هایی هستند که صفحات آنها از متن، تصاویر، صوت و ... تشکیل شده‌اند و احيانا یک یا چند لینک به سایر آدرسها در آن قرار گرفته است. در واقع این نوع صفحات با کاربر تعاملی ندارند و صرفا کاربر می‌تواند موارد داخل این صفحات را مشاهده نماید.

برنامه‌های دینامیک برنامه‌هایی هستند که داده‌ای را از کاربر دریافت، و آن را مورد پردازش قرار می‌دهند، و در انتها نتیجه آن را در اختیار کاربر قرار می‌دهند.

HTML در ساخت صفحات استاتیک مورد استفاده قرار می‌گیرد. همچنین ظاهر صفحات وب را نیز می‌توان با آن تعیین نمود. HTML مبتنی بر متن (Text Base) می‌باشد که ساختار آن را تگ‌ها (Tags) تشکیل می‌دهند. سنگ بنای Web، HTML می‌باشد.

هر یک از تگ‌های HTML معنی و مفهوم خاصی دارند و تاثیر مشخصی بر محتوای صفحه می‌گذارند. مانند تگ‌هایی برای فرمت متن یا لینک به آدرسی دیگر.

پسوند فایل‌های HTML بصورت .htm یا .html می‌باشد. مرورگرهای وب تک تک دستورات این صفحات را که شامل همان تگ‌ها می‌باشد تفسیر نموده و در انتها خروجی را برای کاربر مشخص می‌سازند. HTML یک زبان برنامه‌نویسی نبوده و زبانی برای نشانه‌گذاری متن است. از این زبان برای سازماندهی کردن اطلاعات و تفکیک اجزای یک صفحه می‌توان استفاده نمود. (اجزایی مانند متون، تصاویر و ...). از سوی دیگر نباید از HTML برای صفحه آرایی استفاده کرد (منظور رنگ نوشته‌ها و قالب بندی متون). این وظیفه بر دوش فن‌آوری دیگری با عنوان CSS است که در همین کتاب مورد بحث قرار خواهد گرفت.

تعریف Web Browser:

یک نرم‌افزار است که بصورت مفسر (Interpreter) کار می‌کند. این مفسر قابلیت تفسیر Tagها و همچنین Scriptها را دارد. شاید این سؤال در ذهن شما پیش آمده باشد که Interpreter چه فرقی با Compiler دارد؟

پاسخ این است که Compiler ابتدا همه دستورات برنامه را خوانده و در صورت صحیح بودن همه آنها برنامه را اجرا می‌نماید. اما Interpreterها رفتار متفاوتی دارند. مفسرها هر خط برنامه را اجرا کرده و به خط بعدی می‌روند. در واقع اجرای هر خط همان لحظه خواندن خط اتفاق می‌افتد. اگر خطایی در خطی باشد آن را اعلام نموده و به کار خود ادامه داده و خط‌های بعدی را اجرا می‌نماید.

بنابراین با توجه به توضیح بالا دلیل اینکه چرا از Interpreterها برای نمایش صفحات وب استفاده می‌شود کاملا مشخص می‌گردد. (اگر از Compiler استفاده می‌شد سرعت اجرا بسیار پایین بود و همچنین اگر کوچکترین خطایی در صفحه بود، هیچگاه به آن صفحه نمی‌توانستیم دسترسی داشته باشیم تا همه ایرادهای آن برطرف گردد)

شروع کار با HTML

برای اینکار باید از یک ویرایشگر استفاده نمایید. چون فایل‌های HTML متنی می‌باشند بنابراین می‌توانیم از یک ویرایشگر متنی استفاده نماییم. (ویرایشگرهایی چون Microsoft Word، Notepad، Wordpad و...)

البته برای نوشتن زبان HTML می‌توان از نرم‌افزارهای پیشرفته در این زمینه استفاده کرد که شما را در نوشتن تگ‌ها راهنمایی می‌کنند. اما چون در ابتدای راه هستیم توصیه می‌شود از اینگونه نرم‌افزارها استفاده نکنید تا ساختار تمام تگ‌ها را یاد بگیرید. ما در این کتاب از ویرایشگر Notepad استفاده می‌کنیم.

Start -> All Programs -> Accessories -> Notepad

برای اینکه راحت تر بتوانید در Notepad دستورات را تایپ کرده و آنها را مشاهده کنید، بعد از اجرای نرم افزار از مسیر زیر اندازه فونت خود را به 14 تغییر دهید.

همچنین برای مشاهده نتیجه کار نیاز به یک Web Browser (Firefox, IE, Opera, ...) خواهیم داشت. ما در این کتاب از IE (Internet Explorer) و Firefox استفاده خواهیم نمود.

ساختار Tag ها در HTML

یک Tag داخل علامتهای < > قرار می‌گیرد. برای درک بهتر این موضوع به دستور زیر که برای ضخیم نمودن یک متن بکار می‌رود توجه نمایید:

 Welcome to Tabriz

ساختار کلی Tag ها بصورت زیر است:

<Start tag>

</End tag>

هر tag دارای یک قسمت آغازین و یک قسمت پایانی می‌باشد. دقت کنید که بخش پایانی یک tag دارای یک علامت / می‌باشد.

البته بعضی از Tag ها نیز وجود دارند که قسمتهای آغازین و پایانی آنها در یک قسمت قرار می‌گیرد. در این نوع tag متنی نوشته نمی‌شود و صرفاً یک عملی انجام می‌گیرد. به tag زیر توجه کنید:

 وظیفه این دستور رفتن به سر بعدی خواهد بود

یک tag ممکن است دارای تعدادی ویژگی یا Attribute نیز باشد. این ویژگی‌ها هر کدام خواصی را برای آن tag بیان می‌کنند. این ویژگیها در شروع هر tag نوشته می‌شود و شامل موارد مختلفی هستند که بسته به نوع tag متفاوت است. اکثر tag ها دارای Attribute هستند. به عنوان نمونه به این tag توجه کنید:

<body bgcolor="Pink">

</body>

وظیفه این tag صورتی کردن زمینه صفحه خواهد بود.

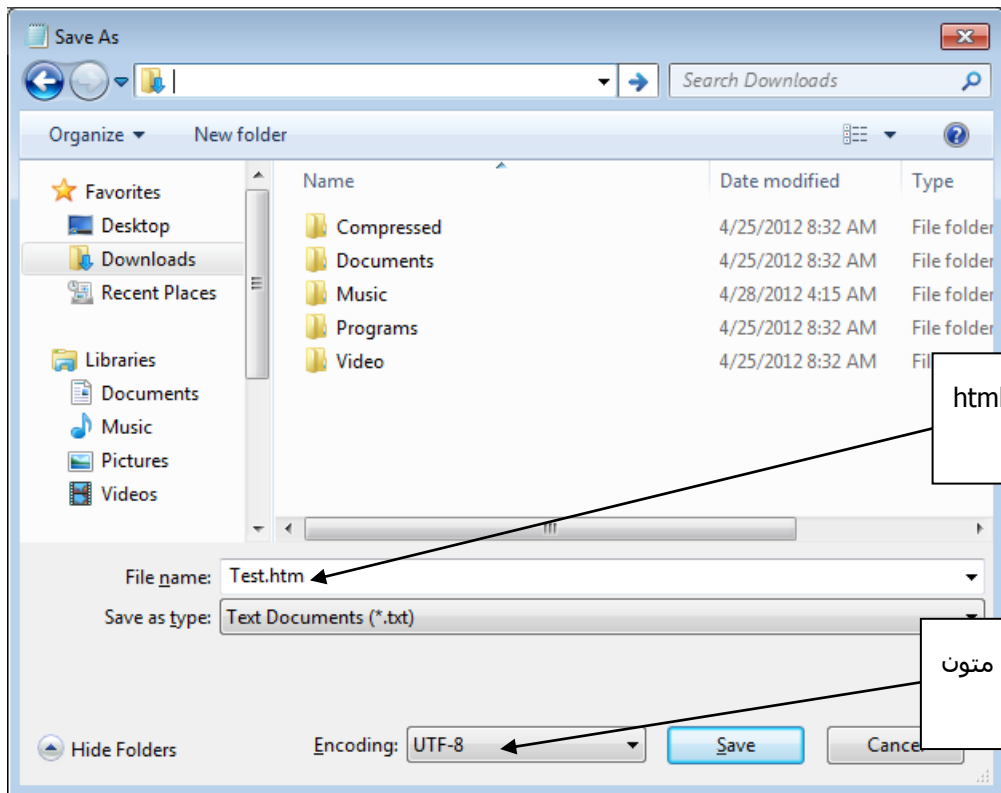
سند HTML

سند HTML ساختاری است که با آن صفحه وب را ایجاد می‌کنیم. به نمونه برنامه زیر توجه کنید:

```

<html>
  <head>
    <title> Welcome to Farhad and Farzad site </title>
  </head>
  <body bgcolor="pink" text="blue">
    <p> Welcome to Tabriz </p>
  </body>
</html>
    
```

بعد از نوشتن کدها در Notepad باید آن را ذخیره کنیم. برای اینکار از منوی File گزینه Save را کلیک نموده تا پنجره زیر ظاهر گردد. سپس گزینه‌های مشخص شده را تنظیم نموده و کلید Save را در انتها کلیک می‌کنیم:



پسوندهای فایل را حتماً htm یا html قرار دهید

حتماً UTF-8 در نظر بگیرید تا متون فارسی قابل رویت باشند

از برنامه Notepad خارج شده و روی فایل ذخیره شده Double Click کنید و نتیجه را ملاحظه فرمائید.

فصل دوم: متن‌ها در HTML

انواع نوشتن متن در HTML:

در این بخش می‌خواهیم ببینیم که متون خود را با چه tag‌هایی می‌توانیم به نمایش درآوریم. دقت نمائید در این کتاب تمامی کدهای برنامه بصورت کامل نوشته شده‌اند و چنانچه مایل بودید می‌توانید آنها را در Notepad کپی کرده و ذخیره کنید و سپس نتیجه کار را مشاهده فرمایید.

نوشتن متن بصورت ساده:

```
<body>
```

```
Welcome to Tabriz
```

```
</body>
```

تگ : با این تگ می‌توانید متن را بصورت ضخیم بنویسید:

```
<body>
```

```
<b> Welcome to Tabriz </b>
```

```
</body>
```

تگ <u>: با این تگ می‌توانید متن را بصورت زیرخط دار بنویسید:

```
<body>
```

```
<u> Welcome to Tabriz </u>
```

```
</body>
```

تگ <i>: با این تگ می‌توانید متن را بصورت مایل بنویسید:

```
<body>
```

```
<i> Welcome to Tabriz </i>
```

```
</body>
```

تگ <strike>: با این تگ می‌توانید متن را بصورت خط خورده بنویسید:

```
<body>
```

```
<strike> Welcome to Tabriz </strike>
```

```
</body>
```


تگ <sub>: با این تگ می‌توانید متن را بصورت زیرنویس بنویسید:

```
<body>
```

The water formula is H₂SO₄

```
</body>
```

تگ <sup>: با این تگ می‌توانید متن را بصورت بالانویس بنویسید:

```
<body>
```

The result of 2¹⁰ is 1024

```
</body>
```

تگ
: با این تگ می‌توانید متن را به سطر بعدی ببرید:

```
<body>
```

Welcome to
Tabriz

```
</body>
```

تگ <center>: با این تگ می‌توانید متن را وسط چین کنید:

```
<body>
```

<center> Welcome to Tabriz </center>

```
</body>
```

تگ Head و انواع سرفصلها:

در تگ های html، تگ های شش گانه‌ای با عنوان تگ‌های سرفصل وجود دارند که برای تیترو عنوان مطالب از آنها استفاده می‌شود. این تگ‌ها با حرف h و یک عدد که نشان دهنده درجه تگ‌ها می‌باشد آغاز می‌گردد.

```
<body>
```

```
<h1> Welcome to Tabriz </h1>
```

```
<h2> Welcome to Tabriz </h2>
```

```
<h3> Welcome to Tabriz </h3>
```

```
<h4> Welcome to Tabriz </h4>
```

```
<h5> Welcome to Tabriz </h5>
```

```
<h6> Welcome to Tabriz </h6>
```

```
</body>
```

پاراگراف بندی متون و تگ br:

برای پاراگراف بندی متون از تگ p استفاده می‌شود و بسیار پرکاربرد است. به خاطر داشته باشید که استفاده از این تگ کاری خواهد کرد که فضای پاراگراف بعدی به اندازه دو تگ br (وظیفه این تگ رفتن به سطر بعدی می‌باشد) از پاراگراف قبلی باشد. در واقع یک تگ p جایگزین دو تگ br می‌شود.

```
<body>
<p> Welcome to Tabriz </p>
<p> How are you? </p>
</body>
```

تعیین ویژگیهای متن:

در این قسمت می‌خواهیم با نحوه تنظیم خصوصیات یک متن آشنا شویم. این خصوصیات عبارتند از نوع فونت، اندازه فونت و رنگ آن. برای ایجاد تنظیمات مورد نظر از تگ font استفاده می‌کنیم:

نوع فونت:

```
<body>
<font face="Arial Black"> Welcome to Tabriz </font face>
</body>
```

اندازه فونت:

```
<body>
<font size="+2"> Welcome to Tabriz </font size> <br>
<font size="-2"> Welcome to Tabriz </font size>
</body>
```

علامت +2 به معنای این است که دو واحد به سایز فونت اضافه شود و علامت -2 یعنی دو واحد از اندازه فونت کسر گردد.

