

آموزش گام به گام

HTML5

Foundations

جلد اول

مؤلف : مهندس شاهرخ مهرآور

پاییز ۹۶



بسم الله الرحمن الرحيم

HTML5

جلد اول

فهرست مطالب

| | |
|--|----|
| ۱. اینترنت چیست؟ | ۶ |
| 1-1 تاریخچه اینترنت: | ۶ |
| ۲-۱ مرورگرها: | ۹ |
| ۳-۱ وب سایت ها: | ۱۰ |
| 1-3-1 هاست (Host): | ۱۱ |
| ۲-۳-۱ نام دامنه (Domain): | ۱۱ |
| ۳-۳-۱ محتوای وب سایت: | ۱۱ |
| ۲. HTML5 اصول اولیه | ۱۴ |
| 1-۲ مقدمه: | ۱۴ |
| 2-2 تگ های HTML: | ۱۵ |
| ۳-۲ نوشتن اولین سند HTML: | ۱۶ |
| 2-4 اصول نوشتن صفحات HTML: | ۱۸ |
| ۵-۲ تگ <HEAD>: | ۱۹ |
| 2-5-1 تگ <title>: | ۲۰ |
| 2-5-2 تگ <base>: | ۲۰ |
| ۳-۵-۲ تگ <meta>: | ۲۲ |
| 2-6 تگ <BODY>: | ۲۳ |
| ۱-۶-۲ کاربرد تگ <header> و <footer>: | ۲۴ |
| 2-6-2 کاربرد تگ <section> و <article>: | ۲۵ |
| 2-6-3 تگ <nav>: | ۲۶ |
| 2-6-4 تگ <aside>: | ۲۸ |
| ۳. قالب بندی متن در HTML5 | ۲۹ |
| ۱-۳ سرتیترها: H1, H2, H3, H4, H5, H6: | ۳۲ |
| ۲-۳ قالب بندی عبارات: | ۳۵ |
| ۳-۳ خط، بالانویس، پایین نویس: | ۴۲ |
| ۴-۳ کاراکترهای خاص HTML5: | ۴۵ |
| ۵-۳ <P>, <WBR> و <PRE>: | ۴۷ |
| ۶-۳ لیست ها: | ۴۹ |
| ۱-۶-۳ لیست نامرتب: | ۴۹ |
| ۲-۶-۳ لیست مرتب: | ۵۰ |
| ۳-۶-۳ لیست تعریف: | ۵۱ |
| ۷-۳ <DFN> و <ABBR>: | ۵۲ |

4. کار با جدول ها.....

| | | |
|----|----------------------------------|-----|
| ۵۵ | | |
| ۵۵ |<TD> و <TR> ، <TABLE> | ۱-۴ |
| ۵۷ |<TH> | 4-2 |
| ۵۸ |<CAPTION> | 4-3 |
| ۵۹ |<TFOOT> و <THEAD> ، <TBODY> | 4-4 |
| ۶۱ |<COLGROUP> و <COL> | 4-5 |

۵. صفت های عمومی

| | | |
|----|-------|--|
| ۶۵ | | |
|----|-------|--|

6. کار با تصاویر، فیلم ها و صداها

| | | |
|----|-------|--|
| ۷۲ | | |
|----|-------|--|

| | | |
|----|---|-------|
| | ۷۲ | ۱-۶ |
| ۷۲ | صفت <i>alt</i> | ۱-۱-۶ |
| ۷۳ | صفت <i>src</i> | 6-1-2 |
| ۷۴ | صفات <i>width</i> و <i>height</i> | ۳-۱-۶ |
| ۷۵ | صفات <i>usemap</i> و <i>ismap</i> | ۴-۱-۶ |
| ۷۵ |<AUDIO> | 6-2 |
| ۷۶ | صفت <i>src</i> | 6-2-1 |
| ۷۶ | صفت <i>autoplay</i> | 6-2-2 |
| ۷۶ | صفت <i>controls</i> | 6-2-3 |
| ۷۷ | صفت <i>loop</i> | 6-2-4 |
| ۷۷ | صفت <i>preload</i> | 6-2-5 |
| | ۷۸ <VIDEO> | ۳-۶ |
| ۸۰ |<SOURCE> | 6-4 |
| ۸۲ |<PARAM> و <OBJECT> | ۵-۶ |
| ۸۴ |<EMBED> | 6-6 |
| ۸۵ |<FIGCAPTION> و <FIGURE> | 6-7 |

۷. ایجاد لینک و برقراری ارتباط با وب

| | | |
|----|-------|--|
| ۸۷ | | |
|----|-------|--|

| | | |
|----|--------------|-----|
| ۸۷ | تگ <A> | ۱-۷ |
|----|--------------|-----|

۱. پیشگفتار

با عرض سلام خدمت مخاطبان گرامی. کتابی که پیش رو دارید، جلد اول از سری مجموعه ی آموزش برنامه نویسی تحت وب است. در این دوره آموزشی قصد داریم مراحل آموزش برنامه نویسی تحت وب را به شما آموزش دهیم.

در جلد اول (کتاب پیش رو) به آموزش مفاهیم اولیه و اصطلاحات برنامه نویسی تحت وب و همچنین مقدمات HTML5 می پردازیم تا نمای روشنی از برنامه نویسی تحت وب برای مخاطب ترسیم گردد. امید آنکه این مجموعه مورد استفاده شما عزیزان قرار بگیرد.

و من الله التوفیق

۲. اینترنت چیست؟

این سوال کلیشه ای بارها و بارها از سوی رسانه های مختلف مطرح میشود و هربار پاسخی کم و بیش تکراری ، مبهم یا ناقص برای آن انتخاب می شود. اما آیا واقعا نیاز نیست بدانیم اینترنت اولین بار به چه منظور به وجود آمد و چه مراحل و سیاست هایی را پشت سر گذاشت تا به این نقطه رسید؟ از آنجا که بهترین راه شناخت یک پدیده مشاهده تاریخچه آن است، بهتر است برای شناخت اینترنت تاریخچه آن را مطالعه کنیم.

۲-۱ تاریخچه اینترنت^۱:

اتحاد جماهیر شوروی آن زمان موشکی با نام «اسپونیک» (Spotnik) را به فضا می فرستد و نشان می دهد دارای قدرتی است که می تواند شبکه های ارتباطی آمریکا را توسط موشک های بالستیک و دوربرد خود از بین ببرد. آمریکایی ها در پاسخگویی به این اقدام روس ها، موسسه پروژه های تحقیقی پیشرفته "ARPA" را به وجود آوردند. هدف از تاسیس چنین موسسه ای پژوهش و آزمایش برای پیدا کردن روشی بود که بتوان از طریق خطوط تلفنی، کامپیوترها را به هم مرتبط نمود. به طوری که چندین کاربر بتوانند از یک خط ارتباطی مشترک استفاده کنند. در اصل شبکه ای بسازند که در آن داده ها به صورت اتوماتیک بین مبدا و مقصد حتی در صورت از بین رفتن بخشی از مسیرها جابه جا و منتقل شوند. در اصل هدف "ARPA" ایجاد یک شبکه اینترنتی نبود و فقط یک اقدام احتیاطی در مقابل حمله احتمالی موشک های اتمی دوربرد بود. هر چند اکثر دانش امروزی ما درباره شبکه به طور مستقیم از طرح آرپانت "ARPPA NET" گرفته شده است. شبکه ای که همچون یک تار عنکبوت باشد و هر کامپیوتر آن از مسیرهای مختلف بتواند با همتایان خود ارتباط داشته باشد و اگر اگر یک یا چند کامپیوتر روی شبکه یا پیوند بین آنها از کار بیفتاد بقیه باز هم بتوانستند از مسیرهای تخریب نشده با هم ارتباط برقرار کنند.

این ماجرا با وجودی که بخشی از حقایق به وجود آمدن اینترنت را بیان می کند اما نمی تواند تمام واقعیات مربوط به آن را تشریح کند. باید بگوییم افراد مختلفی در تشکیل اینترنت سهم داشته اند آقای "Paul Baran" یکی از مهمترین آنهاست. آقای باران که در دوران جنگ سرد زندگی می کرد می دانست که شبکه سراسری تلفن آمریکا توانایی مقابله با حمله اتمی شوروی سابق را ندارد. مثلا اگر رئیس جمهور وقت آمریکا حمله اتمی متقابل را دستور دهد، باید از یک شبکه تلفنی استفاده می کرد که قبلا توسط روس ها منهدم شده بود. در نتیجه طرح یک سیستم مقاوم در مقابل حمله اتمی روس ها ریخته شد. آقای باران (Baran) تشکیل و تکامل اینترنت را به ساخت یک کلیسا تشبیه کرد و معتقد بود، طی سال های اخیر هر کس سنگی به پایه ها و سنگ های قبلی بنا اضافه می کند و انجام هر کاری وابسته به کارهای انجام شده قبلی است. بنابراین نمی توان گفت، کدام بخش از کار مهمترین بخش کار بوده است و در کل پیدایش اینترنت نتیجه کار و تلاش گروه کثیری از دانشمندان است. داستان پیدایش اینترنت با افسانه و واقعیت در هم آمیخته شده است.

در اوایل دهه ۶۰ میلادی آقای باران طی مقالاتی پایه کار اینترنت امروزی را ریخت. اطلاعات و داده ها به صورت قطعات و بسته های کوچکتری تقسیم و هر بسته با آدرسی که به آن اختصاص داده می شود به مقصد خاص خود فرستاده می شود. به این ترتیب بسته ها مانند نامه های پستی می توانند از هر مسیری به مقصد برسند. زیرا آنها شامل

^۱ این مطلب به نقل از سایت <http://fa.wikipedia.org> درج شده است.

آدرس فرستنده و گیرنده هستند و در مقصد بسته‌ها مجدداً یکپارچه می‌شوند و به صورت یک اطلاعات کامل درمی‌آیند.

آقای باران (Baran) طی مقالاتی اینچنینی ساختمان و ساختار اینترنت را پیش‌گویی کرد. او از کار سلول‌های مغزی انسان به عنوان الگو استفاده کرد، او معتقد بود: وقتی سلول‌های مغزی از بین بروند، شبکه عصبی از آنها دیگر استفاده نمی‌کند و مسیر دیگری را در مغز انتخاب می‌کند. از دیدگاه وی این امکان وجود دارد که شبکه‌ای با تعداد زیادی اتصالات برای تکرار ایجاد شوند تا در صورت نابودی بخشی از آن، همچنان به صورت مجموعه‌ای به هم پیوسته کار کند. تا نیمه دهه ۶۰ میلادی کسی به نظرات او توجه‌ای نکرد. تا اینکه در سال ۱۹۶۵ نیروی هوایی آمریکا و «آزمایشگاه‌های بل» به نظرات او علاقه‌مند شدند و پنتاگون با سرمایه‌گذاری در طراحی و ساخت شبکه‌ای براساس نظریات او موافقت کرد.

ولی آقای باران (Baran) بنابر دلایلی حاضر با همکاری با نیروی هوایی آمریکا نشد. در این میان دانشمندی با نام تیلور (Taiton) وارد موسسه آرپا (ARPA) شد. او مستقیماً به آقای هرتسفلد رییس موسسه پیشنهاد کرد: (ARPA) آرپا هزینه ایجاد یک شبکه آزمایشی کوچک با حداقل چهار گره را تأمین کند که بودجه آن بالغ بر یک میلیون دلار می‌شد. با این پیشنهاد تیلور تجربه‌ای را آغاز کرد که منجر به پیدایش اینترنت امروزی شد. او موفق شد در سال ۱۹۶۶، دو کامپیوتر را در شرق و غرب آمریکا به هم متصل کند. با این اتصال انقلابی در نحوه صدور اطلاعات در دنیای ارتباطات رخ داد که نتیجه آن را امروز همگی شاهد هستیم. این شبکه به بسته‌هایی (packet) از داده‌ها که به وسیله کامپیوترهای مختلف ارسال می‌شدند اتکا داشت. پس از آنکه آزمایشها سودمندی انرا مشخص کردند سایر بخش‌های دولتی و دانشگاهها پژوهشی تمایل خود را به وصل شدن به آن اعلام کردند. ارتباطات الکترونیکی به صورت روشی موثر برای دانشمندان و دیگران به منظور استفاده مشترک از داده‌ها در آمد. در همان زمان که ARPAnet در حال رشد بود تعدادی شبکه پوشش محلی (LAN) در نقاط مختلف آمریکا به وجود آمد. مدیران LANها نیز به وصل کردن کامپیوترهای شبکه‌های خود به شبکه‌های بزرگتر اقدام کردند. پروتوکل اینترنت IP ARPAnet زبان استاندارد حکمفرما برای برقراری ارتباط کامپیوترهای شبکه‌های مختلف به یکدیگر شد. تاریخ تولد اینترنت به طور رسمی اول سپتامبر ۱۹۶۹ اعلام شده‌است. زیرا که اولین "IMP"^۱ در دانشگاه "UCLA" واقع در سانتا باربارا در این تاریخ بارگذاری شده‌است.

از اوایل دهه ۱۹۹۰ رشد استفاده از اینترنت به صورت تصاعدی افزایش یافت. یکی از علل چنین استقبالی ابزار جستجویی مانند Gopher وarchie بوده‌است اما اینها در سال ۱۹۹۱ تحت تاثیر word wide web^۲ قرار گرفتند که به وسیله CERN یا آزمایشگاه فیزیک هسته‌ای اروپا ساخته شد. با آن که اینترنت از ابتدا طوری بود که مبادله اطلاعات برای تازه واردان بسیار ساده باشد. بزرگترین جهش در وب در سال ۱۹۹۳ با عرضه نرم‌افزار موزاییک mosaic که نخستین برنامه مرورگر وب گرافیکی بود به وجود آمد. برنامه موزاییک محصول تلاش دانشجویان و استادان بخش "مرکز ملی کاربردهای ابر کامپیوتر" در دانشگاه ایلینویز آمریکا بود. برای نخستین بار موزاییک امکانات اشاره و کلیک (به وسیله موش) را فراهم کرد. کاربران می‌توانستند صفحات وب (web page) یا مجموعه‌ای از متن و گرافیک را کنار هم بگذارند تا هر کسی که می‌خواست آنها را بتواند روی اینترنت ببیند. وقتی با موش روی کلمه‌ها یا تصاویر خاصی که hyper link نامیده می‌شد کلیک می‌کردند برنامه موزاییک به طور خود کار یک صفحه دیگر باز می‌کرد که به کلمه یا تصویر خاص و کلیک شده اختصاص داشت. بهترین بخش این سیستم آنجا بود که hyper linkها می‌توانستند به صفحاتی روی همان کامپیوتر یا هر کامپیوتر دیگر اینترنت با خدمات وب اشاره کنند. صفحات وب هر روز متولد می‌شدند و مفهوم وبگردی یا surfing روی وب متولد شد. اواسط سال ۱۹۹۴ سه میلیون کامپیوتر به اینترنت وصل شده بود و در آن هنگام اجرای عملیات اهسته نشده بود. صفحات جدید وب که شامل

^۱ پردازشگر ارتباط و پیام

^۲ شبکه تار عنکبوتی جهان گستر (که امروزه به صورت مخفف WWW در ابتدای آدرس های اینترنتی دیده می شود).

همه چیز از اسناد دولتی تا مدارک شرکت‌ها و مدل‌های جدید لباس بود در سراسر دنیا چندین برابر شد. موزاییک و جانشینان آن مانند navigator محصول شرکت "نت اسکایپ" اینترنت را از قلمرو علمی به میان مردم آوردند. طبق آخرین امار ۵۱ درصد کاربران بعد از سال ۱۹۹۵ وارد این محیط شده‌اند. میلیون‌ها انسانی که از اینترنت استفاده می‌کنند نیازی ندارند که نکات فنی مانند TCP/IP را بدانند. امروزه شرکت‌های خدمات دهنده اینترنت یا ISP این کار را به عهده دارند. رشد روز افزون آن و ساده تر شدن استفاده از آن همچنان ادامه دارد. هر چه تعداد مردم بیشتری به اینترنت رجوع کنند تعداد شرکت‌های سازنده برنامه‌های اینترنت بیشتر می‌شود. با آنکه بعضی از عاشقان اینترنت آن را نوعی شیوه زندگی می‌دانند. در نظر بیشتر کاربران منبع سرگرمی اطلاعات است ولی بیشترین مصرف آن پست الکترونیکی یا همان email است که یکی از ابزارهای ارتباطی کار آمد به شمار می‌رود. پیام‌ها از کامپیوتری به کامپیوتر دیگر با سرعت پرواز می‌کنند و منتظر میمانند تا شخص فرصت خواندن آنها را پیدا کند. وب امکانات خوبی برای کپی از نرم‌افزارهای مجاز از لحاظ کپی فراهم میسازد. وقتی که می‌بینیم که در مدت کوتاهی اینترنت به چنین رشدی نایل آمده است، مطمئناً دشوار خواهد بود که آینده او را پیش بینی کنیم. طبق نظر کارشناسان ماهانه ۱۰ درصد به تعداد کاربران اینترنت افزوده می‌شود ولی تعداد دقیق کاربران که روزانه از آن استفاده می‌کنند مشخص نیست. هرچند که پاره‌ای از کارشناسان تعداد آنها را تا ۹۰۰ میلیون نفر حدس می‌زنند. تعداد رسمی کاربران اینترنتی را در سال ۲۰۰۰ کارشناسان ۵۰۰ میلیون نفر اعلام کرده بودند. قطعاً در سال‌های آینده تحولات شگرفی را در زمینه شبکه‌های اینترنتی شاهد خواهیم بود. به وسیله اینترنت انسان به راه‌های جدیدی دست پیدا کرد. در کنار این شانس جدید توسط اینترنت، باید بگوییم خطراتی نیز در رابطه با سیاست و اقتصاد و علم به دنبال خواهد داشت. فرم امروزی اینترنت مدیون همکاری تمام کاربران اینترنت در سرتاسر گیتی است که با این تصور که اطلاعات موجود در سطح جهان را به راحتی با یکدیگر مبادله کنند. این تصویری بود که آقای باران (Baran) از اینترنت داشت و امیدواریم در آینده نیز تکامل اینترنت در این مسیر باشد.

تاریخچه اینترنت در ایران

سال ۱۳۷۱: تعداد کمی از دانشگاه‌های ایران، از جمله دانشگاه صنعتی شریف و دانشگاه گیلان، توسط مرکز تحقیقات فیزیک نظری و از طریق پروتکل UUCP به اینترنت وصل می‌شوند تا با دنیای خارج ایمیل رد و بدل کنند.

سال ۱۳۷۲: در سال ۱۳۷۲ هجری شمسی ایران نیز به شبکه اینترنت پیوست. نخستین رایانه‌ای که در ایران به اینترنت متصل شد مرکز تحقیقات فیزیک نظری در ایران بود. در حال حاضر نیز این مرکز یکی از مراکزهای خدمات اینترنت در ایران است. مرکز تحقیقات فیزیک نظری و ریاضیات، بعنوان تنها نهاد ثبت اسامی قلمرو [ir] در ایران به رسمیت شناخته می‌شود. این قلمرو مشخصه تعیین شده برای هویت ایران در فضای اینترنت است.

سال ۱۳۷۳: مؤسسه ندا رایانه تأسیس می‌شود. پس از راهاندازی اولین بولتن بورد (BBS)، در عرض یک سال نیز اولین وب سایت ایرانی داخل ایران را راهاندازی می‌کند. همچنین، این مؤسسه روزنامه «همشهری» را به زبان فارسی در اینترنت منتشر می‌کند، که این اولین روزنامه رسمی ایرانی در وب محسوب می‌شود. در همین سال به دنبال اتصال به اینترنت از طریق ماهواره کانادایی «کد ویژن» (Cadvision)، مؤسسه ندا رایانه فعالیت بازرگانی خود را بعنوان اولین شرکت خدمات سرویس اینترنتی (ISP) آغاز می‌کند.

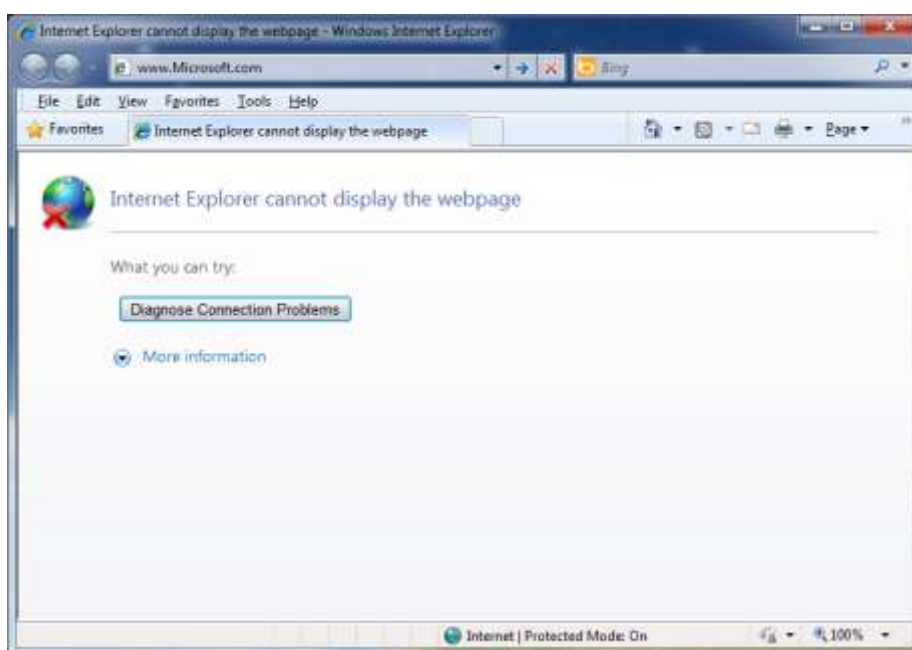
سال ۱۳۷۴: مجلس ایران تأسیس شرکت «امور ارتباطات دیتا» تحت نظر شرکت مخابرات ایران را تصویب می‌کند و مسئولیت توسعه خدمات دیتا در سطح کشور را بطور انحصاری در اختیار آن شرکت قرار می‌دهد.

سال ۱۳۷۷: پروژه یونیکد در ایران با قرارداد شورای عالی انفورماتیک و همکاری بنیاد دانش و هنر واقع در انگلستان و با نظارت و مدیریت فنی دانشگاه صنعتی شریف تحت عنوان «فارسی وب» آغاز می‌شود. هدف پروژه اینست که با گنجاندن کامل و جامع الفبای فارسی در استاندارد یونیکد، نشر فارسی در کامپیوتر، مخصوصاً اینترنت و وب، استاندارد شود و اصولاً مشکل قلم (فونت)‌های غیر استاندارد موجود در نرم‌افزارهای ایرانی حل شود.

حالا که با تاریخچه عمومی اینترنت آشنا شدید بهتر است کمی هم در مورد اصطلاحات فنی آن صحبت کنیم تا مطالب بعدی را آسان تر فرا بگیرید.

۲-۲ مرورگرها

اینترنت در ابتدا تنها برای جابجایی متن بکار می رفت. به تدریج با گسترش این شبکه، نرم افزارهایی با عنوان مرورگر^۱ به وجود آمدند. کار این نرم افزارها دریافت اطلاعات ارسالی از سرور^۲ و نمایش آن در کامپیوتر کاربر بود. امروزه انواع مختلفی از مرورگرها وجود دارند که از معروفترین آنها می توان به اینترنت اکسپلورر^۳ (ساخت شرکت Microsoft)، موزیلا فایرفاکس^۴، گوگل کروم^۵، سافاری^۶ و اپرا^۷ اشاره کرد. در تصاویر زیر، نمایی از مرورگرهای اینترنت اکسپلورر و فایرفاکس را مشاهده می نمایید.



تصویر ۱ – نمایی از مرورگر اینترنت اکسپلورر

^۱ Web Browser

^۲ Server (کامپیوتر سرویس دهنده)

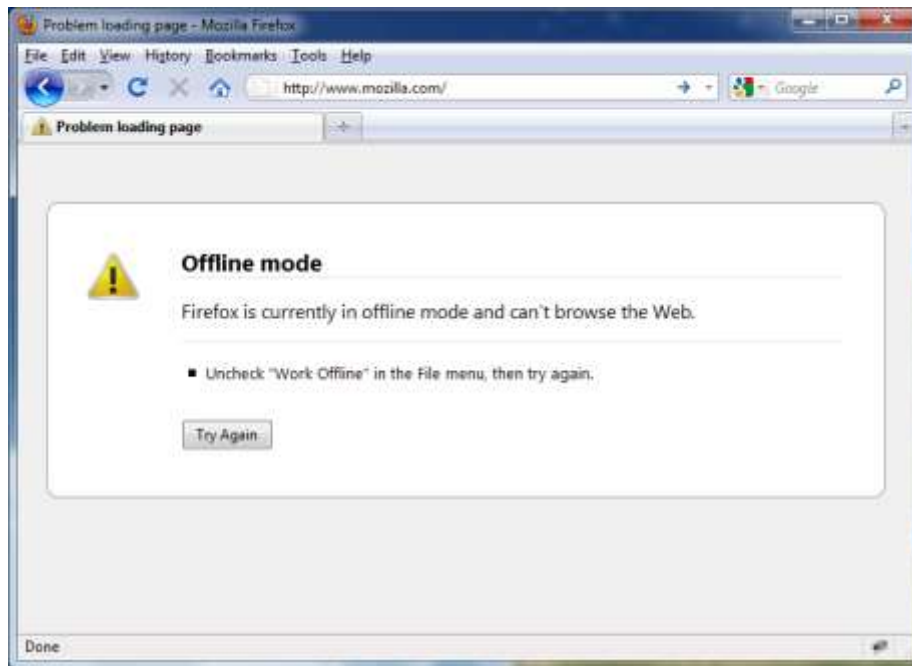
^۳ Internet Explorer

^۴ Mozilla Firefox

^۵ Google chrome

^۶ safari

^۷ opera



تصویر ۲ - نمایی از مرورگر فایرفاکس

با به وجود آمدن نرم افزارهای مرورگر، عناصر تصویری، صوتی و ویدئویی نیز به تدریج پا به عرصه ظهور گذاشتند. نرم افزار های مرورگر از زبان ^۱ HTML برای کار استفاده می کنند که توضیحات مربوط به این زبان را به طور مفصل در فصل های آینده تقدیم حضورتان می کنیم.

۲-۳ وب سایت ها

تا اینجا، با مفاهیم اینترنت آشنا شدیم و دانستیم که اینترنت شبکه ای از کامپیوترهای متصل به هم است که اطلاعات را بین یکدیگر رد و بدل می کنند. بخش مهمی از اینترنت را وب سایت ها تشکیل می دهند. وسعت این بخش به حدی است که بسیاری از افراد این دو مقوله، یعنی اینترنت و وب سایت، را یکی می پندارند. در واقع باید گفت که یک وب سایت، فضای مشخص بر روی هارد دیسک ^۲ یک کامپیوتر متصل به اینترنت است، که صاحب این وب سایت، مطالبی را بر روی آن کامپیوتر قرار می دهد و سایر کاربران می توانند از طریق تمام کامپیوترهای متصل به اینترنت به این مطالب دسترسی داشته باشند.

یک وب سایت از سه قسمت اصلی تشکیل می شود:

- هاست ^۳
- نام دامنه ^۴
- محتوای وب سایت

^۱ Hyper Text Markup Language (زبان نشانه گذاری فوق متن)

^۲ Hard disk

^۳ Host

^۴ Domain

۱-۳-۲ هاست (Host)

هاست یا میزبان، فضایی از حافظه سرور است که مطالب وب سایت بر روی آن ذخیره می شود. هر کامپیوتر متصل به اینترنت می تواند به عنوان هاست یک وب سایت بکار رود اما از آنجا که نیاز است این کامپیوتر همواره روشن باشد و همچنین با پهنای باند بالایی به اینترنت متصل باشد- و تقریباً این امر برای همگان مقدور نیست - معمولاً کاربران از هاست های ارائه شده توسط شرکت های فعال در این زمینه استفاده می کنند. چند عامل در قیمت هاست دخیل هستند که از جمله آن می توان به میزان حافظه ارائه شده، اختصاصی یا غیر اختصاصی بودن سرور، نوع سیستم عامل سرور و نوع سخت افزار سرور می توان اشاره کرد. در ایران نیز شرکت های بسیاری در این زمینه فعالیت می کنند.

۲-۳-۲ نام دامنه (Domain)

نام دامنه یا دامین، آدرس اینترنتی یک وب سایت است که با نوشتن این آدرس در نوار آدرس یک مرورگر، وب سایت مورد نظر برای کاربر نمایش داده می شود. از نام دامنه های معروف اینترنتی می توان به www.google.com، www.yahoo.com، www.ebay.com و www.amazon.com اشاره کرد. باید توجه کرد که هر کامپیوتر در شبکه اینترنت دارای نامی متشکل از یک سری عدد است که به اصطلاح به آن IP^۱ می گویند. در یک شبکه کامپیوتری با داشتن IP یک کامپیوتر، می توان از طریق کامپیوتر های دیگر، به کامپیوتر مورد نظر دسترسی داشت. اما از آنجا که این IP یک عدد چند رقمی است و بخاطر سپردن آن دشوار می باشد، از نام دامنه بجای IP استفاده می کنیم. در واقع نام دامنه نمایانگر IP هاست وب سایت مورد نظر است. دامنه های اینترنتی انواع گوناگون دارند از قبیل .com، .net، .org و غیره. که معمولاً از دامنه .com که مخفف کلمه لاتین company است، برای شرکت ها و کارخانه ها، از دامنه .org که مخفف organization است، برای ارگان ها و سازمانها و از دامنه .net نیز برای سایر کارها استفاده می شود. هر کشور نیز دامنه مخصوص به خود را دارد که از حروف اول نام هر کشور اقتباس شده است. به عنوان مثال دامنه اینترنتی کشوری مثل روسیه .ru و دامنه کشور عزیزمان ایران، .ir است. برای ثبت یک دامین، باید از طریق شرکت های ثبت دامنه اقدام کرد. این شرکت ها، وب سایت هایی دارند که از طریق آنها می توانید آزاد بودن نام دامنه مورد نظر خود را چک کنید^۲. پس از ثبت یک دامین، به مدت یک سال تمام اختیارات یک دامنه به خریدار واگذار می گردد. پس از گذشت این مدت زمان، در صورتی که خریدار تمایلی به تمدید نام دامنه نداشته باشد، دامین، آزاد شده و افراد دیگر میتوانند این دامین را خریداری کرده و مورد استفاده قرار دهند.

