

تصاویر پس زمینه در HTML (HTML Background) :

برچسبهای HTML که جنبه نمایشی دارند مانند `body` و `table` از دو صفت `background` و `bgcolor` پشتیبانی می کنند. این صفات برای تعیین تصویر و رنگ زمینه در محیط `web` بکار می رود. صفت `bgcolor` پیرو `background` است یعنی اگر برای `background` مقداری تعیین نکنیم، آنگاه `bgcolor` تأثیر گذار خواهد بود. مقادیری که به عنوان رنگ می توان تعیین نمود در سه قالب می باشد :

* کد هگزادسیمال رنگ (# 000000) :

که متشکل از یک علامت # و سه جفت عدد هگزادسیمال برای تعیین رنگ است. حداقل عدد هگز در هر جفت 00 (عدد صفر) و حداکثر آن FF (عدد 255) خواهد بود. به عنوان مثال #FFFFFF رنگ سفید و #000000 رنگ سیاه را نمایش می دهد. بقیه رنگها کدی بین این دو مقدار محسوب می شوند. در واقع این یک ترکیب سه رنگ RGB را تعیین می کند.

* تابع RGB :

این تابع به شکل `Rgb(0,0,0)` با دریافت سه رقم بین 0 تا 255 شدت روشنایی سه رنگ قرمز R، سبز G و آبی B را تعیین نمود و با ترکیب سه مقدار داده شده، رنگی را تولید می کند مثلاً `Rgb(255,0,0)` رنگ قرمز و `Rgb(255,255,255)` رنگ سفید را نشان خواهد داد.

* اسامی رنگ :

در HTML ترکیبهای بسیاری که در تمامی مرورگرها بدرستی نشان داده می شوند از پیش تعریف شده و بصورت نام شناسایی می شود مثلاً رنگ `Black` سیاه، `Red` قرمز، `Green` معادل سبز است. اسامی دیگری نیز وجود دارد که بصورت استاندارد تعریف شده اند.

اما بهترین راه تولید رنگ در HTML استفاده از روش اول و دوم است. اگر می خواهید رنگ مورد نظر خود را دقیقاً تولید کنید می توانید از ابزار `color picker` در فتوشاپ بهره مند گردید.

در خصوص انتخاب Background نیز باید نکات زیر را حتما در نظر بگیریم :

- آیا انتخاب تصویر پس زمینه تأثیر زیادی بر سرعت Load صفحه می گذارد و زمان را افزایش می دهد ؟
- آیا تصویر پس زمینه در ترکیب با سایر تصاویر موجود در صفحه خود به نظر می رسد ؟
- آیا تصویر پس زمینه با رنگ متون صفحه هماهنگی لازم دارد ؟
- آیا وجود تصویر پس زمینه تمرکز کافی روی موضوعات متنی را از بیننده سلب می کند ؟

اینها همه سوالاتی است که هنگام انتخاب تصویر پس زمینه باید پاسخ دهید و دقت کنید تا تصویری را انتخاب نمایید که ضمن حفظ زیبایی ظاهر صفحه پارامترهای فوق را نیز پوشش دهد .

در این قسمت به پایان بخش مقدماتی آموزش HTML می رسیم . به خاطر داشته باشید اگر به دنبال پیاده سازی یک گرافیک خوب در سایت خود هستید حتما با یک متخصص گرافیک مشورت کنید و از ابزارهای گرافیکی قدرتمند نظیر فتوشاپ بهره مند گردید .

موفق باشید

بخش دوم (HTML Advanced)

آشنایی با HTML Layout :

لایه بندی صفحه HTML از طریق جدول ممکن است ، شما می توانید کل صفحه HTML را با استفاده از table به بخشهای متنوعی تقسیم بندی کنید و در هر بخش مطلب یا تصویری خاص قرار دهید . بنابراین می توان یک صفحه Web را با کمک جداول مانند یک صفحه مجله به چند ستون دارای متن و تصویر تقسیم نمود و از این طریق دسته بندی راحت تری را برای مطالب ایجاد کرد . استفاده از این ترفند علاوه بر جذابیت صفحه ایجاد شده ، تأثیر عمیقی بر دقت بیننده صفحه می گذارد و او را وادار به پیگیری مطالب می نماید . درست مانند یک مجله که در صفحه اول خلاصه ای از یک مطلب را با تیتراژ درشت بیان می کند و ادامه آن را موقوف به صفحات بعد می نماید و از این طریق مخاطب مجبور به بررسی مجله می باشد .