رنگ فونت:

```
<body>
<font color="blue"> Welcome to Tabriz </font color>
</body>
```

ترکیبی:

```
<font face="Arial Black" size="12" color="red"> Welcome to Tabriz </font>
```

ایجاد خط:

برای اینکار از <hr> استفاده می‌نمائیم:

```
<body>
```

```
<h1> Welcome to Tabriz </h1>
```

```
<hr size="10" width="40" noshade="noshade">
```

```
<h1> What is your name? </h1>
```

```
</body>
```

noshade: توپر

width: عرض خط

Size: ضخامت خط

کاراکترهای خاص:

گاهی اوقات نیاز است که از یک سری کاراکترهای خاص مانند علامت Copyright یا غیره در صفحه استفاده گردد. در آن زمان می‌توانید از جدول زیر برای درج آن کاراکترها استفاده نمایید:

| نتیجه روی صفحه | توضیح | کد وارده در برنامه |
|----------------|----------------------|--------------------|
| | non-breaking space | |
| < | less than | < |
| > | greater than | > |
| & | ampersand | & |
| ¢ | cent | ¢ |
| £ | pound | £ |
| ¥ | yen | ¥ |
| € | euro | € |
| § | section | § |
| © | copyright | © |
| ® | registered trademark | ® |
| ™ | trademark | ™ |

```
<body>
```

```
<h1> Welcome to Tabriz &copy; </h1>
```

```
</body>
```

مرجع کامل کاراکترهای خاص : http://www.w3schools.com/tags/ref_entities.asp

ایجاد لیستهای بولت دار

برای اینکار از تگ استفاده می‌شود.

```
<body>
```

```
<p> The Azarbayjan's cities are: </p>
```

```
<ul>
```

```
<li>Tabriz </li>
```

```
<li>Marand </li>
```

```
<li>Azarshahr </li>
```

```
</ul>
```

```
</body>
```

ایجاد لیستهای شماره دار

برای اینکار از تگ استفاده می‌شود.

```
<body>
```

```
<p> The Azarbayjan's cities are: </p>
```

```
<ol>
```

```
<li>Tabriz </li>
```

```
<li>Marand </li>
```

```
<li>Azarshahr </li>
```

```
</ol>
```

```
</body>
```

ایجاد لیستهای تعریف

برای اینکار از تگ <dl> استفاده می‌شود.

```
<body>
<p> The Azarbayjan's cities are: </p>
<dl>
  <dt>Tabriz </dt>
  <dd> Tabriz is best city in Iran </dd>
</dl>
</body>
```

با استفاده از <dt> می‌توانیم لیست را ساخته و با <dd> برای هر <dt> یک توضیح بنویسیم

درج Comment در دستورات

اگر بخواهید یک توضیح در متن برنامه بنویسید که حالت اجرایی نداشته باشد بصورت زیر عمل کنید:

```
<!-- محل نوشتن توضیح -->
```

نوشتن متون متحرک در صفحه

برای اینکار از تگ <marquee> استفاده می‌شود.

```
<body>
<marquee loop="infinite" bgcolor="red" width="20%">
<h1> Welcome to Tabriz </h1>
</marquee>
</body>
```

width: عرض بخشی که متن در آن به حرکت درمی‌آید

bgcolor: رنگ زمینه

Loop: تعداد دفعات اجرا

فصل سوم: لینکها و تصاویر

آشنایی با تگ <a>

با استفاده از این تگ می‌توان یک لینک بوجود آورده و توسط آن به مسیر دلخواه رفت.

```
<body>
```

```
<a href="http://imany-brothers.blogfa.com/"> Go to our site </a>
```

```
</body>
```

همچنین توسط این تگ می‌توانید به صفحه دیگری از سایت خودتان بروید. دقت کنید که در مثال زیر صفحه آدرس دهی شده در همان مسیری است که صفحه ارجاع دهنده در آن است:

```
<body>
```

```
<a href="ContactMe.html"> Goto Contact Page </a>
```

```
</body>
```

ساخت یک Screentip برای موقعی که نشانگر ماوس روی لینک قرار می‌گیرد:

```
<body>
```

```
<a href="http://imany-brothers.blogfa.com/" title="Click Here to Enter Weblog"> Our Weblog</a>
```

```
</body>
```

قرار دادن تصویر در صفحه

با استفاده از امکان پذیر است

```
<body>
```

```

```

```
</body>
```

به خاطر داشته باشید که این تگ به حروف بزرگ و کوچک در مورد نام فایل تصویری حساس است. در ضمن پسوند فایل‌های تصویری بهتر است jpg یا gif باشد. برای تعیین عرض، ارتفاع، ضخامت دور تصویر، موقعیت تصویر و پیام راهنما که با قرار گرفتن ماوس روی تصویر ظاهر می‌گردد از شناسه‌های زیر استفاده نمایید:

```

```

تگ Body و شناسه های آن

شناسه bgcolor: رنگ زمینه صفحه را مشخص می‌کند.

شناسه text: رنگ متن صفحه را مشخص می‌کند.

شناسه link: رنگ تمام لینکهای کلیک نشده در صفحه را مشخص می‌کند.

شناسه alink: رنگ تمام لینکهای کلیک شده در صفحه را مشخص می‌کند. این رنگ تا زمانی که به صفحه بعدی نرفته‌ایم می‌ماند.

شناسه vlink: بعد از کلیک روی یک لینک رنگ آن به رنگی در می‌آید که در alink تعریف کرده بودیم. ولی به محض رفتن به صفحه مورد نظر رنگ لینک به رنگی که در vlink تعریف شده است تغییر می‌یابد.

شناسه background: تعریف یک تصویر برای پس زمینه صفحه.

```
<body bgcolor="red" text="white" link="green" alink="yellow" vlink="pink">
```

```
<a href="http://imany-brothers.blogfa.com/" title="Click Here to Enter Weblog"> Our Weblog</a>
```

```
</body>
```

ساخت لینک با استفاده از عکس

می‌خواهیم کاری کنیم که اگر روی تصویر کلیک کردیم به صفحه مورد نظر خود برویم. برای اینکار به صورت زیر عمل می‌کنیم:

```
<body>
```

```
<a href="http://imany-brothers.blogfa.com">  </a>
```

```
</body>
```

اگر این احتمال می‌رود، ممکن است تصویر مورد نظر بنا بر هر دلیلی load نشود بهتر است یک متن را مشخص نمود که در صورت پیش آمدن موضوع مطرح شده آن متن به جای تصویر ظاهر گردد. بصورت زیر:

```
<body>
```

```
<a href="http://imany-brothers.blogfa.com">  </a>
```

```
</body>
```

نامگذاری بخشی از یک صفحه و ایجاد یک لینک برای رفتن به آن قسمت

بعضی اوقات در ابتدای یک صفحه html می‌خواهیم لیستی از عناوین داشته باشیم که با کلیک روی آن عناوین به قسمت‌های مربوط به هر کدام از آنها در ادامه صفحه برویم. این برای زمانی مفید است که متون یک صفحه بسیار زیاد بوده و پیدا کردن توضیح مربوط به هر عنوان مشکل است.

برای نامگذاری هر قسمت، از تگ <a> بصورت زیر استفاده خواهیم نمود :

```
<a name="part1"></a>
```

حال در ابتدای متن خود برای ارجاع به بخشی از صفحه که نام آن را part1 گذاشته ایم دستور زیر را وارد می‌نمائیم:

```
<a href="#part1"> Goto Part1 </a>
```

نکته: فرض کنیم که این صفحه را با نام Farhad.html ذخیره نموده ایم. اگر خواستیم از یک صفحه دیگر مانند Farzad.html به بخش مورد نظر در فایل Farhad.html برویم باید در فایل Farzad.html در محل دلخواه دستور زیر را وارد نمائیم:

```
<a href="Farhad.html#part1"> Goto Part1 into Farhad.html </a>
```

لینکهای پست الکترونیک

بعضی اوقات می‌خواهیم لینکی بوجود آوریم که با کلیک روی آن بتوانیم برای شخص مورد نظر نامه الکترونیکی ارسال نمائیم. لازم به توضیح است برای کار کردن صحیح این دستور ابتدا باید نرم افزار Outlook خود را تنظیم کرده باشید.

```
<body>
```

```
<a href="mailto:Farhad.Imany@Gmail.com"> Send me Email </a>
```

```
</body>
```

اگر خواستیم cc,bcc,subject را نیز بصورت پیش فرض داشته باشیم بصورت زیر عمل می‌کنیم:

```
<body>
```

```
<a href="mailto:Farhad.Imany@Gmail.com?cc=Farzad.Imany@Gmail.com&bcc=Sales@imanybrothers.ir&subject=Your Products price">
Send me Email
</a>
```

```
</body>
```

دقت نمائید که اولین جدا کننده لیست علامت ? و بقیه علامت & می‌باشد.

فصل چهارم: کار با فرمها

آشنایی با فرم

برای ارسال اطلاعات وارد شده توسط کاربر به سمت مقصد مورد استفاده قرار می‌گیرد. بصورت کلی هرگاه خواستیم از کاربر اطلاعاتی را دریافت نمائیم باید از فرم در آن صفحه استفاده نمائیم. هر فرم می‌تواند در داخل خود از اجزایی مانند textbox, radio button و ... استفاده نماید.

برای ایجاد یک فرم بصورت زیر استفاده می‌گردد:

```
<form>
```

```
</form>
```

یک فرم دارای شناسه‌های زیادی از جمله action, name, method, target می‌باشد که از میان آنها بکار بردن action ضروری است.

```
<form action="" method="" name="" target="" >
```

```
</form>
```

توضیح شناسه‌ها:

action: وظیفه این شناسه مشخص نمودن آدرس جایی است که اطلاعات فرم به آن آدرس ارسال می‌شود.

method: وظیفه این شناسه مشخص نمودن طریقه ارسال اطلاعات فرم به آدرس مقصد می‌باشد. دو حالت برای ارسال اطلاعات داریم:

حالت اول) **get**: اطلاعات با استفاده از نوار ابزار آدرس مرورگر به سمت مقصد ارسال می‌گردد. از این روش باید در موارد زیر استفاده نمود:

- اطلاعات وارده کوتاه باشد. (حداکثر تعداد کاراکترهای ارسالی ۲۵۵ کاراکتر است)
- از کاراکترهای اصلی استفاده شده باشد.
- اطلاعات وارده سری نیستند و اهمیت چندانی ندارند.

