

# مقدمات مهندسی اینترنت

- سرفصل‌ها:
- ✓ مقدمه
- ✓ تاریخچه
- ✓ فناوری اطلاعات
- ✓ مفاهیم
- ✓ HTML
- ✓ DHTML

## مقدمه:

مبحث اینترنت و زیرشاخه‌های آن - در یک نگاه کلی - شاخه‌ای از علم شبکه‌های کامپیوتری [1] هستند. در واقع اگر بخواهیم به مباحث سطح بالا و کاربردی شبکه‌های کامپیوتری بپردازیم، اینترنت و تکنولوژی‌های آن از اولویت بالایی برخوردارند [2].

در دنیای امروز که جهت‌گیری اکثر دولت‌ها به سوی بیشتر و زودتر دانستن است، به جرأت می‌توان گفت «فناوری اطلاعات» [3] و ابزار قدرتمند آن یعنی اینترنت در میزان پیشرفت علمی و فنی یک ملت نقش بسزایی دارد. با درک اهمیت یادگیری و بکارگیری اینترنت در زندگی عادی مرور مختصری را روی آن خواهیم داشت.

## تاریخچه:

پس از آنکه شوروی سابق اولین ماهواره‌ی مصنوعی را در سال ۱۹۵۷ به فضا پرتاب کرد، به دلیل رقابت تنگاتنگ نظامی میان آمریکا و شوروی، اداره‌ی دفاع [4] آمریکا به دستور آیزنهاور [5] (رئیس‌جمهور وقت ایالات متحده) اقدام به تشکیل آژانس تحقیقاتی پروژه‌های پیشرفته [6] (آرپا) نمود. به دنبال تشکیل آرپا و برای آماده‌باش و کنترل بیشتر بر روی منابع موشکی، پنتاگون بودجه‌ای را برای تحقیقات دانشگاهی در مورد ساختن یک شبکه‌ی ارتباطی بین مناطق مختلف نظامی آمریکا اختصاص داد. حاصل این تحقیقات بعدها آرپانت [7] نامیده شد که در ابتدا صرفاً ۴ کامپیوتر در آمریکا را به هم مرتبط می‌کرد. آرپانت به سرعت گسترش یافت و با طراحی و پیاده‌سازی پروتکل [8] TCP/IP و همکاری دانشگاه برکلی در فراهم کردن رابط نرم‌افزاری مناسب به کمک سیستم عامل یونیکس این دانشگاه [9]، اینترنت متولد شد.

## فناوری اطلاعات:

آنچه امروز اینترنت را با اهمیت جلوه می‌دهد، دیدگاه گسترش ارتباطات و اطلاعات است. با این رویکرد می‌توان به اینترنت با دید یک ابزار قدرتمند برای گسترش فرهنگ، علم و فناوری نگریست. اهمیت فناوری اطلاعات و ارتباطات تا حدی است که کلان‌ترین بودجه‌های دولت‌های پیشرفته و در حال توسعه را به خود اختصاص داده است. حرکت به سوی آموزش الکترونیکی [10]، دولت الکترونیکی [11]، تجارت الکترونیکی [12] و بسیاری از محصولات و فرآورده‌های IT از سوی کشورهای مختلف، چیزی نیست که بتوان به سادگی از کنار آن گذشت.

درک عمیق تمامی این موارد نیازمند داشتن درکی مناسب از اینترنت و در واقع تکنولوژی تور جهان گستر [13] است. به این منظور مروری خواهیم داشت بر روی مفاهیم اولیه‌ی اینترنت.

## مفاهیم:

### اینترنت:

مخفف واژه‌ی Inter Network است. بزرگترین شبکه‌ی دنیا که از به هم پیوستن تعداد زیادی کامپیوتر ایجاد شده است. این تصور که اینترنت بر روی چند کامپیوتر خاص در دنیا سوار شده به کلی غلط است. در واقع هر کامپیوتری که به طریقی به اینترنت متصل می‌شود، جزئی از اینترنت محسوب می‌شود.

### پروتکل:

واژه‌ی پروتکل در لغت به معنای *قرار داد* است و از دیدگاه علم اینترنت، به قوانینی گفته می‌شود که ۲ یا چند کامپیوتر برای ارتباط (رد و بدل کردن اطلاعات) با یکدیگر رعایت می‌کنند. به عنوان مثال هنگامی که کامپیوتر ۱ در حال صحبت کردن است، کامپیوتر ۲ به حال گوش دادن درمی‌آید و ضمناً میان صحبت کامپیوتر ۱ صحبت نمی‌کند. پروتکل اصلی اینترنت، TCP/IP<sup>[14]</sup> است، که قسمت IP آن وظیفه‌ی حمل بسته‌های اطلاعات از طریق کابل‌ها و شبکه‌ی اینترنت و قسمت TCP هم عهده‌دار تعیین صحت داده و رساندن آن به سطح قابل استفاده برای کاربر را دارد. یک سطح بالاتر و برای دسترسی به سایت‌های اینترنتی از پروتکل HTTP<sup>[15]</sup> یا همان مجموعه قواعد دریافت اطلاعاتی شامل متن، صوت، تصویر و ... (در قالب HTML<sup>[16]</sup>) از تور جهان گستر استفاده می‌کنیم.

## :HTML

نوعی زبان توصیف کننده (کاملاً متفاوت و بسیار آسان‌تر از زبان‌های متداول برنامه‌سازی نظیر C و Pascal) است که برای نمایش دادن صفحات وب به کار می‌رود. تقریباً تمامی صفحات دریافتی ما از اینترنت به این زبان نوشته شده‌اند و مرورگرهای وب<sup>[17]</sup> می‌توانند آن زبان را به ما نمایش دهند.

به واحدهای سازنده‌ی یک کد HTML، تگ<sup>[18]</sup> می‌گویند. در واقع تگ‌ها کلماتی کلیدی هستند که در زبان HTML قرار داد شده‌اند و یک مرورگر وب با دیدن هر کدام از این کلمات، متون و یا تصاویر موجود در صفحه را به نوعی نمایش می‌دهد. به عنوان مثال تکه کد HTML زیر متن «سلام» را به کمک تگ <B>، توپُر<sup>[19]</sup> نمایش می‌دهد.