می توان اینطور تصور نمود که لایه بندی یک صفحه مشابه تعریف frame است با این تفاوت که در این مورد برخلاف frame احتیاج به ساخت چند صفحه web مجزا نمی باشد و ما می توانیم در هر سلول از جدول لایه بندی برچسبهای دلخواه را قرار دهیم .

```
<html>
<body>
<table border = "0" width = "100" cellpadding = "10" >
<tr>
<td width = "50%" valign = "top" >
This is some text . This is some text . This is some text </td>
<td width = "50%" valign = "top" bgcolor = "yellow" >
Another text . Another text . Another text . Another text .
Another text . Another text . Another text .
</td>
</tr>
</table>
</body>
</html>
```

همانطور که در مثال بالا مشاهده می کنید با استفاده از این تقسیم بندی به راحتی می توانید صفحه را به بخشهای متفاوت تقسیم کنید و جلوه و تأثیر مطالب آنرا بیفزایید .

آشنایی با تعیین Font در HTML :

در HTML برای تنظیم متن و قالب بندی آن با برچسبهای مختلفی آشنا شدید . برای تغییر ظاهر متن از جمله اندازه ، رنگ و فونت از برچسب استفاده می شود . متنی که درون این برچسب قرارگیرد، از خواص این برچسب بهره می برد . صفات عمده این برچسب عبارتند از :

Size : که باید مقدار عددی بین 1 تا 6 جهت تعیین اندازه فونت باشد . البته با بکاربردن پسوند pt یا px می توان اندازه فونت را بر حسب نقطه یا پیکسل وارد نمود. مثلا `size = 12px` یعنی اندازه فونت برابر با 12 پیکسل باشد .

Face : این صفت نام فونت انتخابی را مشخص می سازد . فونتهای انتخابی می توانند Verdana, Arial, Courier, Tahoma و غیره باشد . البته برای زبان فارسی بیشتر از فونت Tahoma بصورت استاندارد استفاده می شود . اگر شما در Face نام فونتی را وارد کنید که مرورگر نتواند آن را پیدا کند خطایی صادر نمی شود ، بلکه متن با فونت استاندارد مرورگر نمایش داده خواهد شد .

Color : صفت رنگ جهت تعیین رنگ بر اساس همان قواعدی که قبلا مطرح شد کاربرد دارد . شما توسط این صفت یک کد رنگ یا تابع () rgb ویا نام رنگ را برای متن می توانید تعیین کنید .

**دقت کنید که استفاده از این امکانات نیازمند صبر و حوصله و دقت است تا متن خروجی به فرم زیبایی آراسته گردد . به مثال زیر توجه کنید :

```
<html>
<body>
<p><font face=' Tahoma ' color = "#0000cc" size = "12px" >
This is a somple text . <br>
این یک متن آزمایشی است .
</font>
</p>
</body>
</html>
```

اکنون با تغییر دادن مقادیر صفات در مثال قبل حالت‌های مختلف را تجربه کنید و تفاوت بین حروف انگلیسی و فارسی را در نظر بگیرید .

**نکته مهم : با توجه به استانداردهای (W3C) کنسرسیوم جهانی وب، استفاده از برچسب Font در ورژن‌های جدید HTML و مرورگرهای جدید حذف شده و به جای آن از مدل Stylesheet (CSS) که در ادامه توضیح داده می شود، بهره خواهیم برد .