۳-۳-۲ محتوای وب سایت

محتوای وب سایت ها از زمان پیدایش تا کنون رو به گسترش و تکامل بوده اند. همانگونه که قبلاً گفتیم، در ابتدای پیدایش اینترنت، فقط متن های ساده از طریق اینترنت مبادله می شد. اما با پیدایش مرورگرها این روند تغییر محسوسی یافت. بطوری که امروزه شاهد وب سایت هایی پویا، زیبا و کاربردی در دنیای اینترنت هستیم. بطور کل، می توان محتوای وب سایت ها را از لحاظ تکامل به چهار دوره زمانی تقسیم کرد:

^۱ ممکن است یک دامین، قبلاً توسط فرد دیگری ثبت شده باشد. در اینصورت تا زمانی که قرارداد صاحب امتیاز آن دامین اعتبار داشته باشد، شخص دیگری نمی تواند آن دامین را ثبت نماید.

- دوره ساخت وب سایت ها با HTML محض و CSS
- پیدایش زبان سمت کاربر JavaScript
- پیدایش زبان های سمت سرور از قبیل asp، jsp و php
- پیدایش Ajax و سایر تکنولوژی های وب ۲

اولین وب سایتهای ساخته شده، صفحات ایستایی بودند که کاربر فقط می توانست چند سطر نوشته ساده را مطالعه کند. این وب سایت ها فاقد هرگونه دکمه، لینک و یا حتی تصویر بودند. در واقع وب سایت های اولیه، نسخه ای کامپوتری از روزنامه های کاغذی آن دوران بودند. به تدریج با رشد اینترنت، محتوای وب سایت ها دچار دگرگونی اساسی شد. وب سایت ها از حالت خشک و بی روح اولیه تبدیل به صفحاتی شدند که علاوه بر تصاویر و عکسها، شامل لینک ها و دکمه های مختلف بود. کاربر میتواند براحتی بین صفحات وب سایت گشت و گذار کرده و به مطالعه مطالب مورد نظر خود بپردازد. تا این زمان، تنها زبان مورد استفاده در وب سایت ها، زبان محبوب HTML بود. در آن زمان، تمام صفحات جداگانه بر روی هاست ذخیره می شد و کاربران میتوانند آن صفحات را از طریق مرورگر درخواست کرده و مشاهده نمایند. پیدایش CSS باعث زیبا تر شدن و اصولی تر شدن طراحی صفحات HTML شد. امروزه کماکان از HTML و CSS در طراحی صفحات وب استفاده می کنیم. آموزش کامل این دو زبان را در فصل های آتی تقدیم حضورتان می کنیم.

با خلق زبان جاوااسکریپت، دوره جدیدی از طراحی صفحات وب آغاز شد. این زبان، یک زبان سمت کاربر (مشتري^۱) بود. به این معنی که می توانست کارهایی را در مرورگر کاربر انجام دهد که تا پیش از آن، زبان HTML قادر به انجام آنها نبود. کارهایی از قبیل alert^۲، بررسی درخواست های کاربران، ایجاد افکت های قابل برنامه ریزی و گزینه های کاربردی دیگر. در واقع جاوااسکریپت یک زبان ساده اما قدرتمند بود که امکانات زیادی را برای برنامه نویسان ایجاد می کرد تا بتوانند با نوشتن توابع مورد نیاز خود، وب سایتهای پویاتری را نسبت به قبل ایجاد نمایند.

متولد شدن زبانهای سمت سرور و استفاده آنها از بانک های اطلاعاتی^۳، مهمترین تحول وب سایت ها و حتی اینترنت به شمار می رود. با وجود زبانهای سمت سرور برنامه نویسان قادر شدند وب سایتهایی را ایجاد نمایند که تا پیش از آن امکان وجودشان مهیا نبود. دیگر زمان آن رسیده بود که کاربران بتوانند با وب سایت ها تعامل داشته باشند و بتوانند با آن ارتباط برقرار کنند. به این ترتیب وب سایتها از حالت ایستای اولیه – که کاربر فقط می توانست صفحات را مطالعه کند – به حالت پویایی تبدیل شدند که کاربران می توانستند اطلاعاتی را بسته به کاربرد وب سایت در دیتابیس آن وبسایت قرار دهند و مطالب مورد نیاز خود را در اختیار بگیرند. به عنوان مثال مهمترین دستاوردهای این زمان به وجود آمدن وب سایت های ارائه دهنده پست الکترونیکی^۴، سایت های خرید مجازی و بانکهای الکترونیکی بود. ASP (با تلفظ ای اس پی)، JSP و PHP سه زبان معروف سمت سرور هستند. که با توجه به کامل، روان، رایگان و قدرتمند بودن زبان PHP، ما در فصل های آینده این کتاب به آموزش این زبان شیرین می پردازیم.

در سالهای اخیر مفهومی در اینترنت بر سر زبانها افتاد تحت عنوان "وب ۲"^۵. کلمه "وب ۲" اولین بار در یک همایش توسط معاون موسسه oreilly مطرح شد. آنها به دنبال نسل جدیدی از وب بودند که بتواند جذاب، کاربردی و قابل گسترش باشد. اینگونه بود که کنفرانس وب ۲ شکل گرفت و بحث های زیادی

^۱ client

^۲ اخطار

^۳ database

^۴ E-mail

^۵ Web 2

پیرامون این پدیده مطرح شد. امروزه جستجوی این عبارت در گوگل بیش از ۱۰ میلیون نتیجه را بر می‌گراند.

باید گفت که وب ۲، یک زبان برنامه نویسی یا نرم افزار خاصی نیست. "وب ۲" در واقع پدیده‌ای است که در نحوه استفاده از فناوری و طراحی سایت‌ها در اینترنت رایج شده است. سایت‌ها یا خدمات اینترنتی که امکان تبادل اطلاعات را بین کاربران فراهم می‌کنند، یا به آنها اجازه تولید یا دستکاری در اطلاعات را می‌دهند، معمولاً "وب ۲" تلقی می‌شوند. به عبارت دیگر در "وب ۲" دیگر کاربران فقط به خواندن مطالب اکتفا نمی‌کنند، بلکه این خود آنها هستند که مطالب را به وجود آورده و اطلاعات خود را با دیگر کاربران به اشتراک می‌گذارند.

برای درک بهتر مفهوم "وب ۲" بهتر است از مقایسه چند سرویس و وبسایت "وب ۱" با سرویس‌ها و وبسایت‌های مشابه آنها در "وب ۲" استفاده کنیم.

| Web 1 | Web 2 |
|-------------------|-----------------|
| Britannica Online | Wikipedia |
| mp3.com | Last.fm |
| Double Click | Google Ad Sense |
| MapQuest | Google Maps |

جدول ۱ مقایسه سرویس‌ها و سایت‌های وب ۱ با وب ۲

- Wikipedia یک وبسایت "وب ۲" است. این سایت یک دایره المعارف رایگان است که هیچ مولف خاصی ندارد و این کاربران هستند که آن را گسترش می‌دهند. در واقع هر کاربر در این سایت امکان ایجاد و ویرایش محتوای وبسایت را دارد. اما دایره‌المعارف بریتانیکا یک سایت "وب ۱" محسوب می‌شود. زیرا تعداد بسیار محدودی مولف هستند که مقالات این سایت را به روز می‌کنند و بر روی سایت قرار می‌دهند.
 - سایت last.fm یک سایت "وب ۲" است. این سایت، یک سایت مرتبط با موسیقی است که کاربران در آن، به ایجاد بیوگرافی خوانندگان، قراردادن عکس‌های آنها در سایت و امتیاز دادن به موسیقی‌های مختلف می‌پردازند. اما در سایت mp3.com که یک سایت "وب ۱" است، تمام این کارها توسط مسئولان سایت انجام می‌گیرد.
 - Double Click و Google Ad Sense هر دو سرویس تبلیغات را ارائه می‌دهند. سایت Double Click یک سایت "وب ۱" است. این سایت از تبلیغات بنری و آگهی‌های pop-up^۱ که از دید کاربران، بسیار آزار دهنده هستند استفاده می‌کند. در مقابل Google Ad Sense از تبلیغات متنی که بسیار ساده و قابل فهم هستند استفاده می‌کند. می‌توان گفت که در سایت Double Click نظر آگهی دهندگان به نظر کاربران ارجحیت دارد اما در سایت گوگل این کاربران هستند که مورد توجه قرار می‌گیرند.
 - تا همین چند وقت پیش، شرکت MapQuest بزرگترین مرجع نقشه‌های جهان به شمار می‌رفت. اما بدون شک امروزه گوگل تنها مرجع بزرگ نقشه‌های دنیاست. سایت گوگل با استفاده از نیروی کاربران خود، به نامگذاری خیابانها، کوچه‌ها، محله‌ها و مناطق مختلف نقشه‌های جهان پرداخته است و به همین خاطر است که اکنون شما براحتی می‌توانید نقشه مناطق مورد نظر خود را حتی با لغات فارسی جستجو کنید.
- سایت‌های "وب ۲" خصوصیات دیگری نیز دارند. به عنوان مثال این سایتها برای وقت کاربرانشان اهمیت زیادی قائل هستند. بطور مثال شما در Gmail^۲ نیازی به ذخیره کردن ایمیل افرادی که به شما میل زده اند ندارید. گوگل

^۱ احتمالاً برایتان پیش آمده است هنگامی که قصد باز کردن یک سایت را در مرورگر خود دارید، به محض بارگذاری صفحه مورد نظر، یک صفحه جدید در مرورگر بطور خودکار باز می‌شود که حاوی تبلیغات و مطالب دیگر است. به این صفحات که بدون درخواست کاربر بصورت اضافی باز می‌شوند، pop-up (با تلفظ پاپ-آپ) می‌گویند.

^۲ سایت پست الکترونیکی گوگل www.gmail.com

بطور خودکار ایمیل این افراد را در لیست شما ذخیره می کند، هنگامی که شما می خواهید برای دوستانتان ایمیل ارسال کنید، کافیست ابتدای نام او را تایپ کنید تا گوگل ایمیل مربوطه را بطور خودکار برایتان نمایش دهد. مثال دیگری از این خصوصیت، نمایش آخرین عبارات جستجو شده توسط کاربران در سایت گوگل است. هنگامی که شما در این سایت قصد جستجوی عبارتی را دارید، گوگل بطور خودکار لیستی از آخرین عبارات جستجو شده توسط کاربران را برای شما به نمایش می گذارد تا نیاز به تایپ کامل عبارت نداشته باشید.

یکی دیگر از خصوصیات مهم سایتهای "وب ۲"، رابط کاربری قدرتمند است. این سایتهای با استفاده از تکنولوژی هایی مانند Ajax، CSS، JavaScript و XML قادرند صفحاتی را تولید کنند که بدون refresh شدن صفحات، اطلاعات مورد نظر ارسال و دریافت شود. این کار باعث بهبود سرعت سایت و راحتی کاربران شده است زیرا رفرش شدن صفحات علاوه بر وقت گیر بودن، از دید کاربران بسیار ناپسند است.

پس بطور کل میتوان خصوصیات یک وبسایت منطبق با مفهوم "وب ۲" را اینطور خلاصه کرد:

استفاده از نیرو، داشته ها و دانش کاربران، در ارتباط بودن کاربران، در اولویت بودن نظر کاربران و در نهایت راحتی و رضایتمندی کاربران.

حال که با تاریخچه تکامل محتوای وب سایت ها آشنا شدید و اصطلاحات پایه ای و نمای کلی آن را فراگرفتید، دیگر وقت آن رسیده که کار خود را بعنوان یک طراح و برنامه نویس وب سایت آغاز کنید. اما قبل از آغاز کار، چند توصیه مهم به شما دانشجوی عزیز داریم و آن اینکه، برای یادگرفتن عمیق مطالب این کتاب لازم است که پایه پای این کتاب به تمرین و تست مطالب آموزش داده شده بپردازید تا تمام دستورات و کدهای مورد نیاز، در ذهن شما ماندگار شده تا به یاری خدا پس از اتمام این کتاب، تبدیل به یک برنامه نویس با تجربه و کارآموده شوید.

ترتیب آموزش، طبق همان تاریخچه تکامل محتوای وب سایت ها است. یعنی ابتدا به آموزش HTML می پردازیم. سپس CSS، JavaScript و الی آخر.

۳. اصول اولیه HTML5

۳-۱ مقدمه^۱

تاریخچه html:

HTML توسط Tim Berners-Lee در خلال دهه ۹۰ میلادی همراه با گسترش وب، شکوفا شد. این زبان توسط مرورگر Mosaic معروفیت خاصی پیدا کرد. در آن زمان HTML در چند مدل منتشر می شد که آن بستگی داشت به سازنده فایل و انجمنهایی که در زمینه وب فعالیت داشتند. در نوامبر ۱۹۹۵ نسخه HTML 2.0 گسترش یافت و بلافاصله در همان سال HTML 3.0 منتشر شد، ولی استقبالی از آن نشد. در سال ۱۹۹۶ انجمن W3C^۲ شروع به فعالیت بر روی نسخه این زبان کرد که حاصل کار آنها در ۱۴ ژانویه ۱۹۹۷ انتشار HTML 3.02 بود. این نسخه توانست رضایت اکثریت را جلب کند چون هماهنگی بیشتری با مرورگرهای مختلف در سیستمهای عامل متفاوت داشت.

در تمام نسخه های این زبان، سعی بر این شده بود تا نظر کسانی که در زمینه وب سرمایه گذاری کرده بودند جلب

^۱ برگرفته از وب سایت wikipedia

^۲ www.consortium کنسرسیوم جهانی وب

شود و برنامه های تولید شده برای وب بتوانند مدت طولانی تری قابل استفاده باشند. به همین منظور HTML برای اهداف گسترده تری، در وب توسعه یافت تا در کلیه سیستمهای اطلاع رسانی و الکترونیکی کوچک و بزرگ با بکار بردن گرافیک و رنگها، قابلیت بهره برداری بیشتری داشته باشد.

در ۱۸ دسامبر ۱۹۹۷ نسخه HTML 4.0 در وب منتشر شد و در همین بین شرکتهای تولید کننده مرورگر وب یکسری مشخصات منحصر بخود را به این نسخه اضافه کردند که قابل اجرا در مرورگرهای دیگر نبود. بعضی از این تغییرات در W3C مورد تایید قرار گرفت اما بعضی دیگر نه. با تغییرات HTML مرورگرها مجبور به تغییر شدند تا با تحولات جدید سازگار شوند. در تاریخ ۲۴ آوریل ۱۹۹۸ در این نسخه تجدید نظر شد و حاصل آن پیدایش HTML 4.01 بود که با کمی تغییر و رفع یکسری مشکلات، در W3C برسمیت شناخته شد و این انجمن استفاده از آنرا به توسعه دهندگان و طراحان وب، توصیه کرد. بطور کل مجموعه HTML 4 با قابلیت استفاده از /embed/ed objects, frames, scripting, style sheets و با کارآیی بالاتر جدولها و فرمها به وب معرفی شد، همچنین در این نسخه توجه زیادی به افراد با توانایی کم شده بود تا این افراد هم بتوانند از محیط وب استفاده کنند.

اما مهمترین قدمی که در این نسخه برداشته شد، پشتیبانی HTML از زبانهایی بود که از راست به چپ نوشته میشدند مانند زبان فارسی، که در این نسخه با پذیرفتن استاندارد ISO 10646 به هدف بزرگ بین المللی شدن این زبان نزدیک شدند تا همه مردم دنیا در هر کجا و با هر زبانی بتوانند اسناد HTML را منتقل کنند. HTML 4 زبان بسیار قوی است برای طراحان و سازندگان محصولات وب اما در آن توجهی به دستگاههای اطلاع رسانی و الکترونیکی کوچک با قدرت و حافظه کمتر نشده است. به همین منظور W3C در ۲۶ ژانویه ۲۰۰۰ اقدام به معرفی XHTML 1.0 کرد و در ۱۹ دسامبر همان سال آنرا به رسمیت شناخت و تاکید به استفاده از این نسخه کرد تا با بکارگیری ویژگیهایی که دارد دامنه استفاده از زبان محبوب HTML را گسترش دهد.

HTML5 نیز به عنوان نسخه جدید زبان نشانه گذاری ابرمتن برای ایجاد صفحات وب گسترش داده شده است. اولین بار در ژوئن ۲۰۰۴ بحث برای ایجاد نسخه جدیدی از زبان نشانه گذاری اچ تی ام ال آغاز شد و در فوریه ۲۰۱۰ نسخه های تقریباً نهایی، از این استانداردها ایجاد شدند. در HTML5 بحث اصلی ایجاد استانداردهایی برای وب چندرسانه ای است. استانداردهایی جدید که شامل قابلیت هایی شبیه پخش پرونده های ویدیویی و کشیدن و رها کردن که قبل ها به افزونه هایی مانند ادوبی فلش، مایکروسافت سیلور لایت و گوگل گیرز وابسته بودند.

۲-۳ تگ های html

HTML یا "زنگام" (مخفف زبان نشانه گذاری ابر متن)^۱، سنگ بنای وب است. یک زبان نشانه گذاری ابر متن که برای تدوین قالب و طراحی صفحات وب بکار می رود. آنچه که باید مورد توجه قرار گیرد اینست که HTML یک زبان برنامه نویسی نیست بلکه یک زبان نشانه گذاری برای فرم دهی به خروجی صفحات در مرورگر و جداسازی عناصر منطقی یک صفحه -نظیر عکس ها و جدولها- از یکدیگر است.

دستورالعمل های این زبان Tag^۲ نام دارند که این تگ ها، محتوای صفحه وب را نشانه گذاری کرده و بدین ترتیب نحوه نمایش صفحه را برای مرورگر تفسیر می کنند. هر تگ دارای نامی است که این نام مشخص می کند که این تگ دقیقاً چه کاری را انجام می دهد. تگ ها در html به منظور متمایز شدن از متن معمولی با علامت های کوچکتر

^۱ واژه مصوب فرهنگستان زبان و ادب فارسی، دفتر اول تا چهارم مصوبات (۱۳۷۶-۸۵)

^۲ برچسب

و بزرگتر (< و >) مشخص می شوند. به عنوان مثال <p> یک تگ html است که شروع یک پاراگراف را نشان می دهد.

بطور کل در HTML دو نوع تگ وجود دارد:

۱. **تگ های دوتایی** که شروع و پایان دارند. این تگ ها با یک تگ آغازین، شروع شده و پس از نوشتن دستورات و متون مورد نظر، در پایان به یک تگ پایانی ختم می شوند. مانند:

```
<p>Hello,Html5</p>
```

در تگ دوتایی <p></p> - که کار ایجاد پاراگراف را بر عهده دارد - <p> را تگ آغازین و </p> - که حاوی علامت اسلش است - را تگ پایانی می نامیم.

۲. **تگ های تکی**. که بصورت یک تگ منفرد نوشته شده و کار خاصی را انجام می دهند مانند تگ <hr> که یک خط افقی را در صفحه رسم میکند.

در html5 حدود یکصد تگ داریم که بعضی از این تگ ها دارای صفاتی نیز می باشند. این صفات اختیاری هستند و شما می توانید در صورت نیاز از آنها استفاده کنید. هر صفت از دو قسمت تشکیل می شود. یکی نام صفت و دیگری مقدار صفت. نام صفت و مقدار صفت توسط علامت مساوی (=) به هم متصل هستند. در زیر نمونه ای از تگ <p> به همراه صفت class را مشاهده می نمایید.

```
<p class="classic">Hello,Html5</p>
```

همان گونه که می بینید صفت class دارای مقدار classic است که این مقدار در داخل علامت نقل قول نشان داده شده است. در html همواره باید مقدار صفت در داخل علامت نقل قول تکی (' ... ') یا نقل قول دوتایی ("...") قرار داده شود. "صفات" هر تگ مشخص کننده جزئیات اضافه ای است که می خواهیم به محتوی تگ اعمال شود. در این کتاب با معرفی هر تگ، صفات آن را - در صورت وجود - نیز برایتان مشخص کرده ایم.

۳-۳ نوشتن اولین سند html

در شکل زیر نمونه ای ساده از یک سری دستور HTML را مشاهده می کنید.

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

تصویر ۳ نمونه ساده ای از دستورات HTML

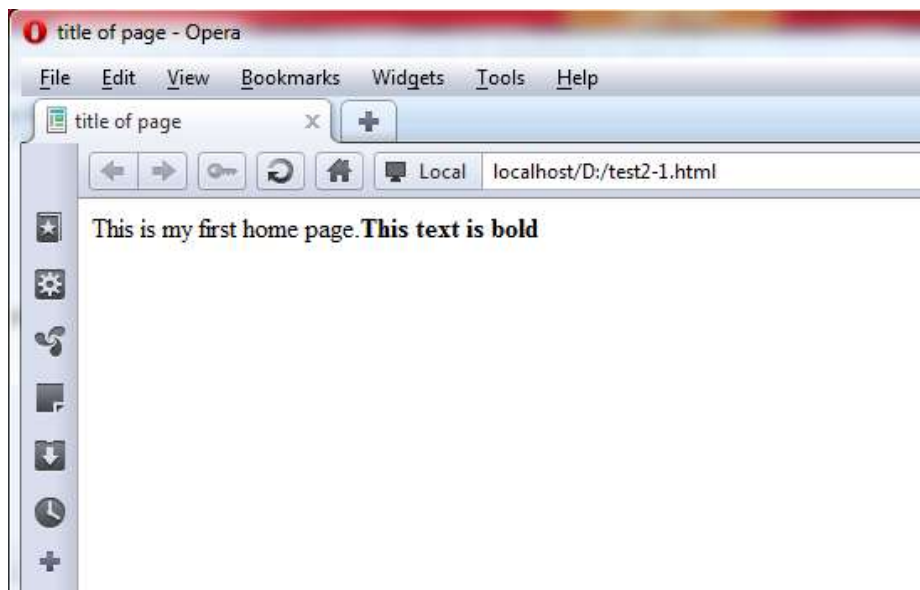
برای نوشتن و ایجاد دستورات HTML نیاز به نرم افزار خاصی نیست. شما می توانید با استفاده از یک ویرایشگر متن ساده مانند نرم افزار Notepad در ویندوز، یک فایل HTML ایجاد کنید.

البته ویرایشگرهایی نیز تحت عنوان ویرایشگرهای WYSIWY ("what you see is what you get") نیز وجود دارند که شما می توانید فایل های HTML خود را با استفاده از آنها ویرایش کنید. این نرم افزارها قابلیت های جالبی دارند کار را برای ایجاد صفحات HTML راحت می کنند مانند نرم افزار های Front Page و Adobe pageMill. اما اگر شما می خواهید یک برنامه نویس وب توانا باشید ما توصیه می کنیم که شما از یک ویرایشگر متنی ساده مانند notepad استفاده کنید تا اصول اولیه HTML را یاد بگیرید و در استفاده از آنها مهارت پیدا کنید.

برای این کار کافی است که دستورات تصویر ۳ را در یک فایل متنی نوشته و این فایل را با پسوند html یا .htm در کامپیوتر خود ذخیره نمایید. زیاد نگران نباشید. تمام دستورات ارائه شده در تصویر بالا را برایتان توضیح میدهیم. اکنون تنها برای آشنا شدن با فایل های HTML این دستورات را در یک فایل متنی نوشته و فایل خود را با پسوند html ذخیره نمایید.

برای ذخیره کردن یک فایل HTML در کامپیوتر، شما میتوانید از هر دو پسوند html و .htm استفاده کنید. دلیل بوجود آمدن پسوند .htm اینست که بیشتر نرم افزار های قدیمی تنها از پسوندهای سه حرفی استفاده می کنند. اما امروزه این مشکل برطرف شده است.

پس از ایجاد فایل HTML نوبت به اجرای آن است. برای اجرای این فایل ابتدا مرورگر خود را باز کرده و از طریق منوی File>Open، فایل خود را انتخاب کرده و اجرا نمایید. ما در این کتاب از مرورگر اپرا (opera) برای تست تمرینات استفاده می کنیم. تصویر زیر نشان دهنده نتیجه اجرای فایل HTML نوشته شده در مرورگر opera است.



تصویر ۴ اجرای دستورات تصویر ۳ در مرورگر اپرا

همانگونه که مشاهده می نمایید، تگ های HTML در خروجی صفحه نمایش داده نمیشوند. در این مثال تنها خط هفتم این دستورات در خروجی صفحه مرورگر نمایش داده شده است. برای دیدن کدهایی که نوشته اید، کافی است در صفحه مرورگر راست کلیک کرده و از منوی باز شده گزینه source را انتخاب کنید^۱. با اینکار پنجره ای باز می شود که کدهای نوشته شده توسط شما بطور کامل در آن به نمایش در می آید.

۳-۴ اصول نوشتن صفحات HTML

نوشتن دستورات HTML اصولی دارد که باید این اصول را در ایجاد صفحات HTML در نظر بگیریم. هر صفحه html با اعلام نوع سند^۲ (doctype) شروع می شود. بخشی مورد نیاز که بیان کننده نسخه^۳ html است که صفحه با آن طراحی می شود. با این کار طراح به مرورگر اعلام می کند که در طراحی صفحه وب خود بعنوان مثال از html4 استفاده کرده است یا html5.

این کار با استفاده از تگی با عنوان <doctype> انجام میگردد. استفاده از این تگ در html4 قدری دشوار بود. اما در html5 این کار به سادگی و تنها با استفاده از عبارت زیر انجام می پذیرد.

```
<!doctype html>
```

با تایپ این تگ در ابتدای صفحه خود، به مرورگر اعلام می کنیم که ما در طراحی صفحه وب خود از html5 استفاده می کنیم.

عموما یک سند HTML شامل دو قسمت کلی می باشد.

^۱ این کار در مرورگرهای مختلف متفاوت است. بعنوان مثال در مرورگر اینترنت اکسپلورر باید پس از راست کلیک کردن، گزینه view source را انتخاب کنید. و در مرورگر فایرفاکس باید گزینه view page source را انتخاب نمایید.

^۲ Document type declaration

^۳ Version (ورژن)

۱. قسمت اعلام نسخه htmlی که صفحه با آن طراحی می شود. (همانطور که گفتیم/این کار با استفاده از تگ `<!doctype html>` انجام می گیرد)
۲. قسمت تعریف صفحه وب که این کار با استفاده از تگ `<html>` انجام می گیرد. در واقع باید گفت که تمامی مندرجات یک صفحه وب، در داخل تگ `<html>` واقع می شود.

```
<!doctype html>
<html>
{کل مندرجات صفحه وب}
</html>
```

تصویر ۵ - دو قسمت اصلی یک صفحه وب

محتویات تگ `<html>` نیز، به دو قسمت اصلی تقسیم می شود:

۱. **قسمت تیترو**، شامل اطلاعاتی در مورد سند HTML مزبور می باشد و در داخل تگ `<head>` درج می شود.
 ۲. **قسمت بدنه**، شامل همه مواردی است که در یک صفحه وب نمایش داده میشود. و در داخل تگ `<body>` درج می شود.
- این دو قسمت را در شکل زیر مشاهده می نمایید :

```
<html>
  <head>
    {قسمت تیترو}
  </head>
  <body>
    {قسمت بدنه}
  </body>
</html>
```

تصویر ۶ - نمایش قسمت های تیترو و بدنه

همانگونه که در شکل مشاهده می نمایید، همواره، تمامی مندرجات صفحات HTML بین دو تگ `<html>` و `</html>` قرار می گیرند.

۳-۵ تگ `<head>`

این تگ مشخص کننده بخشی است که شما باید عنوان صفحه، کلمات کلیدی، و دیگر توضیحات خود برای این صفحه را در آن قرار دهید. جالب ترین نکته در مورد این تگ این است که هیچ کدام از عناصر داخل آن، در صفحه به نمایش در نمی آیند.

تگ `<head>` می تواند تگ های زیر را در بر بگیرد:

`<Title>`, `<base>`, `<link>`, `<meta>`, `<script>`, `<style>`

که در این بخش به توضیح این تگها می پردازیم.

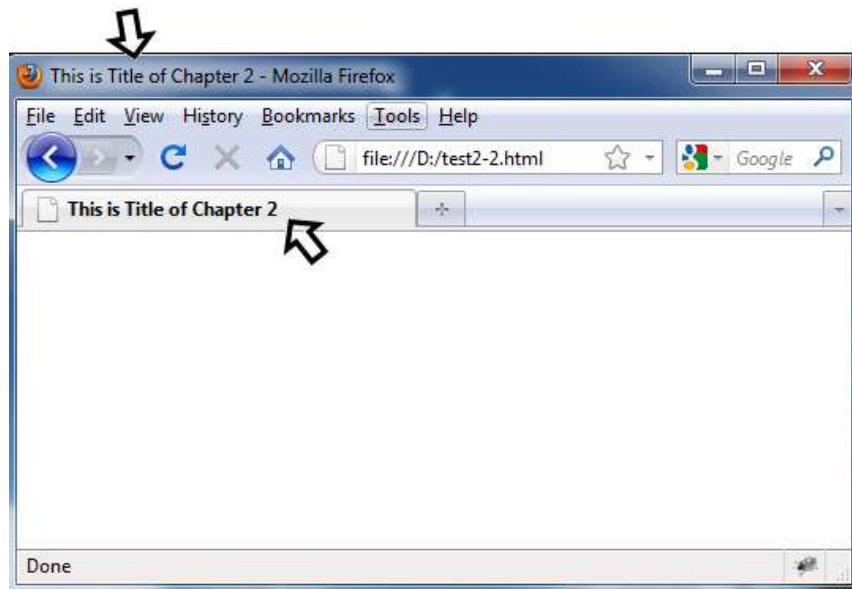
۳-۵-۱ تگ <title>

این تگ به شما این امکان را می دهد که برای صفحه خود یک "عنوان" ایجاد کنید. مرورگر ها معمولاً این مقدار را در نوار عنوان خود نشان می دهند و در بوک مارک ها^۱ از این اسم به عنوان نام پیش فرض استفاده می کنند.

```
<html>
<head>
<title>This is Title of Chapter 2</title>
</head>
<body>
</body>
</html>
```

تصویر ۷ – استفاده از تگ title

در شکل زیر نتیجه اجرای این دستورات را مشاهده می کنید. توجه کنید که حاصل تگ <title>، در تصویر، با استفاده از فلش، مشخص شده است.