<B>سلام</B>

## سرویس دهنده<sup>[20]</sup> / سرویس گیرنده<sup>[21]</sup>

تمامی سایت‌های اینترنتی که مشاهده می‌کنیم، از طریق یک سرور (سرویس دهنده) برای ما یعنی سرویس گیرنده (یا مشتری) فرستاده می‌شود. در واقع سرور منتظر یک تقاضای مشاهده‌ی صفحه‌ی وب از جانب مشتری است تا آن صفحه را برای او ارسال کند. مثلاً هنگامی که ما در نوار آدرس مرورگر وب خود آدرس [www.google.com](http://www.google.com) را وارد می‌کنیم، مرورگر در نقش مشتری، درخواستی را به سروری که صفحه‌ی گوگل را در اختیار دارد می‌فرستد. پس از آن سرور در پاسخ به این درخواست، صفحه‌ی گوگل را برای مشتری ارسال می‌کند و مرورگر پس از دریافت، آن را به ما نمایش می‌دهد. لازم به ذکر است که این درخواست و پاسخ – که به Request و Response مشهور است – از طریق پروتکل HTTP صورت می‌گیرد.

## مرورگر وب (Web Browser)

برنامه‌ای است که می‌تواند صفحات HTML را به ما نمایش دهد. برای استفاده از تور جهان گستر، استفاده از یک مرورگر وب تقریباً اجتناب ناپذیر است، چرا که یک مرورگر وب علاوه بر آشنایی و توان نمایش صفحات HTML، می‌تواند به واسطه‌ی پروتکل HTTP صفحات وب را از سرور دریافت کند. اکثر قریب به اتفاق صفحات وب در دنیا توسط مرورگر IE (Internet Explorer) بازدید می‌شوند، چرا که Windows عام‌ترین سیستم عامل مورد استفاده در جهان است که به‌صورت پیش‌فرض IE را داراست. بعد از IE، پرطرفدارترین مرورگر Netscape Navigator (NN) نام دارد. مرورگرهای دیگری نیز وجود دارند از جمله Opera، Mozilla، Safari و ....

## جاوا اسکریپت<sup>[22]</sup>

یک زبان برنامه‌نویسی که فقط در صفحات وب به کار می‌رود، چرا که برخی از امکانات معمول یک زبان برنامه‌نویسی مانند C را ندارد؛ در واقع JavaScript برای استفاده در وب طراحی شده و به همین دلیل زبان سطح بالایی است، به طوری که می‌توان با نوشتن یک خط کد در آن یک پنجره‌ی سؤال (Ok - Cancel) باز کرد و از کاربر پرسش کرد. از آنجایی که HTML صرفاً شامل Tagهایی برای نمایش صفحات وب است، برای کنترل ساختارهای دیگر یک صفحه‌ی وب و یا استفاده از الگوریتم‌های برنامه‌نویسی و یا حتی ساختن صفحات پویا<sup>[23]</sup> که باید روی مرورگر وب اجرا شوند، به JavaScript نیاز داریم. در واقع JavaScript زبانی است که مرورگر وب مترجم و اجرا کننده‌ی آن است. همچنین یادآوری می‌کنیم که JavaScript زبانی است شامل بسیاری از ملزومات زبان‌های برنامه‌نویسی، نظیر حلقه‌ها، شرط‌ها، توابع، رشته‌ها و حتی رویدادها<sup>[24]</sup>.

حال پس از مروری کوتاه بر روی مفاهیم اصلی وب، به HTML می‌پردازیم.

## :HTML

ساده‌ترین و رایج‌ترین زبان برای ساختن صفحات وب است و بدنه‌ای مشابه مثال زیر دارد:

```
<HTML>
<HEAD>
</HEAD>
<BODY>
</BODY>
</HTML>
```

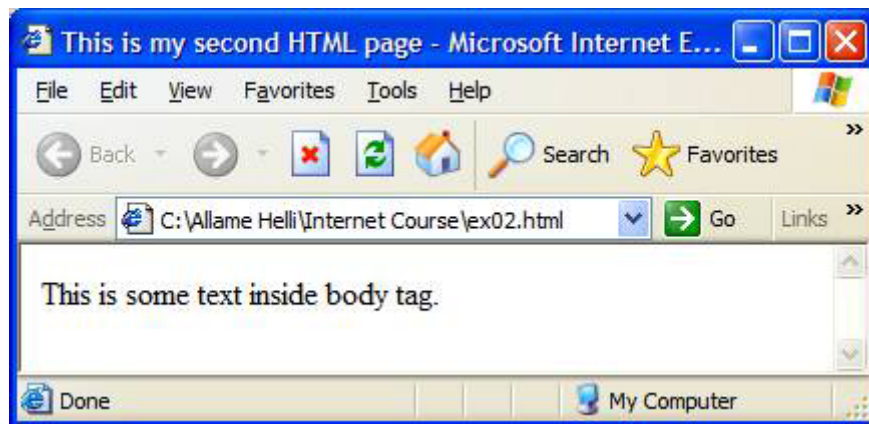
مثال ۱: شکل کلی یک کد HTML (ex01.html)

همان طوری که مثال ۱ نشان می‌دهد، تمامی کد HTML بین دو تگ <HTML> و </HTML> قرار می‌گیرد. ناگفته پیداست که <TAG> نشانگر یک تگ آغازین و </TAG> یک تگ پایانی می‌باشد (به جای TAG هر کدام از تگ‌های رزرو شده‌ی HTML را قرار دهید). شما می‌توانید مثال فوق را در یک فایل خالی نوشته و پسوند آنرا html (و یا htm) بگذارید، سپس آنرا به‌وسیله‌ی یک مرورگر وب (مانند IE) باز کنید؛ با این کار یک صفحه‌ی HTML خالی را خواهید دید [25].  
حال به مثال دوم توجه کنید:

```
<HTML>
<HEAD>
  <TITLE>This is my second HTML page</TITLE>
</HEAD>
<BODY>
  This is some text inside body tag.
</BODY>
</HTML>
```

مثال ۲: یک کد به‌مرد بخور! (ex02.html)

همان‌طور که مشاهده می‌کنید در این مثال از تگ‌های HEAD، TITLE و BODY استفاده کردیم. برای درک بهتر، این کد را در IE باز کریم و نتیجه به صورت زیر بود:



می‌توان حدس زد هر کدام از تگ‌ها چه وظیفه‌ای دارند:

- HTML: تمامی کد ما باید بین این تگ قرار گیرد.
- HEAD: اطلاعات مربوط به کد HTML و یا عنوان صفحه و سایر تنظیم‌ها در این قسمت قرار می‌گیرند.
- TITLE: عنوان صفحه؛ در شکل مشاهده می‌کنید که متنی که بین این تگ قرار گرفته، بر روی نوار عنوان IE ظاهر شده است.
- BODY: قسمتی از HTML که در داخل HEAD نمی‌آید، باید میان تگ BODY قرار گیرد. هر متنی که در قسمت BODY نوشته شود، عیناً در صفحه‌ی ما جا می‌گیرد.