حالت دوم) **post**: اطلاعات بصورت محرمانه به سمت مقصد ارسال می‌گردد و در نوار ابزار مرورگر موردی دیده نمی‌شود. از این روش در موارد زیر استفاده می‌شود:

- اطلاعات وارده طولانی باشد.
- از کاراکترهای غیراصلی نیز استفاده شده باشد.
- اطلاعات وارده سری باشد. مانند رمز کارت اعتباری و ...

name: وظیفه این شناسه مشخص نمودن نام فرم مورد نظر است.

target: وظیفه این شناسه مشخص نمودن نحوه باز شدن صفحه پردازشگر اطلاعات وارده است که در بخش action آن را معرفی نموده ایم. در ادامه به مفادیری که این شناسه می پذیرد اشاره می کنیم:

_blank: صفحه پردازشگر در یک پنجره جدید باز می شود.

_self: صفحه پردازشگر در فریمی که فرم در آن قرار دارد باز خواهد شد.

_top: صفحه پردازشگر در تمام پنجره کنونی باز خواهد شد.

در ضمن تگ form شناسه های دیگری نیز دارد که در ادامه این بخش با آنها آشنا خواهیم شد.

استفاده از textbox :

با استفاده از textbox می توان جعبه های دریافت متن را در صفحه قرار داد تا کاربران بتوانند مقادیر دلخواه خود را در آنها وارد نمایند. برای درج یک جعبه متنی در صفحه بصورت زیر عمل می کنیم:

```
<form>
```

```
<input type="text">
```

```
</form>
```

شناسه های textbox:

name: اختصاص نام به جعبه متن برای استفاده از آن در برنامه نویسی

maxlength: تعیین تعداد کاراکتر قابل پذیرش در جعبه متنی

size: تعیین طول کادر جعبه متن بر اساس تعداد کاراکتر یا سایز فونت (مرورگرهای مختلف برخورد متفاوتی با این گزینه دارند)

value: تعیین متن پیش فرض داخل جعبه متنی.

readonly: باعث می شود نتوان در محتویات جعبه متنی تغییری ایجاد کرد. ولی با ارسال فرم به سرور محتویات آن ارسال می گردد

disabled: باعث می شود غیرفعال شدن جعبه متنی می گردد. همچنین با ارسال فرم به سرور محتویات آن ارسال نمی گردد.

مثال:

```
<input type="text" name="text0" maxlength="2" size="2" value="test" readonly="readonly" disabled="disabled">
```

سؤال: فرمی را تشکیل دهید که توسط آن در دو سطر مجزا بتوان نام و نام خانوادگی را وارد نمود.

پاسخ:

<form>

First name: <input type="text" name="txt_fname"> </br>

First name:

Last name: <input type="text" name="txt_lname"> </br>

Last name:

</form>

نکته یک : هرگاه خواستید در جعبه متن، کاربر رمز عبور خود یا رمز کارت اعتباری خود را وارد نماید آن را بصورت زیر تعریف کنید:

Password: <input type="password" name="txt_pass">

نکته دو : هرگاه خواستید علاوه بر اطلاعاتی که کاربر وارد نموده، یکسری اطلاعات را نیز شما به سمت سرور (بدون اطلاع کاربر) ارسال نمایید از کادر متنی مخفی بصورت زیر استفاده نمایید:

<input type="hidden" name="txt_pass" value="test">

حال هنگام ارسال اطلاعات به سمت سرور مقدار test نیز به همراه اطلاعات وارد شده توسط کاربر به سمت سرور ارسال می‌گردد بدون آنکه کاربر متوجه این امر گردد.

استفاده از checkbox :

با استفاده از checkbox می‌توان برای انتخاب یک یا چند گزینه بصورت همزمان عمل نمود. برای درج یک checkbox در صفحه بصورت زیر عمل می‌کنیم:

<form>

<input type="checkbox">

</form>

شناسه های checkbox:

name: اختصاص نام به checkbox برای استفاده از آن در برنامه نویسی

value: مقداری است که در صورت انتخاب شدن checkbox به سمت server ارسال می‌گردد. استفاده از این شناسه ضروری است.

checked: باعث می‌شود هنگام باز شدن صفحه، checkbox بصورت پیش فرض انتخاب شده باشد

disabled: باعث غیرفعال شدن checkbox می‌گردد. همچنین با ارسال فرم به سرور محتویات آن ارسال نمی‌گردد.

مثال:

```
<input type="checkbox" name="check0" value="ok" checked="checked" disabled="disabled">
```

سؤال: فرمی را تشکیل دهید که در آن بتوان علاقمندیهای فرد را مشخص نمود.

پاسخ:

```
<form>
```

What is your Favorites?

Movie : <input type="checkbox" name="moviecheck" value="movie">

Sport : <input type="checkbox" name="sportcheck" value="sport">


```
</form>
```

What is your Favorites?

Movie :

Sport :

استفاده از radiobutton :

با استفاده از radiobutton می‌توان برای انتخاب صرفاً یک گزینه از بین چند گزینه عمل نمود. برای درج یک radiobutton در صفحه بصورت زیر عمل می‌کنیم:

```
<form>
```

```
<input type="radio" name="r1">
```

```
</form>
```

نکته مهم: دقت کنید که شناسه name را اگر قرار ندهید به هیچ عنوان کار نخواهد کرد (در حالیکه در مورد textbox و checkbox اینطور نیست)

شناسه های radiobutton:

name: اختصاص نام به radiobutton برای استفاده از آن در برنامه نویسی

value: مقداری است که در صورت انتخاب شدن radiobutton به سمت server ارسال می‌گردد. استفاده از این شناسه ضروری است.

checked: باعث می‌شود هنگام باز شدن صفحه، radiobutton بصورت پیش فرض انتخاب شده باشد

disabled: باعث غیرفعال شدن radiobutton می‌گردد. همچنین با ارسال فرم به سرور محتویات آن ارسال نمی‌گردد.

مثال:

```
<input type="radio" name="radio0" value="ok" checked="checked" disabled="disabled">
```

سؤال: فرمی را تشکیل دهید که در آن بتوان محل کار فرد را مشخص نمود.

پاسخ:

```
<form>
```

```
What is your unit name? <br>
```

```
Line A: <input type="radio" name="line" value="linea">
```

```
Line B: <input type="radio" name="line" value="lineb">
```

```
</form>
```

```
What is your unit name?
```

```
Line A:  Line B: 
```

نکته بسیار مهم: برای تعریف یک گروه از radiobuttonها که بتوان یکی از آنها را انتخاب کرد باید name آنها را یکی داد.

استفاده از button :

با استفاده از button می‌توان یک کلید عملیاتی را تعریف نمود. این کلیدها به سه حالت تقسیم می‌شوند:

حالت اول) Submit: از این کلید زمانی استفاده می‌شود که قصد داریم داده‌های یک فرم را به سمت سرور ارسال نمائیم. طرز قرار دادن کلید بصورت زیر است:

```
<form action="test.asp">
```

```
<input type="submit" value="Submit">
```

```
</form>
```

عبارت value مشخص کننده متنی است که باید روی کلید به نمایش درآید. حال با اجرای این برنامه کلید مربوطه ساخته شده و با کلیک روی آن محتویات صفحه جاری به صفحه test.asp که در بخش action فرم خود تعریف نموده‌ایم ارسال می‌گردد.

حالت دوم) Reset: از این کلید زمانی استفاده می‌شود که قصد داریم داده‌های یک فرم را پاک نمائیم. طرز قرار دادن کلید بصورت زیر است:

```
<form action="test.asp">
```

```
<input type="reset" value="Reset">
```

```
</form>
```

حالت سوم) Button: از این کلید زمانی استفاده می‌شود که قصد داریم دستورات JavaScript را در صفحه خود با فشردن یک کلید انجام دهیم. طرز قراردادن این کلید بصورت زیر است:

```
<form action="test.asp">
<input type="button" value="button">
</form>
```

توجه کنید این نوع کلید عمل خود را صرفاً در صفحه‌ای که خود در آن است انجام داده و داده‌ای را به سمت سرور ارسال نمی‌کند.

شناسه های button:

name: اختصاص نام به button برای استفاده از آن در برنامه نویسی

value: مشخص کننده متنی است که باید روی کلید به نمایش درآید.

disabled: باعث غیرفعال شدن button می‌گردد.

استفاده از textarea :

با استفاده از textarea می‌توان جعبه متنی را بوجود آورد که امکان دریافت متن از کاربر را در چندسطر فراهم می‌آورد. نحوه قرار دادن آن بصورت زیر است:

```
<form>
<textarea> </textarea>
</form>
```

شناسه های textarea:

name: اختصاص نام به جعبه متن برای استفاده از آن در برنامه نویسی

cols: تعیین تعداد کاراکتر قابل پذیرش در عرض. (برای تعیین عرض کادر بر اساس تعداد کاراکتر نیز به کار می‌رود).

rows: تعیین تعداد سطرها قابل مشاهده در هر مرحله.

readonly: باعث می‌شود نتوان در محتویات جعبه متنی تغییری ایجاد کرد. ولی با ارسال فرم به سرور محتویات آن ارسال می‌گردد

disabled: باعث می‌شود غیرفعال شدن جعبه متنی می‌گردد. همچنین با ارسال فرم به سرور محتویات آن ارسال نمی‌گردد.

مثال: <textarea name="textarea0" cols="10" rows="5" readonly="readonly" disabled="disabled"></textarea>

اگر خواستید متن پیش فرض برای textarea داشته باشید کافیست آن را بین دو تگ بنویسید. (<textarea>Test </textarea>)

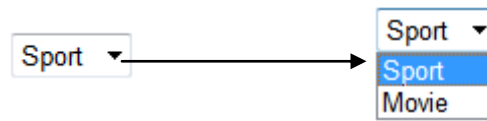
استفاده از dropdownlist :

با استفاده از dropdownlist می‌توان یک جعبه بازشدنی ایجاد کرده و این امکان را فراهم ساخت تا کاربر مقدار دلخواه خود را از یک لیست بازشدنی انتخاب نماید. طرز قرار دادن آن بصورت زیر است:

```
<form>
<select>
  <option>
  </option>
</select>
</form>
```

مثال:

```
<form>
<select name="select_favorites">
  <option value="sport"> Sport </option>
  <option value="movie"> Movie </option>
</select>
</form>
```



شناسه های select:

name: اختصاص نام به dropdownlist برای استفاده از آن در برنامه نویسی

size: مشخص می‌کند چه تعداد از گزینه‌های لیست هنگام بازشدن دیده شوند.

multiple: باعث می‌شود کاربر بتواند چند گزینه را با استفاده از کلیدهای ctrl یا shift بصورت پراکنده یا پشت سرهم انتخاب نماید. در این حالت دیگر لیست بازشدنی نخواهد بود.

disabled: باعث غیرفعال شدن dropdownlist می‌گردد. همچنین با ارسال فرم به سرور محتویات آن ارسال نمی‌گردد.

مثال:

```
<select name="select_favorites" multiple="multiple" disabled="disabled">
</select>
```

شناسه های option :

value: مشخص می‌کند با انتخاب آیتم مورد نظر چه عبارتی به سمت سرور ارسال شود.

Selected: مشخص می‌کند گزینه مورد نظر به عنوان پیش فرض انتخاب شده باشد.

optgroup: اگر تعداد گزینه‌های لیست زیاد باشد می‌توان آنها را دسته بندی کرد. به مثال زیر توجه کنید:

```
<form>
```

```
<select name="select_favorites">
```

```
<optgroup label="sport">
```

```
<option value="sport"> Sport </option>
```

```
<option value="movie"> Movie </option>
```

```
</optgroup>
```

```
<optgroup label="computer">
```

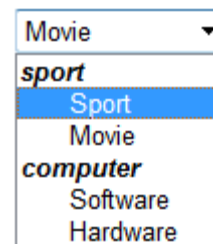
```
<option value="soft"> Software </option>
```

```
<option value="hard"> Hardware </option>
```

```
</optgroup>
```

```
</select>
```

```
</form>
```



فصل پنجم: قابها و جداول

آشنایی با قاب

تا کنون یاد گرفتیم که چگونه در یک صفحه مرورگر یک سند html داشته باشیم. حال در ادامه می‌خواهیم این موضوع را بررسی نمائیم که چگونه می‌توان در یک صفحه مرورگر چند صفحه html داشته باشیم. این امر با ایجاد قاب امکان پذیر است.