تصویر ۸ – اجرای دستورات در مرورگر فایرفاکس

۳-۵-۲ تگ <base>

این تگ باعث کوتاه تر شدن لینک ها می شود. شکل عمومی این تگ به صورت زیر است:

<base href="URL"^۲>

^۱ Bookmark هنگامی که در اینترنت در حال وبگردی هستید و به صفحه ای برخورد می کنید که نیاز دارید آدرس آن صفحه را حفظ کنید، می توانید از قابلیت بوک مارک استفاده کنید. با این کار، مرورگر آدرس صفحه مورد را در لیستی ذخیره می کند که به آن لیست بوک مارک می گویند و شما می توانید با مشاهده لیست و کلیک بر روی بوک مارک مورد نظر، به صفحه مورد نظر بروید.

^۲ Uniform Resource Locator آدرس دهی منبع یکنواخت، آدرس اینترنتی

همانگونه که در فصل های آتی خواهید آموخت، هنگامی که شما در صفحه خود نیاز به نمایش تصاویر داشته باشید، باید مشخص کنید که این تصویر در کجا قرار دارد. زمانی که از یک تصویر استفاده می کنید، این تگ به کار نمی آید. اما هنگامی که قصد دارید از چندین تصویر که در یک پوشه قرار دارند، در صفحه خود استفاده کنید، این تگ با کوتاه کردن آدرس، کار را بسیار راحت خواهد کرد. همچنین اگر تصمیم گرفتید محل پوشه تصویر را تغییر دهید، دیگر نیاز نیست که تگ لینک ها را اصلاح کنید. تنها کافیست که پارامتر href در تگ <base> را تغییر دهید.

به عنوان مثال فرض کنید می خواهیم از تعدادی تصویر در صفحه خود استفاده کنیم. در حالت عادی (و بدون استفاده از تگ <base>، کد ما به صورت زیر خواهد بود:

```
<html>
<head>
  <title>This is Title of Chapter 2</title>
</head>
<body>
  
  
  
  
  
  
</body>
</html>
```

تصویر ۹ - نمایش تصاویر بدون استفاده از تگ <base>

حال همین کد را با استفاده از تگ <base> بازنویسی می کنیم.

```
<html>
<head>
  <base href="www.example.com/page_images/lesson1">
  <title>This is Title of Chapter 2</title>
</head>
<body>
  
  
  
  
  
  
</body>
</html>
```

تصویر ۱۰ - نمایش تصاویر با استفاده از تگ <base>

همانگونه که مشاهده می کنید، استفاده از تگ <base> باعث کوتاه تر شدن آدرس مربوط به تصاویر شده است. زمانی که مرورگر به تگ - که مخصوص نمایش تصویر است - می رسد، URL اولیه را از تگ <base> گرفته و با فایل درخواست شده ترکیب می کند. استفاده از تگ <base> موجب کمتر شدن تعداد خطوط دستورات و در نتیجه کاهش حجم کلی صفحه خواهد شد.

بطور حتم تا کنون از موتورهای جستجو از قبیل گوگل و یاهو استفاده کرده اید. هرگاه در این سایت ها عبارتی را جستجو می کنید، لیستی از سایتهایی که عبارت مورد نظر در آنها قرار دارد، به صورت صفحه به صفحه برای شما به نمایش در می آید. در لیست نمایش داده شده، هر سایت که بیشترین ارتباط را به عبارت مورد جستجو داشته باشد، در بالای لیست قرار می گیرد. قرار گرفتن در صفحه نخست جستجو، برای مدیران وب سایتهای اهمیت حیاتی دارد. به همین منظور تمام تلاششان را انجام می دهند که موتورهای جستجو، این سایت ها را در رده بالای لیست جستجو قرار دهند. برای این کار یک سری کارها باید انجام شود و یک سری استانداردها نیز باید رعایت گردد که مجموعه اینها تخصص جدیدی را در دنیای اینترنت به وجود آورده اند به نام "SEO".

متخصصین "سئو" باید آشنایی کامل با موتورهای جستجو و نحوه امتیاز دهی آنها داشته باشند تا بتوانند سایت های خود را در صدر نتایج جستجو قرار دهند. موتورهای جستجو نیز برای بهینه سازی و سرعت بخشیدن به عمل جستجو، یک سری استانداردهایی را برای سازندگان وب سایت ها تعریف کرده اند تا با پیروی از این استانداردها، موتورهای جستجو راحت تر بتوانند مطالب وب سایت ها را جستجو و امتیاز بندی کنند. یکی از مواردی که باعث بهبود وضعیت یک سایت در نتایج جستجو می شود، استفاده از تگ <meta> است.

این تگ، اطلاعاتی در مورد صفحه HTML مهیا می کند که موتورهای جستجو از این اطلاعات برای دسته بندی صفحات وب سایت ها استفاده می کنند. شما می توانید در این تگ، کلمات کلیدی و یا توضیحاتی در مورد وب سایت خود قرار دهید تا موتورهای جستجو با استفاده از آن بتوانند صفحه شما را به نحو بهتری فهرست کنند.

یکی از کاربردهای دیگر تگ meta، تعیین نوع کدینگ صفحه می باشد. حتماً تا بحال با سایت هایی مواجه شده اید که به زبان فارسی نوشته شده اند اما هنگام بازدید از آنها، صفحه با کلمات به هم ریخته و غیر قابل فهم نمایش داده می شود. دلیل این مشکل ناسازگار بودن encoding انتخاب شده در مرورگر با زبان فارسی است. برای رفع این مشکل و تنظیم خودکار encoding مرورگر، از تگ meta استفاده می کنیم. این کار در نسخه های قبلی html کمی دشوار تر بود اما در html5 به سادگی با استفاده از کد زیر انجام می شود:

```
<meta charset='utf-8'>
```

تگ <meta> یک کاربرد دیگر نیز دارد و آن به روز رسانی^۲ خودکار صفحه توسط مرورگر با استفاده از پارامتر http-equiv است.

حالا برای آشنایی کامل شما با این تگ مثالهای زیر را مشاهده نمایید.

- نوشتن کلمات کلیدی در تگ <meta>:
<meta name='keyword' content='HTML,CSS,JAVASCRIPT,PHP,AJAX'>
- نوشتن توضیحات مربوط به سایت در تگ <meta>:
<meta name='description' content='This is a book for learning web programing'>
- تنظیم encoding صفحه:
<meta charset='utf-8'>
- به روز رسانی خودکار صفحه در هر ۳۰ ثانیه:
<meta http-equiv='refresh' content='۳۰'>

با استفاده از این کد، صفحه وب مورد نظر، هر ۳۰ ثانیه یکبار refresh می شود.

البته http-equiv تگهای دیگری نیز دارد می توانید آنها را در سایت http://www.w3schools.com/html5/att_meta_http-equiv.asp مشاهده نمایید.

^۱ Search engine optimization . بهینه سازی برای موتور جستجو

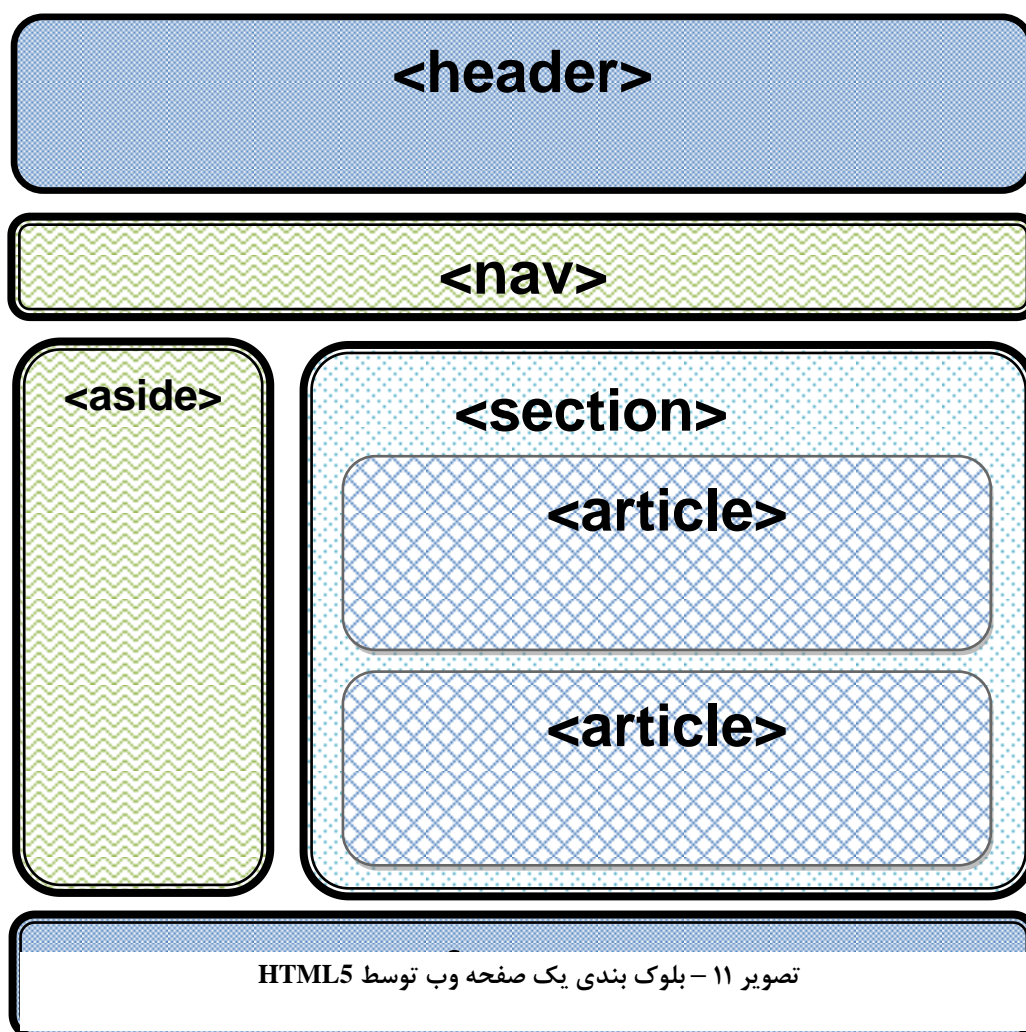
^۲ refresh

تا این قسمت با سه تگ `base.title` و `meta` آشنا شدید، تگهای `script.link` و `style` را در فصل های آینده توضیح می دهیم.

۶-۳ تگ `<body>`

همانگونه که در ابتدای این بخش گفتیم، بدنه اصلی صفحه وب سایت، درون تگ `<body>` قرار می گیرد و هر عنصر و متنی که خارج از این تگ قرار بگیرد، در صفحه مرورگر قابل مشاهده نخواهد بود. ساختار قالب صفحات وب در `html5` نسبت به نسخه های پیشین تغییر محسوسی کرده است. در `html5` محتوا توسط طراح صفحه مشخص می شود که به این کار بلوک بندی می گویند. یعنی اینکه طراح هر قسمت از صفحه را در داخل یک بلوک قرار می دهد. این کار باعث ایجاد تحولی در عرصه موتورهای جستجو می گردد. به این ترتیب که شما با مشخص کردن مکان دقیق مطالب خود در صفحه، آنها را از سایر قسمت های صفحه که همیشه ثابت هستند جدا کرده و باعث دسته بندی صحیح صفحات برای موتور های جستجو می شوید.

تصویر زیر، بلوک بندی یک صفحه وب بر اساس `html5` را نمایش می دهد.



همانگونه که در تصویر فوق مشاهده می کنید، در HTML5، صفحه وب توسط ۶ تگ مختلف به بلوک های متفاوتی تقسیم شده است. این تگ ها هر کدام مفهوم خاص خود را دارند و صفحه را از لحاظ منطقی به قسمت های :
header,nav,section,article,aside و footer تقسیم بندی می کند.

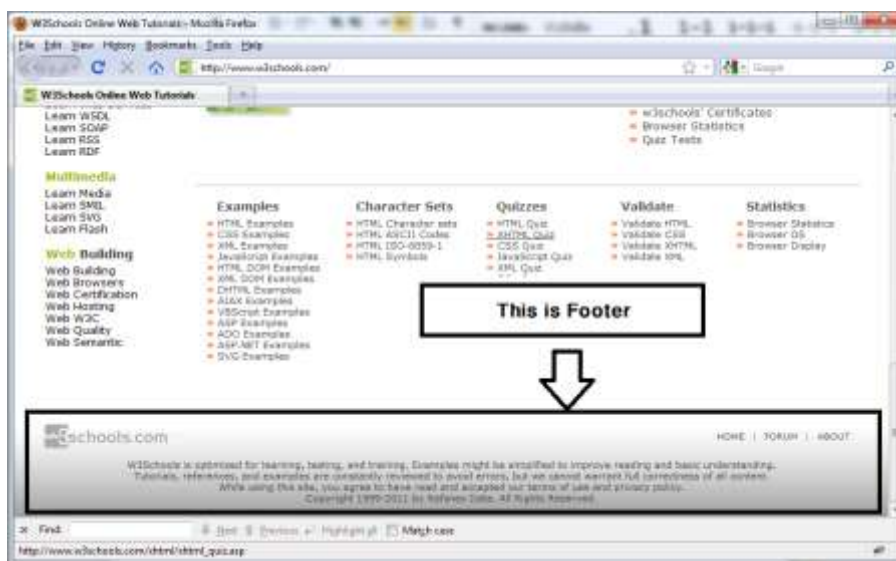
۱-۶-۳ کاربرد تگ <header> و <footer>

قبل از توضیح باید بگویم که تگ <header> با تگ <head> که در بخش قبل معرفی کردیم، تفاوت دارد. تگ <header> و <footer> برای ایجاد سر صفحه و پا صفحه بکار می روند.
اگر به کتابی که در دست خود دارید توجه کنید، می بینید که در بالا و پایین تمام صفحات آن یک کادر و یا یک نوشته یکسانی وجود دارد. به متن، عکس و یا کادری که در بالای تمام صفحات تکرار می شود و معمولاً عنوان فصل یا نام کتاب را در بر دارد، سر صفحه (یا header) و به متن، عکس و یا کادری که در پایین صفحات تکرار می شود و معمولاً شماره صفحه را در بر می گیرد، پا صفحه (یا footer) می گویند.
در صفحات اینترنتی نیز معمولاً از header برای نمایش نام و لوگوی^۱ سایت و از footer نیز برای قرار دادن اطلاعات کلی راجع به سایت مانند جزئیات سایت و متن حق نشر استفاده می شود. تصاویر زیر نمونه ای از header و footer یک سایت را نمایش میدهند.



تصویر ۱۲ – Header یک سایت

^۱ Logo تصویر کوچکی است که شرکت ها و سایت ها به عنوان نماد موسسه خود انتخاب می کنند.



تصویر ۱۳ – footer یک سایت

برای ایجاد header و footer از تگ مربوط به هر یک استفاده می کنیم. برای نمونه تکه کد زیر، چگونگی استفاده از تگ header را html5 نمایش می دهد.

```
<header>
  <img src = "/images/logo.png">
</header>
```

تکه کد زیر نیز نحوه استفاده از تگ footer در html5 را نمایش می دهد.

```
<footer>
  <p>Copyright © 2011 , company name.</p>
</footer>
```

۲-۶-۳ کاربرد تگ <section> و <article>

تگ <section> برای تعیین "بخش ها" در صفحه بکار می رود. و در واقع مطالب صفحه توسط این تگ به بخش های مجزا از یکدیگر بلوک بندی می شوند. تگ <article> در داخل تگ <section> قرار میگیرد و محتوی موجود در تگ <section> را بلوک بندی کرده و هر کدام را به عنوان یک عنصر مجزا در نظر می گیرد. تصویر زیر این موضوع را مشخص می کند.



تصویر ۱۴ – section و article در یک صفحه وب

همانگونه که در تصویر ملاحظه می فرمائید، تگ article فضای داخل تگ section را بلوک بندی می کند. ما در درس های آتی، از سایر تگ ها در داخل تگ article استفاده می کنیم. کد زیر، نمونه ای از تگ section و article را به نمایش می گذارد.

```
<section>
  <article>
    <p>This is first Article</p>
    A sample for article and section tags.
  </article>
  <article>
    <p>This is second Article</p>
    Thanks.
  </article>
</section>
```

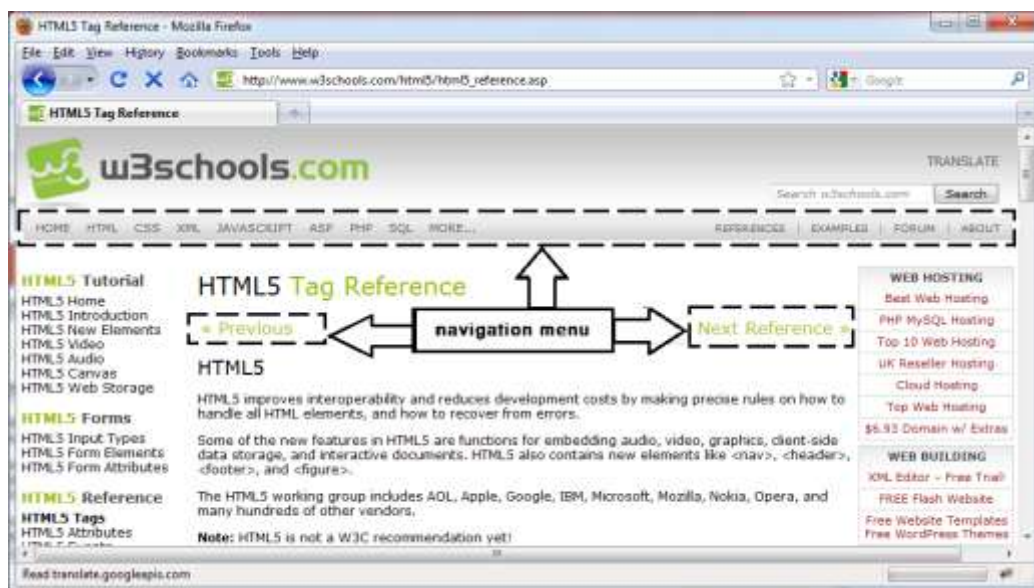
۳-۶-۳ تگ <nav>

این تگ به منظور ایجاد "منوهای ناوبری"^۱ استفاده می گردد. منظور از "منوهای ناوبری"، لینک هایی است که در بالای صفحه و یا در قسمت footer قرار گرفته و کاربر می تواند از طریق این لینک ها، به قسمت های مختلف سایت و یا به صفحات قبلی^۲ و بعدی^۳ دسترسی داشته باشد. در تصاویر زیر منوهای ناوبری که توسط خط چین مشخص شده اند را مشاهده می نمایید.

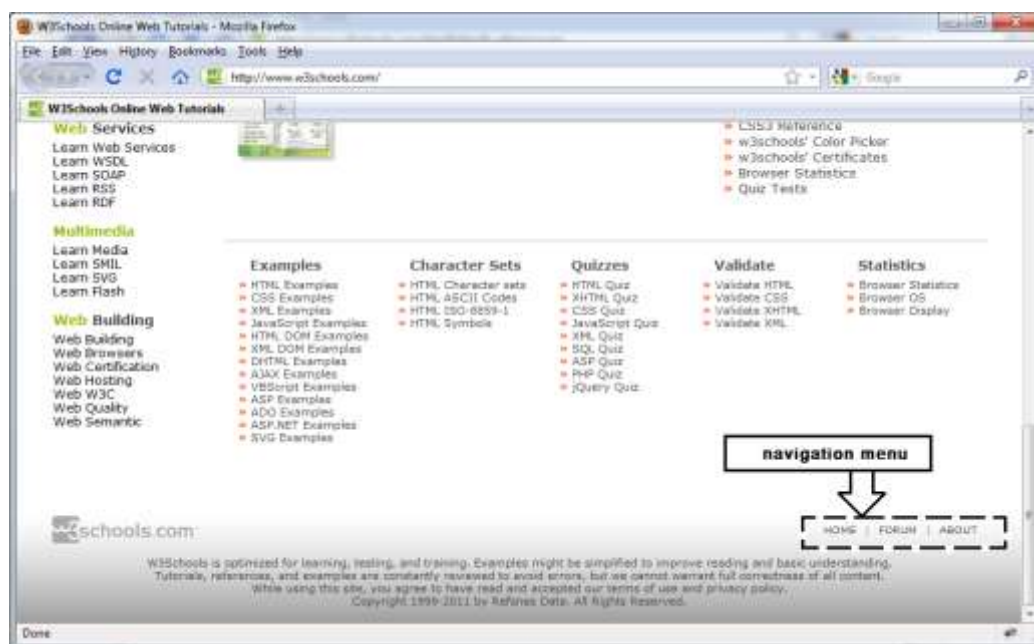
^۱ Navigation menu

^۲ previous

^۳ next



تصویر ۱۵ - منوهای ناوبری بالای صفحه



تصویر ۱۶ - منوهای ناوبری پایین صفحه

مثال زیر کاربرد تگ <nav> را نشان می دهد^۱:

```
<nav>
  <a href="home.html">HOME</a> |
  <a href="forum.html">FORUM</a> |
  <a href="about.html">ABOUT</a>
</nav>
```

^۱ این مثال برای نشان دادن کاربرد تگ <nav> می باشد، تگ <a> در بخش های بعدی بطور کامل آموزش داده خواهد شد.

تگ <aside> ۴-۶-۳

محتوی سایت که در کنار صفحه و جدا از محتوی اصلی سایت تعریف شده و با محتوی قسمت های دیگر صفحه مرتبط می باشند، در تگ <aside> قرار می گیرند. معمولاً لینک های صفحه در این قسمت قرار می گیرند. در تصویر زیر نمونه ای از تگ <aside> را مشاهده می نمایید که با استفاده از خط چین مشخص شده است.



تصویر ۱۷ - تگ <aside>

تکه کد زیر، کاربرد تگ aside را نمایش می دهد:

```
<aside>
  <nav>
    <p><a href="...">learn html5</a></p>
    <p><a href="...">learn css3</a></p>
    <p><a href="...">learn javaScript</a></p>
    <p><a href="...">learn php</a></p>
  </nav>
</aside>
```

۴. قالب بندی متن در html5

تا اینجا با نحوه ساختمان صفحات html و دو تگ اصلی آن، یعنی تگ های <head> و <body>، آشنا شدید. همانگونه که گفتیم تگ <head> اطلاعات مربوط به صفحه وب را تعیین کرده و تگ <body> بدنه اصلی صفحه وب و محتویات آن را دربر دارد. اما اینکه چگونه یک عکس یا یک لینک و یا یک متن در داخل تگ <body> ایجاد می شود را در این قسمت به شما آموزش می دهیم. به عنوان مثال به متن زیر توجه کنید.



تصویر ۱۸- فرمت بندی متن در html5

همانگونه که می بینید، در این تصویر، متون به حالات مختلفی نمایش داده شده است. قسمتی از آن، پر رنگ تر از سایر قسمت هاست. قسمتی دارای "زیر خط"^۱ است، قسمتی با استفاده از یک خط به حالت "حذف شده"^۲ در آمده است و ...

اگر بخواهیم، صفحه ای همانند این صفحه ایجاد کنیم، باید چه مراحل را طی کنیم؟ باید گفت که به سادگی می شود این کار را انجام داد. فقط کافی است برای هر قسمت، متن مورد نظر را در داخل تگ مربوطه قرار دهیم. حالا برای آشنایی با این تگ ها، همین مثال را با کمک هم، طراحی می کنیم. همانگونه که گفتیم، تمام صفحات html در داخل دو تگ <html> قرار می گیرند. پس ابتدا ویرایشگر متن خود را باز کرده و این تگ را داخل آن تایپ کنید. البته قبل از این تگ، تگ <doctype> را نیز تایپ می کنیم تا به مرورگر بگوییم که ما در این صفحه از html نسخه ۵ استفاده می کنیم.

```
<!doctype html>
<html>
</html>
```

^۱ underscore

^۲ delete

تصویر ۱۹ – گام اول در فرمت بندی متون

```
<!doctype html>
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

پس از تایپ این دو تگ، دو تگ اصلی یعنی تگ <head> و تگ <body> را نیز به آن اضافه می کنیم.

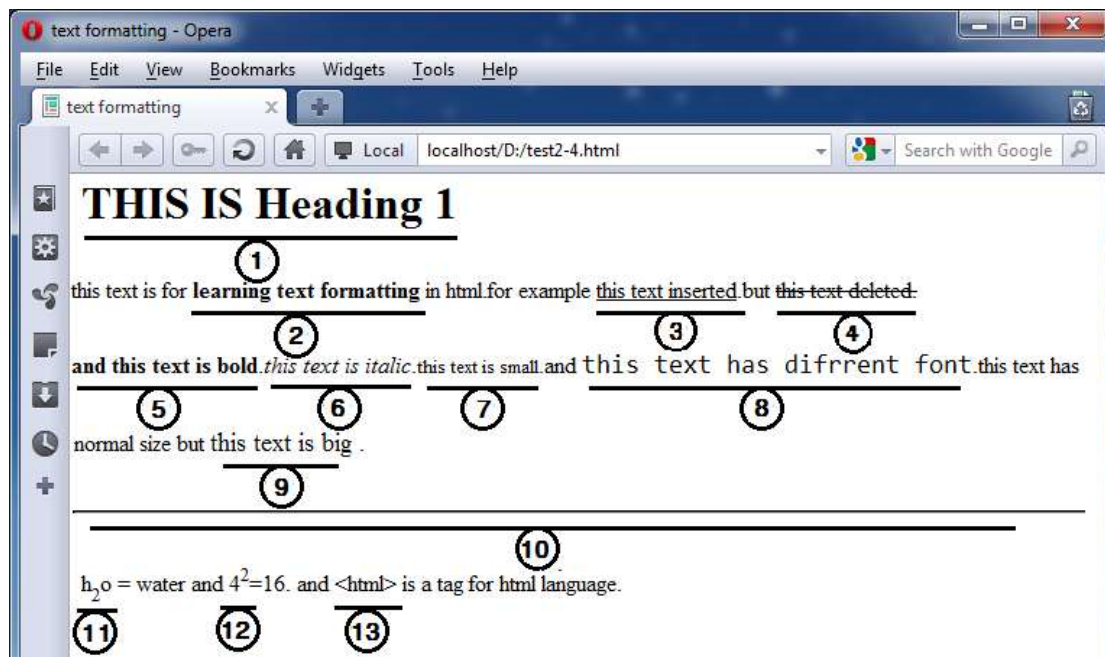
تصویر ۲۰ – گام دوم: افزودن تگ head و body

همانگونه که در تصویر ۱۸ مشاهده می کنید، "عنوان صفحه" (یا همان title) در این مثال، "text formatting" است. پس طبق آموزش هایی که دیدیم، تگ title را در داخل تگ head اضافه می کنیم.

```
<!doctype html>
<html>
  <head>
    <title>text formatting</title>
  </head>
  <body>
  </body>
</html>
```

تصویر ۲۱ – گام سوم: مشخص کردن title

حال به ایجاد مطالب بصری صفحه می پردازیم. همانگونه که قبلا نیز ذکر کردیم، هر مطلبی برای دیده شدن در صفحه وب، باید در داخل تگ <body> قرار بگیرد. در این صفحه، تعدادی نوشته به سبک های مختلف نوشته شده است. با دقت در این تصویر می توانیم این متن را از لحاظ شکل و ترکیب، به ۱۳ قسمت مختلف تفکیک کنیم. شکل زیر این موضوع را به تصویر می کشد.



تصویر ۲۲ - تفکیک متن از لحاظ شکل و ترکیب

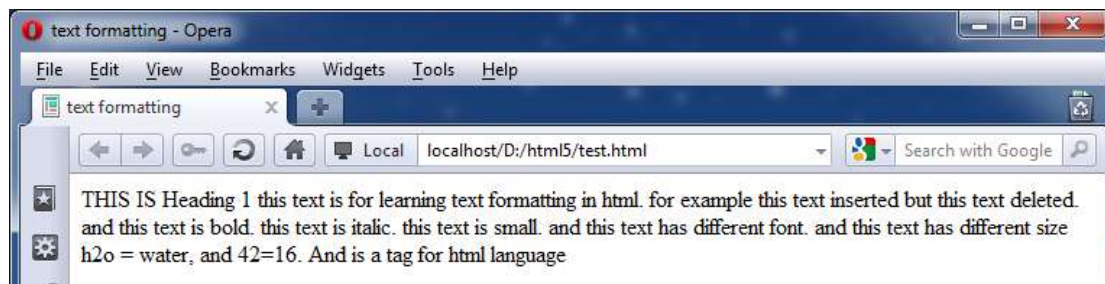
برای ایجاد هر کدام از این قسمت ها، باید، متن مورد نظر را در داخل تگ مربوطه قرار دهیم. پس دست بکار می شویم.

ابتدا متنی که در تصویر ملاحظه می کنید را در داخل تگ body تایپ کنید.