لا بد تا اینجا متوجه شده‌اید که HTML چیزی نیست جز یک سری تگ برای نمایش دادن یک صفحه. حال یک مثال جالبتر را ببینیم:

```
<HTML>
<HEAD>
```

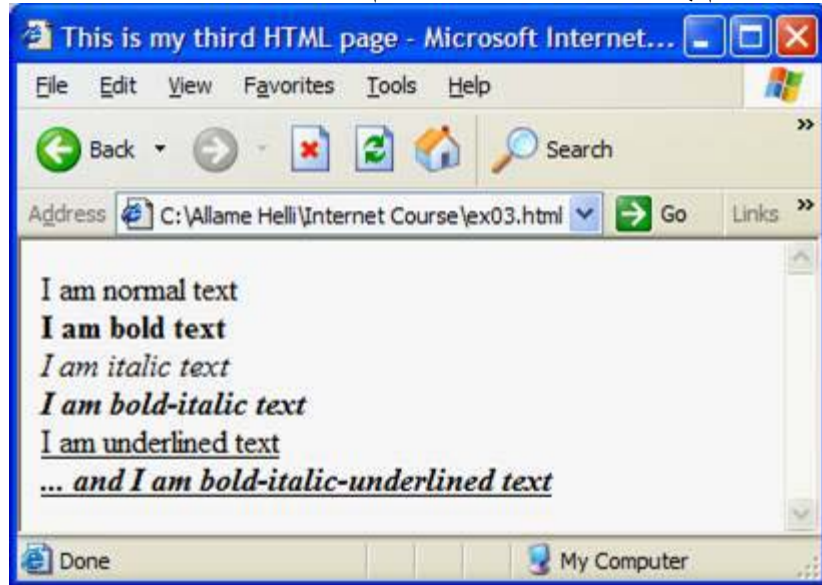
```

<TITLE>This is my third HTML page</TITLE>
</HEAD>
<BODY>
  I am normal text<BR>
  <B>I am bold text</B><BR>
  <I>I am italic text</I><BR>
  <B><I>I am bold-italic text</I></B><BR>
  <U>I am underlined text</U><BR>
  <B><I><U>... and I am bold-italic-underlined text</U></I></B>
</BODY>
</HTML>

```

مثال ۳: چند جور متن قشنگ! (ex03.html)

اگر این مثال را در IE باز کنیم چیزی شبیه شکل زیر را خواهیم دید:



عملکرد تگ‌های B، I و U را به عهده‌ی شما می‌گذاریم (واضحند!). تگ BR، مخفف واژه‌ی line Break است. این تگ خط متن را قطع می‌کند و به خط بعدی می‌رود. نکته‌ی دیگر این‌که تگ BR همواره به صورت تنها — نه جفت ابتدا و انتها— ظاهر می‌شود (به دلیل عمل‌کردش باید اینگونه باشد).

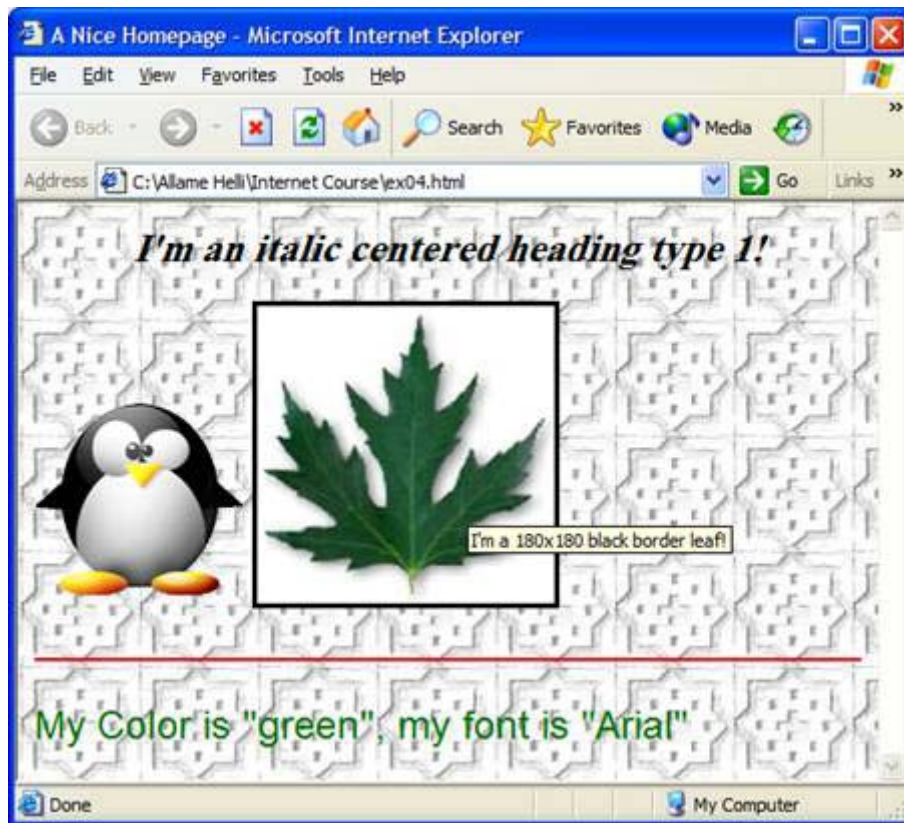
در HTML، هر تگ می‌تواند تعدادی صفت [26] (و یا مشخصه) داشته‌باشد. صفت‌ها در HTML به صورت کلماتی کلیدی هستند که برای هر تگ رزرو شده‌اند و می‌توان هنگام نوشتن تگ آن‌ها را مقاردهی کرد. به مثال زیر توجه کنید:

```

1 <HTML> <!-- You can put some comments here! -->
2 <HEAD>
3 <TITLE>A Nice Homepage</TITLE>
4 </HEAD>
5 <BODY background="pix/wallpaper.gif" dir="ltr">
6 <P>
7 <H2 align="center"><I>I'm an italic centered
8 heading type 1!</I></H1>
9 <IMG src="pix/penguin.gif" title="I'm a penguin!">
10 <IMG src="pix/leaf.jpg" height="180" width="180"
11 border="3" title="I'm a 180x180 black border leaf!">
12 </P>
13 <HR color="red"> <!-- Draw a Horizontal Rule -->
14 <P>
15 <FONT color="green" face="Arial"
16 size="5">My Color is "green", my font is "Arial"</FONT>
17 </P>
18 </BODY>
19</HTML>