به مثال زیر توجه کنید:

```
<html>
<frameset cols="25%,75%">
  <frame src="test1.htm">
  <frame src="test2.htm">
</frameset>
</html>
```

در این مثال صفحه به دو قسمت عمودی تقسیم می‌شود. بخش اول ۲۵ درصد و بخش دوم ۷۵ درصد کل صفحه را به خود اختصاص داده و با استفاده از تگ frame محتویات دو فایل test1.htm و test2.htm در آنها به ترتیب نمایش داده خواهند شد. دقت کنید که از تگ body استفاده نشد.

اگر مرورگر نتوانست قابها را نشان دهد، باید کاری کرد که این مساله حل شود. تگ noframes زمانی راه اندازی خواهد شد که بنابه هر دلیلی مرورگر نتواند قابها را نشان دهد. به مثال زیر توجه کنید:

```
<html>
<frameset cols="25%,75%">
  <frame src="test.htm">
  <frame src="test.htm">
</frameset>
<noframes>
  <body> Your Browser does not handle frames! </body>
</noframes>
</html>
```

دقت کنید که در تگ noframes، تگ body را هم داریم.

اگر خواستیم صفحه بصورت افقی به دو بخش تقسیم شود، کفایت به جای شناسه cols از شناسه rows استفاده نمائیم:

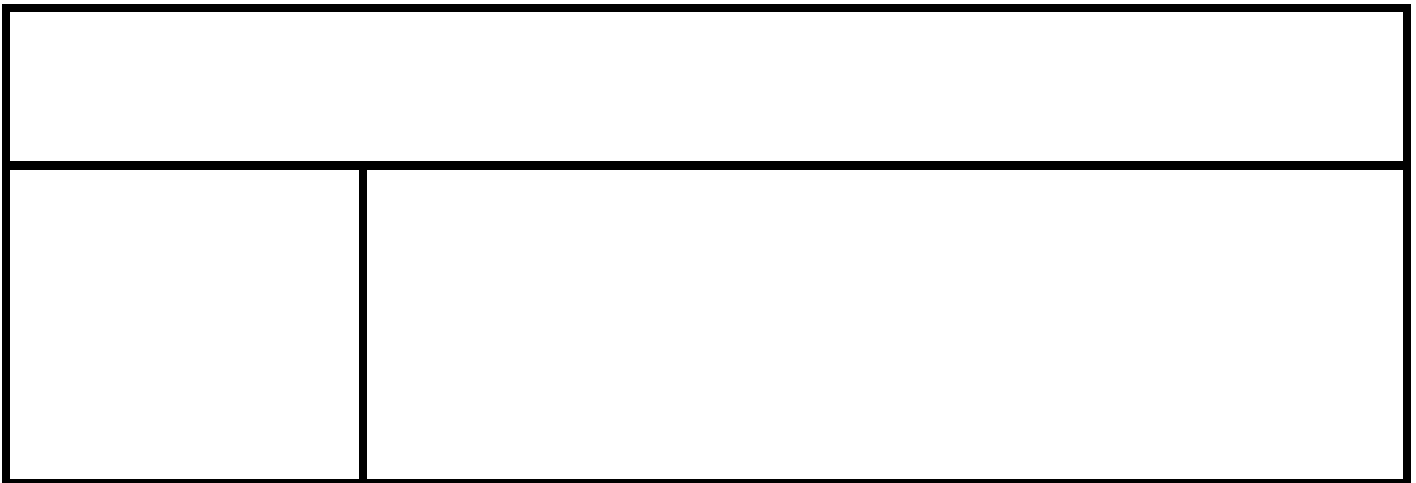
```
<html>
<frameset rows="25%,75%">
  <frame src="test1.htm">
  <frame src="test2.htm">
</frameset>
</html>
```

حال فرض می‌کنیم صفحه را به دو بخش افقی تقسیم کرده‌ایم (مورد بالا) و می‌خواهیم بخش پایینی را به دو قسمت عمودی تقسیم کنیم:

```
<html>
<frameset rows="25%,75%">
  <frame src="test1.htm">
  <frameset cols="25%,75%">
    <frame src="test3.htm">
    <frame src="test4.htm">
  </frameset>
</frameset>
</html>
```

جایگزین شده است با

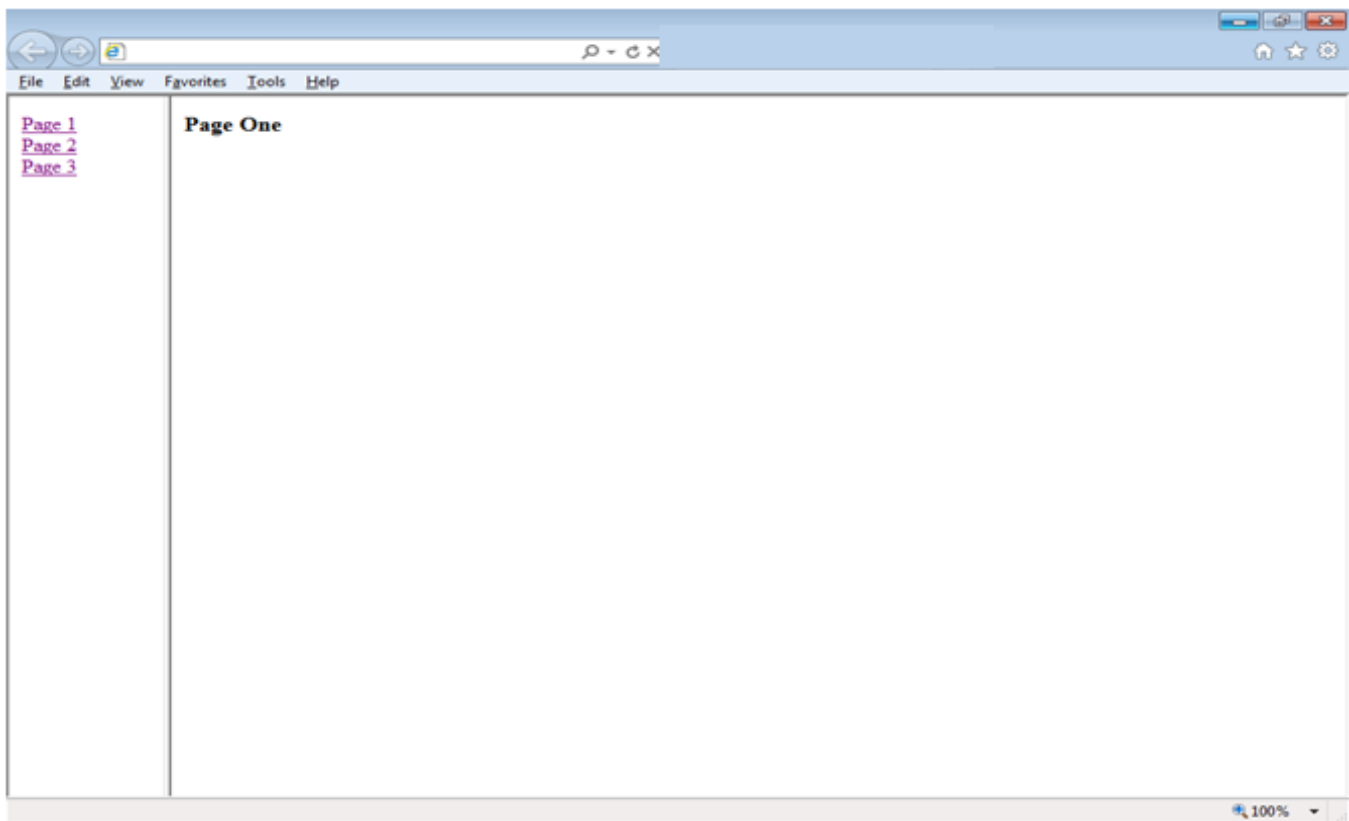
خروجی:



اگر خواستیم قابهای ایجاد شده در مرورگر توسط ماوس تغییر اندازه را قبول نکنند از گزینه noresize استفاده خواهیم کرد:

```
<html>  
<frameset rows="25%,75%" >  
<frame src="test.htm">  
    <frameset cols="25%,75%" >  
        <frame src="test.htm" noresize="noresize">  
        <frame src="test.htm" noresize="noresize">  
    </frameset>  
</frameset>  
</html>
```

حال فرض می‌کنیم که می‌خواهیم در صفحه مرورگر خود دو قاب را بوجود آوریم که در قاب سمت چپ عناوین باشد و به ما اجازه دهد با کلیک روی هر عنوان محتویات مربوط به آن را در قاب دوم مشاهده کنیم. برای اینکار از مثال زیر استفاده می‌کنیم:



فرضها:

نام صفحه ای که قابها در آن تعریف شده است: Frame.html

نام صفحه ای که شامل لینکها به صفحات زیر مجموعه است: Titles.html

نام صفحات زیر مجموعه: Page1.html و Page2.html و Page3.html

کدهای فایل Frame.html

```
<html>
<frameset cols="25%,75%">
  <frame src="Titles.html">
  <frame src="Page1.html" name="showframe">
</frameset>
</html>
```

کدهای فایل Titles.html

```
<html>
<body>
  <a href="Page1.htm" target="showframe">Page 1</a><br>
  <a href="Page2.htm" target="showframe">Page 2</a><br>
  <a href="Page3.htm" target="showframe">Page 3</a>
</body>
</html>
```

محتویات فایل‌های Page1.html و Page2.html و Page3.html بستگی به عنوان مربوطه خود تعیین می‌گردد. (هر چه کاربر خود بخواهد می‌تواند در این فایلها به عنوان توضیح وارد نماید.)

آشنایی با تگ iframe

این تگ باعث ایجاد یک قاب می‌شود که درون آن می‌توان یک سایت یا تصویر را نشان داد. به مثال زیر توجه کنید:

```
<html>  
  
  <iframe src="http://imany-brothers.blogfa.com">  
  
  </iframe>  
  
</html>
```

شناسه های تگ iframe:

width: تعیین عرض قاب

height: تعیین ارتفاع قاب

مثال:

```
<iframe src="http://imany-brothers.blogfa.com" width="400" height="600"></iframe>
```

آشنایی با جدول

برای ایجاد یک جدول یک سطری که شامل چند فیلد است بصورت زیر عمل می‌کنیم:

```
<html>  
  
<body>  
  
  <table border="2">  
  
    <td> Name </td>  
  
    <td> Family </td>  
  
    <td> Unit_name </td>  
  
    <td> Salary </td>  
  
  </table>  
  
</body>  
  
</html>
```

| | | | |
|------|--------|-----------|--------|
| Name | Family | Unit_name | Salary |
|------|--------|-----------|--------|

حال اگر خواستیم ردیفهای دیگری به جدول اضافه نمائیم بصورت زیر عمل می‌کنیم:

```
<html>
<body>
<table border="2">
  <tr>
    <td> Name      </td>
    <td> Family    </td>
    <td> Unit_name  </td>
    <td> Salary     </td>
  </tr>

  <tr>
    <td> Farzad    </td>
    <td> Imany     </td>
    <td> Behdari   </td>
    <td> 12000000 </td>
  </tr>

  <tr>
    <td> Farhad    </td>
    <td> Imany     </td>
    <td> TPC       </td>
    <td> 12000000 </td>
  </tr>
</table>
</body>
</html>
```

| Name | Family | Unit_name | Salary |
|--------|--------|-----------|----------|
| Farzad | Imany | Behdari | 12000000 |
| Farhad | Imany | TPC | 12000000 |

شناسه های تگ table:

cellsapcing: برای ایجاد فاصله بین سلولها.

cellpadding: برای ایجاد فضا داخل سلولها.

width: تعیین عرض جدول

height: تعیین ارتفاع جدول

bgcolor: تعیین رنگ زمینه

شناسه های تگ td:

align: برای چینش متن داخل سلول در راستای افقی بکار می‌رود. مقادیر زیر را می‌توان برای آن وارد کرد:

left: چپ چین

right: راست چین

center: وسط چین

valign: برای چینش متن داخل سلول در راستای عمودی بکار می‌رود. مقادیر زیر را می‌توان برای آن وارد کرد:

top: بالا چین

middle: میان چین

bottom: پایین چین

bgcolor: تعیین رنگ زمینه

تگ th: اگر خواستید ردیف عنوان برای جدول داشته باشید، آن را در این تگ بنویسید:

```
<table border="2">
```

```
<tr>
```

```
<th> Name </th>
```

```
<th> Family </th>
```

```
<th> Unit_name </th>
```

```
<th> Salary </th>
```

```
</tr>
```

```
</table>
```