```
<!doctype html>
<html>
  <head>
    <title>text formatting</title>
  </head>
  <body>
    THIS IS Heading 1
    this text is for learning text formatting in html. for example this text inserted but
    this text deleted.
    and this text is bold. this text is italic. this text is small. and this text has different
    font. and this text has different size
    h2o = water, and 42=16. And <html> is a tag for html language
  </body>
</html>
```

تصویر ۲۳ - گام چهارم: نوشتن متن مورد نظر

خب! بهتر است یکبار این کد را با هم اجرا کنیم. چه می بینید؟ در تصویر زیر نتیجه اجرای این کد را مشاهده می کنید:



تصویر ۲۴ - اجرای کد تصویر ۲۳

چه اتفاقی افتاده است؟ چرا کل متن، پشت سر هم نشان داده می شود؟ ما که در تایپ متن از دکمه اینتر و تب (tab) استفاده کرده بودیم پس چرا متن اینطوری نمایش داده میشود؟

باید گفت که این اولین نکته ای است که شما باید در مورد قالب بندی متون در html یاد بگیرید.

مرورگرها همیشه متون را پشت سر هم و ساده نمایش می دهند مگر اینکه ما تغییراتی که در نظر داریم را به آنها بگوییم.

در واقع مرورگرها، کاراکترهای اینتر و تب (tab) را در خروجی نمایش نمی دهند. دلیل اینکار اینست که در بعضی از سیستم ها برای هر کاراکتر تب (tab) چهار فضای خالی نمایش داده میشود و در بعضی از سیستم ها، هشت فضای خالی نمایش داده می شود. همین امر موجب می شود که نتیجه اجرای کدهای html در سیستم های کاربران، همیشه آنطوری نباشد که ما انتظار داریم. و این موضوع در طراحی صفحات وب یک فاجعه محسوب می شود. برای دکمه اینتر هم این مشکلات وجود دارد. برای همین در html برای ایجاد قالب متنی مورد نظر خود، باید از دستورات مربوطه استفاده کنیم.

حال که این نکته را فرا گرفتید به ادامه طراحی می پردازیم.

۴-۱ سرتیترها: h1, h2, h3, h4, h5, h6

تگ های <h1> تا <h6>، سرتیترها را مشخص می کنند. سرتیترها برای تعیین قسمت جدیدی از محتوا بکار می روند. <h1> مهمترین سرتیتر و <h6> کم اهمیت ترین سرتیتر را مشخص می کنند. تکه کد زیر، نمونه ای از سرتیترهای مختلف را نمایش میدهد:

```
<!doctype html>
<html>
  <head>
    <title>h1 to h6 </title>
  </head>
  <body>
    <h1>Learning HTML5</h1>
    Html5 is new version of html. We want to learn html5 to you.html5 has a serious
```



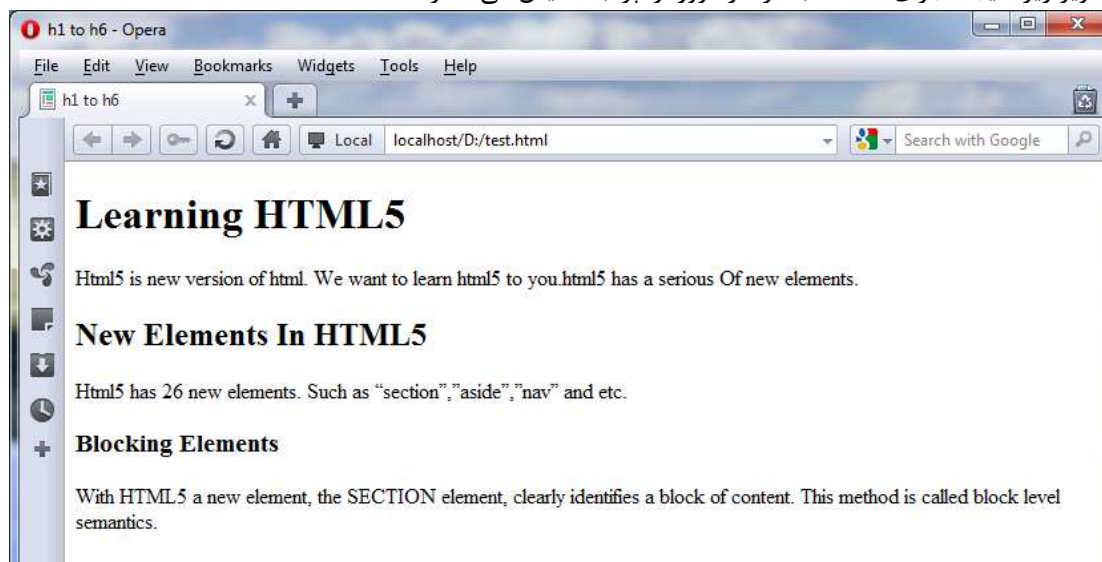
```

Of new elements.
<h2>New Elements In HTML5</h2>
Html5 has 26 new elements. Such as "section","aside","nav" and etc.
<h3>Blocking Elements</h3>
With HTML5 a new element, the SECTION element, clearly
identifies a block of content. This method is called block level
semantics.
</body>
</html>

```

تصویر ۲۵ - سر تیترها

مرورگرها، تیترها را به صورت درشت، و در سایزهای مختلف نمایش می دهند.البته می توان از CSS برای تغییر ظاهر و سایز تیترها استفاده نمود. تصویر زیر نتیجه اجرای تکه کد بالا را در مرورگر اپرا به نمایش می گذارد.



تصویر ۲۶ - سر تیترها

همانگونه که مشاهده می کنید، سر تیتر h1 که مهمترین سر تیتر محسوب می شود، درشت تر از سایر سر تیترها نمایش داده می شود. نکته اصلی که باید در مورد سر تیترها گفت اینست که تگ های h1 تا h6 عناصر سطح بلوک^۱ هستند.

عناصر از لحاظ استقرار در صفحه به دو نوع خطی و سطح بلوک تقسیم می شوند.عناصر خطی می توانند در یک خط و پشت سر هم ظاهر شوند اما عناصر سطح بلوک هر کدام در خط جداگانه ای ظاهر می شوند.به عبارت دیگر، یک عنصر سطح بلوک مثل h1 ، وقتی به یک عبارت اعمال می شود، قبل و بعد از عبارت ، یک خط خالی اضافه می کند تا عنصر مذکور در یک خط جداگانه نمایش داده شود.

خب! حالا که کار با "سر تیتر" ها را فرا گرفتید، به مثال طراحی برمی گردیم.

^۱ Block-Level



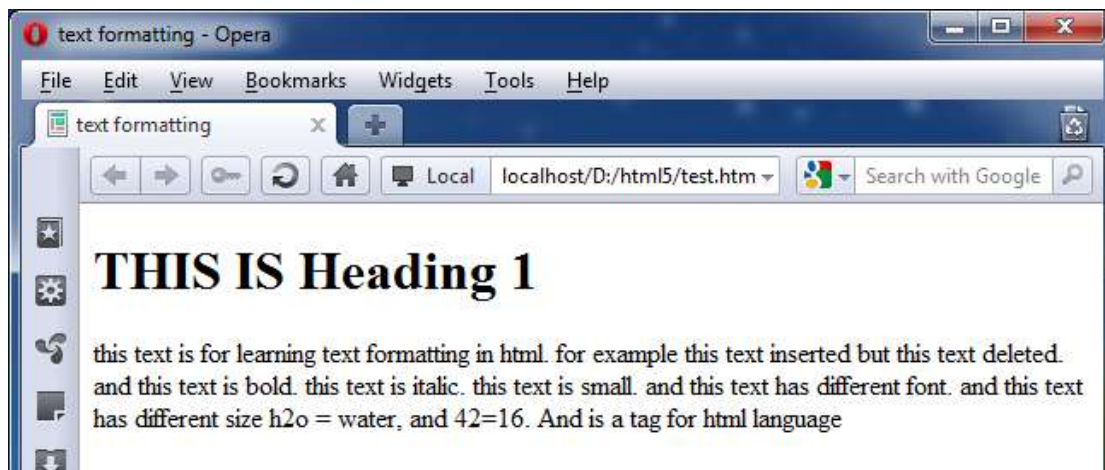
تصویر ۲۷ – کار با سرتیترها

در این مثال، خط اول، یک "سرتیتر" است. و با توجه به اینکه این سرتیتر، در ابتدای صفحه آمده و مهمترین تیتر محسوب می شود، باید از تگ `<h1>` برای آن استفاده نماییم. پس تکه کد تصویر ۲۴ را به صورت زیر تغییر می دهیم:

```
<!doctype html>
<html>
  <head>
    <title>text formatting</title>
  </head>
  <body>
    <h1> THIS IS Heading 1</h1>
    this text is for learning text formatting in html. for example this text inserted
    but this text deleted.
    and this text is bold. this text is italic. this text is small. and this text has different
    font. and this text has different size
    h2o = water, and 4^2=16. And <html> is a tag for html language
  </body>
</html>
```

تصویر ۲۸ – افزودن تگ `<h1>`

با افزودن تگ `<h1>` به خط هفتم و اجرای این کد، تصویر زیر حاصل می شود.



تصویر ۲۹ - نتیجه افزودن تگ <h1>

همانگونه که مشاهده می نمایید، با افزودن تگ <h1>، عبارت مربوطه از لحاظ فونت، سایز و شکل تغییر کرده تا در صفحه متمایز از سایر عبارات نمایش داده شود. و چون h1 یک عنصر سطح بلوک است، عبارتی که توسط این تگ مشخص شده، در یک خط جداگانه نمایش داده می شود و ادامه متن در خط بعدی نشان داده می شود.

۴-۲ قالب بندی عبارات

با توجه به تصویر ۲۲ (صفحه ۳۱)، تا اینجا اولین تغییر را در صفحه ایجاد کردیم. اکنون به ایجاد دومین تغییر می پردازیم. همانگونه که در تصویر مذکور مشاهده می نمایید، عبارت "learning text formatting" نسبت به متن اطرافش کمی پررنگ تر نمایش داده شده است. "پر رنگ کردن"، "تغییر اندازه دادن" و "تغییر دادن فونت"، متداول ترین راهها برای تاکید بر روی یک عبارت و بطور کلی، متمایز کردن آن عبارت در یک متن است. حتما تا بحال کتاب های برنامه نویسی گوناگونی مطالعه کرده اید. قطعاً دقت کرده اید که در اینگونه کتاب ها، فونت مربوط به "کدهای کامپیوتر" با فونت متن کتاب تفاوت دارد. یا اینکه در بعضی از قسمت های متن، عبارات اصلی را پررنگ تر از بقیه نمایش می دهند و یا اینکه اصطلاحات جدید را بصورت مورب به نمایش می گذارند. همه اینکارها برای این است که قسمتی از متن باید متفاوت از سایر قسمت ها دیده شود. در html5، ۹ تگ برای اینکار پیشنهاد میشود که هر کدام از لحاظ منطقی برای مورد مشخصی کاربرد دارد. تگهای ، <i>، ، ، <code>، <samp>، <kbd>، <var> و <mark> تگهایی هستند که در html5 برای این کار در نظر گرفته شده اند.

- **b**
این تگ برای توپر (bold) نشان دادن یک عبارت بکار می رود.
- **i**
این تگ برای مورب (italic) نشان دادن یک عبارت بکار می رود.
- **em**
این تگ باعث ایجاد تاکید بر روی کلمه و یا عبارت می شود. در اکثر مرورگر ها این تاکید با مورب کردن متن مذکور نشان داده می شود. اما ممکن است این تاکید در برخی مرورگرها به شیوه دیگری نمایش داده شود. بعنوان مثال نرم افزار هایی که برای خواندن متن صفحه (برای افراد نابینا) استفاده می شود، این متن را کمی بلند تر از بقیه متن می خواند.

- **Strong**
این تگ نیز باعث ایجاد تاکید بر روی عبارت می شود. البته تاکید بیشتری نسبت به عنصر em به عبارت اضافه میکند. مرورگر ها این تاکید را با توپر (bold) نشان دادن متن نشان می دهند.
- **Code**
همانگونه که گفتیم، در کتابهای آموزش برنامه نویسی کامپیوتر، فونت مربوط به "کدهای کامپیوتری" با فونت مربوط به متن کتاب تفاوت دارد. در سایت های اینترنتی نیز، این موضوع برقرار است. در سایت های آموزش زبانهای برنامه نویسی، "کدهای کامپیوتر" از لحاظ شکل و فونت با سایر قسمتهای متن تفاوت دارد. این کار باعث خوانایی مطلب می شود. در html از تگ <code> برای این کار استفاده می شود. این تگ باعث می شود، متن به صورت کدهای کامپیوتر نمایش داده شود. در ادامه مثالی از این تگ ها تقدیم حضورتان می شود.
- **Samp**
واضح است که هر زبان برنامه نویسی، یک خروجی دارد. اگر با زبان هایی مثل c و pascal برنامه نویسی کرده باشید قطعاً با شکل خروجی این نوع برنامه ها آشنایی دارید. خروجی برنامه های کامپیوتری معمولاً با فونت هم عرض (فونتی که فضای لازم برای تمام کاراکترهای آن یکسان است) نشان داده می شوند. در html5 تگ <samp>، خروجی ساده برنامه را نشان می دهد.
- **Kbd**
با استفاده از این تگ، متن و دستوراتی را طراحی می کنیم که کاربر باید وارد کند. زیرا این متن باید خوانا بوده و در واقع باید شبیه به کاراکترهای صفحه کلید (keyboard) باشد.
- **Var**
برای طراحی متغیر زبان برنامه نویسی بکار می رود و معمولاً به صورت مورب نشان داده می شود.
- **Mark**
این تگ به تازگی به html5 افزوده شده است و برای هایلایت کردن متن استفاده می شود. (البته در حال حاضر تنها مرورگر اینترنت اکسپلورر ۹ از این تگ پشتیبانی می کند).

در جدول زیر، تگهای فوق را به طور خلاصه به همراه کاربردهایشان ملاحظه می نمایید.

| Tag | Description |
|----------|---|
| | توپر (bold) کردن متن |
| <i> | مورب کردن (italic) متن |
| | ایجاد تاکید بر روی کلمه یا عبارت |
| | تعیین یک متن بعنوان متن مهم (تاکید بیشتر نسبت به em) |
| <code> | تعیین یک متن بعنوان کد کامپیوتر |
| <samp> | خروجی ساده برنامه را نشان می دهد |
| <kbd> | متن و دستوراتی را طراحی می کند که کاربر باید وارد کند |
| <var> | برای طراحی متغیر زبان برنامه نویسی بکار می رود |
| <mark> | هایلایت کردن متن |

جدول ۲ – تگهای مربوط به تغییر در شکل متن

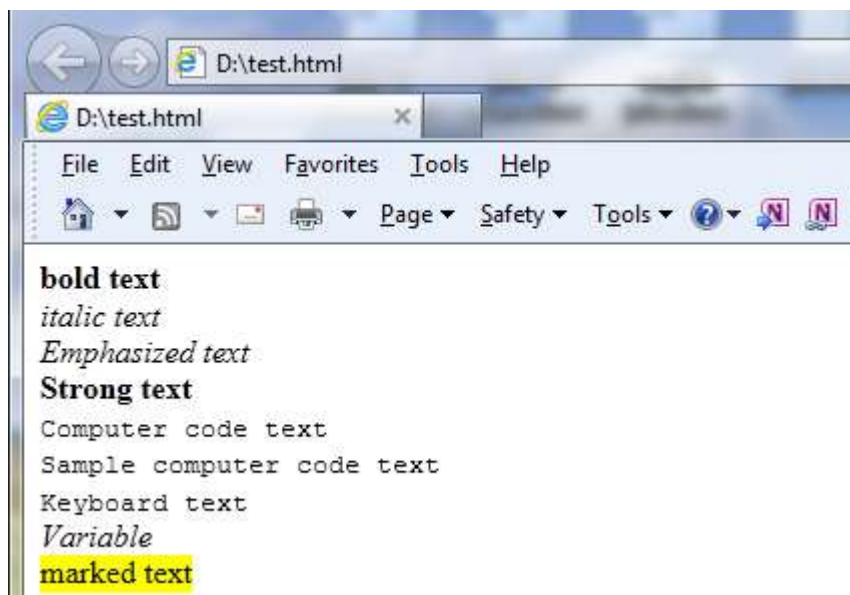
در زیر نمونه ای از اجرای این تگ ها را مشاهده می کنید:

```

<!doctype html>
<html>
  <body>
    <b>bold text</b><br/>
    <i>italic text</i><br/>
    <em>Emphasized text</em><br />
    <strong>Strong text</strong><br />
    <code>Computer code text</code><br />
    <samp>Sample computer code text</samp><br />
    <kbd>Keyboard text</kbd><br />
    <var>Variable</var><br />
    <mark>marked text</mark>
  </body>
</html>

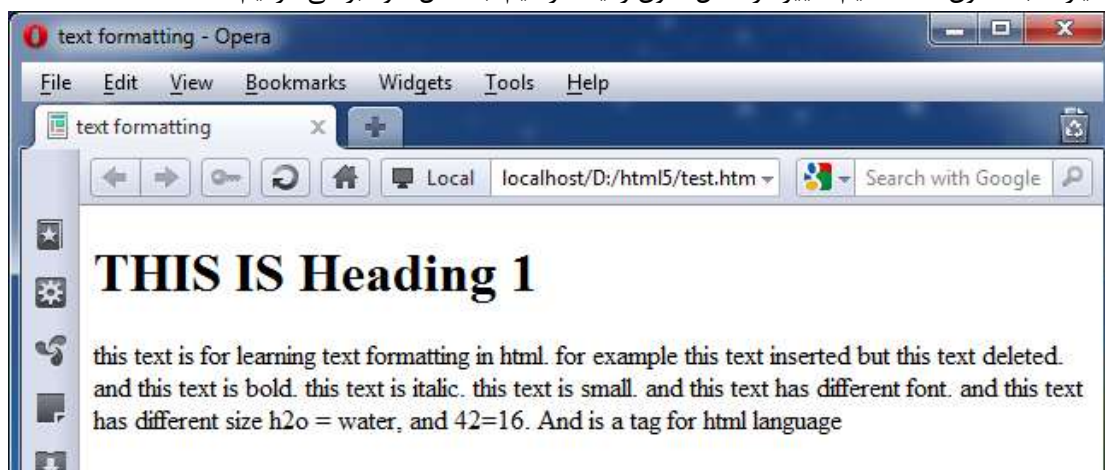
```

تصویر ۳۰ - نمونه کد تغییر در شکل متون



تصویر ۳۱ - نتیجه نمایش کد تغییر در شکل متن

بسیار خب! اکنون که مفاهیم تغییر در شکل متون را یاد گرفتیم، به مثال خود بر می گردیم.



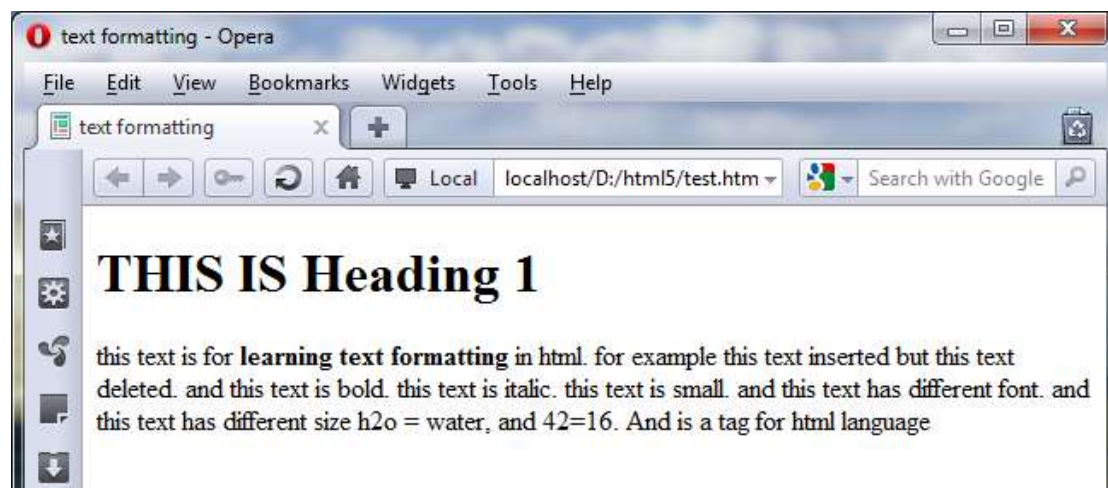
تصویر ۳۲ - ادامه مثال قالب بندی متن

همانگونه که در تصویر فوق مشاهده می کنید (و با توجه به تصویر ۲۲ صفحه ۳۱)، اکنون باید دومین قسمت را طراحی کنیم. یعنی اینکه عبارت "learning text formatting" را به صورت "تاکید دار" تبدیل کنیم. طبق تگ هایی که در صفحات قبل آموزش دادیم، ما برای این کار می توانیم از تگ های **strong**، **bold**، **em** و استفاده کنیم. با توجه به متن، چون این عبارت یک عبارت اصلی است، ترجیحا از تگ **strong** برای تاکید بیشتر استفاده می کنیم. اکنون کد تصویر ۲۸ تصویر ۲۸ صفحه ۳۴ را به صورت زیر تغییر میدهیم.

```
<!doctype html>
<html>
<head>
<title>text formatting</title>
</head>
<body>
<h1> THIS IS Heading 1</h1>
this text is for <strong>learning text formatting</strong> in html. for example this
text inserted. but this text deleted.
and this text is bold. this text is italic. this text is small. and this text has different
font. and this text has different size
h2o = water, and 42=16. And <html> is a tag for html language
</body>
</html>
```

تصویر ۳۳ – افزودن تگ

همانگونه که در تصویر فوق مشاهده می کنید، در خط هشتم، عبارت "learning text formatting" در داخل تگ **** قرار داده شده است. در تصویر زیر، نتیجه حاصل از افزودن این تگ را مشاهده می نمایید.



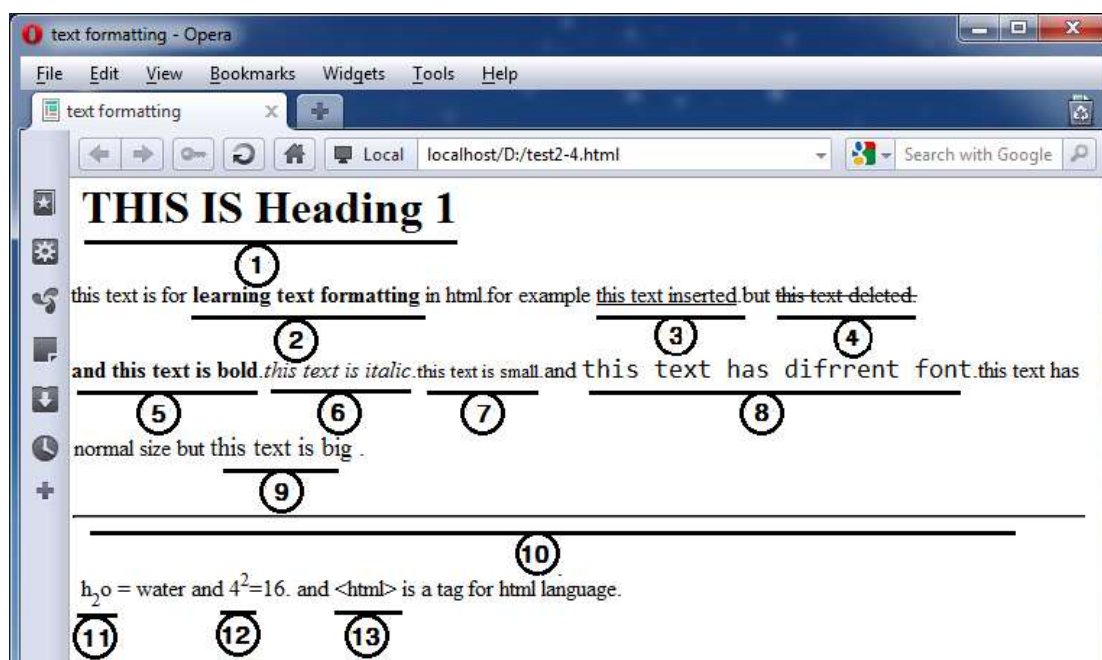
تصویر ۳۴ – نتیجه حاصل از افزودن تگ

همانگونه که در تصویر فوق می بینید، عبارت "learning text formatting" به صورت تاکید دار تبدیل شده است. تبریک می گویم. شما موفق شدید، دومین قسمت طراحی را با موفقیت انجام دهید. حال به سراغ قسمت سوم (طبق قسمت های تصویر ۲۲ صفحه ۳۱) می رویم. همانگونه که در تصویر مذکور مشاهده می کنید، در قسمتی که با شماره ۳ مشخص شده است، متنی را می بینیم با عنوان **this text inserted**، که زیر آن یک خط کشیده شده است و به این صورت مشاهده می شود: **"this text inserted"**.

در صفحات html، گاهی نیاز داریم جمله ای را که به متن افزوده ایم را به صورت زیرخط دار به نمایش بگذاریم. در html5 ما از تگ `<ins>` برای زیر خط دار کردن یک عبارت استفاده می کنیم.

در ادامه این تگ را نیز به کد html خود می افزاییم اما اجازه دهید قبل از اعمال تگ `<ins>` به کد، قسمت چهارم طراحی (که بی ارتباط با تگ `<ins>` نیست) را نیز بیان کنیم.

همان طور که گفتیم، از تگ `<ins>` برای زیرخط دار کردن یک عبارت استفاده می کنیم. اما گاهی هم نیاز داریم تا یک عبارت را بصورت خط خورده نمایش دهیم. همان طور که در تصویر زیر مشاهده میکنید، عبارت شماره ۴ یک عبارت "خط خورده" یا به اصطلاح "حذف شده" است. و عبارت ۳ نیز همان عبارت "زیرخط دار" یا به اصطلاح "اضافه شده" می باشد.



تصویر ۳۵ - تفکیک متن از لحاظ شکل و ترکیب

در html5، برای به حالت "خط خورده" در آوردن یک متن باید از تگ `` استفاده کنیم. که در ادامه نتیجه حاصل از افزودن هر دوی این تگ ها (`<ins>` و ``) را تقدیم حضورتان می کنیم.

یکی از کاربردهای عمومی تگ های `<ins>` و `` درج قیمت های جدید و قدیم، در یک سایت فروش است. بعنوان مثال حتما تا کنون با صفحه ای شبیه به این تصویر برخورد کرده اید:

| 50% off New product |
|---------------------|
| Price : 2500 \$ |
| New price : ۱۲۵۰ \$ |

تصویر ۳۶ - استفاده از تگ `del` و `ins`

در تصویر فوق از تگ های و <ins> استفاده شده است.

- <ins> در html5 از تگ <ins> برای زیر خط دار کردن یک عبارت استفاده می کنیم.
- همانگونه که بیان شد، از تگ برای "خط خورده" کردن متن استفاده می شود.

حال با استفاده از این توضیحات، کد تصویر ۳۳ را به صورت زیر تغییر می دهیم.

```
<!doctype html>
<html>
  <head>
    <title>text formatting</title>
  </head>
  <body>
    <h1> THIS IS Heading 1</h1>
    this text is for <strong>learning text formatting</strong> in html. for example
    <ins>this text inserted.</ins> but <del>this text deleted</del>.
    and this text is bold. this text is italic. this text is small. and this text has different
    font. and this text has different size
    h2o = water, and 42=16. And <html> is a tag for html language
  </body>
</html>
```

تصویر ۳۷ – افزودن تگ های و <ins>

حالا این کد را اجرا می کنیم تا تغییرات ایجاد شده را مشاهده نماییم:



تصویر ۳۸ – نتیجه افزودن تگ های و <ins>

همانگونه که مشاهده می کنید با افزودن تگ <ins>، عبارت "this test inserted" به صورت "زیر خط دار" و عبارت "this text deleted" به صورت "خط خورده" تبدیل شده است.

بسیار خب! تا اینجا قسمت های ۱، ۲، ۳ و ۴ را تکمیل کردیم (به تصویر ۳۵ صفحه ۳۹ دقت کنید). اکنون به تکمیل قسمت های ۵، ۶ و ۷ می پردازیم.

همانگونه که در تصویر مذکور مشاهده می کنید (و از خود متن نیز همین امر استنباط می شود)، قسمت ۵، یک عبارت bold (توپر)، و قسمت ۶، یک عبارت italic (مورب) است.

همانگونه که در صفحات پیشین آموختیم (و در جدول ۲ نیز به اختصار بیان کردیم)، برای توپر (bold) نشان دادن یک عبارت باید از تگ ****، و برای مورب (italic) نشان دادن یک عبارت باید از تگ *<i>* استفاده کنیم. پس برای قسمت های ۵ و ۶ به تگ جدیدی نیاز نداریم. اما برای قسمت ۷، یکی دیگر از تگ های قالب بندی متن در html را معرفی می کنیم.

• **<small>**

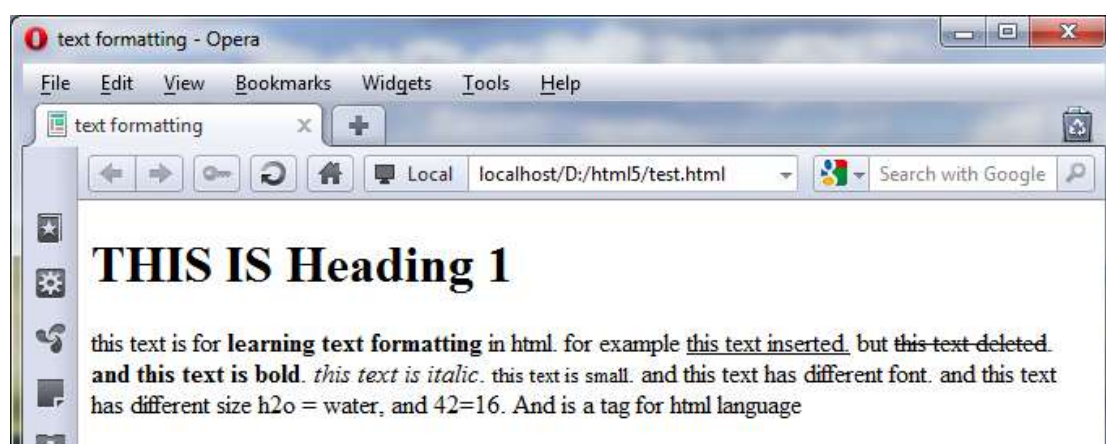
در html از تگ **<small>** برای کوچکتر کردن سایز یک عبارت نسبت به سایز متن اطراف آن استفاده می شود. به عبارت دیگر استفاده از این تگ موجب می شود که یک عبارت نسبت به نوشته های چپ و راستش کوچکتر دیده شود. (در متون انگلیسی از این تگ برای درج توضیحات نیز استفاده می شود).

اکنون که تگ **<small>** را شناختید، این تگ را به همراه دو تگ **** و **<i>** به کد html خود اضافه می کنیم تا قسمت های ۵، ۶ و ۷ را تکمیل کرده باشیم.