```

اگر کد بالا را در IE نمایش دهیم، با فرض اینکه سه فایل تصویری wallpaper.gif (پس‌زمینه‌ی صفحه)، penguin.png و leaf.jpg در فولدر pix موجود باشند، چیزی شبیه شکل زیر را خواهیم داشت:



۲۴

کمی به کد و صفحه‌ی معادل آن توجه کنید. آیا می‌توانید وظیفه‌ی تگ‌های جدید و صفت‌های آن‌ها را بیابید؟ شاید تا حالا متوجه شده باشید که صفت چیست. background برای تگ BODY، یک صفت است. برخی از تگ‌ها بدون صفت کارایی خاصی ندارند؛ به عنوان مثال تگ IMG که مخفف image است، به منظور نمایش دادن یک تصویر (از نوع gif، bmp، jpg، png و ...) به کار می‌رود و بدون صفت src - که مشخص کننده‌ی آدرس تصویر است - کاربردی ندارد. همانطور که می‌بینید IMG یک تگ تنه‌است و تگ پایانی ندارد (مانند BR). همانند تمامی زبان‌های برنامه‌نویسی، HTML هم روشی را برای نادیده گرفتن یک تکه متن در کد در اختیار ما قرار می‌دهد. همان‌طور که در C به‌وسیله‌ی /\* \*/، و یا در Pascal توسط { } کامنت [27] می‌زیم، در HTML هم می‌توان توسط شبه‌تگ زیر، عبارت Something را از دید مرورگر مخفی کرد و به اصطلاح کامنت کرد:

```
<!-- Something -->
```

در مثال ۴ ما تمامی صفت‌ها را در داخل دو-کویتیئن [28] قرار دادیم. مثلاً در تگ HR (خط ۴) صفت color را برابر red قرار دادیم: `<HR color="red">`. این کار اجباری نیست ولی توصیه می‌شود، چرا که گاهی اوقات ممکن است مقدار یک صفت، حاوی کاراکتر فضای خالی باشد که در آن صورت استفاده از کوتیشن ("..." یا '...') اجتناب‌ناپذیر است. `<HR>` مخفف Horizontal Rule است و برای رسم خط‌های افقی در اندازه و رنگ دلخواه به‌کار می‌رود.

در تگ BODY با استفاده از صفت background، تصویر پس‌زمینه و با استفاده از تگ dir (مخفف direction و به معنی جهت)، جهت نمایش دادن کل BODY را تعیین کرده‌ایم. در واقع ltr (left to right) برای متن‌های انگلیسی به کار می‌رود. اگر بخواهیم در BODY -و یا هر تگ دیگر- فارسی بنویسیم باید جهت را rtl (right to left) قرار دهیم.

تگ P برای مشخص کردن یک پاراگراف به کار می‌رود. تگ‌های H1 تا H6 هرکدام نوعی تیتر (Heading) را مشخص می‌کنند. صفت align در تگ H2 روش چیدمان متن میان تگ را تعیین می‌کند. چیدمان می‌تواند راست‌چین (right)، چپ‌چین (left) و یا وسط‌چین (center) باشد. در مورد تگ IMG هم صفت src، آدرس فایل تصویری را معین می‌کند. می‌توانید به جای آدرس موجود ("pix/leaf.jpg" در خط ۱۰)، URL زیر را قرار دهید:

"http://www.google.com/images/logo\_sm.gif"

که با این کار هنگامی که به اینترنت متصل باشید لوگوی کوچک گوگل جایگزین تصویر برگ می‌شود:



صفت title در تگ IMG وظیفه‌ی جالبی دارد! آیا تا به حال چیزی در مورد Tool Tip شنیده‌اید؟ Tool Tip متنی است که برای کمک به کاربر در محیط‌های شبیه Windows ظاهر می‌شوند. مثلاً اگر در Windows، اشاره‌گر ماوس را روی Start Menu ببرید و مدت کمی صبر کنید، Tool Tipی با این مضمون ظاهر می‌شود که Click here to begin. کاری که صفت title در بسیاری از تگ‌های HTML از جمله تگ IMG انجام می‌دهد، تعیین Tool Tip است. مشاهده می‌کنید که در شکل مثال ۴، با نگه داشتن اشاره‌گر ماوس بر روی تصویر برگ، Tool Tip آن ظاهر می‌شود. هنگامی که می‌خواهیم یک تصویر را با استفاده از تگ IMG نمایش دهیم، می‌توانیم از دو صفت width و height به منظور تعیین اندازه‌ی تصویر استفاده کنیم؛ در غیر این صورت تصویر در اندازه‌ی طبیعی خود نمایش داده می‌شود. علاوه بر این‌ها می‌توان به تصویر حاشیه داد. این کار را با استفاده از صفت border انجام می‌دهیم و ضخامت حاشیه‌ی تصویر را مشخص می‌کنیم. پیش‌فرض این صفت، صفر است.

تگ دیگری که در خط ۱۴ مورد استفاده قرار گرفته، HR است و به منظور کشیدن خط افقی به‌کار می‌رود. وظیفه‌ی صفت color هم ناگفته پیداست.

در پاراگراف دیگری که از خط ۱۴ تا ۱۷ کد قرار دارد، از تگ FONT برای مشخص کردن نوع قلم، رنگ و اندازه استفاده کردیم. صفت face برای تعیین نام قلم (در مثال ما Arial)، size برای اندازه و بالاخره color هم برای تعیین رنگ متن میان تگ FONT است.