اهمیت استفاده از استاندارد HTML 4.0 :

به جهت آنکه عمل نمودن به مدل HTML 3.2 که تاکنون آموختید نیازمند استفاده از برچسب‌های زیادی برای قالب بندی و انتخاب، رنگ و سایر مسائل برای متن می باشد ، نوشتن صفحات بزرگ وب بسیار مشکل می شود و هزینه تولید چنین صفحاتی بالا می رود . لذا مطابق استاندارد HTML 4.0 عمل نموده و کل برچسب‌های قالب بندی و Formatting را از بدنه HTML جدا می کنیم و در قالب فایل تحت عنوان Style sheet ذخیره می نماییم . در این صورت قادر خواهیم بود تنها یکبار فرمت‌های مختلف را ایجاد نموده و درون کدهای خود از این فرمت‌های از پیش تعیین شده بهره ببریم . برای اینکه فایل HTML خود را بصورت یک فایل HTML 4.0.1 معرفی کنیم، لازم است در اولین خط فایل با استفاده از برچسب `<!DOCTYPE>` با عنوان (Type Definition Document DTD) شناخته میشود و قبل از `<html>` در فایل قرار می گیرد، آنرا معرفی کنیم . برای معرفی فایل خود به صورت HTML 4.01 باید یکی از 3 دستور زیر را برحسب مورد در ابتدای فایل قرار دهیم :

* دستور تعریف فایل برحسب HTML 4.01 جهت صفحاتی که از Frame استفاده نمی کنند و همراه با CSS یا کاربرد تنظیمات آبخاری ، بکار می روند .

```
<! Doctype HTML Public “ - //W3C// DTD HTML 4.01 //EN” >
```

* دستور تعریف فایل برحسب HTML 4.01 به فرم Transitional جهت صفحاتی که از frame استفاده نمی کنند و نیز از برچسب‌هایی که W3C پشتیبانی نمی کند ، بهره برده اند و با CSS کار نمی کنند .

```
<! Doctype HTML Public “ - //W3C// DTD HTML 4.01 Transitional //EN” >
```

* دستور تعریف فایل برحسب HTML 4.01 برای صفحاتی که از fromset استفاده می کنند .

```
<! Doctype HTML Public “ _ //W3C// DTD HTML 4.01 fromset //EN” >
```

آشنایی با فرمت بندی بیرونی (HTML styles) :

همانطور که بیان شد در HTML 4.0 می توان کل برجسبهای فرمت بندی را حذف و آنها را درون یک فایل بیرونی ذخیره کرد. بطور کلی ایجاد Style sheet برای عناصر فایل HTML از سه طریق امکان پذیر است :

External style sheet : در این مدل ، محتوی Style sheet درون یک فایل External با پسوند CSS قرار گرفته و توسط برجسب <link> در بدنه <Head> به فایل افزوده می شود این مورد برای مواقعی مناسب است که صفحات web در سایت از یک مدل پیروی می کنند و شما می توانید به راحتی و با تعویض فایل CSS ، حالت‌های نمایشی از جمله فونت ، رنگ ، اندازه و غیره را تغییر دهید :

```
<head>
<Link rel = “stylesheet” type = “text / CSS” href = “mystyle.CSS” >
</head>
```

برای فراگیری چگونگی ساخت CSS (Cascade Style Sheet) بیرونی کتاب جداگانه ای داریم که به ان خواهیم پرداخت .

Internal style sheet : در این حالت style sheet توسط برجسب <style> درون heading تعریف می گردد و برای وقتی مناسب است که هر صفحه حالت مخصوص به خود را دارد برای این منظور مشابه فرم کلی زیر عمل می کنیم :

```
<head>
<Style type = “text/CSS” >
{ ... ; مقدار : صفت ; مقدار : صفت } نام برجسب
</style>
</head>
```

به عنوان مثال مجموعه زیر را به heading یک فایل HTML اضافه کنید و مشاهده نمایید که رنگ زمینه قرمز خواهد شد :

```
<head>
<Style type = "text/CSS" >
Body { background - color : red }
</style> </head>
```

Inline styles : فرم حالت درونی می تواند برای هر برچسب قابل فرمت بندی ، بکار رود، مثلا شما می توانید صفت Style را در برچسب <p> یا <td> یا برچسبهای مشابه استفاده کنید :

```
<p style = "color : red ; margin - left : 20px" >
```

در رابطه با نحوه تعیین style ها در سه حالت بالا نگران نباشید، با مطالعه مجموعه کار با stylesheet با فنون این مورد و قواعد آن آشنا می شوید . در ادامه این مبحث لازم است با دو برچسب دیگر نیز آشنا شوید .