```
<!doctype html>
<html>
  <head>
    <title>text formatting</title>
  </head>
  <body>
    <h1> THIS IS Heading 1</h1>
    this text is for <strong>learning text formatting</strong> in html. for example
    <ins>this text inserted.</ins> but <del>this text deleted</del>.
    <b> and this text is bold</b>.<i> this text is italic</i>. <small>this text is
    small</small>. and this text has different
    font. and this text has different size
    h2o = water, and 42=16. And <html> is a tag for html language
  </body>
</html>
```

تصویر ۳۹ - افزودن تگ های ****، **<i>** و **<small>**

همانگونه که در کد فوق مشاهده می کنید، تگ های ****، **<i>** و **<small>** را به کد html اضافه کردیم. حال این کد را در مرورگر اجرا می کنیم تا شاهد تغییرات داده شده باشیم:



تصویر ۴۰ - نتیجه افزودن تگ های ****، **<i>** و **<small>**

همانگونه که می بینید عبارت "and this text is bold" به صورت bold و عبارت "this text is italic" به صورت italic و عبارت "this text is small" کوچکتر از سایر عبارات این متن نمایش داده شده اند.

بسیار خب! تا اینجا ۷ قسمت از ۱۳ قسمت مربوط به مثال قالب بندی متن را تکمیل کرده ایم. اکنون به قسمت های ۸ و ۹ می پردازیم.

قسمت ۸ با این عبارت مشخص شده است: "this text has different font" و با نگاه کردن به آن متوجه می شویم که فونت این عبارت با فونت بقیه متن متفاوت است.

در نسخه های قبلی html، تگی با عنوان برای تغییر دادن "فونت" و "سایز" عبارات وجود داشت. اما با توجه به اینکه این کار از لحاظ طراحی، غیر استاندارد بود، در html5 این تگ دیگر پشتیبانی نمی شود. و کار تغییر فونت و سایز، به CSS واگذار شده است.

قسمت ۹ نیز که با عبارت "this text has different size" مشخص شده است و دارای سایز بزرگتری نسبت به بقیه متن است، در نسخه های قبل html، با استفاده از تگ تغییر سایز می داد که با حذف تگ از html5 این کار نیز به CSS محول شده است.

پس فعلا از قسمت ۸ و ۹ عبور می کنیم تا در فصل های آینده (در بخش مربوط به آموزش CSS) به آنها بپردازیم. اکنون به سراغ ادامه طراحی می رویم.

۴-۳ خط، بالانویس، پایین نویس

به دهمین قسمت طراحی رسیده ایم. اگر به تصویر ۳۵ صفحه ۳۹ دقت کنید می بینید که قسمت ۱۰، یک "خط افقی" است. بله! درست است! در html می توانیم با استفاده تگ <hr>، یک خط افقی رسم کنیم.

برای این کار کافی است در هر نقطه ای از برنامه که دوست دارید، یک تگ منفرد <hr> بگذارید. این تگ در مرورگر بصورت یک خط افقی نمایش داده خواهد شد و باعث می شود که مطالب صفحه، مجزا از یکدیگر نمایش داده شوند.

• <hr>

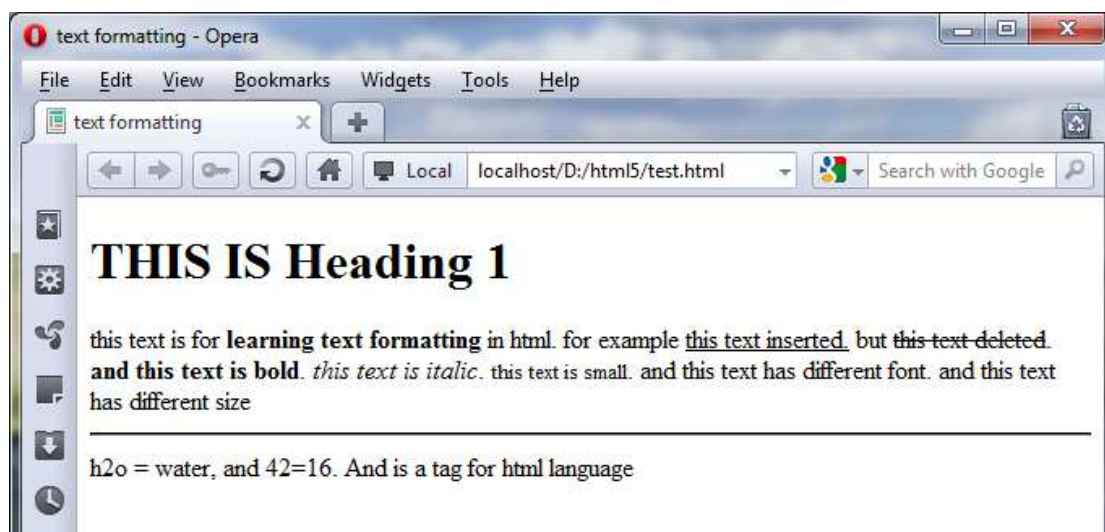
از تگ <hr> برای ترسیم خط افقی در html استفاده می کنیم.

اکنون، این تگ را نیز به کد خود اضافه می کنیم:

```
<!doctype html>
<html>
  <head>
    <title>text formatting</title>
  </head>
  <body>
    <h1> THIS IS Heading 1</h1>
    this text is for <strong>learning text formatting</strong> in html. for example
    <ins>this text inserted.</ins> but <del>this text deleted</del>.
    <b> and this text is bold</b>. <i> this text is italic</i>. <small>this text is
    small</small>. and this text has different
    font. and this text has different size<hr>
    h2o = water, and 42=16. And <html> is a tag for html language
  </body>
</html>
```

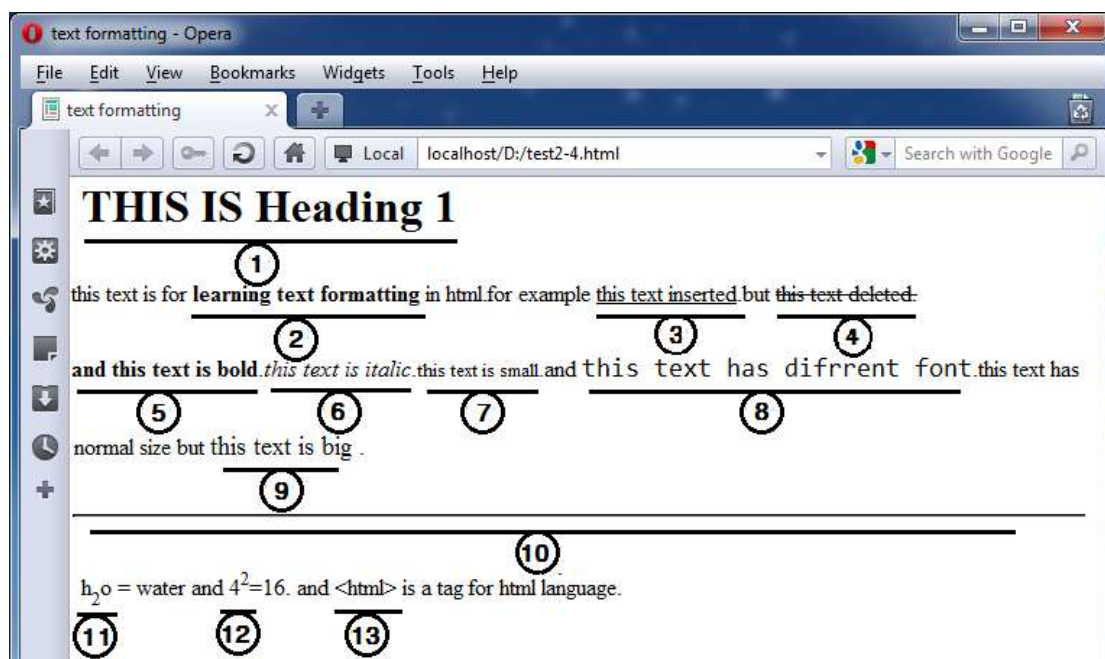
تصویر ۴۱ - افزودن تگ <hr>

با توجه به تصویر فوق مشاهده می کنید، که در انتهای خط دوازدهم، تگ منفرد `<hr>` را درج کردیم، نتیجه افزودن این تگ را می توانید در تصویر زیر مشاهده کنید:



تصویر ۴۲ - نتیجه افزودن تگ `<hr>`

همانگونه که در تصویر مشاهده می کنید، تگ `<hr>` باعث ایجاد یک خط افقی در صفحه شد. بسیار خب! تا کنون ۱۰ قسمت از مثال طراحی را تکمیل کرده ایم. اکنون به قسمت ۱۱ و ۱۲ می پردازیم.



تصویر ۴۳ - تفکیک متن از لحاظ شکل و ترکیب

همانگونه که در تصویر مشاهده می کنید، قسمت های ۱۱ و ۱۲ متن هایی هستند که دارای "زیرنویس" و "بالانویس" می باشند. در `html5` از تگ `<sup>` برای ایجاد "بالانویس" و از تگ `<sub>` برای ایجاد "زیرنویس" استفاده می کنیم.

• `<sub>`

ایجاد متن " زیرنویس دار " مانند X_1, H_2O و ...

- `<sup>`

ایجاد متن " بالانویس دار " مانند $X^2, 4^2$ و

اکنون این دو تگ را نیز به کد خود اضافه می کنیم.

```
<!doctype html>
<html>
<head>
<title>text formatting</title>
</head>
<body>
<h1> THIS IS Heading 1</h1>
this text is for <strong>learning text formatting</strong> in html. for example
<ins>this text inserted.</ins> but <del>this text deleted</del>.
<b> and this text is bold</b>.<i> this text is italic</i>. <small>this text is
small</small>. and this text has different
font. and this text has different size<hr>
h<sub>2</sub>o = water, and 4<sup>2</sup>=16. And <html> is a tag for html
language
</body>
</html>
```

تصویر ۴۴ – افزودن تگ های `<sub>` و `<sup>`

همانگونه که در کد فوق ملاحظه می کنید، تگ های `<sup>` و `<sub>` را در خط سیزدهم درج کردیم. حال به نتیجه اجرای این کد دقت کنید:



تصویر ۴۵ – نتیجه افزودن تگ های `<sub>` و `<sup>`

همانگونه که می بینید، افزودن تگ `<sub>` باعث زیرنویس شدن کاراکتر "۲" در عبارت " H_2O " شده است. همچنین افزودن تگ `<sup>` نیز باعث بالانویس شدن کاراکتر "۲" در عبارت " 4^2 " شده است.

۴-۴ کاراکترهای خاص html5

بسیار خب! به آخرین قسمت تکمیل مثال طراحی می پردازیم. قسمت ۱۳ طراحی، نحوه ایجاد کاراکترهای خاص را آموزش میدهد.

فرض کنید، یک سایت آموزش html5 راه اندازی کرده اید. و می خواهید در مورد تگ `<hr>` توضیح دهید. اما همانگونه که می دانید، اگر این تگ را در هر جای صفحه بنویسید، یک خط افقی در صفحه نمایش داده خواهد شد. پس باید چه کار کرد؟ در واقع ما می خواهیم یک تگ را در صفحه نمایش دهیم بدون اینکه این تگ اجرا شود. نگران نباشید. راه حل ساده است. برای این کار باید از کاراکترهای خاص ^۱ html استفاده کرد.

همانگونه که تا اینجا آموختیم، تگ های html در میان علامت های بزرگتر و کوچکتر (`<` `>`) نوشته می شوند تا از متن معمولی تشخیص داده شوند. زمانی که مرورگر این علامت ها را می بیند، می فهمد که با یک تگ روبرو شده است. اما این موضوع یک مشکل را در بر دارد: اگر بخواهیم در متن خود از علامت های بزرگتر و کوچکتر استفاده کنیم، یا بخواهیم یک تگ را نشان دهیم (بدون اینکه بخواهیم این تگ اجرا شود) با مشکلی روبرو می شویم که راه حل آن، استفاده از "کاراکترهای خاص html" است.

"کاراکترهای خاص" با علامت امپرسند "&" شروع و پس از نوشتن چند حرف، با علامت نقطه ویرگول ";" پایان می یابد.

در جدول زیر تعدادی از کاراکترهای خاص به همراه کد سازنده آنها را مشاهده می کنید:

| نام کاراکتر | شکل ظاهری کاراکتر | کد سازنده کاراکتر |
|-------------------|-------------------|-------------------|
| امپرسند | & | & |
| علامت کوچکتر از | < | < |
| علامت بزرگتر از | > | > |
| علامت تجاری | ™ | ™ |
| علامت کپی رایت | © | © |
| علامت ثبت شده | ® | ® |
| خط فاصله ام | — | — |
| علامت درجه | ° | ° |
| نقطه میانی | · | · |
| علامت یورو | € | € |
| علامت پوند انگلیس | £ | £ |
| علامت تقسیم | ÷ | ÷ |
| علامت رادیکال | √ | √ |

جدول ۳ - کاراکترهای خاص در HTML

بعنوان مثال برای نمایش علامت کوچکتر باید از کاراکتر خاص آن یعنی "<" استفاده کرد. یا برای نمایش علامت "کپی رایت" باید "©" را در کد خود بنویسیم.

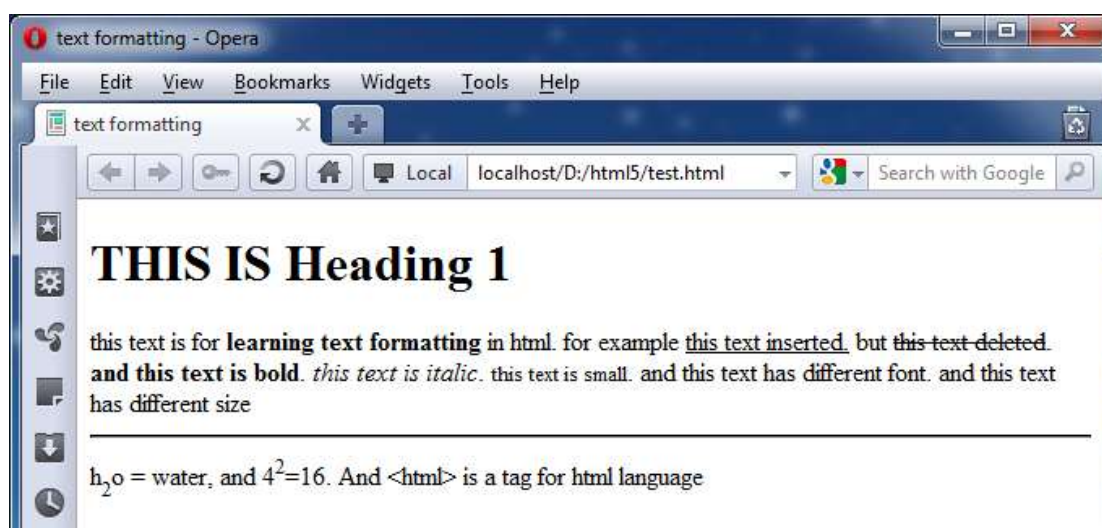
^۱ Special characters

بهتر است خودتان یک بار این کد ها را در یک فایل html نوشته و نتیجه اجرای آن را ببینید.
 خب! حالا به مثال خود بازمی گردیم. همانگونه که دیدیم، برای نمایش عبارت "<html>" در صفحه مرورگر، نباید مستقیماً علامت های کوچکتر و بزرگتر را بنویسیم (زیرا همانگونه که گفتیم این کار باعث می شود که مرورگر این علامت ها را با تگ اشتباه بگیرد) بلکه باید از کاراکترهای "<" و ">" استفاده کنیم. اکنون این دو کاراکتر را به کد مثال اضافه کرده و نتیجه را بررسی می کنیم.

```
<!doctype html>
<html>
  <head>
    <title>text formatting</title>
  </head>
  <body>
    <h1> THIS IS Heading 1</h1>
    this text is for <strong>learning text formatting</strong> in html. for example
    <ins>this text inserted.</ins> but <del>this text deleted</del>.
    <b> and this text is bold</b>.<i> this text is italic</i>. <small>this text is
    small</small>. and this text has different
    font. and this text has different size<hr>
    h<sub>2</sub>o = water, and 4<sup>2</sup>=16. And &lt;html&gt; is a tag for
    html language
  </body>
</html>
```

تصویر ۴۶ - افزودن کاراکترهای خاص < و >

همانگونه که در خط سیزدهم مشاهده می کنید، بجای استفاده از علامت های کوچکتر و بزرگتر از کاراکترهای خاص "<" و ">" استفاده کرده ایم. نتیجه این تغییر را در تصویر زیر مشاهده می کنید:



تصویر ۴۷ - نتیجه افزودن کاراکترهای خاص < و >

بسیار خب! تبریک می گویم. شما موفق شدید، مثال را با موفقیت طراحی کنید. بهتر است تصویر فوق را با تصویر ابتدایی (تصویر ۱۸ صفحه ۲۹) مقایسه کنید. می بینید که تمام قسمت ها را بدرستی طراحی کرده ایم (البته به جز دو مورد مربوط به فونت و سایز که گفتیم باید با CSS طراحی شود).

۵-۴ <p>، <wbr> و <pre>

تا اینجا، شما را با اکثر تگ های مربوط به قالب بندی متن در HTML5 آشنا کرده ایم. در این قسمت، سه تگ دیگر را تقدیم حضورتان می کنیم.

این سه تگ عبارتند از <p>، <pre> و <wbr> که در ادامه، توضیح مربوط به هر یک را مشاهده می نمایید.

• <p>

تگ <p> برای ایجاد پاراگراف بکار می رود.

```
<p>
The p element automatically creates some space before and after itself. The space is
automatically applied by the browser, or you can specify it in a style sheet.
</p>
```

تصویر ۴۸ - تگ <p>

• <wbr>

این تگ برای تعیین مکان شکستگی خط در یک عبارت بکار می رود. مرورگرها همواره تمامیت یک کلمه را حفظ می کنند. یعنی اینکه هیچگاه یک کلمه را نمی شکنند و تمام حروف آن کلمه را در یک خط نمایش می دهند. و اگر احيانا این کلمه در انتهای جمله ظاهر شد و جا برای نوشتن آن کلمه کم بود، کل کلمه را به خط بعد برده و فاصله ایجاد شده در خط قبلی را بین فواصل کلمات آن خط تقسیم می کند. این کار باعث می شود که در بعضی مواقع ظاهر خطوط بهم ریخته شود. بعنوان مثال این کد را در نظر بگیرید:

```
<!DOCTYPE HTML>
<html>
<body>
  <p style="text-align:justify;direction:ltr;">
    To learn AJAX, you must be familiar with the XMLHttpRequest Object. Ajax is a
    way to connect to server without page refreshing. this is very good for users. websites
    with Ajax are very fast and attractive.
  </p>
</body>
</html>
```

تصویر ۴۹ - عدم شکسته شدن کلمات با کم شدن عرض مرورگر

این کد را توسط مرورگر اجرا کنید. سپس با استفاده از ماوس، عرض مرورگر را کم و زیاد کنید. در همین حال به کلمه "XMLHttpRequest" دقت کنید. همانگونه که می بینید، تمام حروف این کلمه همواره در یک خط ظاهر می شوند. به عبارت دیگر، تمام حروف این کلمه، یا در خط بالایی نمایش داده می شود و یا در خط پایینی و هیچگاه این کلمه شکسته نمی شود.

این از خصوصیات مرورگرهاست که هیچگاه یک کلمه را نمی شکنند. بلکه جملات را فقط از حد فاصل بین کلمات می شکنند زیرا ممکن است با شکسته شدن یک کلمه، معنی آن تغییر کند که این امر قطعا کار اشتباهی خواهد بود. اما گاهی مواقع، شکسته شدن یک کلمه، مشکلی در معنی آن بوجود نمی آورد.

بعنوان مثال اگر کلمه “XMLHttpRequest” از بین “XML” و “HttpRequest” شکسته شود، مشکلی خاصی پیش نمی آید. در اینگونه مواقع بهتر است این مطلب را برای مرورگر بازگو کنیم. که ما این کار را با استفاده از تگ `<wbr>` انجام میدهیم. یعنی اینکه ما با استفاده از این تگ به مرورگر می گوییم که در صورتی که نیاز باشد، می تواند از نقطه ای که این تگ در آن قرار دارد، کلمه را بشکند. در کد زیر این موضوع را مشاهده می کنید:

```
<!DOCTYPE HTML>
<html>
<body>
  <p style="text-align:justify;direction:ltr;">
    To learn AJAX, you must be familiar with the XML<wbr>Http<wbr>Request
    Object. Ajax is a way to connect to server without page refreshing.this is very good
    for users.websites with Ajax are very fast and attractive.
  </p>
</body>
</html>
```

تصویر ۵۰ – استفاده از تگ `<wbr>`

این کد را توسط مرورگر فایرفاکس^۱ اجرا کرده و نتیجه حاصل را با کم و زیاد کردن عرض مرورگر مشاهده کنید. همانگونه که شاهد هستید، اینبار مرورگر قادر است کلمه “XMLHttpRequest” را از نقاطی که با تگ `<wbr>` مشخص کرده ایم بشکند.

• `<pre>`

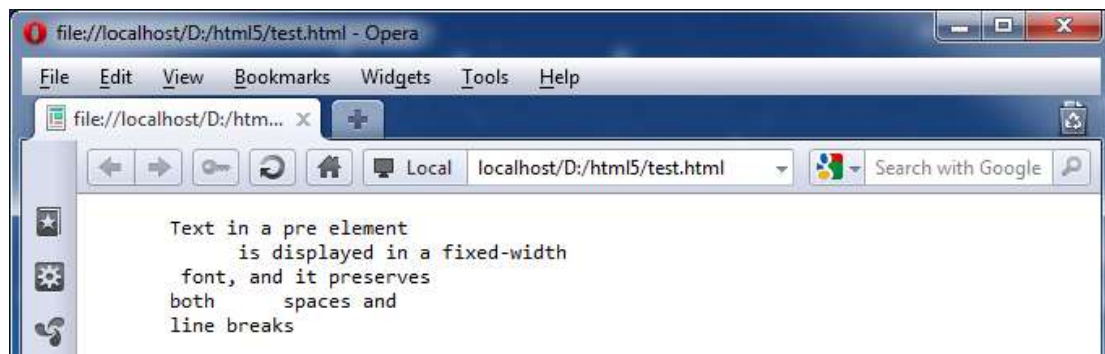
همان گونه که در صفحات قبل آموختید فضاهای خالی در سند html در هنگام ترجمه سند توسط مرورگر نادیده گرفته شده و تمام آنها به عنوان یک فضای خالی در نظر گرفته می شوند (همان مشکلی که در زیر تصویر ۲۴ صفحه ۳۲ به تشریح آن پرداختیم). اما راهی برای حل این مشکل وجود دارد که آنهم استفاده از تگ `<pre>` می باشد. تگ `<pre>` باعث میشود که متون و عبارات به همان حالتی که در کد می نویسید - با حفظ فاصله ها و شکستگی خطوط - در خروجی نمایش داده شوند. این تگ برای نوشتن اشعار و یا نوشتن کدهای کامپیوتری در صفحه کاربرد دارد. کد زیر نمونه استفاده از این تگ را نمایش می دهد:

```
<!DOCTYPE HTML>
<html>
<body>
<pre>
  Text in a pre element
  is displayed in a fixed-width
  font, and it preserves
  both spaces and
  line breaks
</pre>
</body>
</html>
```

تصویر ۵۱ – استفاده از تگ `<pre>`

در تصویر زیر نتیجه استفاده از این تگ را مشاهده می کنید:

^۱ زیرا سایر مرورگرها از این تگ پشتیبانی نمی کنند



تصویر ۵۲ - نتیجه استفاده از تگ <pre>

۴-۶ لیست ها

قسمت دیگری که برای کار با متون در html از آن استفاده می کنیم، "لیست ها" هستند. بطور کل، سه نوع لیست در html وجود دارد که عبارتند از: لیست های نامرتب، لیست های مرتب و لیست های تعریف.

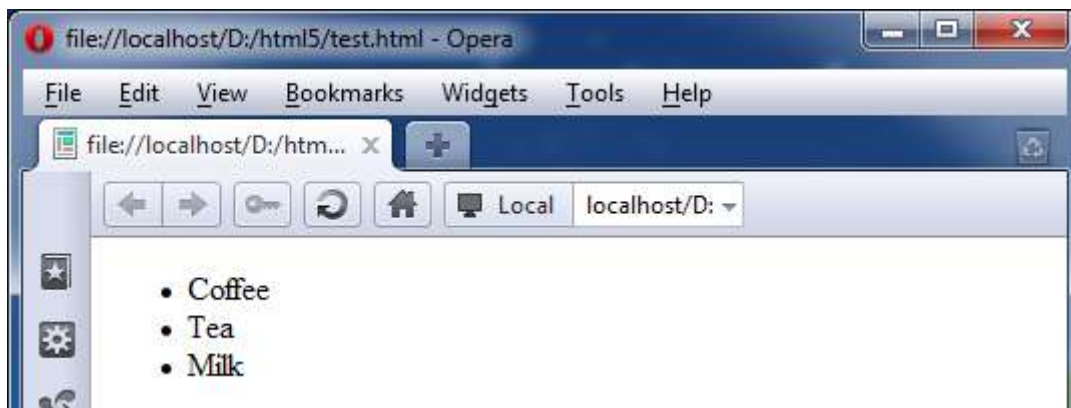
۴-۶-۱ لیست نامرتب

لیست نامرتب توسط تگ ساخته می شود و برای لیست کردن مواردی است که ترتیب در آنها اهمیت ندارد. هر گزینه لیست توسط عنصر ساخته می شود. تکه کد زیر، نحوه ساخت یک لیست نامرتب را به شما نمایش می دهد:

```
<!DOCTYPE HTML>
<html>
<body>
  <ul>
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
  </ul>
</body>
</html>
```

تصویر ۵۳ - لیست نامرتب

نتیجه اجرای کد فوق را در تصویر زیر مشاهده کنید:



تصویر ۵۴ - لیست نامرتب

همانگونه که در تصویر می بینید، به طور پیش فرض لیست های نامرتب در مرورگرها، کمی تو رفته نشان داده می شوند و علامت های دایره توپر در ابتدای هر مورد نشان داده می شود. البته در فصل های آینده خواهید آموخت که چگونه با استفاده از CSS می توانید این دایره های توپر را عوض کرده و بجای آن از کاراکتر های دیگر و حتی عکس استفاده کنید

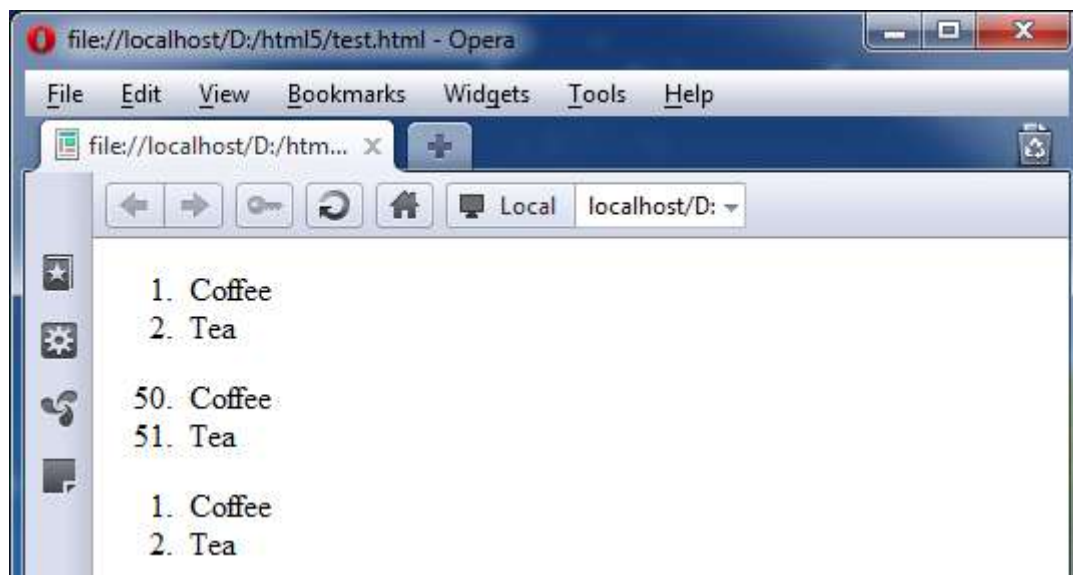
۴-۶-۲ لیست مرتب

لیست مرتب با عنصر `` نمایش داده می شود و برای لیست کردن مواردی بکار می رود که ترتیب در آنها اهمیت دارد. این ترتیب با استفاده از شماره گذاری مشخص می شود. البته می توانید این شماره گذاری را از عدد مشخصی شروع کنید. برای این کار از صفت `"start"` استفاده می کنیم. صفت دیگر این تگ، `"reversed"` است. در صورتی که این صفت در تگ نوشته شود، شمارش لیست از آخر به اول و به صورت نزولی خواهد بود. البته این صفت هنوز توسط هیچ مرورگری پشتیبانی نمی شود. همانند تگ `` در تگ `` نیز از `` برای ایجاد گزینه ها استفاده می کنیم. تکه کد زیر این موارد را نمایش می دهد:

```
<!DOCTYPE HTML>
<html>
<body>
  <ol>
    <li>Coffee</li>
    <li>Tea</li>
  </ol>
  <ol start="50">
    <li>Coffee</li>
    <li>Tea</li>
  </ol>
  <ol reversed="reversed">
    <li>Coffee</li>
    <li>Tea</li>
  </ol>
</body>
</html>
```

تصویر ۵۵ - لیست مرتب

تصویر زیر در مرورگر اپرا نمایش داده شده است. همانگونه که در این تصویر می بینید، صفت reserved توسط مرورگر اپرا پشتیبانی نمی شود، در نتیجه این لیست هم بصورت صعودی شمارش شده است:



تصویر ۵۶ - لیست های مرتب

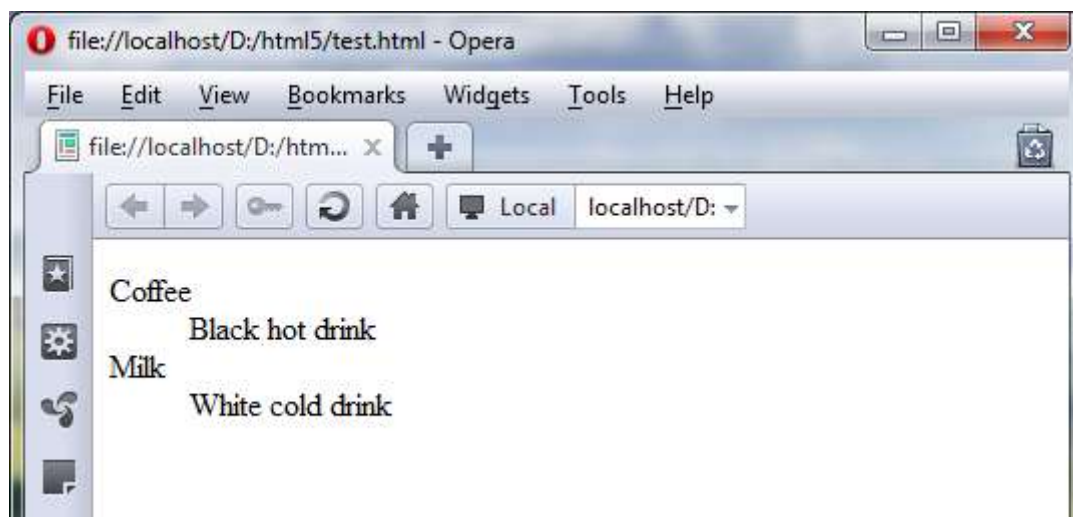
۳-۶-۴ لیست تعریف

لیست تعریف با تگ <dl> مشخص می شود. در لیست تعریف برای گزینه هایی که نمایش می دهیم، تعریفی را نیز می آوریم. در نتیجه در کدنویسی - بر خلاف تگهای و که گزینه ها تنها با استفاده از مشخص می شدند - از <dt> برای مشخص کردن گزینه و از <dd> برای مشخص کردن تعریف گزینه استفاده می کنیم. در زیر یک لیست تعریف را مشاهده میکنید:

```
<!DOCTYPE HTML>
<html>
<body>
  <dl>
    <dt>Coffee</dt>
    <dd>Black hot drink</dd>
    <dt>Milk</dt>
    <dd>White cold drink</dd>
  </dl>
</body>
</html>
```

تصویر ۵۷ - لیست تعریف

در تصویر زیر نتیجه اجرای این تکه کد را ملاحظه می کنید:



تصویر ۵۸ - لیست تعریف

همانگونه که در تصویر فوق می بینید، برای هر یک از دو گزینه "Coffee" و "Milk"، عبارتی به عنوان "تعریف" آمده است.

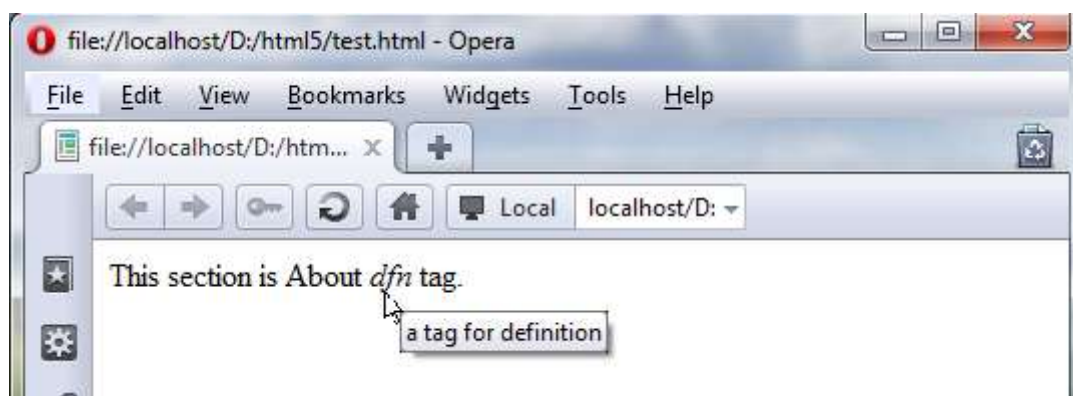
۷-۴ <dfn> و <abbr>

بعضی مواقع مطلبی در سایت می نویسیم که در این مطلب اصطلاحات جدیدی وجود دارد که این اصطلاحات نیاز به توضیح دارند. عنصر dfn برای نمایش تعریف کوتاهی از یک عبارت بکار می رود. برای این کار باید از صفت title موجود در این تگ، همانند تکه کد زیر استفاده کنیم:

```
<!DOCTYPE HTML>
<html>
<body>
This section is About <dfn title="a tag for definition">dfn</dfn> tag.
</body>
</html>
```

تصویر ۵۹ - کاربرد تگ dfn

تصویر زیر نتیجه اجرای این کد را به نمایش می گذارد.



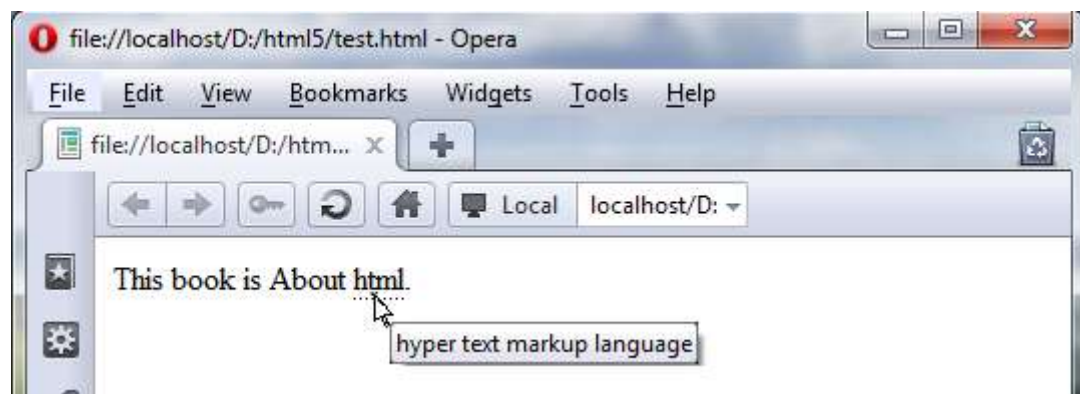
تصویر ۶۰ - کاربرد تگ dfn

مرورگرها معمولاً محتوی تگ dfn را بصورت مورب نمایش می دهند و همانگونه که در تصویر می بینید، با قرار گرفتن ماوس بر روی عبارتی که با این تگ مشخص شده است، یک تولتیپ^۱ ظاهر شده که توضیحات نوشته شده در صفت title این تگ را نمایش می دهد.

عنصر abbr^۲ نیز همانند عنصر dfn است. از این عنصر در مواقعی استفاده می کنیم که بخواهیم یک کلمه مخفف را مشخص کنیم. مثلاً html مخفف hyper text markup language است. تکه کد زیر نحوه استفاده از این تگ را نمایش می دهد.

```
<!DOCTYPE HTML>
<html>
<body>
This book is About <abbr title= "hyper text markup language">html</abbr>.
</body>
</html>
```

تصویر ۶۱ – تگ abbr



تصویر ۶۲ – اجرای تگ abbr

همانگونه که می بینید مرورگر محتوی این تگ را به صورت ساده با یک خط نقطه چین در زیرش نمایش می دهد. با قرار گرفتن ماوس بر روی این عبارت، یک تولتیپ حاوی متن داخل صفت title این عنصر نمایش داده می شود.

بسیار خب! شما اکنون تمام تگ های مربوط به قالب بندی متن را فرا گرفته اید. در فصل های آینده به کار با جداول و تصاویر می پردازیم.

^۱ Tooltip – کادر کوچکی که در کنار ماوس ظاهر می شود و معمولاً توضیحاتی در مورد چیزی که ماوس بر روی آن قرار گرفته ارائه می کند

^۲ Abbreviation به معنی مخفف

۵. کار با جدول ها

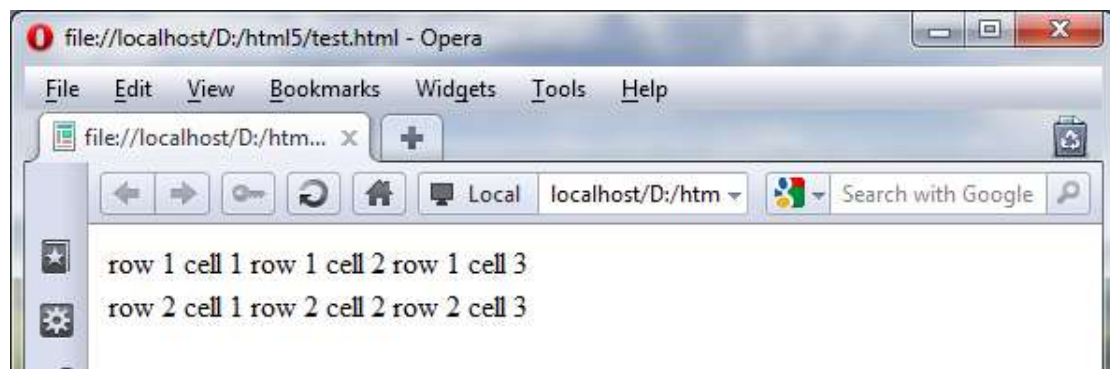
۱-۵ <table> ، <tr> و <td>

جداول یکی از زیباترین بخش های html است. در این فصل نحوه ایجاد جدول های گوناگون و تگهای مربوط به آنها را فرا می گیرید.

همانطور که میدانید جدول عنصری است برای نظم بخشیدن به اطلاعات موجود در صفحه . ما در html با استفاده از تگ <table> می توانیم یک جدول ایجاد کنیم همانطور که می دانید هر جدول حاوی تعدادی سطر و ستون است. در زبان html برای ایجاد سطر از تگ <tr> و برای ایجاد سلول در هر ردیف از تگ <td> استفاده می کنیم. تکه کد زیر نحوه ایجاد یک جدول ساده را به شما نشان می دهد:

```
<!DOCTYPE HTML>
<html>
<body>
  <table>
    <tr>
      <td>row 1 cell 1</td>
      <td>row 1 cell 2</td>
      <td>row 1 cell 3</td>
    </tr>
    <tr>
      <td>row 2 cell 1</td>
      <td>row 2 cell 2</td>
      <td>row 2 cell 3</td>
    </tr>
  </table>
</body>
</html>
```

تصویر ۶۳ - نمونه ساده جدول html



تصویر ۶۴ - اجرای کد ایجاد جدول

در تصویر بالا جدولی که ساخته ایم را مشاهده می کنید. خب! حتما می گوئید این دیگر چه جور جدولی است. بله من هم با شما موافقم. این جدول یک مشکل دارد و آن هم اینست که کادر جدول نشان داده نشده است و جدول بی مفهوم است.

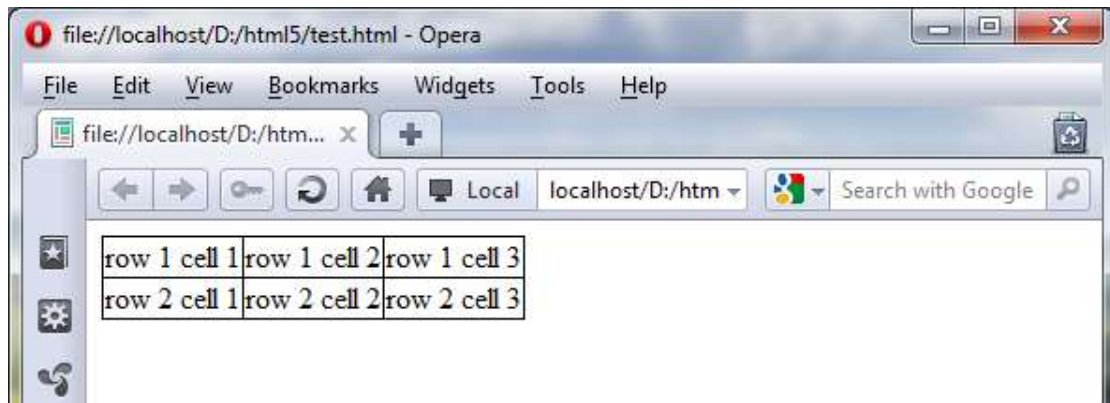
باید گفت که در نسخه های پیشین html تگ <table> حاوی صفات گوناگون بود که از آن جمله همین مورد نمایش کادر جدول بود اما به دلیل اینکه کار ظواهر عناصر در html5 به CSS واگذار شده است، دیگر در تگ <table> تنها یک صفت داریم و آنهم صفت summary است که کار آن مشخص کردن خلاصه ای از جدول برای مرورگرهایی است که محتوی صفحات را بصورت صوتی برای کاربران می خوانند. اما چون ما قصد داریم CSS را جداگانه به شما آموزش دهیم تا مطالب آن را بهتر بیاموزید، در اینجا فقط یک تکه کد CSS را به کد قبلی می افزاییم که البته توضیحات مربوط به آن را به فصل CSS ماکول می کنیم. تکه کد مذکور را در زیر مشاهده می کنید:

```
<style type="text/css">
  table{border-collapse:collapse;}
  table, td, th{border:1px solid black;}
</style>
```

شما تا آن زمان کافی است این تکه کد را در قسمت <head> فایل خود قرار دهید. به صورتی که در کد زیر مشاهده می کنید:

```
<!DOCTYPE HTML>
<html>
<head>
  <style type="text/css">
    table{border-collapse:collapse;}
    table, td, th{border:1px solid black;}
  </style>
</head>
<body>
  <table>
    <tr>
      <td>row 1 cell 1</td>
      <td>row 1 cell 2</td>
      <td>row 1 cell 3</td>
    </tr>
    <tr>
      <td>row 2 cell 1</td>
      <td>row 2 cell 2</td>
      <td>row 2 cell 3</td>
    </tr>
  </table>
</body>
</html>
```

تصویر ۶۵ – اصلاح کد table



تصویر ۶۶ – اصلاح کد table

همانطور که می بینید با افزودن کد مذکور، جدول شکل واقعی خودش را پیدا کرد. به یاد داشته باشید که از هم اکنون تا پایان فصل این تکه کد را به فایل خود اضافه کنید.

۲-۵ <th>

همانگونه که می دانید در هر جدول همواره بالاترین سلول بعنوان سربرگ هر ستون مشخص می شود که این سربرگ ها مشخص می کنند که هر ستون حاوی چه اطلاعاتی است. تگ <th> یک سلول سربرگ^۱ را در یک جدول html ایجاد می کند. بطور کلی در یک جدول دو نوع سلول وجود دارد:

- سلول سربرگ : که با استفاده از تگ <th> ایجاد میشود.
- سلول استاندارد : که همانطور که دیدیم با استفاده از تگ <td> ایجاد می شود.

متون در یک سلول سربرگ معمولاً بصورت توپُر و در مرکز سلول نمایش داده می شوند در حالیکه در یک سلول استاندارد بصورت معمولی و در سمت چپ سلول نمایش داده می شوند. در زیر نحوه استفاده از این تگ را مشاهده میکنید:

```
<!DOCTYPE HTML>
<html>
<head>
  <style type="text/css">
    table{border-collapse:collapse;}
    table, td, th{border:1px solid black;}
  </style>
</head>
<body>
  <table>
    <tr>
      <th>student name</th>
      <th>Mathematics</th>
      <th>physics</th>
    </tr>
    <tr>
```

^۱ header

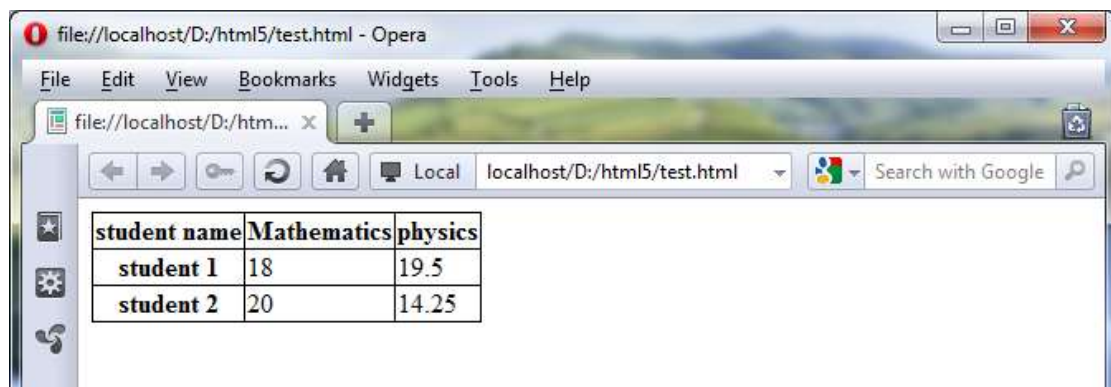
```

<th>student 1</th>
<td>18</td>
<td>19.5</td>
</tr>
<tr>
<th>student 2</th>
<td>20</td>
<td>14.25</td>
</tr>
</table>
</body>
</html>

```

تصویر ۶۷ – استفاده از تگ <th>

همانگونه که در کد فوق مشاهده می کنید، برای تولید سلول های سربرگ از تگ <th> و برای تولید سلول های استاندارد از تگ <td> استفاده کرده ایم. نتیجه حاصل از اجرای این کد را در تصویر زیر مشاهده می نمایید:



تصویر ۶۸ – استفاده از تگ <th>

با توجه به تصویر فوق، در ردیف اول هر سه سلول جزء سلول های سربرگ (<th>) هستند اما در ردیف دوم و سوم تنها سلول اول، سلول سربرگ بوده و دو سلول دیگر این ردیف ها، سلول استاندارد (<td>) هستند.

۳-۵ <caption>

تگ <caption> یک "عنوان" برای جدول در نظر می گیرد. این عنوان معمولاً در بالای جدول نشان داده می شود. این تگ باید بلافاصله پس از تگ آغازین <table> نوشته شود و لازم به ذکر است که هر جدول تنها یک caption می تواند داشته باشد. کد زیر نحوه استفاده از این تگ را به شما نشان می دهد.

```

<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
table{border-collapse:collapse;}
table, td, th{border:1px solid black;}
</style>
</head>
<body>
<table>

```

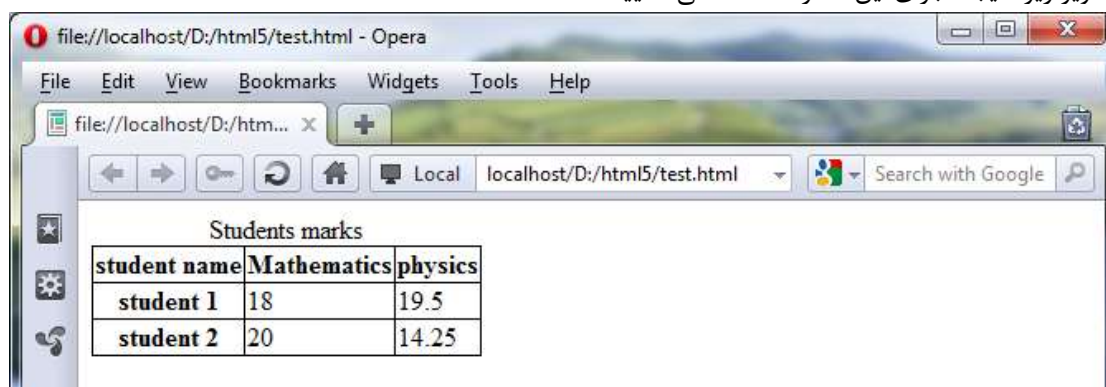
```

<caption>Students marks</caption>
<tr>
  <th>student name</th>
  <th>Mathematics</th>
  <th>physics</th>
</tr>
<tr>
  <th>student 1</th>
  <td>18</td>
  <td>19.5</td>
</tr>
<tr>
  <th>student 2</th>
  <td>20</td>
  <td>14.25</td>
</tr>
</table>
</body>
</html>

```

تصویر ۶۹ – استفاده از تگ caption

همانطور که در کد فوق می بینید، تگ <caption> بلافاصله پس از تگ آغازین <table> نوشته شده است. در تصویر زیر نتیجه اجرای این کد را مشاهده می نمایید:



تصویر ۷۰ – استفاده از تگ <caption>

همانگونه که در تصویر فوق مشاهده می کنید، عبارت “students marks” بعنوان caption این جدول در قسمت بالا و در وسط جدول نمایش داده شده است.

۵-۴ <thead> ، <tbody> و <tfoot>

- تگ های <tbody>، <thead> و <tfoot> برای گروه بندی ردیف ها و خانه های جدول بکار می روند. این گروه بندی باعث می شود که کار زیباسازی و سبک دهی توسط CSS راحت و آسان گردد.
- تگ <tbody> ردیف هایی از جدول را به عنوان بدنه اصلی جدول، در یک گروه قرار می دهد. این تگ باید حتما با تگ های <thead> و <tfoot> بکار گرفته شود.
 - تگ <thead> معمولا ردیف اول از جدول را به عنوان سربرگ جدول مشخص می کند.

به یاد دارید که در صفحات گذشته از تگ <th> برای ایجاد یک سلول سربرگ استفاده می کردیم. توجه کنید که تگ <thead> با تگ <th> متفاوت است. همانطور که گفتیم، تگ <th> یک سلول سربرگ ایجاد می کند درحالیکه تگ <thead> یک ردیف از جدول را گروه بندی کرده که این گروه، "گروه سربرگ" نام دارد.

- تگ <tfoot> معمولا آخرین ردیف جدول را به عنوان "گروه ته برگ جدول" مشخص می کند

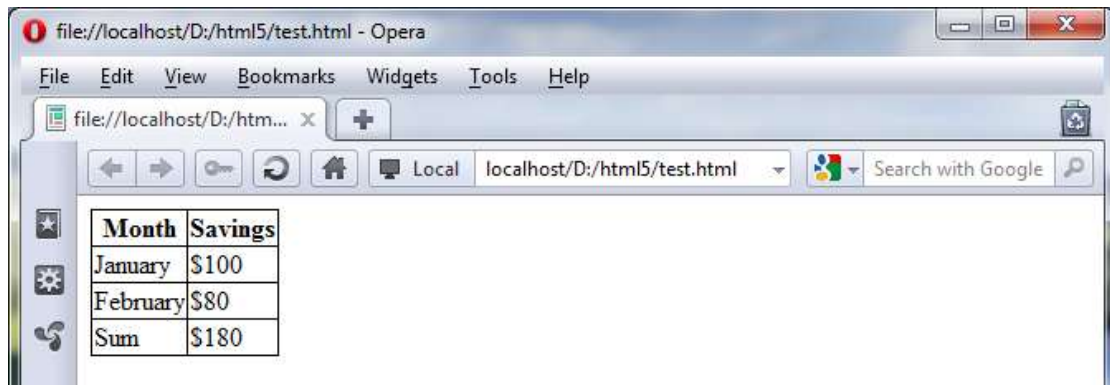
نکته ای که باید در استفاده از این تگ ها مورد توجه قرار بگیرد اینست که ترتیب نوشتن این سه تگ ، مهم است. یعنی ابتدا باید تگ <thead> ، سپس تگ <tfoot> و در انتها تگ <tbody> را بنویسیم. کد زیر نحوه استفاده از این تگ ها را نمایش می دهد. آنچه که مشخص است اینست که هنر اصلی این تگ ها، پس از آموزش CSS مشخص می گردد.

```
<!DOCTYPE HTML>
<html>
<head>
  <style type="text/css">
    table{border-collapse:collapse;}
    table, td, th{border:1px solid black;}
  </style>
</head>
<body>
  <table>
    <thead>
      <tr>
        <th>Month</th>
        <th>Savings</th>
      </tr>
    </thead>
    <tfoot>
      <tr>
        <td>Sum</td>
        <td>$180</td>
      </tr>
    </tfoot>
    <tbody>
      <tr>
        <td>January</td>
        <td>$100</td>
      </tr>
      <tr>
        <td>February</td>
        <td>$80</td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

تصویر ۷۱ – استفاده از تگ های <thead> ، <tbody> و <tfoot>

همانگونه که در تصویر فوق می بینید، سه قسمت <thead> ، <tbody> و <tfoot> را با رنگ متمایز نشان داده ایم. همانطور که می بینید، قسمت <tfoot> قبل از قسمت <tbody> نوشته می شود و قسمت <thead> نیز قبل از این دو.

در تصویر زیر می توانید نتیجه اجرای این کد را مشاهده نمایید:



تصویر ۷۲ - استفاده از تگ های `<thead>`، `<tbody>` و `<tfoot>`

در تصویر فوق ردیف اول جدول با استفاده از تگ `<thead>` در یک گروه با عنوان "گروه سربرگ" قرار داده شده است. ردیف آخر نیز با استفاده از تگ `<tfoot>` بعنوان "گروه ته برگ" و دو ردیف وسط نیز توسط تگ `<tbody>` به عنوان "گروه بدنه جدول"، گروه بندی شده اند.

۵-۵ `<colgroup>` و `<col>`

تگ `<col>` صفتی را برای یک یا گروهی از ستون های یک جدول مهیا می کند. این تگ معمولاً برای اعمال یک style به یک ستون مشخص به کار می رود. بعنوان مثال اگر جدول داشته باشیم که این جدول حاوی سه ستون باشد و ما بخواهیم کل ستون دوم را به رنگ قرمز درآوریم چه باید بکنیم؟ لابد می گوئید که در هر ردیف، سلول دوم را به رنگ قرمز در می آوریم. خب! این یک راه نسبتاً دشوار است. یک راه بهتر برای این کار استفاده از تگ `<col>` است. با استفاده از این تگ می توانید یک صفت را به کل ستون مورد نظر خود اعمال کنید. این تگ، یک تگ منفرد است و یک صفت اختصاصی دارد با عنوان `span` که این صفت مشخص کننده تعداد ستون هایی است که قرار است استایل را به آنها اعمال کنیم. به عنوان مثال توسط کد زیر، دو ستون اول یک جدول سه ستونی را به رنگ قرمز در می آوریم.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      table{border-collapse:collapse;width:100%}
      table, td, th{border:1px solid black;}
    </style>
  </head>
  <body>
    <table>
      <col span="2" style="background-color:red" />
      <tr>
        <th>ISBN</th>
        <th>Title</th>
        <th>Price</th>
      </tr>
      <tr>
        <td>3476896</td>
        <td>My first HTML</td>
        <td>$53</td>
```

```

</tr>
</table>
</body>
</html>

```

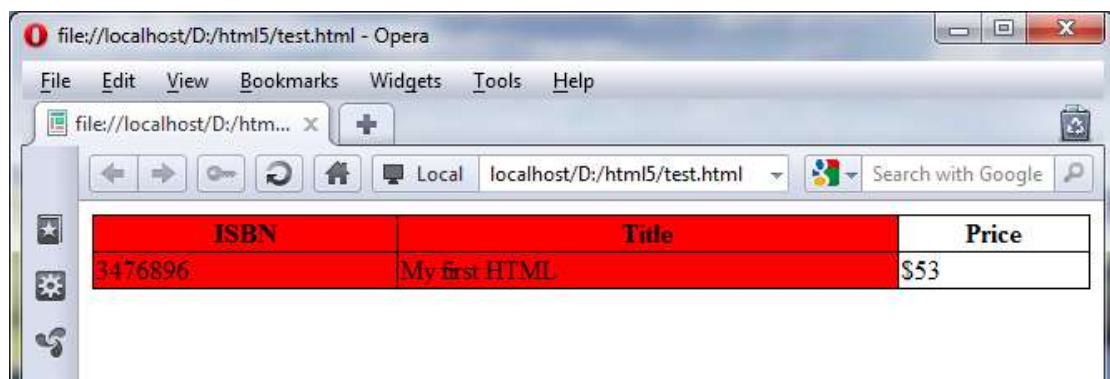
تصویر ۷۳ – استفاده از تگ <col/>

همانگونه که در کد فوق می بینید، در خط یازدهم یک تگ <col/> به همراه صفت span مربوط به آن نوشته ایم. به این صورت :

<col span="2" style = "background-color : red">
توجه کنید که صفت style یک صفت عمومی در تگ های html است. صفت های عمومی html را در فصل های آتی به طور مفصل توضیح خواهیم داد اما در اینجا فقط به ذکر این نکته بسنده می کنیم که توسط صفت style می توان شکل، رنگ و ظاهر یک عنصر html را تغییر داد که در اینجا این صفت باعث قرمز رنگ شدن پس زمینه ستون های جدول شده است.

صفت span نیز همانطور که ذکر کردیم، صفت اختصاصی تگ <col/> است. همانطور که می بینید ، در کد فوق مقدار این صفت برابر ۲ است که معنی آن اینست که از ستون فعلی، تعداد دو ستون را تحت تاثیر قرار بده. توجه کنید که اگر ما می خواستیم بجای دو ستون اول، دو ستون دوم را تحت تاثیر قرار دهیم، باید یک تگ <col/> بدون هیچ صفتی را قبل از تگ <col> کنونی می نوشتیم. به این صورت:

<col/>
<col span="2" style = "background-color : red">
به این ترتیب می توان از ستون اول پرش کرده و به ستون دلخواه رسید.
تصویر زیر نتیجه اجرای کد تصویر ۷۳ را به نمایش می گذارد:



تصویر ۷۴ – استفاده از تگ <col/>

همانگونه که در تصویر مشاهده می کنید، رنگ پس زمینه دو ستون اول به رنگ قرمز درآمده است.

اما تگ دیگری که قرار است در این قسمت یاد بگیرید، تگ <colgroup> است. این تگ برای گروه بندی ستون های جدول به منظور اعمال استایل بکار می رود. همانگونه که دیدیم، تگ <col/> استایل مورد نظر شما را به یک یا چند ستون اعمال می کرد. حال با استفاده از تگ <colgroup> می توانید این ستون ها را گروه بندی کرده تا کار استایل دهی آنها سریع تر و راحت تر انجام شده یا تغییر داده شود. در زیر نحوه استفاد از این تگ را مشاهده می کنید:

```

<!DOCTYPE HTML>
<html>
<head>

```

```

<style type="text/css">
  table{border-collapse:collapse;width:100%}
  table, td, th{border:1px solid black;}
</style>
</head>
<body>
  <table>
    <colgroup style="background-color:red" />
      <col/>
      <col/>
    </colgroup>
    <colgroup style="background-color:yellow" />
      <col/>
      <col/>
    </colgroup>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Mathematics</th>
      <th>Physics</th>
    </tr>
    <tr>
      <td>reza</td>
      <td>darrudi</td>
      <td>17.25</td>
      <td>16.5</td>
    </tr>
    <tr>
      <td>somayyeh</td>
      <td>maddah</td>
      <td>17</td>
      <td>18.25</td>
    </tr>
  </table>
</body>
</html>

```

تصویر ۷۵ - استفاده از تگ <colgroup>

در کد فوق، دو ستون اول که مربوط به نام و نام خانوادگی دانشجو هستند را در یک گروه و دو ستون دوم که مربوط به نمرات دروس هستند در یک گروه دیگر قرار داده ایم. همانگونه که مشاهده می کنید، تگ های <col> و <colgroup>، ردیف یا خانه ای را به جدول اضافه نمی کنند بلکه تنها همان ستون های موجود را تحت تاثیر قرار می دهند.

نکته دیگری که لازم به ذکر است اینست که تگ <colgroup> همانند تگ <col> دارای صفت خصوصی span می باشد که دقیقاً مشابه با صفت span در تگ col عمل می کند. در زیر می توانید نتیجه اجرای کد فوق را مشاهده نمایید:

file:///localhost/D:/html5/test.html - Opera

File Edit View Bookmarks Widgets Tools Help

file:///localhost/D:/htm... x +

Local localhost/D:/html5/test.html Search with Google

| First Name | Last Name | Mathematics | Physics |
|------------|-----------|-------------|---------|
| reza | darrudi | 17.25 | 16.5 |
| somayyeh | maddah | 17 | 18.25 |

تصویر ۷۶ - استفاده از تگ `<colgroup>`

۶. صفت های عمومی

اگر بخاطر داشته باشید در فصل قبل اشاره ای به صفت عمومی تگ `<col>` داشتیم و گفتیم که این صفت ها را در فصلی جداگانه به شما آموزش می دهیم. بسیار خب! اکنون وقت آن است که در مورد این صفات بیشتر بدانیم. تا اینجا با تگ های بسیاری آشنا شده اید و همانطور که دیدید بعضی از این تگ ها، صفت هایی نیز دارند بعنوان مثال حتما صفت `span` در تگ `<col/>` را بخاطر دارید یا صفت `start` در تگ `` که عدد آغاز یک لیست مرتب را مشخص می کرد.

به این صفت ها که برای هر تگ، متفاوت است، صفت های خصوصی می گویند که هر کدام از این صفت ها، وظیفه مخصوصی در تگ خود بر عهده دارند.

اما بعضی از صفت ها هستند که در تمام تگ ها وجود دارند. بعنوان مثال صفت `title` یکی از این صفت هاست. اگر بخاطر داشته باشید این صفت را اولین بار در تگ `<dfn>` بکار بردیم و سپس در تگ `<abbr>` نیز از آن مجددا استفاده کردیم. یا صفت دیگری مثل صفت `style` که این صفت را نیز تا بحال در چندین تگ بکار برده ایم. اینگونه صفتها که در همه تگ ها وجود دارند، تگ های عمومی نام دارند. یعنی اینکه تمام تگ ها می توانند از این صفت ها استفاده کنند.

در `html5` چهارده نوع صفت عمومی داریم که عبارتند از:

• Title

این صفت اطلاعات بیشتری در مورد عنصر ارائه می دهد که این اطلاعات معمولا به هنگام قرار گرفتن ماوس بر روی عنصر بصورت یک `tooltip` نمایش داده می شود. و بصورت زیر مورد استفاده قرار می گیرد `<element title = "value">` که بجای `element` نام تگ مورد نظر را نوشته و بجای `value` متن مورد نظر خود را بنویسید. این متن همان متنی است که در `tooltip` نمایش داده می شود. کد زیر نمونه ای از کاربرد این صفت در دو تگ `<abbr>` و `<p>` را نمایش می دهد:

```
<p>
  <abbr title="World Health Organization">WHO</abbr> was founded in 1948.
</p>
<p title="Free Web tutorials">
  W3Schools.com
</p>
```

تصویر ۷۷ – کاربرد صفت `title`

• Style

از این صفت به منظور ایجاد یک `inline style` برای عنصر مورد نظر استفاده می کنیم. همانگونه که در آینده خواهیم دید برای اعمال استایل به یک عنصر، چندین راه وجود دارد که استایل دهی از طریق صفت `style`، استایل دهی داخلی یا `inline` نامیده می شود. این نوع استایل دهی باعث می شود سایر استایل های اعمال شده به عنصر نادیده گرفته شده و استایل درج شده در این صفت به عنصر اعمال شود. در فصل های آینده در مورد `style` ها به تفصیل سخن خواهیم گفت. نحوه استفاده از صفت `style` به صورت زیر است:

`<element style = "value">`

که بجای `element` نام تگ مورد نظر را نوشته و بجای `value`، استایل مورد نظر خود را می نویسید. کد زیر نمونه ای از این کد را به شما نمایش می دهد:

<h1 style="color:blue;text-align:center">This is a header</h1>
<p style="color:green">This is a paragraph.</p>

تصویر ۷۸ – استفاده از صفت style

• Class

این صفت یک "نام کلاس" برای عنصر فراهم می کند که این "نام کلاس" در استایل دهی مورد استفاده قرار می گیرد.طریقه استفاده :

<element class = "value">

که بجای *element* نام تگ مورد نظر را نوشته و بجای *value*، نام کلاس مورد نظر خود را می نویسید. توضیحات کامل در مورد این تگ را در فصل CSS می توانید مشاهده کنید.

• Id

بوسیله این صفت می توان یک "نام شناسه" برای عنصر مورد نظر ایجاد کرد که این "نام شناسه" در JavaScript و CSS مورد استفاده قرار می گیرد. تعیین نام شناسه قواعدی دارد و مهمترین آن اینست که نام شناسه باید یکتا باشد. یعنی اینکه در فایل html خود نباید عنصر دیگری با این نام وجود داشته باشد. قواعد نامگذاری دیگری نیز وجود دارند که عبارتند از:

۱. هر نام شناسه باید با یکی از حروف (A تا Z) یا (a تا z) شروع شود. به عبارت دیگر، نام شناسه نمی تواند با اعداد یا کاراکترهای دیگر آغاز شود.

۲. نام شناسه تنها می تواند شامل حروف (a-z,A-Z)، اعداد (۰-۹)، خط فاصله (-)، خط زیر (_)، کولن (:) و نقطه (.) باشد. یعنی اینکه اگر کاراکتر دیگری غیر از این کاراکترها در نامگذاری استفاده شود، این نام شناسه نامعتبر خواهد بود.

۳. نام شناسه ، case-sensitive یا "حساس به متن" است. یعنی اینکه در نامگذاری، حروف کوچک و بزرگ با هم فرق دارند. به عبارت دیگر شناسه "my_Element" با شناسه "MY_element" تفاوت دارد.

نحوه استفاده از صفت id بصورت زیر است:

<element id = "value">

که بجای *element* نام تگ مورد نظر را نوشته و بجای *value*، نام شناسه مورد نظر خود را وارد کنید.

• tabIndex

هنگامی که از کلید Tab صفحه کلید کامپیوتر برای پرش بین عناصر صفحه استفاده می کنید، توسط این صفت می توانید، ترتیب فعال شدن این عناصر را تعیین کنید. نحوه استفاده از این صفت به شرح زیر است:

<element tabindex = "value">

که بجای *element* نام تگ مورد نظر را نوشته و بجای *value*، باید یک عدد وارد کنید. کد زیر کاربرد این کد را در لینک ها نشان می دهد:

```
<a href="http://www.w3schools.com/" tabindex="2">W3Schools</a><br />
<a href="http://www.google.com/" tabindex="1">Google</a><br />
<a href="http://www.microsoft.com/" tabindex="3">Microsoft</a>
```

تصویر ۷۹ – استفاده از صفت tabIndex

به مقادیر tabIndex دقت کنید. با اجرای این کد در مرورگر و فشردن کلید tab از روی صفحه کلید، لینک دوم به صورت فعال در می آید. زیرا صفت tabIndex، اولویت این عنصر را برابر ۱ قرار داده است.

حال اگر دکمه tab را مجدداً بفشارید، لینک اول فعال می شود. زیرا اولویت این عنصر با استفاده از صفت tabindex برابر ۲ تعیین شده است. فعال شدن به این معنی است که اگر شما دکمه Enter صفحه کلید را بفشارید، عنصر فعال شده، عکس العمل نشان خواهد داد و در این مثال با فشردن دکمه Enter، لینک دوم اجرا خواهد شد. فعال بودن یک عنصر در مرورگرهای مختلف بصورت های متفاوتی نمایش داده می شود. به عنوان مثال در مرورگر فایرفاکس، یک کادر نقطه چین در اطراف لینک فعال شده توسط کلید tab نمایش داده می شود.

• hidden

عنصری که از این صفت استفاده کنند در صفحه نمایش مخفی می شوند. یعنی اینکه وجود دارند اما نمایش داده نمی شوند. یکی از کاربردهای این صفت جلوگیری از مشاهده برخی عناصر توسط کاربر است. تا پس از ملاحظه ضوابط مربوطه (به عنوان مثال تیک زدن یک check box) این عناصر برای کاربر قابل رؤیت شود. نحوه استفاده از این صفت به صورت زیر است:

<element hidden = "hidden">

که بجای element باید نام عنصری که قصد مخفی کردن آن را دارید، بنویسید.

• dir

همانگونه که می دانید بعضی از زبان ها مانند زبان انگلیسی از چپ به راست نوشته می شوند و بعضی دیگر مانند زبان فارسی از راست به چپ. این صفت، جهت متن را مشخص می کند و دارای سه مقدار rtl^۱ (راست به چپ)، ltr^۲ (چپ به راست) و auto (تعیین جهت متن بصورت خودکار توسط مرورگر) است. این گزینه زمانی کاربرد دارد که به عنوان مثال در یک متن فارسی یک کلمه انگلیسی قرار داده می شود. در این صورت، یک نرم افزار خواندن متن، به وجود این صفت، نیاز مبرم خواهد داشت. در زیر نحوه استفاده از این صفت را مشاهده می کنید:

<element dir = "value">

که بجای element نام عنصر مورد نظر را نوشته و بجای value، باید یکی از سه مقدار rtl، ltr یا auto را قرار دهید.

• lang

صفت lang مشخص کننده زبان محتوا می باشد. زبان ها بصورت اختصار در این صفت قرار داده می شوند مثلاً fr برای زبان فرانسوی، en برای انگلیسی و fa برای فارسی بکار می رود. در آدرس زیر می توانید لیست کاملی از نام اختصاری زبان ها را مشاهده کنید:

http://w3schools.com/tags/ref_language_codes.asp

در زیر نحوه استفاده از این صفت را مشاهده می کنید:

<element lang = "value">

که بجای element نام عنصر مورد نظر را نوشته و بجای value، باید نام اختصاری زبان (مانند fa یا en یا) را بنویسید.

• contenteditable

این صفت به تازگی به html اضافه شده است. همانگونه که می دانید تا پیش از این، در یک صفحه وب امکان پاک کردن، تغییر دادن و ویرایش عناصر وجود نداشت اما اکنون با استفاده از این صفت می توان به

^۱ Right to left

^۲ Left to right

کاربر اجازه داد تا بتواند محتوی یک عنصر را در یک صفحه وب دستکاری و ویرایش کند. نحوه استفاده از این صفت به صورت زیر است:

`<element contenteditable = "value">`

که بجای *element* نام عنصر مورد نظر را نوشته و بجای *value* می توان یکی از سه مقدار `true`، `false` یا `inherit` را قرار داد. که `true` امکان ویرایش کردن را می دهد. `false`، امکان ویرایش را نمی دهد و `inherit` در صورتی امکان ویرایش را می دهد که عنصر `"parent"`^۱، نیز قابل ویرایش باشد در غیر اینصورت این عنصر غیر قابل ویرایش خواهد بود.

مفهوم `parent` و `child` یا پدر و فرزند در زبان های برنامه نویسی، مفهومی برای توصیف زیر مجموعه بودن یا ریشه بودن عناصر نسبت به یکدیگر است. به عنوان مثال در تکه کد زیر:

```
<html>
  <p>
    Hello world!
  </p>
</html>
```

عنصر `<p>`، فرزند عنصر `<html>` است زیرا در داخل این عنصر قرار گرفته و عنصر `<html>` نیز، پدر عنصر `<p>` است زیرا این عنصر را در خود جای داده است. (به همین سادگی)

در کد زیر نمونه استفاده از این صفت را ملاحظه می کنید:

```
<!DOCTYPE HTML>
<html>
  <body>
    <p contenteditable = "true">This is a paragraph. It is editable.</p>
    <p contenteditable = "false">This is a paragraph. It is not editable.</p>
    <b contenteditable = "true">
      <p contenteditable = "inherit">
        this is inherit that parent element is editable.
      </p>
    </b>
    <b contenteditable = "false">
      <p contenteditable = "inherit">
        this is inherit that patent element is not editable
      </p>
    </b>
  </body>
</html>
```

تصویر ۸۰ - کاربرد صفت `contenteditable`

همانگونه که در کد فوق مشاهده می کنید تمام حالت های استفاده از این صفت را ذکر کرده ایم. در تصویر زیر نمونه ای از اجرای این کد توسط مرورگر اپرا به نمایش گذاشته شده است.

^۱ عنصر پدر



تصویر ۸۱ – کاربرد صفت contenteditable

با اجرای کد فوق در کامپیوتر خود، تصویر مشابه تصویر فوق را مشاهده خواهید کرد. در این تصویر خطوط اول و سوم قابل ویرایش بوده اما خطوط دوم و چهارم غیر قابل ویرایش اند.

• spellcheck

صفت دیگری که به تازگی به جمع صفتهای عمومی html وارد شده است، صفت spellcheck است. اگر کد تصویر ۸۰ (کد قبلی) را در سیستم خود اجرا کرده باشید، متوجه خواهید شد که با تغییر متون قابل ویرایش، گاهی اوقات خط قرمز رنگی در زیر کلمات در حال ویرایش، ظاهر می شود. این خطوط بخاطر وجود قابلیت spell check است به نام "spell check" یا "تست املاء". و کار آن اینست که هنگامی که یک لغت از لحاظ املائی، نادرست است، در زیر آن یک خط قرمز رنگ رسم می کند تا شما را از وجود این اشکال مطلع سازد. با رفع این غلط املائی، خط قرمز نیز ناپدید می گردد. برای فعال یا غیر فعال کردن این قابلیت می توانید از صفت spellcheck استفاده کنید. این صفت دو مقدار را می پذیرد، true برای فعال کردن این قابلیت و false برای غیرفعال کردن آن. نحوه استفاده از این عنصر را در زیر مشاهده می کنید:

`<element contenteditable = "value">`

که بجای *element* نام عنصر مورد نظر را نوشته و بجای *value*، باید یکی از دو مقدار true یا false را بنویسید. کد زیر نمونه ای از کاربرد این صفت را نشان می دهد:

```
<p contenteditable="true" spellcheck="false">This is a praggagraph. It is editable
without spellcheck capability.</p>
```

تصویر ۸۲ – کاربرد صفت spellcheck

در کد فوق، متن مورد نظر قابل ویرایش بوده اما قابلیت spellcheck را ندارد.

• accesskey

این صفت، یک shortcut (کلید میانبر) برای عنصر درست می کند تا دستیابی به این عنصر از طریق صفحه کلید راحت تر و سریع تر شود. مقداری که برای این صفت در نظر گرفته می شود همان کاراکتر دستیابی به این عنصر است. نحوه استفاده از این صفت به شکل زیر است:

`<element accesskey = "value">`

که بجای *element* نام عنصر مورد نظر را نوشته و بجای *value*، یک کاراکتر را می نویسیم. بعنوان مثال اگر مقدار *value* برابر با "w" باشد، هنگام اجرای صفحه با فشردن کلید های Alt و w از صفحه کلید،

عنصر مذکور به حالت فعال در می آید. البته این صفت، در مرورگرهای امروزی کاربرد کمی دارد زیرا معمولاً مرورگرها از کلیدهای ترکیبی برای اجرای دستورات خود استفاده می کنند.

• **draggable**

صفت تازه ای که به html اضافه شده است، صفت **draggable** می باشد. با استفاده از این صفت می توانید قابلیت کشیدن و رها کردن با ماوس (drag and drop) را به عناصر اضافه کنید. نحوه استفاده از این صفت به صورت زیر است:

`<element draggable = "value">`

که بجای *element* نام عنصر مورد نظر را نوشته و بجای *value*، می توان یکی از سه مقدار **true**، **false** و یا **auto** را قرار داد. تکه کد زیر شما را بیشتر با این صفت آشنا می کند:

```
<!DOCTYPE HTML>
<html>
  <body>
    <p draggable = true >This is a paragraph. It is editable.</p>
    <p draggable = false >This is a paragraph. It is not editable.</p>
    <p draggable = auto >this is inherit that parent element is editable.</p>
  </body>
</html>
```

تصویر ۸۳ – استفاده از صفت **draggable**

- **dropzone**
- **contextmenu**

بسیار خب! اکنون که با همه ۱۴ مورد صفت های عمومی html5 آشنا شدید، به شما تبریک می گوئیم. شما می توانید این ۱۴ مورد را بطور خلاصه در جدول زیر مشاهده کنید:

| Attribute | Value | Description |
|--|-----------------------|--|
| accesskey | character | تعیین یک کلید میانبر از صفحه کلید برای عنصر مربوطه |
| class | classname | تعیین یک نام کلاس برای استفاده در CSS |
| contenteditable New | true false | اجازه به کاربر برای ویرایش محتوی عنصر |
| contextmenu New | menu_id | تعیین context menu برای عنصر |
| dir | ltr rtl | تعیین جهت متن |
| draggable New | true false auto | اجازه به کاربر برای کشیدن و رها کردن عنصر با ماوس |
| dropzone New | copy move link | Specifies what happens when dragged items/data is dropped in the element |
| hidden New | hidden | مخفی کردن عنصر |

| | | |
|---------------------------------------|-------------------------|--|
| id | <i>id</i> | تعیین نام شناسه برای عنصر |
| lang | <i>language_code</i> | مشخص کردن زبان محتوی |
| spellcheck New | true false | ایجاد قابلیت تست املاء |
| style | <i>style_definition</i> | تعیین یک استایل inline برای عنصر |
| tabindex | <i>number</i> | تعیین اولویت برای دستیابی از طریق دکمه tab |
| title | <i>text</i> | مشخص کردن اطلاعات اضافی در مورد عنصر |

جدول ۴ - صفت های عمومی html5

۷. کار با تصاویر، فیلم ها و صداها

۷-۱

در فصل های گذشته با متون و جدول ها آشنا شدید. حال وقت آن رسیده است که صفحات خود را با استفاده از تصاویر تزئین کنید. تصاویر یکی از مهمترین عناصر در زیبایی صفحات وب است. در html5، سه فرمت تصویری پشتیبانی می شود:

- فرمت jpeg: برای عکس های رنگی
- فرمت gif یا 8-bit png: برای عکس های شفاف (که پس زمینه ندارند)
- فرمت png: برای حروف شفاف (که پس زمینه ندارند)

برای قرار دادن تصویر در صفحه، از عنصر استفاده می کنیم. این عنصر علاوه بر صفت های عمومی که در فصل قبل معرفی کردیم، دارای ۶ صفت اختصاصی نیز می باشد که از این ۶ صفت، ۴ مورد اختیاری و ۲ مورد اجباری هستند به این معنی که این ۲ صفت حتما باید در تگ درج شوند. در جدول زیر این صفت ها را مشاهده می کنید، که دو صفت اول، صفتهای اجباری و ۴ صفت بعدی، صفتهای اختیاری هستند:

| Attribute | Value | Description |
|------------------------|-------------|--|
| alt | text | توضیحی در مورد مفهوم تصویر ارائه می کند |
| src | URL | آدرس URL مربوط به تصویر را تعیین می کند |
| height | pixels % | ارتفاع عکس را تعیین می کند |
| ismap | ismap | تعیین یک عکس به عنوان نقشه عکس سمت سرور |
| usemap | #mapname | تعیین یک عکس به عنوان نقشه عکس سمت کاربر |
| width | pixels % | تعیین عرض یک عکس |

جدول ۵ - صفت های تگ

در زیر، نحوه استفاده از این تگ را مشاهده می کنید:

همان طور که مشاهده می نمایید این عنصر از دو صفت اجباری src و alt استفاده می کند. صفت src منبع فایل عکس را مشخص می کند و صفت alt متنی را برای عکس در نظر می گیرد تا در مواقعی که عکس در دسترس نیست یا مرورگر نمی تواند آن را نمایش دهد، این متن بجای عکس نمایش داده شود. در ادامه توضیحات کامل در مورد این صفت ها را ملاحظه می کنید.

۷-۱-۱ صفت alt

هر عنصر باید صفت alt را داشته باشد تا در زمانی که کاربر قادر به دیدن عکس نیست، این متن بجای عکس نمایش داده شود. دیده نشدن عکس ممکن است دلایلی مختلفی داشته باشد. ممکن است

کاربری که در حال بازدید از سایت شماست، مشکل بینایی داشته و از نرم افزار های خواندن صفحه استفاده می کند، و یا اینکه مرورگر کاربر، تصاویر را نشان ندهد و یا فایل عکس در URL مشخص شده پیدا نشود و بسیاری دلایل دیگر. در این مواقع متن درون صفت alt می تواند تا حدی مقصود عکس را منتقل کند.

۷-۱-۲ صفت src

این صفت، منبع فایل تصویر را مشخص می کند. خوب است بدانید که در زبان های برنامه نویسی دو روش برای آدرس دهی وجود دارد. یکی "آدرس دهی مطلق" و دیگری "آدرس دهی نسبی". آدرس دهی مطلق، آدرس کامل و دقیق یک فایل را مشخص می کند. این نوع آدرس دهی معمولاً زمانی استفاده می شود که بخواهید یک فایل (تصویر، ویدئو، صوت و...) را از یک سایت و یا منبع دیگر فراخوانی کنید. به عنوان مثال URL زیر یک URL مطلق است:

`www.atada.ir/images/picture1.jpg`

همانگونه که مشاهده می کنید در آدرس دهی مطلق، آدرس کامل یک فایل به همراه نام میزبان مشخص شده و هیچ نکته مبهمی در مورد مکان فایل باقی نمی ماند. اما در مورد آدرس دهی نسبی وضع فرق می کند.

در آدرس دهی نسبی، آدرس فایل نسبت به مکانی که فایل کد را ذخیره می کنیم، آدرس دهی می شود. از این روش برای آدرس دهی فایل های وب سایت خود استفاده می کنیم. بعنوان مثال برای آدرس دهی نسبی یک فایل عکس تکه کد زیر را داریم:

```

```

در این شرایط سه حالت مختلف برای آدرس دهی نسبی پیش می آید:

۱. شما فایل عکس را در کنار فایل کد ذخیره کرده اید.

در این حالت کافی است فقط نام فایل عکس (بهمراه پسوند) را در صفت SRC بنویسید. در این حالت مرورگر، پوشه ای که فایل کد را در آن ذخیره کرده اید، جستجو کرده، فایل عکس را پیدا کرده و نمایش می دهد. پس در این حالت داریم:

```

```

۲. شما فایل عکس را درون یک پوشه در کنار فایل کد ذخیره کرده اید.

در این حالت شما باید ابتدا نام پوشه ای که فایل عکس در آن قرار دارد را نوشته، سپس یک علامت اسلش (/) قرار داده و در نهایت نام فایل عکس (بهمراه پسوند) را بنویسید. پس داریم:

```

```

۳. شما فایل عکس را درون یک پوشه در پوشه ای بالاتر از پوشه ای که فایل کد در آن

قرار دارد ذخیره کرده اید.

در این حالت ابتدا باید به یک پوشه بالاتر برویم و سپس از آنجا آدرس دهی کنیم. در html برای رفتن به یک پوشه بالاتر از علامت "دو نقطه اسلش" (../) استفاده می کنیم. مرورگر با مشاهده این علامت به یک پوشه بالاتر رفته و ادامه آدرس را از آنجا دنبال می کند. پس در این حالت داریم:

```

```

برای تمرین، دو پوشه با نام های source و images ایجاد کنید. تکه کد زیر را در پوشه source

```
<!DOCTYPE HTML>
<html>
  <body>
    <img alt = "html5 and css3" src = "../images/picture1.jpg" />
  </body>
</html>
```

ذخیره کرده و یک تصویر با نام picture1.jpg نیز در پوشه images قرار دهید.

تصویر ۸۴ – آدرس دهی نسبی

هنگامی که مرورگر این کد را اجرا می کند، با مشاهده علامت "دو نقطه اسلش" (../) ابتدا یک پوشه (نسبت به پوشه ای که فایل کد در آن قرار دارد) به عقب رفته سپس با مشاهده عبارت images به پوشه images رفته و با مشاهده عبارت picture1.jpg این فایل را جستجو کرده و پس از یافتن، آن را نمایش می دهد.



تصویر ۸۵ – نمایش تصویر با آدرس دهی نسبی

همانطور که در تصویر فوق ملاحظه می کنید، فایل test.html در پوشه source در حالیکه تصویری که قصد نمایش آن را داریم در پوشه images قرار دارد که بوسیله آدرس دهی نسبی، مرورگر این تصویر را نمایش می دهد.

۳-۱-۷ صفات width و height

این دو صفت برای تعیین عرض و ارتفاع یک تصویر بکار می روند. هنگامی که این دو صفت برای عنصر تعیین شده باشند، مرورگر قبل از دانلود تصویر، فضای مورد نیاز برای این تصویر را می داند و مقدار فضای لازم برای این تصویر را در چیدمان عناصر صفحه در نظر می گیرد. اما اگر این صفات تعیین نشده باشند، مرورگر مقدار فضای لازم برای این تصویر را نمی داند و چیدمان صفحه در طول دانلود تصویر تغییر می کند و کلیه عناصر موجود تا بارگذاری کامل صفحه، بطور مرتب جابجا می شوند.

نکته ای که باید در مورد این دو صفت متذکر شد اینست که اگر ما تصویر بزرگی داشته باشیم و بخواهیم از سایز کوچک این تصویر در صفحه استفاده کنیم ابتدا باید با نرم افزارهای مناسب (مانند فتوشاپ) اندازه

آن را کاهش داده و سپس از آن استفاده کنیم. زیرا کوچک کردن سایز تصویر با استفاده از صفت های width و height باعث کوچک کردن اصل تصویر نمی شود بلکه همان تصویر بزرگ دانلود شده و فقط در هنگام نمایش این تصویر کوچک تر نمایش داده می شود.

نکته دیگری که باید بدانید اینست که مقدار دهی صفت های width و height به دو گونه است. یکی بر اساس "پیکسل"^۱ و دیگری بر اساس "درصد"^۲. هنگامی که از روش درصدی استفاده می کنید باید در جلوی مقدار درج شده علامت درصد (%) را بگذارید. مانند زیر:

```

```

اما هنگامی که از روش پیکسلی استفاده می کنید باید بعد از مقدار درج شده، عبارت "px" را بنویسید. مانند زیر:

```

```

توجه کنید در صورتی که هیچکدام از علامت های درصد (%) و (px) را نگذارید، مرورگر شیوه مقدار دهی را، مقدار دهی پیکسلی در نظر می گیرد.

۴-۱-۷ صفات ismap و usemap

از این دو صفت در کار با نقشه عکس ها استفاده می شود. با استفاده از مفهوم نقشه عکس می توانیم نواحی قابل کلیک کردن در روی یک عکس ایجاد کنیم. توضیحات کامل مربوط به عنصر نقشه و دو صفت usemap و ismap را در فصل های آتی تقدیم حضورتان می کنیم.

۲-۷ <audio>

اگر بخواهید صدایی را به صفحه html خود اضافه کنید باید از تگ <audio> استفاده کنید. این تگ که به تازگی به html افزوده شده است یک تگ دو تایی می باشد و طریقه استفاده آن به صورت زیر است:

```
<audio src = "URL"> browser does not support the audio element </audio>
```

همانگونه که می بینید، یک عبارت بین دو تگ <audio> و </audio> قرار دارد. این عبارت هنگامی که مرورگر از تگ <audio> پشتیبانی نکند، در صفحه به نمایش در می آید و در غیر اینصورت این عبارت نمایش داده نمی شود. به همین خاطر معمولاً از عباراتی مانند "مرورگر شما از عناصر صوتی پشتیبانی نمی کند" در داخل تگ <audio> استفاده می کنیم تا در صورتی که مرورگر کاربر از این تگ پشتیبانی نکند، توسط این عبارت، موضوع برای کاربر توضیح داده شود.

در حال حاضر، Html5 از چهار فرمت صوتی mp3، wav، Acc و ogg پشتیبانی می کند. همانگونه که در جدول زیر مشاهده می کنید، عنصر <audio> پنج صفت خاص دارد که از بین این صفات تنها صفت SRC اجباری بوده و مابقی صفات اختیاری هستند.

| Attribute | Value | Description |
|--------------------------|----------|--|
| src | url | تعیین آدرس url فایل صوتی |
| autoplay | autoplay | اگر این صفت وجود داشته باشد، فایل صوتی بطور خودکار اجرا می شود |
| controls | controls | اگر وجود داشته باشد، دکمه های کنترلی نمایش داده می شوند |

^۱ pixels

^۲ percent

| | | |
|-------------------------|--------------------------|--|
| loop | loop | اگر وجود داشته باشد، فایل صوتی پس از خاتمه دوباره تکرار می شود |
| preload | auto metadata none | تعیین نحوه لود فایل صوتی |

جدول ۶ - صفات عنصر audio

۱-۲-۷ صفت src

صفت src در این تگ دقیقا مانند تگ عمل می کند و آدرس فایل صوتی را برای مرورگر مشخص می کند. نحوه استفاده از این صفت به صورت زیر است:

```
<audio src = "sound.mp3">
  browser does not support the audio element
</audio>
```

۲-۲-۷ صفت autoplay

در اینترنت دو روش برای پخش یک فایل صوتی وجود دارد. یکی اینکه ابتدا کل فایل، لود شده و سپس پخش شود. و دیگری اینکه مقداری از فایل لود شده و شروع به پخش شود و در همین حال، باقی مانده فایل نیز لود شده و در ادامه پخش گردد. استفاده از صفت autoplay به معنی استفاده از روش دوم است. یعنی اینکه با درج این صفت در داخل تگ audio، مرورگر به محض آماده شدن فایل صوتی آن را بصورت خودکار اجرا می کند.