در ادامه به اختصار چند کد HTML به همراه صفحه‌ی معادل آن‌ها آورده شده. شما می‌توانید به سادگی به وظیفه‌ی تگ‌ها و صفت‌های هر کدام پی ببرید.

```
<HTML>
<HEAD dir="rtl">
<META http-equiv="Content-Language" content="fa">
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
<TITLE>یک مثال خوب از جدول‌ها</TITLE>
</HEAD>
<BODY dir="rtl"><font face="Tahoma">
<P><I>لیست نمرات دانش‌آموزان یک کلاس تخیلی</I></P>
  <TABLE border="3" width="100%" background="pix/paper1.gif">
<TR>
<TD align="center">نام</TD>
<TD align="center">نام خانوادگی</TD>
  <TD align="center">معدل</TD>
</TR>
<TR>
<TD>علی</TD>
<TD>علوی</TD>
<TD>18.5</TD>
</TR>
<TR>
<TD>احمد</TD>
<TD>احمدی</TD>
<TD>17.6</TD>
</TR>
<TR>
<TD>محسن</TD>
<TD>محسنی</TD>
<TD>19</TD>
</TR>
</TABLE></FONT>
</BODY>
</HTML>
```

مثال ۴: یک جدول (TABLE) فارسی و صفت‌های آن (ex05.html)

دقت کنید که تگ META که در خطهای سوم و چهارم مورد استفاده قرار گرفته، صرفاً اطلاعاتی (از قبیل زبان متن و یا کدینگ کاراکترهای آن) را در اختیار مرورگر قرار می‌دهد. تگ‌های TD و TR فقط داخل TABLE کاربرد دارند. TR مشخص‌کننده‌ی یک ردیف و TD نمایانگر یک سلول آن ردیف است. این هم صفحه‌ی تولید شده توسط IE:



تا به حال در مورد لینک [29] در صفحه‌های اینترنت شنیده‌اید؟ لینک‌ها یا بهتر بگوییم Hyper Link ها اشیایی (اعم از متن، تصویر، جدول و ...) هستند که با کلیک ماوس بر روی آنها می‌توانند ما را از صفحه‌ای به صفحه‌ی دیگر ببرند. به مثال زیر توجه کنید:

```
<HTML>
<HEAD dir="rtl">
<META http-equiv="Content-Language" content="fa">
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
<TITLE>Hyperlink و کاربردهای متنوع آن</TITLE>
</HEAD>
<BODY dir="rtl"><FONT face="Tahoma">
<P align="center">
<A href="ex04.html" target="">بر ۴ به صفحه‌ی مثال</A></P>
<P align="center">
<A href="ex04.html" target="_self">
<IMG src="pix/penguin.png" border="0"
title="با کلیک روی من به صفحه‌ی مثال ۴ بروید"></A>
</P>
</FONT>
</BODY>
</HTML>
```

مثال ۶: Hyperlink و کاربردهای آن (ex06.html)

مهمترین تگ برای ایجاد یک Hyperlink، تگ A (مخفف Anchor) می‌باشد. با مقدار دهی به صفت href می‌توان صفحه‌ی مقصد را (که در صورت کلیک کردن روی لینک به آن می‌رویم) تعیین کرد.



صفت target هم استفاده‌های بسیار متنوعی دارد. اگر آنرا \_self قرار دهیم، صفحه‌ی جدید در همان پنجره‌ای که صفحه‌ی اصلی قرار داشت، باز می‌شود؛ و در صورتی که آنرا برابر \_blank قرار دهیم، هنگام کلیک روی لینک، صفحه‌ی مقصد (در اینجا ex04.html) در پنجره‌ای جدید باز می‌شود. دقت کنید که در این مثال هم تصویر و هم متن، به صورت لینک می‌باشند. با بردن ماوس بر روی تصویر پنگوئن، ماوس به حالت دست در می‌آید و Tool Tip آن ظاهر می‌شود. به محض کلیک بر روی عکس، صفحه‌ی مثال ۴ بر روی صفحه‌ی اول باز می‌شود (\_self).

یکی دیگر از تگ‌های بسیار پرکاربرد و جالب در HTML، تگ INPUT است. تمامی فرم‌هایی که بر روی وب مشاهده می‌کنید (نظیر فرم‌های ثبت نام، وارد کردن نام کاربری و کلمه‌ی عبور، فرم‌های نظر خواهی و ...)، کم و بیش از این تگ استفاده می‌کنند. با استفاده از تگ INPUT می‌توان اشیائی مانند دکمه (Button)، جعبه‌ی متن (Text Box)، جعبه‌ی تیک (Check Box) و ... را به‌وجود آورد. برای درک بهتر به مثال زیر توجه کنید:

```
<HTML>
<HEAD dir="rtl">
<META http-equiv="Content-Language" content="fa">
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
<TITLE>فرم درخواست کار</TITLE>
</HEAD>

<BODY dir="rtl">
<FONT face="Tahoma">
<TABLE width="440">
<TR>
<TD width="170">نام و نام خانوادگی:</TD>
<TD><input type="text" value="نام!" width="24"></TD>
</TR>
<TR>
<TD>رمز عبور:</TD>
<TD><INPUT type="password" value="12345" width="24"></TD>
</TR>
<TR>
<TD>جنسیت:</TD>
<TD>
<INPUT type="radio" name="Gender">مرد
<INPUT type="radio" name="Gender">زن
</TD>
</TR>
<TR>
<TD>مایل به همکاری هستم</TD>
<TD><INPUT type="checkbox"></TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

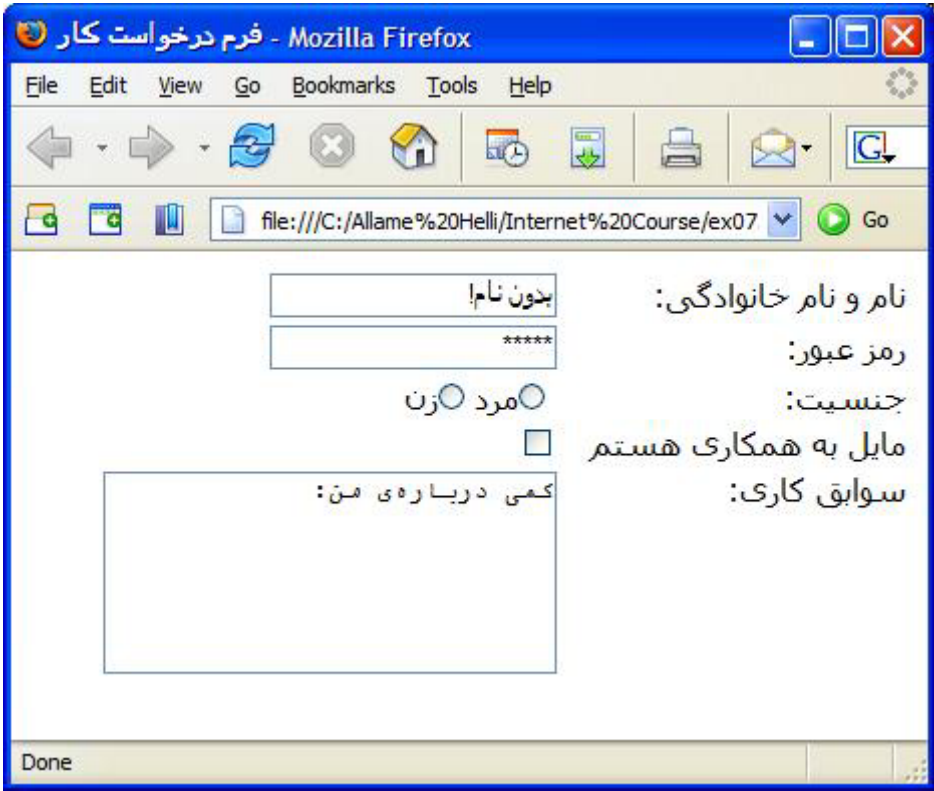
```