فصل ششم: نکات تکمیلی

قرار دادن موسیقی در صفحه

اگر بخواهید هنگام باز شدن صفحه یک موسیقی بصورت اتوماتیک پخش شود بصورت زیر عمل کنید:

```
<html>
<body>

    <embed src="2012.mp3"> </embed>

</body>
</html>
```

قرار دادن متاگ

این تگ در بخش head مورد استفاده قرار می‌گیرد. موتورهای جستجو به این بخش صفحه نگاه کرده و آن را شاخص گذاری می‌نمایند. بنابراین کلمات کلیدی متناسب با متن صفحه را در آن قرار می‌دهیم تا هنگام جستجو صفحه ما نیز جز صفحات یافت شده توسط موتور جستجو باشد.

```
<html>
  <head>
    <meta name="" content="">
  </head>
  <body>
  </body>
</html>
```

شناسه های متاگ:

name: مشخص کننده نوع متاگ. مقادیر قابل پذیرش در آن عبارت است از:

author: مشخص کننده نام نویسنده صفحه

Generator: مشخص کننده نام برنامه‌ای است که برای ایجاد صفحه وب خود از آن استفاده کرده‌ایم.

keywords: موتورهای جستجو را از وجود یک دسته کلمات کلیدی در این صفحه آگاه می‌کند.

description: برای توصیف محتویات صفحه استفاده می‌شود.

content: مشخص کننده محتویات متاگ. این محتویات توسط موتورهای جستجو مورد استفاده قرار می‌گیرند.

متاتگ refresh

گاهی اوقات می‌خواهید کاربر به محض ورود به صفحه مورد نظر شما بصورت اتوماتیک به صفحه دیگری ارجاع داده شود. در این اوقات از این متاتگ بصورت زیر استفاده نمائید:

```
<html>
  <head>
    <meta http-equiv="refresh" content="3;url=http://imany-brothers.blogfa.com">
  </head>
  <body>
    You are going automatically to http://imany-brothers.blogfa.com
  </body>
</html>
```

حال کاربر به محض ورود به این صفحه بصورت اتوماتیک بعد از ۳ ثانیه به صفحه <http://imany-brothers.blogfa.com> ارجاع داده می‌شود.

رویدادهای صفحه html:

onload: زمانی که صفحه در حال فراخوانی است.

onunload: زمانی که صفحه در حال بسته شدن است.

onresize: زمانی که صفحه در حال تغییر سایز است.

بطور مثال می‌خواهیم پیامی مناسب در راستای سه رویداد فوق ظاهر گردد. به مثال توجه کنید:

```
<body onload="alert('Welcome')" onunload="alert('Goodbye')" onresize="alert('Resizing')"> </body>
```

رویدادهای فرم:

onchange: فراخوانی اسکریپتی که در این رویداد تعریف شده است، هنگام تغییر عناصر موجود در فرم است.

onsubmit: فراخوانی اسکریپتی که در این رویداد تعریف شده است، هنگام ارسال فرم به سمت سرور است.

onreset: فراخوانی اسکریپتی که در این رویداد تعریف شده است، هنگام فشردن کلید Reset در فرم است.

onselect: فراخوانی اسکریپتی که در این رویداد تعریف شده است، هنگام انتخاب بخشی از فرم است.

Onfocus: فراخوانی اسکریپتی که در این رویداد تعریف شده است، هنگام استفاده کاربر از تگی است که حاوی این شناسه است.

Onblur: عکس Onfocus عمل می‌کند.

حال مثالی در مورد رویدادهای فرم را با هم می‌بینیم:

```
<html>
<body>
<form action="result.asp" onsubmit="alert('onSubmit')" onreset="alert('onReset')">
  Your Name:
    <input type="text" onfocus="alert('onFocus')" onblur="alert('onBlur')"><br>
  Select Option:
    <select name="select" onchange="alert('onChange')">
      <option>First Option</option>
      <option>Second Option</option>
      <option>Third Option</option>
      <option>Fourth Option</option>
    </select>
    <textarea onSelect="alert('onSelect')">onselect</textarea><br>
    <input type="reset" name="reset" value="Reset">
    <input type="submit" name="Submit" value="Submit">
</form>
</body>
</html>
```

خروجی:

Your Name:

Select Option:

رویدادهای صفحه کلید:

onkeydown: فراخوانی اسکریپتی که در این رویداد تعریف شده است، به محض فشردن یک کلید از صفحه کلید است.

onkeypress: فراخوانی اسکریپتی که در این رویداد تعریف شده است، بعد از فشردن یک کلید و رها کردن آن است.

onkeyup: فراخوانی اسکریپتی که در این رویداد تعریف شده است، بعد از رها کردن کلید است.

```
<body>
```

```
<form action="result.asp">
```

This text for testing onkeypress:

```
<input type="text" onkeypress="alert('onkeypress')" onkeydown="alert('onkeydown')">
```

This text for testing onkeyup:

```
<input type="text" onkeyup="alert('onkeyup')">
```

This text for testing onkeydown:

```
<input type="text" onkeydown="alert('onkeydown')">
```

```
</form>
```

```
</body>
```

رویدادهای ماوس:

onclick: فراخوانی اسکریپتی که در این رویداد تعریف شده است، هنگام کلیک ماوس است.

ondblclick: فراخوانی اسکریپتی که در این رویداد تعریف شده است، هنگام دابل کلیک ماوس است.

onmousemove: فراخوانی اسکریپتی که در این رویداد تعریف شده است، هنگام حرکت ماوس است.

onmouseover: فراخوانی اسکریپتی که در این رویداد تعریف شده است، هنگام قرار گیری نشانگر ماوس روی قسمتی است که حاوی این شناسه است.

onmouseout: فراخوانی اسکریپتی که در این رویداد تعریف شده است، هنگام خروج نشانگر ماوس از روی قسمتی است که حاوی این شناسه است.

```
<body>
```

```
<input type="button" value="OnClick" onclick="alert('onclick')"> <br><br>
```

```
<input type="button" value="Ondblclick" ondblclick="alert('ondblclick')"> <br><br>
```

```
<input type="button" value="Onmousemove" onmousemove="alert('onmousemove')"> <br><br>
```

```
<input type="button" value="Onmouseover" onmouseover="alert('onmouseover')"> <br><br>
```

```
<input type="button" value="Onmouseout" onmouseout="alert('onmouseout')"> <br><br>
```

```
</body>
```

بخش دوم

CSS

فصل هفتم: شروع کار با CSS

آشنایی با CSS (Cascade Styling Sheet):

روشی است برای قالب بندی و طراحی اجزای صفحه.

مزایای CSS:

* طراحی ظاهر صفحات بدون استفاده از دستورات HTML.

* جلوگیری از تکرار دستورات.

* سرعت بارگذاری بیشتر برای صفحات.

* استفاده از فایل CSS خارجی جهت تغییر ظاهر هر تعداد صفحه‌ای که مایل هستیم.

طریقه نوشتن کدهای CSS:

برای نوشتن دستورات CSS باید با مفوم‌های زیر آشنا شویم:

الف) Selector

ب) Property

ج) Value

هر دستور CSS شامل این سه قسمت است. به نمونه دستور زیر دقت کنید:

```
h1 {color:blue}
```

این دستور بیانگر این موضوع است که تمام متونی که با تگ h1 نوشته می‌شوند به رنگ آبی باشند. در این دستور h1 همان Selector است. عبارت color همان Property می‌باشد. در نهایت Blue نیز Value می‌باشد.

آشنایی با دو شناسه بسیار پرکاربرد class و id:

اکثر تگ‌های html دارای این دو شناسه می‌باشند. فرض کنید می‌خواهیم در صفحه دو نوع پاراگراف داشته باشیم، یکی با متن آبی و دیگری با رنگ قرمز. برای مشخص نمودن این دو نوع پاراگراف از شناسه class استفاده می‌کنیم. ابتدا استایل را برای تگ p می‌نویسیم:

```
p.blue {color:blue}
```

```
p.red {color:red}
```

در ادامه:

```
<p class="blue"> Farhad Imany </p>
```

```
<p class="red"> Farzad Imany </p>
```

حال اگر خواستیم از دو کلاس در تگ خود استفاده نمائیم باید از سیستم زیر استفاده نمائیم:

```
p.blue {color:blue}
```

```
p.red {color:red}
```

```
.center{text-align:center}
```

```
<p class="blue center"> Farhad Imany </p>
```

```
<p class="red center "> Farzad Imany </p>
```

حال اگر خواستیم برای تگ هایی که از شناسه id استفاده می کنند، یک استیل بنویسیم باید آن را بصورت زیر تعریف کنیم:

```
#border{border:green solid}
```

با این دستور در تمام صفحه تگ هایی که id آنها دارای مقدار border است، با کادری سبز رنگ نمایش داده می شوند.

وارد کردن CSS با استفاده از شناسه style:

ساده ترین راه برای افزودن یک استیل به عضوی در صفحه، افزودن شناسه style به تگ آن عضو است. به مثال زیر دقت کنید:

```
<body>
```

```
<p style="color:green">Farhad and Farzad Imany </p>
```

```
</body>
```

اگر خواستید بیش از یک حالت در بخش استیل تعریف کنید بین آنها از علامت ; استفاده نمائید.

```
<body>
```

```
<p style="color:green ; font-style:italic">Farhad and Farzad Imany </p>
```

```
</body>
```

استفاده از CSS در تگ head:

اگر بخواهیم در بخش head صفحه یک استایل را تعریف نمائیم باید از تگ style در آن قسمت استفاده نمائیم. به مثال زیر توجه کنید:

```
<html>
<head>
    <style>
        h1 {color:green ; font-style:italic}
    </style>
</head>
<body>
    <h1>Farhad and Farzad Imany </h1>
</body>
</html>
```

حال در هر جایی از صفحه تگ h1 مورد استفاده قرار گیرد متن آن سبز و ایتالیک خواهد بود

استفاده از CSS بصورت یک فایل خارجی:

اگر بخواهیم در یک فایل مجزا مشخصات استایل را تعریف کنیم و سپس از آن در فایل html خود استفاده نمائیم باید بصورت زیر عمل کرد:

ابتدا باید در یک فایل استایل های خود را تعریف کرده و آن را با پسوند CSS ذخیره نمائیم. به عنوان مثال دستورات زیر را در یک فایل با نام Style1.css نوشته و آن را ذخیره می‌کنیم:

```
h1 {color:green ; font-style:italic}
h2 {color:red ; font-style:italic}
```

حال در فایل html خود باید بصورت زیر برای استفاده از فایل Style1.css عمل کنیم:

```
<html>
    <head>
        <link rel="stylesheet" type="text/css" href="Style1.css"
    </head>
    <body>
        <h1>Farhad and Farzad Imany </h1>
        <h2>Farhad and Farzad Imany </h2>
    </body>
</html>
```

تعریف فایل Style1.css به عنوان منبع استایل صفحه جاری

حال در هر جایی از صفحه تگهای h1 و h2 مشخصات خود را از فایل Style1.css خواهند گرفت.

تنظیم عرض و ارتفاع با استفاده از CSS:

در این بخش می‌خواهیم بدانیم چگونه می‌توان عرض و ارتفاع عناصر صفحه را با استفاده از CSS تعیین کرد. به مثال زیر دقت کنید:

```
<html>
<body>
  <div style="width:100px;height:100px;overflow:hidden">
    Farhad and Farzad Imany </br>
    Farhad and Farzad Imany </br>
    Farhad and Farzad Imany </br>
    Farhad and Farzad Imany </br>
    Farhad and Farzad Imany </br>
    Farhad and Farzad Imany
  </div>
</body>
</html>
```

توضیح شناسه‌ها:

width: تعیین عرض بخش

height: تعیین ارتفاع بخش

overflow: سه مقدار زیر را می‌پذیرد که عبارتند از :

hidden: باعث می‌شود که متنی که از اندازه بخش تعریف شده تجاوز می‌کند دیده نشود.

visible: عکس hidden عمل خواهد کرد.

scroll: باعث ایجاد scrollbar در صورت بیشتر بودن اندازه متن از محدوده تعریف شده می‌باشد.