```
<audio src = "sound.mp3" autoplay="autoplay">
  browser does not support the audio element
</audio>
```

۳-۲-۷ صفت controls

هنگامی که از این صفت استفاده می کنید در واقع به مرورگر اعلام می کنید که باید از پلیر^۱ پیش فرض خود استفاده کند. زیرا همانگونه که خواهیم آموخت طراح می تواند از پلیر طراحی شده خود بجای پلیر پیش فرض مرورگر استفاده کند.

در هر صورت، یک پلیر صوتی در html5 باید قابلیت انجام کارهای زیر را داشته باشد:

- اجرا (play)
- توقف (stop)
- جلو و عقب بردن فایل صوتی (seeking)
- تنظیم صدا (volume)
- حالت نمایش تمام صفحه (full screen)
- نمایش عنوان و زیرعنوان در صورت وجود (caption & subtitle)
- نمایش شماره آهنگ در صورت وجود (audio track)

^۱ player

در ادامه نحوه استفاده از این صفت را مشاهده می نمایید:

```
<audio src = "sound.mp3" controls = "controls">  
  browser does not support the audio element  
</audio>
```

۴-۲-۷ صفت loop

در شرایط عادی با پایان یافتن پخش یک فایل صوتی، این فایل دیگر اجرا نمی شود مگر اینکه کاربر دوباره بر روی دکمه play کلیک کند. اما گاهی اوقات نیاز داریم که فایل صوتی بطور خودکار تکرار شود. در این شرایط می شود از صفت loop استفاده کرد. نحوه استفاده از این صفت در زیر نمایش داده شده است:

```
<audio src = "sound.mp3" loop="loop">  
  browser does not support the audio element  
</audio>
```

۵-۲-۷ صفت preload

این صفت مشخص می کند که پس از لود شدن صفحه html، وضعیت لود شدن فایل صوتی چه می شود. این صفت سه مقدار را می پذیرد و بر اساس هر مقدار، شرایطی را تعیین می کند که در جدول زیر این موارد را مشاهده می نمایید:

| Value | Description |
|----------|--|
| auto | مشخص می کند پس از لود صفحه باید کل فایل صوتی لود شود. |
| metadata | مشخص می کند پس از لود صفحه تنها باید اطلاعات مربوط به فایل صوتی لود شود |
| none | مشخص می کند که پس از لود صفحه نباید فایل صوتی و اطلاعات مربوط به آن لود شود. |

جدول ۷- مقادیر صفت preload

البته باید توجه داشته باشید در صورتی که از صفت autoplay استفاده کنید، این صفت نادیده گرفته می شود. در زیر نحوه استفاده از این صفت را نیز مشاهده می نمایید:

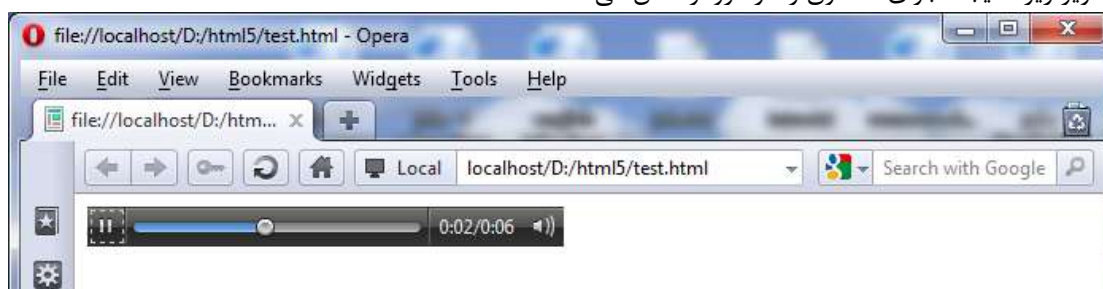
```
<audio src = "sound.mp3" preload = "auto">  
  browser does not support the audio element  
</audio>
```

بسیار خب! اکنون که با عنصر audio و صفات مربوط به آن آشنا شدید بهتر است مثالی از نحوه استفاده از آن را نیز مشاهده کنید:

```
<!DOCTYPE HTML>  
<html>  
  <body>  
    <audio src="sound.wav" controls="controls">  
      Your browser does not support the audio element.  
    </audio>  
  </body>  
</html>
```

تصویر ۸۶ – استفاده از تگ <audio>

تصویر زیر، نتیجه اجرای کد فوق را در مرورگر نشان می دهد:



تصویر ۸۷ – استفاده از تگ <audio>

البته لازم به ذکر است که در حال حاضر مرورگرها از هر چهار فرمت صوتی پشتیبانی نمی کنند. بعنوان مثال مرورگرهای فایرفاکس و اپرا از فرمت mp3 و مرورگر ie9 از فرمت wav پشتیبانی نمی کنند. تصویر زیر نمایانگر شرایط پشتیبانی مرورگرهای مختلف از فرمت های صوتی html5 است:

| | WIN | | | | | | | | | | MAC | | | | |
|-------------------|---|---|---|----|---|-------|---|---|---|---|---|---|---|---|---|
| |  | |  | |  | |  |  | | |  |  |  |  | |
| | CHROME | | OPERA | | FIREFOX | | SAFARI | IE | | | CHROME | OPERA | FIREFOX | SAFARI | |
| | 10 | 9 | 10, 63 | 11 | 3, 6 | 4, 03 | 5 | 6 | 7 | 8 | 9 | 7 | 10, 63 | 3, 6 | 5 |
| Audio: ogg/vorbis | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ |
| Audio: mp3 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ |
| Audio: wav | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ |
| Audio: AAC | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ |

تصویر ۸۸ – پشتیبانی مرورگرها از فرمت های صوتی

همانگونه که مشاهده می کنید، فرمت های ogg و mp3 بیشترین پشتیبانی را از سوی مرورگرهای مختلف دارا هستند پس بهتر است در طراحی بیشتر از این دو فرمت استفاده کنید.

۳-۷ <video>

این تگ نیز همانند تگ <audio> به تازگی به html افزوده شده است و امکان درج فایل های ویدئویی بدون استفاده از نرم افزارهای جانبی را فراهم آورده است. html5 از کدکهای^۱ ogg, webM و H.264 برای تگ ویدئو پشتیبانی می کند.

البته پشتیبانی مرورگرهای مختلف با هم متفاوت است. این تفاوت را در جدول زیر مشاهده می کنید:

^۱ codec

| | WIN | | | | | | | | | | | MAC | | | | |
|-------------------|---|---|---|----|---|------|---|---|---|---|---|---|-------|---|---|-----|
| |  | |  | |  | |  |  | | |  |  | |  | | |
| | 10 | 9 | 10.63 | 11 | 3.6 | 4.03 | 5 | 6 | 7 | 8 | 9 | 7 | 10.63 | 3.6 | 5 | |
| Video: ogg/theora | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | 70% |
| Video: H.264 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | 37% |
| Video: WebM | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | 27% |

تصویر ۸۹ – پشتیبانی مرورگرها از کدک های مختلف برای تگ <video>

همانند تگ audio ، این تگ نیز صفات خاص خود را دارد که این صفات را نیز در جدول زیر مشاهده می نمایید.

| Attribute | Value | Description |
|--------------------------|----------|---|
| audio | muted | اگر وجود داشته باشد، صدای ویدئو قطع می شود |
| autoplay | autoplay | اگر وجود داشته باشد، ویدئو بلافاصله پخش می شود |
| controls | controls | اگر وجود داشته باشد، دکمه های کنترلی نمایش داده می شود. |
| height | pixels | ارتفاع ویدئو پلیر را مشخص می کند |
| loop | loop | اگر وجود داشته باشد، ویدئو پس از پایان یافتن دوباره پخش می شود. |
| poster | url | آدرس url عکس مربوط به پیش نمایش ویدئو |
| preload | preload | مشخص میکند که بعد از کامل لود شدن صفحه ویدئو لود شود یا نه؟ |
| src | url | آدرس url فایل ویدئویی |
| width | pixels | عرض ویدئو پلیر را مشخص می کند |

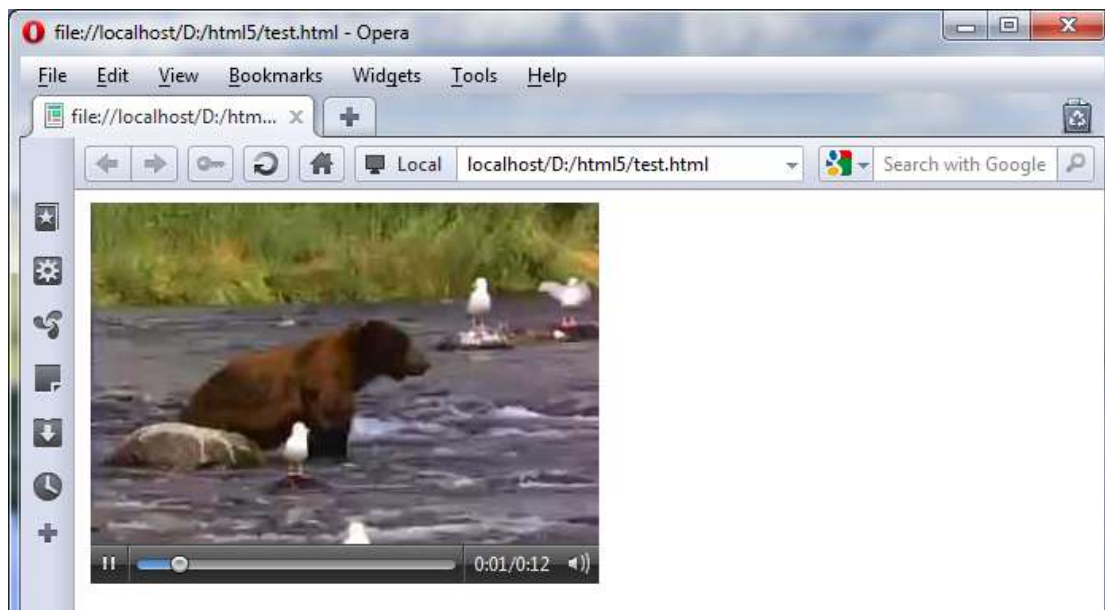
جدول ۸ – صفات تگ <video>

با توجه به اینکه صفات تگ <video> مشابه صفات دو عنصر قبل است، از توضیح مجدد خودداری کرده و توجه شما را به مثالی در رابطه با این تگ جلب می کنیم:

```
<!DOCTYPE HTML>
<html>
<body>
  <video src="movie.ogg" controls="controls">
    your browser does not support the video tag
  </video>
</body>
</html>
```

تصویر ۹۰ – استفاده از تگ <video>

در شکل زیر نمونه اجرای این کد را ملاحظه می نمایید:



تصویر ۹۱ – استفاده از تگ <video>

۷-۴ <source>

همانگونه که گفتیم، همه مرورگرها، تمام فرمت های صوتی و تصویری را پشتیبانی نمی کنند. به عنوان مثال مرورگر ie9 از فرمت ogg پشتیبانی نمی کند. این مسئله ممکن است مشکل ساز شود. یعنی اینکه اگر شما یک فایل صوتی با پسوند ogg. را در سایت خود قرار دهید، کاربرانی که از مرورگر ie استفاده می کنند، قادر به شنیدن این فایل صوتی نخواهند بود.

راه حل این مشکل استفاده از تگ <source> است. این تگ در داخل تگ های چند رسانه ای مانند <audio> و <video> قرار گرفته و آدرس فایل صوتی یا تصویری را مشخص می کند. با استفاده از این تگ، دیگر نیازی به استفاده از صفت src در داخل تگ های چند رسانه ای نخواهد بود. مانند زیر:

```
<"audio controls="controls">
  </ "source src="song.ogg" type="audio/ogg">
  </ "source src="song.mp3" type="audio/mpeg">
  Your browser does not support the audio element
</audio>
```

همانگونه که در تکه کد فوق مشاهده می کنید، ما دو بار از تگ <source> در داخل تگ <audio> استفاده کرده ایم. با این کار، مرورگر از بین پسوند های موجود، فایلی را که می تواند اجرا کند را انتخاب کرده و لود می کند. در صورتی که هیچ یک از این فایلها توسط مرورگر پشتیبانی نشود، متن داخل تگ به نمایش در می آید. این تگ مانند سایر تگ ها، یک سری صفات خاص دارد. که این صفات را در جدول زیر مشاهده می نمایید:

| Attribute | Value | Description |
|---------------------------------|------------------|-----------------------------|
| src New | <i>url</i> | فایل چند رسانه ای URL آدرس |
| type New | <i>MIME_type</i> | تعیین نوع فایل چند رسانه ای |

جدول ۹ – صفات تگ <source>

صفت src را در بخش های قبل معرفی کردیم. اما صفت type همانگونه که در جدول فوق مشاهده می کنید، نوع فایل را با استفاده از عبارتی مخفف (که در اصطلاح به آن MIME_type می گویند) مشخص می کنند. MIME_type های رایج برای فرمت ویدئویی عبارتند از:

- video/ogg
- video/mp4
- video/webm

و برای فرمت صوتی نیز عبارتند از:

- audio/ogg
- audio/mpeg

که نحوه استفاده از این MIME_type ها به صورت زیر است:

```
<source type="value">
```

که بجای value ، یکی از MIME_type های بالا را می نویسیم. به عنوان مثال داریم:

```
<source type="audio/ogg">
```

در ادامه، مثالی از کاربرد تگ <source> را مشاهده می نمایید:

```
<!DOCTYPE HTML>
<html>
  <body>
    <audio controls="controls">
      <source src="song.ogg" type="audio/ogg" />
      <source src="song.mp3" type="audio/mpeg" />
      Your browser does not support the audio element.
    </audio>
  </body>
</html>
```

تصویر ۹۲ – کاربرد تگ <source>

البته تگ <source> یک صفت دیگر هم دارد بنام media که چون بطور معمول از این صفت کمتر استفاده می شود از ذکر آن خودداری کردیم. در صورت تمایل می توانید در آدرس زیر جزئیات این صفت را مطالعه نمایید:
http://w3schools.com/html5/att_source_media.asp

۵-۷ <object> و <param>

با استفاده از تگ <object> می توانیم عناصر چند رسانه ای از قبیل فایل های فلش (swf)، فایل های pdf، اپلت های جاوا، تصاویر، فایل های ویدئویی و صوتی را به صفحه خود اضافه کنیم^۱. استفاده از این تگ به طراح این امکان را می دهد تا پارامترهایی را برای فایل چند رسانه ای تعیین کند که این پارامترها و اطلاعات می تواند توسط فایل مورد استفاده قرار بگیرد.

تگ <object> یک تگ دو تایی است و همانند تگ های <audio> و <video>، متنی را که بین تگ آغازین و تگ پایانی آن درج می کنیم، تنها در صورتی نمایش داده می شود که مرورگر از این تگ پشتیبانی نکند. این تگ، هفت صفت دارد که در جدول زیر این صفات را مشاهده می نمایید:

| Attribute | Value | Description |
|------------------------|-------------|---|
| data | URL | آدرس URL فایل چند رسانه ای |
| form | formname | مشخص کردن فرم یا فرمهایی که عنصر جاری به آنها تعلق دارد |
| height | pixels | تعیین ارتفاع عنصر |
| name | unique_name | تعیین یک نام برای عنصر |
| type | MIME_type | تعیین MIME_type فایل چند رسانه ای |
| usemap | URL | تعیین یک نقشه عکس برای عنصر |
| width | pixels | تعیین عرض عنصر |

جدول ۱۰ - صفات عنصر <object>

بجز صفت های form و name (که در فصل مربوط به فرم ها به توضیح آنها می پردازیم) با سایر صفات این عنصر، قبلاً آشنایی کامل پیدا کرده اید.

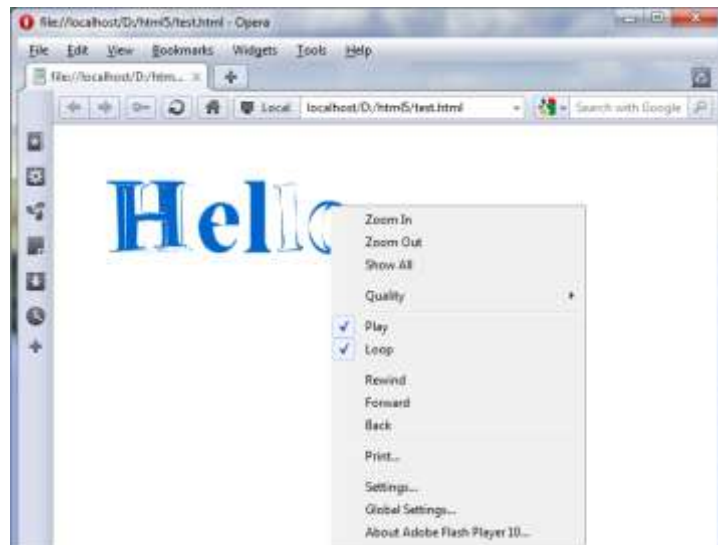
کد زیر نمونه ای از کاربرد تگ <object> است که یک فایل فلش با پسوند swf را در صفحه قرار می دهد:

```
<!DOCTYPE HTML>
<html>
  <body>
    <object data="helloworld.swf">
      Your browser does not support object element.
    </object>
  </body>
</html>
```

تصویر ۹۳ - استفاده از تگ <object>

تصویر زیر نیز نتیجه اجرای این کد را نمایش می دهد:

^۱ هرچند که با تگ object می توان عناصر تصویری، ویدئویی و صوتی را به صفحه افزود اما بهتر است برای این عناصر از تگ مخصوص به خودشان استفاده کنیم.



تصویر ۹۴ – استفاده از تگ <object> برای قرار دادن فایل فلش در صفحه

با توجه به تصویر می بینید که فایل فلش در صفحه لود شده است. و همانگونه که در تصویر مشخص است، با راست کلیک کردن بر روی این عنصر، منوی مربوط به نرم افزار flash player نمایان شده است.

بسیار خب! حالا به معرفی تگ <param> می پردازیم. اگر بخاطر داشته باشید در ابتدای معرفی تگ <object> گفتیم که در تگ <object> می توانیم پارامترها یا متغیرهایی را برای فایل، تعیین کنیم تا فایل چند رسانه ای از این پارامترها در زمان اجرا استفاده کند. این پارامترها توسط تگ منفرد <param> برای عنصر <object> مشخص می شود. نحوه استفاده از تگ <param> به صورت زیر است:

```
< param name = "parameter_name" value = "value">
```

که در هنگام استفاده از این تگ، بجای *parameter_name* نام پارامتر یا متغیر مورد نظر خود را نوشته و بجای *value* نیز مقدار مورد نظر خود را قرار می دهیم. هنگامی که فایل چند رسانه ای توسط تگ <object> در صفحه قرار میگیرد، مرورگر پارامترهایی را که توسط تگ <param> مشخص شده است را به این فایل اعمال می کند. به مثال زیر دقت کنید:

```
<!DOCTYPE HTML>
<html>
  <body>
    <object data="song.ogg">
      <param name="autoplay" value="false" />
    </object>
  </body>
</html>
```

همانگونه که در کد فوق مشاهده می کنید، مرورگر فایل صوتی را با استفاده از تگ <object> در صفحه قرار میدهد اما قبل از آن، پارامتر تعیین شده توسط تگ <param> را به این عنصر اعمال می کند. در مثال فوق تگ <param> ، پارامتر *autoplay* را برابر *false* قرار داده است که با اعمال این پارامتر (و مقدار آن) ، اجرای خودکار فایل صوتی بصورت غیر فعال در می آید.

۶-۷ <embed>

اگر بخواهید یک نرم افزار ساخته شده غیر html را در اینترنت قرار بدهید، می توانید از تگ <embed> استفاده کنید. برای این کار، سازنده نرم افزار باید یک plug-in تهیه کرده تا کاربران با نصب این plug-in بر روی مرورگر خود، قادر باشند نرم افزار مورد نظر را در صفحه مرورگر خود اجرا کنند.

پلاگین ها (plug-in) قابلیت های جانبی هستند که سازندگان نرم افزار در اختیار کاربران قرار می دهند تا این پلاگین ها، کار اجرا و گسترش نرم افزار را توسعه بخشند. به عنوان مثال برای اینکه یک مرورگر قادر به نمایش فایل های ویدئویی با پسوند mov. باشد باید ابتدا پلاگین quick Time بر روی مرورگر نصب شود.

تگ <embed> دارای چهار صفت مخصوص به خود است که در جدول زیر این صفات را مشاهده می نمایید:

| Attribute | Value | Description |
|------------------------|---------------|--|
| height | <i>pixels</i> | ارتفاع محتوای الحاقی را مشخص می کند |
| src | <i>url</i> | آدرس URL مربوط به فایل الحاقی را مشخص می کند |
| type | <i>type</i> | MIME_type فایل الحاقی را مشخص می کند |
| width | <i>pixels</i> | عرض محتوای الحاقی را مشخص می کند |

جدول ۱۱ – صفات تگ <embed>

MIME_type های تگ <embed> انواع مختلف و رو به گسترشی دارند که برای مشاهده لیست کامل آن می توانید به این آدرس مراجعه کنید:

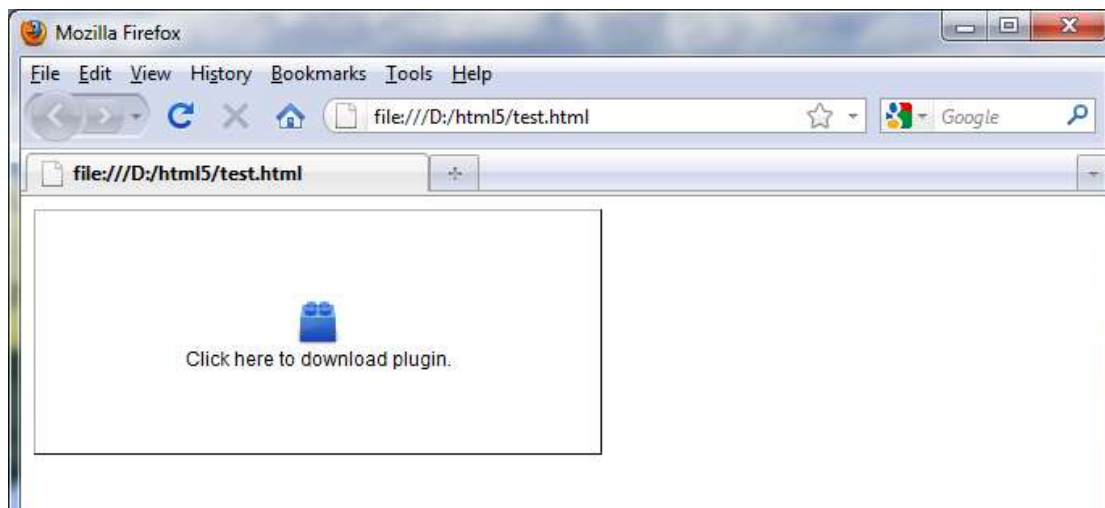
<http://www.iana.org/assignments/media-types/index.html>

بسیار خب! برای آشنایی بیشتر با این تگ به مثال زیر توجه فرمایید:

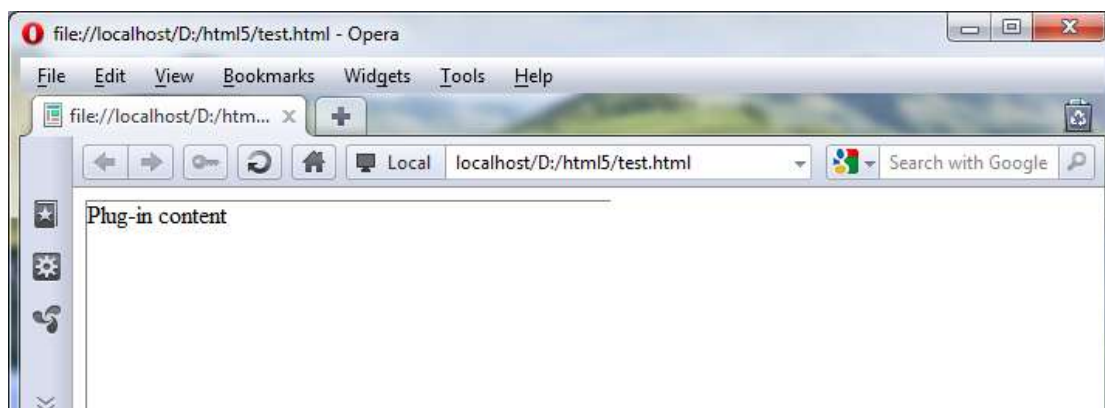
```
<!DOCTYPE HTML>
<html>
  <body>
    <embed type="video/quicktime" src="clip1.mov" width="340" height="140" />
  </body>
</html>
```

تصویر ۹۵ – استفاده از تگ <embed>

همانگونه که می بینید در کد فوق یک فایل ویدئویی با پسوند mov. را بوسیله تگ <embed> در صفحه قرار داده ایم. اگر پلاگین مربوطه بر روی مرورگر نصب باشد، این فایل ویدئویی اجرا می شود اما اگر پلاگین مورد نیاز نصب نباشد، مرورگر به شما پیغام می دهد که پلاگین مربوطه را نصب کنید. این پیغام در مرورگرهای مختلف متفاوت است. به عنوان مثال، نتیجه اجرای کد فوق، در مرورگرهای فایرفاکس و اپرا (که پلاگین مربوطه بر روی آنها نصب نبوده است) را در تصاویر زیر مشاهده می نمایید:



تصویر ۹۶ – پیغام نصب پلاگین در فایرفاکس



تصویر ۹۷ – پیغام نصب پلاگین در اپرا

همانگونه که در تصاویر فوق مشاهده می نمایید، پلاگین مربوط به نمایش فایل ویدئویی با پسوند mov. در مرورگر نصب نبوده به همین دلیل مرورگر پیغامی مبنی بر این موضوع به کاربر نمایش می دهد. برای نصب این پلاگین کافی است بر روی پیغام نمایش داده شده، کلیک کرده و پلاگین مورد نظر را نصب کنید.

۷-۷ <figure> و <figcaption>

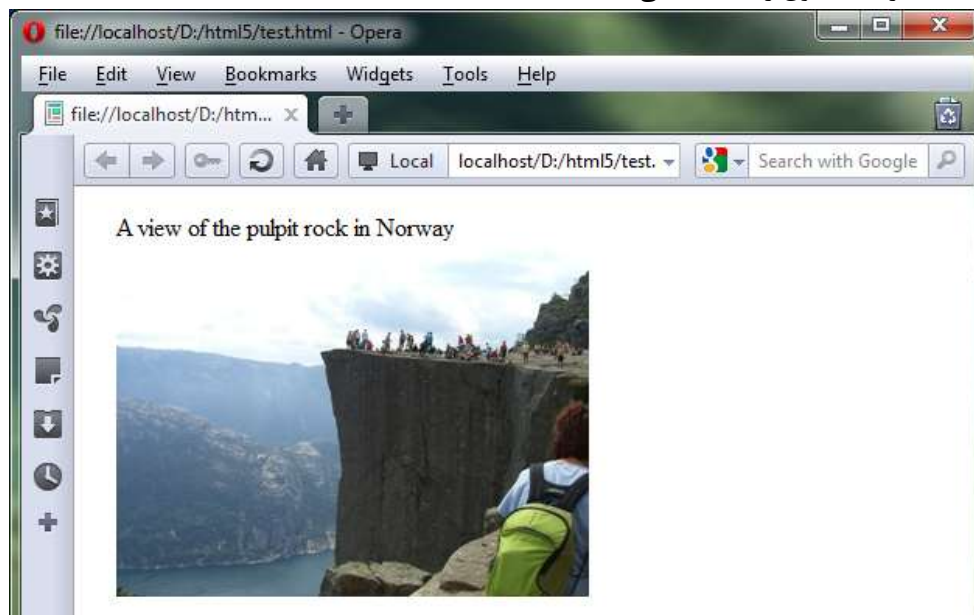
تگ <figure> به تازگی به html5 آمده است و برای فراهم کردن یک محتوی بصری از قبیل عکس، نمودار، کد و غیره بکار می رود. تگ <figcaption> نیز یک برچسب برای عنصر <figure> فراهم می کند. لازم به ذکر است که این تگ نیز به تازگی به html5 پا نهاده است.

```
<!DOCTYPE HTML>
<html>
  <body>
```

```
<figure>
  <figcaption>A view of the pulpit rock in Norway</figcaption>
  
</figure>
</body>
</html>
```

تصویر ۹۸ - استفاده از تگهای <figure> و <figcaption>

در زیر نتیجه اجرای کد فوق را ملاحظه می کنید:



تصویر ۹۹ - استفاده از تگ <figure> و <figcaption>

۸. ایجاد لینک و برقراری ارتباط با وب

در این فصل می آموزید که چگونه از طریق یک لینک^۱ (پیوند) با اسناد مختلف ارتباط پیدا می کنید. بطور پیش فرض، لینک ها بصورت متن هایی با خطی در زیر آنها نشان داده می شوند. هنگامی که ماوس را بر روی لینک قرار دهید شکل آن تغییر می کند که این امر نشان دهنده "لینک" بودن آن متن است. هنگامی که شما بر روی لینک کلیک می کنید، مرورگر شما را به جایی می برد که لینک به آن اشاره می کند. همانطور که در ادامه خواهید دید، برای ایجاد یک لینک در صفحه از تگ <a> استفاده می کنیم.

۸-۱ تگ <a>

که برای ایجاد لینک در صفحه استفاده می شود به صورت زیر است: <a>گرامر تگ

```
<a href="URLسایت مورد نظر">متن مورد نظر</a>
```

در html4 این تگ صفات متعددی داشت اما در html5 بخاطر استاندارد شدن کار طراحی، این تگها کاهش یافت و در حال حاضر تگ <a> دارای ۶ صفت می باشد که در جدول زیر مشاهده می کنید:

باید تکمیل شود:

۱. صفات draggable ، dropzone و contextmenu در فصل صفت های عمومی

۲. صفات ismap و usemap در فصل کار با تصاویر

۳.

^۱ Link

آموزش گام به گام

HTML5

Foundations

جلد اول

مؤلف : مهندس شاهرخ مہرآور

پاییز ۹۶