</TR>
<TR>
<TD valign="top">سوابق کاری:</TD>
<TD>
<TEXTAREA cols="25" rows="5">کمی درباره‌ی من</TEXTAREA>
</TD>
</TR>
</TABLE>
</FONT>
</BODY>
</HTML>

```

مثال ۷: INPUT و کاربردهای آن (ex07.html)

و این هم شکل HTML ما در یک مرورگر وب دیگر به نام Mozilla Firefox<sup>[30]</sup>:



شاید متوجه شده باشید که کل صفحه‌ی بالا در یک جدول (TABLE) قرار دارد و ضخامت خطوط جدول (یعنی صفت border در جدول) به صورت پیش‌فرض صفر است. همچنین به جعبه‌ی متنی مقابل نام و نام خانوادگی قرار دارد، مقدار پیش‌فرض «بدون نام!» و به جعبه‌ی متن رمز عبور هم «۱۲۳۴۵» را نسبت داده‌ایم. در ادامه صفت valign در تگ TD جلب توجه می‌کند. خیلی خلاصه اینکه v مخفف vertical و به معنی عمودی است. از آنجایی که متن «سوابق کاری» به صورت پیش‌فرض روی خط تقارن (فرضی) افقی جعبه‌ی بزرگ TEXTAREA قرار می‌گیرد، خواستیم آنرا به انتها الیه بالایی سلول خودش در جدول منتقل کنیم. برای اینکه بتوانید درک بهتری از align و valign داشته باشید، بهتر است اولاً به TABLE بالا صفت border="1" را اضافه کنید و ثانیاً به این دو صفت تراز کننده (یعنی align و valign)، مقادیر مختلف center, right, left, top, middle, bottom را نسبت دهید.

خوب! تا حالا مقدار زیادی HTML یاد گرفته‌اید. چند مسئله‌ی مهم دیگر در مورد HTML باقی است که از سطح این جزوه بالاتر است. می‌توانید برای اطلاعات بیشتر، در مورد مطالب زیر خودتان تحقیق کنید (حتماً!!):

- FRAME & FRAMESET tags
- FORM tag and actions
- DIV & SPAN tags & Layers
- APPLET & SCRIPT tags
- Cascading Style Sheets (CSS)

## تمرین‌ها

به نظر شما اگر در کد مثال ۳، تگ‌های <BR> را برداریم چه می‌شود؟ آیا Enterهایی که در متن HTML می‌خورند، رندر [31] می‌شوند؟ Spaceهای اضافه چطور؟ این مطلب را با استفاده از تگ <PRE> هم امتحان کنید (متنی با تعدادی Enter یا کاراکتر New line و تعدادی هم Space و یا Tab در میان تگ‌های آغازی و پایانی PRE بنویسید).

آیا تگ‌های HTML، حساس به متن (بزرگی و کوچکی حروف انگلیسی) [32] است؟ صفت‌های آن‌ها چطور؟ اگر در یک کد HTML، یک تگ غلط بی معنا بنویسیم، چه می‌شود؟ اگر برای یک تگ با معنا، صفت‌های غلط بنویسیم چه خواهد شد؟ اگر به صفت‌ها مقادیر نامربوط نسبت دهیم چه؟ در مثال ۷ به جای دو تا Radio Button [33] از تگ SELECT و درون آن هم OPTION استفاده کنید. حاصل باید چیزی شبیه شکل زیر باشد:



آیا می‌توانید کاراکترهایی شبیه به <، > و یا © را در یک صفحه‌ی HTML نمایش دهید؟

## 🏠 (Dynamic HTML) DHTML

HTML تا ورژن کنونی‌اش یعنی ۴، استاندارد نشد! بگذارید کمی به عقب برگردیم؛ از حدود سال ۱۹۹۳ که اولین مرورگر گرافیکی اینترنت یعنی موزایک [34] در دانشگاه ایلینویز [35] آمریکا توسعه داده شد [36] و پس از آن جای خود را به Netscape Navigator یا همان NN داد، بازار داغ مرورگرها در دست چند شرکت محدود (از جمله Netscape) بود. از آنجایی که استاندارد معینی برای HTML وجود نداشت، شرکت‌هایی که بازار مرورگرها را در اختیار داشتند، خودشان برای مرورگرهایشان تگ و صفت تعیین می‌کردند و شرکت‌های کوچک‌تر ناگزیر از تبعیت آن بودند (مانند الان که بازار مرورگر یا سیستم عامل‌های خانگی در اختیار مایکروسافت است و شرکت‌های تولیدکننده‌ی نرم‌افزار معمولاً برای فروش رفتن محصول خود به ویندوز روی می‌آورند). زبان JavaScript زاینده‌ی همین دوران است. صفحات وب نیاز به پویایی بیشتری داشتند و HTML به هیچ وجه پاسخگوی آن نبود. سرانجام Netscape زبانی را برای مرورگر خود ابداع کرد که پس از مدتی به JavaScript شهرت یافت. دلیل آن هم این بود که در آن مفاهیم برنامه‌نویسی شی‌گر [37] و تا حدی هم دستور زبان Java به‌کار رفته بود. شرکت‌هایی که قصد توسعه‌ی مرورگرهای خود را داشتند ناچار بودند برای فروش نرم‌افزارهای خود، اولاً تگ‌های NN و ثانیاً JavaScript را پشتیبانی کنند. به مرور بازار مرورگرهای وب از دست Netscape خارج شد و در اختیار مایکروسافت قرار گرفت. دلیل اصلی آن هم طراحی بهتر و کامل‌تر نسخه‌ی اسکریپت مایکروسافت بود. مایکروسافت زبان خاص مرورگرهای خود را که به JScript شهرت یافت، توسعه داد ولی بسیار کامل‌تر و آسان‌تر از JavaScript. امروزه تقریباً تمامی مرورگرها JavaScript را به خوبی پشتیبانی می‌کنند ولی JScript را نه، چرا که اولاً قدمت و ثانیاً گستردگی زیادی دارد. البته JScript تفاوت‌های عمده‌ای با JavaScript ندارد جز اینکه از آن کامل‌تر است و به جرأت می‌توان گفت هر کدی که به JavaScript نوشته شده باشد، با مرورگرهای سازگار با JScript هم قابل اجراست.

گزینه‌ی دیگری که برای توسعه‌ی صفحات پویای وب موجود است، استفاده از تکنولوژی جاوا برای صفحات وب یعنی Java Applet است. جاوا آپلت راه حل توسعه‌ی صفحات وب به زبان جاواست که توسط خالق زبان جاوا یعنی شرکت Sun Microsystems ارائه شده‌است و تقریباً قدمتی به اندازه‌ی خود جاوا دارد. مزیت استفاده از اپلت این است که شما واقعاً کد خود را به زبان جاوا می‌نویسید و مطمئن هستید که این کد روی هر سیستم عامل موجود [38] که از طرف جاوا حمایت می‌شود، قابل اجراست. البته در یک اپلت جاوا، تمامی امکانات زبان جاوا (از جمله IOها) در اختیار اپلت قرار نمی‌گیرد، و این به خاطر لحاظ کردن مسائل امنیتی در معماری لایه‌های ماشین مجازی جاوا (JVM) است [39]. برای یادگیری اپلت جاوا باید تا حدی بر زبان جاوا و همچنین مفاهیم برنامه‌نویسی شی‌گرا تسلط داشت، و در ضمن همیشه این نگرانی وجود دارد که آیا JVM بر روی کامپیوتری که صفحه‌ی وب حاوی اپلت را مرور می‌کند وجود دارد یا خیر.

با توجه به مطالب گفته‌شده، به نظر می‌رسد که در حال حاضر بهترین راه ممکن برای ساختن صفحات پویای وب، استفاده از JavaScript است.

شاید تا به حال واژه‌ی DHTML و یا صفحات پویای وب را شنیده باشید. در یک نگاه کلی می‌توان گفت، DHTML، تلفیقی از HTML ایستا (Static)، CSS و JavaScript است:

DHTML = HTML + CSS + JavaScript

در ابتدای بحث یک تعریف مقدماتی از JavaScript ارائه کردیم و اکنون می‌خواهیم چند مثال ساده را ببینیم.