ایجاد کادر با استفاده از CSS:

به مثال زیر توجه فرمائید:

```
<html>
<body>
  <div style="border-style:solid;border-width:5px;border-color:red"> Farhad and Farzad Imany </div>
</body>
</html>
```

مقادیری که **border-style** قبول می‌کند عبارتند از: solid/dashed/dotted/double/groove/hidden/inset/outset/ridge

تنظیم ارتفاع و فاصله از بالا با استفاده از CSS:

به مثال زیر توجه فرمائید:

```
<html>
<body>
  <div style="border:double;padding:100px;margin:100px"> Farhad and Farzad Imany </div>
</body>
</html>
```

توضیح شناسه ها:

padding: تعیین ارتفاع بخش.

margin: تعیین فاصله بخش از بالای صفحه

تنظیم موقعیت عناصر با استفاده از CSS:

به مثال زیر توجه فرمائید:

```
<html>
<body>
  <div style="position:absolute;top:50px;right:100px;border-style:double"> Farhad and Farzad Imany </div>
</body>
</html>
```

توضیح شناسه ها:

position: مقادیر زیر را قبول می‌کند:

static: اجزا در همان جایی که تعریف می‌شوند می‌مانند

absolute: می‌توانیم موقعیت جز مورد نظر را نسبت به طرفین صفحه تنظیم نماییم.

fixed: موقعیت عنصر تحت هر شرایط ثابت می‌ماند.

در ادامه برای هر کدام از حالت‌های فوق یک مثال می‌زنیم. دقت نمائید که برای تست کدها از Firefox استفاده نمائید.

مثال برای fixed:

```
<html>
<body>
<div style="position:fixed;top:50px;right:100px;border-style:double"> Farhad and Farzad Imany </div>
<p> Farhad and Farzad Imany <p></br>
<p> Farhad and Farzad Imany <p></br>
<p> Farhad and Farzad Imany <p></br>
<p> Farhad and Farzad Imany <p></br>
<p> Farhad and Farzad Imany <p></br>
<p> Farhad and Farzad Imany <p></br>
<p> Farhad and Farzad Imany <p></br>
<p> Farhad and Farzad Imany <p></br>
</body>
</html>
```

در زمان تست سعی کنید اندازه مرورگر را طوری تنظیم کنید که نوارهای اسکرول ظاهر گردند. حال اگر عمل اسکرول را انجام دهید خود متوجه خواهید شد که وظیفه گزینه fixed چیست

مثال برای absolute:

```
<html>
<body>
<div style="position:absolute;top:50px;right:100px;border-style:double"> Farhad and Farzad Imany </div>
</body>
</html>
```

حال هنگام تغییر سایز پنجره مرورگر خواهید دید بخش مورد نظر موقعیت خود را نسبت به صفحه ثابت نگاه می‌دارد.

فصل هشتم: ویژگیهای متن در CSS

در این بخش می‌خواهیم بخشی از دستورات CSS را که به متن مربوط می‌شود بررسی نمائیم:

شناسه letter-spacing: با استفاده از این شناسه می‌توانید فاصله بین کاراکترهای یک متن را تنظیم نمائید:

```
<html>
<body>
<P style="letter-spacing:10"> Farhad and Farzad Imany </P>
</body>
</html>
```

شناسه text-align: با استفاده از این شناسه می‌توانید چینش متن را مشخص نمائید. این شناسه مقادیر left یا right یا center را می‌پذیرد.

```
<html>
<body>
<P style="text-align:center"> Farhad and Farzad Imany </P>
</body>
</html>
```

شناسه text-decoration: با استفاده از پارامترهایی که این شناسه قبول می‌کند حالت‌های زیر را روی متن اعمال کنید:

underline: زیر خط دار

overline: بالاخط دار

line-through: خط زمینه

blink: چشمک زن

```
<html>
<body>
<P style="text-decoration:blink"> Farhad and Farzad Imany </P>
</body>
</html>
```

شناسه text-transform: با استفاده از پارامترهایی که این شناسه قبول می‌کند حالت‌های زیر را روی متن اعمال کنید:

lowercase: کلا با حروف کوچک

uppercase: کلا با حروف بزرگ

capitalize: حرف اول هر کلمه بزرگ

```
<html>
<body>
<P style="text-transform:capitalize"> Farhad and Farzad Imany </P>
</body>
</html>
```

شناسه text-indent: با استفاده از این شناسه می‌توانید دندان‌دار کردن یک متن را انجام دهید.

```
<html>
<body>
<P style="text-indent:10"> Farhad and Farzad Imany </P>
</body>
</html>
```

شناسه font-weight: با استفاده از این شناسه می‌توانید دندان‌دار کردن یک متن را انجام دهید.

```
<html>
<body>
<p style="font-weight:800"> Farhad and Farzad Imany </p>
</body>
</html>
```

شناسه font-family: با استفاده از این شناسه می‌توانید نوع فونت متن را مشخص کنید.

```
<html>
<body>
<p style="font-family:tahoma"> Farhad and Farzad Imany </p>
</body>
</html>
```

شناسه font-size: با استفاده از این شناسه می‌توانید سایز فونت متن را مشخص کنید.

```
<html>
<body>
<p style="font-size:20"> Farhad and Farzad Imany </p>
</body>
</html>
```

شناسه font-style: با استفاده از این شناسه می‌توانید حالت متن را مشخص نمایید:

```
<html>
<body>
<p style="font-style:italic"> Farhad and Farzad Imany </p>
</body>
</html>
```

شناسه color: با استفاده از این شناسه می‌توانید رنگ متن را مشخص نمایید:

```
<html>
<body>
<p style="color:red"> Farhad and Farzad Imany </p>
</body>
</html>
```

نکته: اگر خواستید رنگ بخشی از متن را تغییر دهید از تگ span استفاده کنید:

```
<html>
<body>
<p style="color:red"> Farhad and <span style="color:blue">Farzad </span> Imany </p>
</body>
</html>
```

شناسه background-color: با استفاده از این شناسه می‌توانید رنگ متن را مشخص نمایید:

```
<html>
<body>
<div style="color:white;background-color:red;width:160"> Farhad and Farzad Imany </div>
</body>
</html>
```

شناسه `background-image`: با استفاده از این شناسه می‌توانید تصویر زمینه را مشخص کنید:

```
<html>
<body style="background-image:url(OurPic.jpg)">
<p> Farhad and Farzad Imany </p>
</body>
</html>
```

شناسه `background-repeat`: با استفاده از پارامترهایی که این شناسه قبول می‌کند حالت‌های زیر را اعمال کنید:

`no-repeat`: تصویر در زمینه تکرار نشود.

`repeat-x`: تصویر بصورت افقی تکرار شود.

`repeat-y`: تصویر بصورت عمودی تکرار شود.

```
<html>
<body style="background-image:url(OurPic.jpg);background-repeat:no-repeat">
<p> Farhad and Farzad Imany </p>
</body>
</html>
```

شناسه `background-attachment`: تعیین موقعیت تصویر زمینه زمان اسکرول کردن صفحه. با استفاده از پارامترهایی که این

شناسه قبول می‌کند حالت‌های زیر را اعمال کنید:

`fixed`: تصویر هنگام اسکرول صفحه ثابت می‌ماند.

`scroll`: تصویر هنگام اسکرول صفحه حرکت می‌کند.

```
<html>
<body style="background-image:url(OurPic.jpg);background-attachment:fixed">
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
<p> Farhad and Farzad Imany </p>
</body>
</html>
```

شناسه `cursor`: تعیین شکل ماوس هنگام رفتن روی قسمت مورد نظر.

```
<html>
<body>
<p style="cursor:wait"> Farhad and Farzad Imany </p>
</body>
</html>
```

پارامترهایی که این شناسه قبول می‌کند عبارتند از:

| | |
|------------------|---|
| auto | این گزینه نشانگر ماوس را با توجه به تنظیمات مرورگر کاربر تغییر می‌دهد. |
| crosshair | نشانگر ماوس به صورت به علاوه در می‌آید. |
| default | نشانگر ماوس به صورت همان فلش همیشگی در می‌آید. |
| move | نشانگر به شکلی در می‌آید که در هنگام جا به جا کردن چیزی به خود می‌گیرد. |
| pointer | نشانگر به شکل دست در می‌آید. |
| help | یک علامت سؤال کنار پیکان نشانگر ماوس قرار می‌گیرد. |
| text | به شکلی در می‌آید که بر روی متنها به خود می‌گیرد. |
| wait | به شکل ساعت شنی در می‌آید. |
| n-resize | یک فلش به سمت شمال |
| s-resize | یک فلش به سمت جنوب |
| e-resize | یک فلش به سمت مشرق |
| w-resize | یک فلش به سمت مغرب |
| ne-resize | یک فلش به سمت شمال شرقی |
| nw-resize | یک فلش به سمت شمال غربی |
| se-resize | یک فلش به سمت جنوب شرقی |
| sw-resize | یک فلش به سمت جنوب غربی |

بخش سوم

Java Script

فصل نهم: شروع کار با Java Script

همانطور که قبلا گفتیم برنامه‌های تحت وب به دو بخش تقسیم می‌شوند:

‡ برنامه‌های استاتیک (Static)

‡ برنامه‌های دینامیک (Dynamic)

برنامه‌های استاتیک برنامه‌هایی هستند که صفحات آنها از متن، تصاویر، صوت و ... تشکیل شده‌اند و احيانا یک یا چند لینک به سایر آدرسها در آن قرار گرفته است. در واقع این نوع صفحات با کاربر تعاملی ندارند و صرفا کاربر می‌تواند موارد داخل این صفحات را مشاهده نماید. (در مورد این برنامه‌ها به تفصیل شرح دادیم)

برنامه‌های دینامیک برنامه‌هایی هستند که داده‌ای را از کاربر دریافت، و آن را مورد پردازش قرار می‌دهند، و در انتها نتیجه آن را در اختیار کاربر قرار می‌دهند. خود برنامه‌های دینامیک به دو بخش تقسیم می‌شوند:

‡ برنامه‌های سمت کاربر (Client side)

‡ برنامه‌های سمت سرور (Server side)

برنامه‌های Client side آنهایی هستند که پردازش روی داده‌ها در سمت کاربر توسط Web Browser انجام می‌شود. برای نوشتن اینگونه برنامه‌ها از زبانهایی چون VB Script, Java Script, ... استفاده می‌شود.

برنامه‌های Server side آنهایی هستند که پردازش روی داده‌ها در سمت سرور انجام می‌شود. بدینگونه که داده‌ها از سمت کاربر به سمت سرور فرستاده می‌شود (به عنوان مثال با Submit کردن فرم سمت کاربر). سپس در سمت سرور عمل پردازش روی داده‌ها انجام شده و نتیجه پردازش به سمت کاربر فرستاده می‌شود. برای نوشتن اینگونه برنامه‌ها در سمت سرور از زبانهایی چون ASP, Net, PHP, ... استفاده می‌شود.

در ادامه این کتاب آموزشی به برنامه نویسی Client Side توسط Java Script اشاره خواهیم کرد. بزودی کتابی در مورد برنامه نویسی PHP تهیه خواهد شد و در آن به برنامه نویسی Server Side توسط PHP اشاره خواهیم نمود.

آشنایی با Java Script

Java Script یک زبان برنامه‌نویسی سمت سرور می‌باشد. این زبان توسط شرکت Netscape اختراع شد و به عنوان اولین زبان اسکریپت نویسی ارایه شد. شرکت میکروسافت نیز برای اینکه از شرکت Netscape عقب نیافتد نسخه دیگری مشابه این زبان را با عنوان Jscript تولید نموده است که خیلی‌ها فکر می‌کنند همان Java Script است. از زبان Java Script برای کارهای زیر می‌توان استفاده نمود:

‣ اعتبارسنجی فرمهای HTML قبل از ارسال به سمت سرور

‣ شناسایی مرورگر کاربران

‣ ایجاد انواع منوها

‣ ...