**برچسب <div>:

این برچسب برای ایجاد یک تقسیم بندی درون متن بکار می رود و شما می توانید متن را بدون استفاده از جدول یا درون سلول یک جدول تقسیم بندی کنید :

```
<p> This some text
<div style = "color : red ; " >
<h4> This is header in div </h4>
<p> This is paragraph in div </p>
</div>
</p>
```

**برچسب :

این برچسب برای گروه بندی اجزاء درونی یک برچسب بکار می رود تا آنها را با style حالت دهی نماید . به مثال زیر توجه کنید :

```
<p> This is paragraph
<Span style = “color : green “ > This is a paragraph </span>
This is paragraph
</p>
```

در مثال فوق می بینید برخلاف <div> که به صورت خودکار یک Newline در ابتدا و انتهای آن قرار می گیرد هیچ شکستی در متن درون پاراگراف ایجاد نمی کند ، بلکه فقط رنگ جمله درون خود را تغییر می دهد . بنابراین از DIV برای حالت دهی اجزای یک بلوک HTML و از SPAN برای حالت دهی اجزای درون یک برچسب استفاده می شود.

کار با HTML HEADER

این بخش از HTML قبل از BODY در فایل قرار می گیرد و به جهت مشخص نمودن اطلاعات فایل HTML بکار می رود.

درون برچسب <Head> شما می توانید اعمال زیر را انجام دهید :

* استفاده از برچسب <title> جهت تعیین عنوان صفحه که در نوار عنوان مرورگر دیده خواهد شد که قبلا بیان شد .

* استفاده از برچسب <Base> جهت تعیین مسیردسترسی یا مقصد یکسان برای کلیه لینکها و برچسبهای صفحه

* استفاده از برچسب <link> برای تعریف و الصاق یک منبع بیرونی مانند Style sheet به فایل HTML

* استفاده از برچسب <meta> برای تعریف توضیحات و کلمات کلیدی مورد استفاده موتورهای جستجوی اینترنتی به شکل meta – information .

** برچسب <base> : این برچسب دو کار برعهده دارد که طی مثالهای زیر نمایش داده می شود :

تعریف مسیر دسترسی یکسان برای برچسبها :

```
<head>
<Base href = “www.yoursite.com / images “ >
```

```
</herad>
<body> <img src = "home.gif" > </body>
```

در مثال بالا چنانچه فایل home.gif که در img tag آمده در مسیر جاری فایل HTML نباشد، در آدرسی که درون Base آمده، جستجو می شود. لذا با این مدل می توان برای صفحه HTML آدرسهای متناظر تعریف کرد که در هنگام دسترسی به فایلها از همه آدرسهای فایل استفاده نماید .

تعریف مقصد یکسان برای لینکها :

```
<head>
<Base target = "_blank" >
</head>
<body >
<a href = "www.yahoo.com" > yahoo site </a>
</body>
</html>
```

در این مثال با وجود اینکه صفت Target در anchor عنوان نشده ،اما چون برچسب <Base> درون <Head> مقدار Target را برابر blank_ در نظر گرفته ، بنابراین هر لینکی در صفحه انتخاب شود ، در صفحه جدید باز خواهد شد .

برچسب <link> : این برچسب ارتباط بین فایل HTML جاری و فایلهای مرتبط را ایجاد می کند : کاربرد این tag بیشتر برای افزودن style sheet به فایل HTML است ، منتهی می تواند حالتهاى دیگری را نیز پشتیبانی کند.

```
<head>
<link rel = "style sheet" type = "text / CSS" href = "themes.CSS" />
</head>
```

همانطور که مشاهده می کنید این tag دارای صفاتی است که برخی از آنها در جدول ذیل آمده است :

شرح	مقدار	نام صفت
ادرس اینترنتی فایل منبع Resource را مشخص می کند	URL	href
نوع رابطه بین سند HTML فعلی با فایل منبع را تعیین می کند	Arternate Style sheet Section Copy right glossary	rel
مقصد بازشدن فایل منبع را تعیین می کند	- blank - top -self -parent	Target
نوع محتوی مقصد (MIME) را باگو می کند مثلا text/CSS به مرورگر می گوید فایل منبع متنی (text) و از نوع style shee (CSS) می باشند	Text/CSS Text/javascript Image/gif	Type