```
<HTML>
<HEAD>
<META http-equiv="Content-Language" content="fa">
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
<TITLE>Recursive Factorial Using JavaScript</TITLE>
</HEAD>
<SCRIPT>
function Factorial(num) {
if(isNaN(num) || num == 0) return 1;
return num * Factorial(num - 1);
}

function showValue(inputTextBox, outputTextBox) {
if(isNaN(inputTextBox.value) || inputTextBox.value.length==0) {
alert ("لطفاً در جعبه‌ی اول عدد وارد کنید!");
return;
}
outputTextBox.value = Factorial(inputTextBox.value);
}
</SCRIPT>

<STYLE>
.TahomaFont {
font-family: Tahoma;
}
</STYLE>

<BODY class="TahomaFont" dir="rtl">
<FORM>
لطفاً یک عدد وارد کنید: <INPUT type="text" name="input_num" size="3"
maxlength="3" dir="ltr">
<INPUT class="TahomaFont" type="button" value="مقدار فاکتوریل ==>"
onclick="showValue(this.form.input_num,
this.form.factorial_result)">
<INPUT class="TahomaFont" type="text" value="فعلاً هیچی"
name="factorial_result" size="10" readonly="yes">
</FORM>
</BODY>
</HTML>
```

مثال ۸: یک مثال مختصر و مفید از CSS و JavaScript (ex08.html)



خوب! در این مثال از تگ‌ها به صورت پیشرفته‌تری استفاده شد؛ به علاوه اولین مثالی است که هم CSS و هم JavaScript دارد. همان‌طور که ملاحظه می‌کنید، برای نوشتن JavaScript از تگ <SCRIPT> و برای نوشتن CSS از تگ <STYLE> استفاده کردیم. در قسمت اول (یعنی همان تگ <SCRIPT>)، دو تابع (function) تعریف کردیم. یکی Factorial() و دیگری showValue(). وظیفه‌ی تابع اول، محاسبه‌ی فاکتوریل عددی است که به آن فرستاده می‌شود (num). این تابع پس از محاسبه‌ی مقدار فاکتوریل (به صورت بازگشتی)، آن را برمی‌گرداند (return). تابع

دوم هم با دریافت اشیاء متناظر دو جعبه‌ی مقدار و حاصل، ابتدا مقدار موجود در جعبه‌ی اول را تعیین صحت می‌کند و سپس در صورت صحت با فرستادن این مقدار به تابع فاکتوریل، خروجی تابع را در جعبه‌ی حاصل قرار می‌دهد. این کد، جزئیات زیادی برای توضیح دارد که اکثر آن‌را می‌توانید خودتان با اجرا کردن برنامه در مرورگرتان و تغییرات در کد دریابید. در اینجا صرفاً به توضیح درباره‌ی صفت class و onclick، بسنده می‌کنیم.

هر [\[40\]](#) Style Sheet می‌تواند تعدادی کلاس در خود داشته باشد که هر کدام از این کلاس‌ها، به تعدادی از صفت‌های Style Sheet مقدار می‌دهند. در مثال فوق یک کلاس به نام TahomaFont تعریف کردیم و در آن با استفاده از صفت font-face، نام قلم [\[41\]](#) را تعیین کردیم. اکنون می‌توانیم این کلاس را به هر کدام از تگ‌های HTML نسبت دهیم تا در آن تگ، از قلم Tahoma برای نمایش دادن محتویاتش استفاده کند. این کار را با استفاده از صفت class انجام دادیم. تقریباً همه‌ی تگ‌های HTML به جز موارد جزئی - دارای صفت class می‌باشند.

صفت دیگری که نیاز به توضیح دارد، صفت onclick است. این صفت مشخص کننده‌ی یک رویداد است. ما از onclick دکمه برای صدا زدن تابع () showValue استفاده کردیم. هنگامی که دکمه فشرده شود، مرورگر به تگ دکمه نگاه می‌کند که آیا رویداد فشرده شدن کلید، ثبت شده است یا نه. در این حال مرورگر، تابع JavaScriptی که به onclick نسبت داده شده باشد (ثبت شده باشد) را اجرا می‌کند.

جزئیات بیشتری که به مدل اشیاء در JavaScript و با صفت‌ها و کلاس‌های CSS مربوط می‌شود، فراتر از حد این جزوه است.

برای تسلط بیشتر بر روی مدل اشیاء در JavaScript مثال زیر را در مرورگر خود اجرا کنید:

```

<HTML>
<HEAD>
<META http-equiv="Content-Language" content="fa">
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
<TITLE>A Nice JavaScript Example</TITLE>
</HEAD>

<SCRIPT>
function response(test_form) {
if(isNaN(test_form.age.value) || test_form.age.value.size == 0) {
alert(" !در جعبه‌ی سن، عدد وارد کنید");
return;
}
var person = test_form.full_name.value;
var years = eval(test_form.age.value) + 1;
document.open(); // open a new html document
document.writeln("<HTML> <BODY dir='rtl'>");
document.writeln("سلام " + person + "!<BR>");
document.writeln("پیشگویی: سال دیگر شما" + years + " ساله می‌شوید");
document.writeln("</BODY> </HTML>");
document.close();
}
</SCRIPT>
<BODY dir="rtl">
<FONT face="Tahoma">
<FORM action="">
لطفاً نام خورا وارد کنید: <INPUT type="text" name="full_name"><P>
چند سال دارید؟ <INPUT type="text" name="age" size="9"><P>
<INPUT type="button" value="پیشگویی" onclick="response(this.form)">
</FORM>
</FONT>
</BODY>
</HTML>

```

مثال ۹: مثال دیگری از JavaScript (ex09.html)

با اجرا کردن این کد در IE، صفحه‌ای شبیه این را خواهیم داشت:



و اگر در قسمت نام، «علی» و در قسمت سن، ۲۱ را وارد کنیم، با فشردن و رها کردن دکمه‌ی «پیشگویی»، صفحه‌ای شبیه صفحه‌ی زیر را خواهیم دید:



پایان

<sup>[1]</sup> Computer Networks

<sup>۲</sup> در حال حاضر به جهت گسترده‌ی مطالب مربوط به اینترنت، می‌توان مهندسی اینترنت را -به تنهایی- یک علم مستقل قلمداد کرد.

<sup>[3]</sup> Information Technology (IT)

<sup>[4]</sup> The Department of Defense (DoD)

<sup>[5]</sup> Dwight D. Eisenhower

<sup>[6]</sup> Advanced Research Projects Agency (ARPA)

<sup>[7]</sup> ARPANET

<sup>[8]</sup> Protocol

<sup>۸</sup> طراحی و پیاده‌سازی سیستم Socket Programming به همراه پشتیبانی از پروتکل TCP/IP در سیستم عامل BSD نسخه‌ی ۴.۲.

<sup>[10]</sup> eLearning

<sup>[11]</sup> eGovernment

<sup>[12]</sup> eCommerce

- [13] World Wide Web (WWW)
- [14] Transmission Control Protocol / Internet Protocol
- [15] HyperText Transfer Protocol
- [16] HyperText Markup Language
- [17] Web Browsers

[18] Tag: در فارسی گاهی به آن برجسب هم می‌گویند؛ ولی ممکن است در آن صورت با واژه‌ی Label به معنای برجسب اشتباه شود. در این جزوه ما از خود واژه‌ی تگ استفاده کرده‌ایم.

- [19] Bold
- [20] Server
- [21] Client
- [22] JavaScript
- [23] Dynamic
- [24] Event

[25] این کار را در Windows می‌توانید به این صورت انجام دهید: notepad را اجرا کنید، متن مثال را در آن نوشته و پس از آن از منوی File، با انتخاب Save As، فایل را به نام دلخواه (مثلاً ex01.html) ذخیره کنید. اکنون آن فایل را در My Computer بیاورید و روی آن دوبار کلیک کنید!

- [26] Attribute
- [27] Comment
- [28] Double Quotation
- [29] Link

<sup>۲۸</sup> برای اطلاعات بیشتر از پروژه‌ی Mozilla، سری به [www.Mozilla.org](http://www.Mozilla.org) بزنید.  
<sup>۲۹</sup> رندر شدن (Rendering) به پروسه‌ای گفته می‌شود که حاصل آن نمایش یک شکل گرافیکی بر روی صفحه‌ی نمایش است. مثلاً برای ساختن تصاویر متحرک ابتدا یک سری ورودی و اشیاء اولیه برای کامپیوتر مهیا می‌کنیم (به عنوان مثال در برنامه‌ی 3D Studio Max) و کامپیوتر با استفاده از اطلاعات و محاسبات پیچیده‌ی ریاضی، شکل حاصل را به ما نشان می‌دهد. اصولاً در علم کامپیوتر رندر شدن را به معنای نمایش دادن هم به کار می‌برند.

- [32] Case Sensitive

<sup>۳۱</sup> دگمه‌ی رادیویی. یعنی اگر یکی انتخاب شود، دیگری بیرون می‌پرد (مانند دگمه‌های ضبط صوت و یا تنظیم‌کننده‌ی امواج رادیوهای قدیمی!).

- [34] Mosaic
- [35] Illinois

<sup>۳۳</sup> فکرش را بکنید تا قبل از آن صفحات اینترنت به صورت متن ساده بودند و تمامی مرورگرها برای محیط‌های متنی (شبیه DOS) نوشته شده بودند. از عکس و فیلم و Flash و صدا و JavaScript و فونت‌های بزرگ و کوچک و ... هم خبری نبود.

- [37] Object Oriented Programming (OOP)

<sup>۳۶</sup> در حال حاضر یک کد جاوا را (با داشتن JVM یا همان Java Virtual Machine) می‌توان روی سیستم عامل‌های Windows، Linux و Mac OS، Unix، و سیستم عامل شرکت Sun اجرا کرد. برای اطلاعات بیشتر در مورد جاوا به آدرس زیر مراجعه کنید:

<http://sun.java.com>

<sup>۳۷</sup> تصور کنید اگر می‌شد توسط یک اپلت جاوا یک فایل روی کامپیوتر ایجاد کرد، چه عواقبی داشت. اپلت می‌توانست یک فایل حاوی ویروس را به جای یکی از فایل‌های مهم ما روی کامپیوتر ایجاد کند و ... شما صرفاً یک صفحه‌ی وب را باز می‌کنید ولی ناگهان ویروس وارد کامپیوتر شما می‌شود!

<sup>۴۰</sup> در طول این درس مقصودمان از Style Sheet همان CSS و صفت‌های آن است

- [41] Font

منبع: <http://khorshid.ut.ac.ir>