کدهای نوشته در Java Script توسط کاربرهای دیگر نیز قابل مشاهده هستند. بدین معنا که هر کاربری می‌تواند کدهای نوشته شده در صفحه HTML را ببیند. به عنوان مثال:

View -> Source (Internet Explorer)

Right Click On Loaded Page -> Click On View Page Source item (Firefox)

بنابراین توصیه می‌شود از آن برای برنامه‌نویسی های مهم چون اتصال به Database و ... استفاده نشود

کدهای Java Script هم می‌توانند درون یک صفحه HTML بصورت ترکیبی با کدهای دیگر قرار گیرند و یا در یک فایل جداگانه با پسوند .js.

تفاوتهای Java و Java Script

Java Script مربوط به شرکت Netscpae بوده ولی Java متعلق به شرکت Oracle است.

Java Script یک زبان اسکریپت نویسی است اما Java یک زبان برنامه نویسی است.

Java Script توسط مفسر تفسیر می‌گردد اما Java توسط کامپایلر، کامپایل می‌شود.

بطورکلی: Java Script در صفحات وب تاثیرگذار است اما برنامه‌های Java برای خلق برنامه‌های پیچیده بکار می‌رود.

اصول نوشتن کدهای Java Script

Java Script یک زبان حساس به متن است. در ضمن فاصله بین تگها باید متناسب باشد در غیر اینصورت کار نخواهد کرد. بنابراین زمان نوشتن دستورات خیلی باید دقت کرد. محل قرارگیری کدهای Java Script بین تگهای HTML می باشد. برای اینکه مفسر مرورگر بداند کجا باید کدهای Java Script اجرا و در کجا به آنها خاتمه دهد باید از تگهای `<Script></Script>` استفاده نمود.

شناسه های تگ Java Script :

language: مشخص کننده زبان اسکریپت نویسی است.

type: برای زبان Java Script باید مقدار آن را برابر text/javascript فرار داد.

بطور کلی محل نوشتن کدهای Java Script در سه بخش صورت می گیرد:

۱ - در بخش head ۲-در بخش body ۳-بصورت یک فایل خارجی

نحوه نوشتن کدهای Java Script

الف) درون تگ head: از این مورد زمانی باید استفاده کنید که مایلید کدهایی که وارد می کنید قبل از بقیه کدها اجرا شود. به مثال زیر دقت کنید:

```
<html>
<head>
  <script language="javascript" type="text/javascript">
    function message()
    {
      alert ("Welcome to Farhad and Farzad imany site.");
    }
  </script>
</head>
<body onload="message()">
<h1> Hello Everybody </h1>
</body>
</html>
```

زمان فراخوانی این صفحه ابتدا پنجره ای باز شده و پیام زیر را نشان می دهد:

Welcome to Farhad and Farzad imany site.

ب) درون تگ body: از این مورد زمانی باید استفاده کنید که مایلید کدهایی که وارد می‌کنید در بین برنامه مورد استفاده قرار گیرد. به مثال زیر دقت کنید:

```
<html>
<body>
  <script language="javascript" type="text/javascript">
    document.write("Hello Everybody");
  </script>
</body>
</html>
```

زمان فراخوانی این صفحه پیام زیر داخل صفحه دیده خواهد شد:

Hello Everybody

می‌توانید از دستورات HTML داخل اسکریپت‌های Java Script استفاده نمائید. مانند مثال زیر:

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

ج) بصورت فایل خارجی: از این مورد زمانی باید استفاده کنید که مایلید کدهایی که وارد می‌کنید در یک فایل خارجی با پسوند js باشد و در جایی که از فایل HTML نیاز بود آن را فراخوانی نمائید. این برای زمانی مفید است که قصد دارید از کدهای خود در چندین صفحه استفاده نمائید. به مثال زیر توجه کنید:

ابتدا دستور زیر را در یک فایل با نام message.js ذخیره می‌کنیم:

```
document.write("<h1 align='center'> Hello Everybody </h1>");
```

حال برای فراخوانی message.js در فایل HTML خود بصورت زیر عمل می‌کنیم:

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

فصل دهم: متغیرها و دستورات شرطی در Java Script

متغیرها

برای تعریف یک متغیر در Java Script بصورت زیر عمل می‌کنیم:

```
Var نام متغیر
```

دقت کنید که نام متغیر نباید با یک عدد شروع شود و همچنین هیچگونه فاصله‌ای هم نباید داشته باشد. در ضمن همانگونه که گفته شد در Java Script حالت حساس بودن به حروف وجود دارد. بنابراین متغیری با نام z با متغیری با نام Z کاملاً تفاوت دارد.

اگر متغیری دارای مقدار متنی باشد، باید متن را داخل کوتیشن قرار داد. به مثالهای زیر توجه کنید:

```
Var x;
```

```
Var t="iran";
```

مقدار دهی متغیر می‌توان بصورت جداگانه صورت گیرد. به مثال توجه کنید:

```
Var x;
```

```
X=10;
```

اگر ابتدا به متغیر مقداری بدهید و بعد آن را تعریف کنید، مقدار قبلی آن نگاه داشته می‌شود:

```
X=10;
```

```
Var x;
```

انواع متغیرها

بطور کلی دو نوع متغیر وجود دارد که تفاوت آنها در محل مورد استفاده می‌باشد:

متغیرهای Global: در ابتدای اسکریپت تعریف شده و مقداردهی می‌شوند و سپس در همه جای کد قابل استفاده می‌باشند.

متغیرهای Local: فقط در یک قسمت از اسکریپت کاربرد داشته و معمولاً در توابع استفاده می‌شوند.

انواع عملگر

Arithmetic Operators

عملگرهای ریاضی

| نتیجه | مثال | توصیف | عملگر |
|---------|-----------------------------|-----------|-------|
| 4 | $x = 2, y = 2$ $x + y$ | جمع | + |
| 2 | $x = 5, y = 3$ $x - y$ | تفریق | - |
| 20 | $x = 5, y = 4$ $x * y$ | ضرب | * |
| 5 | $x = 20, y = 4$ x / y | تقسیم | / |
| 3 | $x = 19, y = 4$ $x \% y$ | خارج قسمت | % |
| $x = 6$ | $x = 5$ $x++$ | افزودن | ++ |
| $x = 4$ | $x = 5$ $x--$ | کم کردن | -- |

Assignment Operators

عملگرهای انتصاب

| برابر است با | مثال | عملگرها |
|--------------|------------|---------|
| $x = y$ | $x = y$ | = |
| $x = x + y$ | $x += y$ | += |
| $x = x - y$ | $x -= y$ | -= |
| $x = x * y$ | $x *= y$ | *= |
| $x = x / y$ | $x /= y$ | /= |
| $x = x \% y$ | $x \% = y$ | %= |

عملگرهای مقایسه ای

| مثال | توصیف | عملگرها |
|-----------------------------------|--|--------------------|
| $x = 5, y = 5$ $x == y$ | مقادیر اگر برابر باشند نتیجه درست خواهد بود. | <code>==</code> |
| $x = 5$ $y = "5"$ $x === y$ | مقایسه بین مقادیر و جنس متغیر که در مثال روبرو نتیجه غلط بازگردانده می شود | <code>===</code> |
| $5 != 8$ | اگر مقادیر برابر نباشند نتیجه درست است | <code>!=</code> |
| $8 > 5$ | بزرگتر | <code>></code> |
| $8 < 10$ | کوچکتر | <code><</code> |
| $5 >= 8$ | بزرگتر یا مساوی که در این مثال نتیجه غلط است | <code>>=</code> |
| $5 <= 8$ | کوچکتر یا مساوی که در این مثال نتیجه درست است | <code><=</code> |

عملگرهای منطقی

`&&` عملگری که به معنای ترکیب (و) دو مقدار است $(x < 10 \ \&\& \ y > 1)$

`||` عملگری که به معنای ترکیب (یا) دو مقدار است $(x == 5 \ || \ y == 5)$

`!` عملگری که به معنای (مخالف) است $! (x == y)$

تاریخ و زمان در Java Script

آبکت (new Date()) برای نمایش تاریخ و ساعت جاری بصورت زیر عمل کنید:

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

حال اگر خواستیم جزییات مختلف این آبجکت نظیر سال و ماه و ... را استخراج کنیم بصورت زیر عمل می‌کنیم:

```
<html>
<body>

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

    d= new Date();
    document.write(d+"<br>");
    document.write(d.getFullYear()+"<br>");
    document.write(d.getMonth()+"<br>");
    document.write(d.getDay()+"<br>");
    document.write(d.getHours()+"<br>");
    document.write(d.getMinutes()+"<br>");
    document.write(d.getSeconds());

  </script>

</body>
</html>
```

اگر خواستید سال تاریخ جاری را صرفاً تنظیم کنید بصورت زیر عمل کنید:

```
<html>
<body>

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

    d=new Date();
    d.setFullYear(2007);
    document.write(d);

  </script>

</body>
</html>
```


دستورات شرطی در Java Script

ساختار دستور بصورت زیر است:

If (شرط)

```
{
```

بلوک دستوراتی که در صورت برقراری شرط باید اجرا گردند

```
}
```

else

```
{
```

بلوک دستوراتی که در صورت عدم برقراری شرط باید اجرا گردند

```
}
```

مثال) برنامه‌ای بنویسید که اگر ساعت جاری سیستم از ۱۰ کوچکتر باشد پیام Good Morning را نشان دهد و در غیر اینصورت پیام Good Afternoon را نشان دهد.

```
<html>
<body>

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

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

    if (t<10)
      {
        document.write("Good Morning");
      }
    else
      {
        document.write("Good Afternoon");
      }
  </script>

</body>
</html>
```

If های تودرتو در Java Script

اگر خواستید چند شرط در if مورد بررسی قرار گیرد از elseif استفاده می‌گردد. به مثال توجه کنید:

```
<html>
<body>

  <script language="javascript" type="text/javascript">
    t=21;

    if (t>=0 && t<=5)
    {
      document.write("greater than or equal to 0 and less than or equal 5");
    }

    else if (t>5 && t<=10)
    {
      document.write("greater than 5 and less than or equal to 10");
    }

    else if (t>10 && t<=15)
    {
      document.write("greater than 10 and less than or equal to 15");
    }

    else if (t>15 && t<=20)
    {
      document.write("greater than 15 and less than or equal to 20");
    }

    else
    {
      document.write("undefine range");
    }

  </script>

</body>
</html>
```

Java Script در switch

همانطور که در بالا ملاحظه کردید استفاده از else if خوانایی برنامه را پایین می‌آورد. در این موارد بهتر است از ساختار switch استفاده گردد. به مثال زیر توجه کنید:

```
<html>
<body>
<script language="javascript" type="text/javascript">
var day=new Date().getDay();

switch (day)
{
case 0:
document.write("Today its Sunday");
break;

case 1:
document.write("Today its Monday");
break;

case 2:
document.write("Today its Tuesday");
break;

case 3:
document.write("Today its Wednesday");
break;

case 4:
document.write("Today its Thursday");
break;

case 5:
document.write("Today its Friday");
break;

case 6:
document.write("Today its Saturday");
break;

default:
document.write("Error in processing");
}

</script>

</body>
</html>
```

در این مثال خروجی دستور `var day=new Date().getDay();` که شماره روز تاریخ جاری است در متغیر `day` قرار می‌گیرد. بر اساس مقدار `day` نام روز هفته چاپ می‌گردد. در Java Script عدد صفر معرف یکشنبه و عدد یک معرف دوشنبه و به همین صورت تا آخر ادامه می‌یابد.

پنجره‌های هشدار در Java Script

نوع اول) پنجره صرفا با ظاهر کردن یک پیام فعال می‌شود و کاربر با زدن OK این پنجره را می‌بندد:

```
<html>
<head>
  <script language="javascript" type="text/javascript">
    function message()
    {
      alert ("Welcome to Farhad and Farzad imany site.");
    }
  </script>
</head>
<body>
<input type="button" onclick="message()" value="show alert window">
</body>
</html>
```

نوع دوم) پنجره با ظاهر کردن یک پیام فعال می‌شود و کاربر با زدن OK یا Cancel می‌تواند روند ادامه عملیات را تعیین کند:

```
<html>
<head>
  <script language="javascript" type="text/javascript">
    function disp_confirm()
    {
      var r=confirm("Press a button!")
      if (r==true)
      {
        alert("You pressed OK!")
      }
      else
      {
        alert("You pressed Cancel!")
      }
    }
  </script>
</head>
<body>
<input type="button" onclick="disp_confirm()" value="Display a confirm box" />
</body>
</html>
```

نوع سوم) پنجره با ظاهر کردن یک پیام فعال می‌شود و کاربر می‌تواند مقداری را وارد نماید و سپس بر اساس مقدار وارده تصمیم گیری شود:

```
<html>
<head>
  <script language="javascript" type="text/javascript">
    function disp_confirm()
    {
      var r=prompt("What is your name?","Guest");
      alert ("Hello "+r);
    }
  </script>
</head>
<body>
<input type="button" onclick="disp_confirm()" value="Display a confirm box" />
</body>
</html>
```

فصل یازدهم: سایر دستورات در Java Script

توابع پویا در Java Script

تا کنون توابعی که ذکر شدند صرفاً برای انجام یک عمل خاص بود و نمی‌توانستیم به آنها انعطاف بدهیم. حال می‌خواهیم تابعی بنویسیم که سه عدد مختلف را به آن بدهیم و تابع آنها را در هم ضرب و نتیجه را نشان دهد. برای اینکار:

```
<html>
<head>
  <script language="javascript" type="text/javascript">
    function delta(a,b,c)
    {
      return a*b*c;
    }
  </script>
</head>
<body>
<input type="button" onclick="document.write(delta(2,3,4))" value="Display a confirm box" />
</body>
</html>
```

حلقه‌ها در Java Script

هرگاه بخواهیم یک سری کد به دفعات مشخص تکرار شوند و یا بخواهیم یک سری کد تا زمانی که شرط مورد نظر ما برقرار است اجرا شود باید از حلقه‌ها استفاده کرد. بطور کلی در Java Script دو نوع حلقه داریم: (حلقه for - حلقه while)

(مثال برنامه‌ای با استفاده از حلقه for بنویسید که اعداد یک تا ده را روی صفحه زیر هم نشان دهد:

```
<html>
<head>
  <script language="javascript" type="text/javascript">
    function ring1()
    {
      var i=1;
      for (i=1;i<=10;i++)
      {
        document.write(i);
        document.write("<br>");
      }
    }
  </script>
</head>
<body>
<input type="button" onclick="ring1()" value="run"/>
</body>
</html>
```

مثال) برنامه‌ای با استفاده از حلقه while بنویسید که اعداد یک تا ده را روی صفحه زیر هم نشان دهد:

```
<html>
<head>
  <script language="javascript" type="text/javascript">
    function ring1()
    {
      var i=1;
      while (i<11)
      {
        document.write(i);
        document.write("<br>");
        i++;
      }
    }
  </script>
</head>
<body>
<input type="button" onclick="ring1()" value="run"/>
</body>
</html>
```

این حلقه شرط را در ابتدا تست کرده و تا زمانی که برقرار باشد کار را ادامه خواهد داد. یک نوع دیگر این حلقه بصورت زیر است که خواهید دید. این حلقه بدون توجه به شرط که در انتهای آن است دستکم یکبار اجرا می‌شود. به مثال دقت کنید:

```
<html>
<head>
  <script language="javascript" type="text/javascript">
    function ring1()
    {
      var i=1;
      do
      {
        document.write(i);
        document.write("<br>");
        i++;
      }
      while (i<11);
    }
  </script>
</head>
<body>
<input type="button" onclick="ring1()" value="run"/>
</body>
</html>
```

در این حلقه اگر مقدار متغیر i را برابر 11 قرار دهید خواهید دید که عدد 11 روی صفحه نمایش داده خواهد شد و بعد حلقه توقف خواهد نمود. یعنی شرط را در انتها تست می‌کند.

آشنایی با دستور break

به مثال زیر توجه کنید. مقدار i برابر با 1 است. شرط حلقه نیز بر این موضوع استوار است تا زمانی که مقدار متغیر i از صفر بیشتر است به کار خود ادامه بده. به این حلقه، حلقه بینهایت می‌گویند و هیچگاه متوقف نخواهد شد. وظیفه دستور break کنترل این موضوع در میان حلقه است و از افتادن حلقه به چرخه بینهایت جلوگیری می‌کند.

```
<html>
<head>
  <script language="javascript" type="text/javascript">
    function ring1()
    {
      var i=1;
      while (i>0)
      {
        document.write(i);
        document.write("<br>");
        i++;
        if (i==11) break;
      }
    }
  </script>
</head>
<body>
  <input type="button" onclick="ring1()" value="run"/>
</body>
</html>
```


آشنایی با دستور continue

فرض کنید می‌خواهیم حلقه‌ای بنویسیم که اعداد یک تا ده را به استثنای عدد سه روی صفحه نشان دهد. در آن صورت باید از continue استفاده کرد. به مثال دقت کنید:

```
<html>
<head>
  <script language="javascript" type="text/javascript">
    function ring1()
    {
      var i=1;
      for (i=1;i<=11;i++)
      {
        if (i==3) continue;
        document.write(i);
        document.write("<br>");
      }
    }
  </script>
</head>
<body>
  <input type="button" onclick="ring1()" value="run"/>
</body>
</html>
```

دستور continue هر جای حلقه باشد عملیات به ابتدای حلقه باز می‌گردد.

آرایه‌ها

از آرایه زمانی استفاده می‌کنیم که قصد داریم داخل یک متغیر چندین مقدار را ذخیره نماییم. طرز تعریف آرایه بصورت زیر است:

```
var i = new array();
```

مثال) برنامه ای بنویسید که با استفاده از حلقه for اعداد صفر تا ده را در آرایه پر کرده و با یک حلقه for دیگر آنها را نمایش دهد.

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

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

```
var myarr = new Array();
```

```
for (i=0;i<11;i++) myarr[i]=i
```

```
for (j=0;j<11;j++) document.write(myarr[j]+"<br>")
```

```
</script>
```

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

متدهای آرایه‌ها

متد length: نشان دهنده تعداد خانه‌های آرایه است.

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

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

```
var myarr = new Array();
```

```
for (i=0;i<11;i++) myarr[i]=i
```

```
document.write(myarr.length);
```

```
</script>
```

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

متد concat: برای نمایش دو آرایه در امتداد همدیگر مورد استفاده قرار می‌گیرد:

```
<html>
<body>
<script language="javascript" type="text/javascript">
  var a1=["Reza","Hasan"];
  var a2=["Sara","Shabnam"];
  var a3=a1.concat(a2);
  document.write (a3);
</script>
</body>
</html>
```

متد join: زمانی که محتویات یک آرایه را با document.write (array name); نمایش می‌دهیم، بین آیتمها از علامت ویرگول استفاده می‌شود. اگر بخواهیم این علامت را ما تعیین کنیم از این متد بهره خواهیم گرفت:

```
<html>
<body>
<script language="javascript" type="text/javascript">
  var a1=["Reza","Hasan"];
  document.write (a1.join("&"));
</script>
</body>
</html>
```

متد pop: با بکارگیری این متد آخرین عنصر آرایه از آن بیرون کشیده می‌شود و طول آرایه یک واحد کم می‌شود

```
<html>
<body>
<script language="javascript" type="text/javascript">
  var a1=["Reza","Hasan"];
  document.write (a1.pop());
</script>
</body>
</html>
```

متد push: با بکارگیری این متد یک عنصر به آخر آرایه اضافه می‌شود و تعداد خانه‌های آرایه را نشان می‌دهد.

```
<html>
<body>

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

    var a1=["Reza","Hasan"];

    document.write (a1.push("Hamid"));

    document.write (a1);

</script>

</body>
</html>
```

متد shift: با بکارگیری این متد اولین عنصر آرایه از آن بیرون کشیده می‌شود و طول آرایه یک واحد کم می‌شود.

```
<html>
<body>

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

    var a1=["Reza","Hasan"];

    document.write (a1.shift());

</script>

</body>
</html>
```

متد unshift: با بکارگیری این متد یک عنصر به اول آرایه اضافه می‌شود و تعداد خانه‌های آرایه را نشان می‌دهد.

```
<html>
<body>

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

    var a1=["Reza","Hasan"];

    document.write (a1.unshift("Hamid"));

    document.write (a1);

</script>

</body>
</html>
```

متد unshift در Internet Explorer کار نمی‌کند. برای دیدن نتیجه از
firefox استفاده کنید

متد reverse: بالعکس نمودن محتویات آرایه وظیفه این متد است.

```
<html>
<body>

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

    var a1=["Reza","Hasan"];
    document.write (a1.reverse());
</script>
```

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

متد slice: نقطه ای را دریافت و از آن نقطه به بعد آرایه را بیرون می‌کشد و نمایش می‌دهد.

```
<html>
<body>

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

    var a1=["Reza","Hasan","Ali","Sara","Ahmad","Bahman"];
    document.write (a1.slice(3));
</script>
```

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

متد sort: مرتب سازی آرایه را برعهده دارد

```
<html>
<body>

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

    var a1=["Reza","Hasan","Ali","Sara","Ahmad","Bahman"];
    document.write (a1.sort());
</script>
```

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

فصل دوازدهم: آشنایی با توابع رشته‌ای

تابع length: برای استخراج طول رشته بکار می‌رود.

```
<html>
<body>

<script type="text/javascript">

    var txt = "Farhad and Farzad Imany";

    document.write(txt.length);

</script>

</body>
</html>
```

تابع charAt: برای استخراج یک کاراکتر با شماره اندیس معین از یک رشته متنی بکار می‌رود.

```
<html>
<body>

<script type="text/javascript">

    var txt = "Farhad and Farzad Imany";

    document.write(txt.charAt(2));

</script>

</body>
</html>
```

تابع concat: برای ترکیب دو رشته بکار می‌رود.

```
<html>
<body>

<script type="text/javascript">

    var str1 = "Farhad and ";
    var str2 = "Farzad Imany";

    document.write(str1.concat(str2));

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

تابع indexOf: نقطه شروع یک زیررشته در رشته را برمی‌گرداند.

```
<html>
<body>

<script type="text/javascript">

    var str1 = "Farhad and Farzad Imany";
    document.write (str1.indexOf("Farzad"));

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

تابع lastIndexOf: نقطه شروع یک زیررشته در رشته را برمی‌گرداند. (اگر آن زیررشته چندبار در متن باشد نقطه شروع آخرین

زیررشته را بیرون خواهد داد.)

```
<html>
<body>

<script type="text/javascript">

    var str="Hello planet earth, you are a great planet.";
    document.write (str.lastIndexOf("planet"));

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

تابع substr: برای بیرون کشیدن یک زیررشته از رشته اصلی با دادن نقطه شروع و تعداد برش مورد استفاده قرار می‌گیرد.

```
<html>
<body>

<script type="text/javascript">

    var str="Hello planet earth, you are a great planet.";
    document.write (str.substr(6,6));

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

تابع toLowerCase: تبدیل کل رشته به حروف کوچک.

```
<html>
<body>

<script type="text/javascript">

    var str="Hello planet earth, you are a great planet.";

    document.write(str.toLowerCase());

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

تابع toUpperCase: تبدیل کل رشته به حروف بزرگ.

```
<html>
<body>

<script type="text/javascript">

    var str="Hello planet earth, you are a great planet.";

    document.write(str.toUpperCase());

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

با احترام

فرهاد - فرزاد ایمانی

دانشجویان کارشناسی ناپیوسته مهندسی تکنولوژی فن آوری اطلاعات

(گرایش طراحی صفحات وب)

Farhad.imany@Gmail.com

Far_1779@Yahoo.com

دانشگاه جامع علمی کاربردی - تبریز