**** برچسب <meta> :** همانطور که قبلا هم بیان شد یکی از ویژگی هایی که در تعداد بیننده و توجه به سایت تأثیر گذار است ، امکان جستجو توسط موتورهای جستجو نظیر yahoo و google می باشد که با استفاده از <meta> می توان اطلاعات لازم برای این که منظور یا اطلاعات مربوط به نویسنده صفحه و غیره را در هدر فایل HTML قرار داد .

برچسب <meta> دارای دو صفت http-equiv و name است که همراه با صفت content برای معرفی هدر یا اطلاعات است . صفت http-equiv برای متصل نمودن یک مقدار درون content به هدر HTML است و دارای مقادیر زیر می باشد :

- * Content- type :** تعیین نوع محتوی صفحه که توسط صفت content کد گذاری می شود .
- * Expires :** تاریخ اعتبار و عمل صفحه web را در content آمده را مشخص می کند .
- * Refresh :** مدت زمان بازخوانی صفحه را در content آمده تعیین می نماید .
- * Set – cookie :** محتوی درون content را بعنوان نام فایل مخفی مربوط به صفحه اعلام می نماید .

****به عنوان مثال دستور زیر باعث می شود که صفحه به طور خودکار هر 5 ثانیه Refresh شود .**

<meta http – equiv = “Refresh” content = “5” >

صفت name رشته درون content را به یک نام متصل می کند. یعنی مقادیر برای name عبارتند از :

* author : نام مولف و نویسنده صفحه

* Description : توضیحاتی پیرامون محتوای صفحه

* Key word : کلماتی کلیدی که برای موتورهای جستجو مورد استفاده قرار می گیرد . مثلا صفحه ای که در رابطه با آموزش مطلب دارد با تعیین یک key word با محتوی learn در هنگام جستجو درون موتورهای جستجوی وب نمایش داده می شود .

* Generator : نام برنامه سازنده صفحه (معمولا در برنامه های ادیتور مخصوص HTML اتوماتک به صفحه اضافه می شود) البته name می تواند هر تعدادی خارج از موارد فوق نیز باشد .

بخش سوم : XHTML

XHTML در واقع نسخه ای حجیم تر و شفاف تر از HTML است . XHTML سرنام کلمات Extensible Hyper Text Markup به معنای HTML قابل توسعه می باشد . XHTML بعنوان HTML 4.01 شناخته می شود و HTML را در قالب یک برنامه XML تعریف می نماید . همچنین XHTML مورد تأیید و پیشنهاد کنسرسیوم جهانی (W3C) Web می باشد . اکنون که با HTML آشنا شدید، می توانید با بکارگیری قواعد XHTML در tag هایی که شناخته اید در قالب جدیدتر کار کنید و صفحات خود را به استانداردهای روز نزدیک نمایید . به خاطر داشته باشید که با بکارگیری استانداردها در هنگام حجیم شدن وب سایت خود و یا ارتقاء آن از مشکلات کمتری برخوردار خواهید بود . در حال حاضر تمامی مرورگرهای جدید با XHTML 1.0 همراه HTML 4.01 سازگاری دارند . هدف از این بخش دستیابی به موارد ذیل است :

- ✓ چرا از XHTML استفاده می کنیم.
- ✓ قواعد دستوری برچسبهای Xhtml را بشناسیم و تفاوتهای آن با html را بررسی کنیم .
- ✓ با طریقه تعریف اسناد xhtml آشنا شویم .
- ✓ ساختار مازولار و تقسیم شده xhtml را فرا بگیریم .
- ✓

* چرا از xhtml استفاده می کنیم :

XHTML ترکیبی از HTML و XML می باشد . XML یا Extensible Markup language به معنای زبان علائم قابل توسعه نوع، پرکاربردی از زبان برچسب می باشد که توسط آن سیستمهای زیادی با هم ارتباط برقرار می کنند . از XML برای ردوبدل اطلاعات استفاده زیادی می شود بویژه ابزارهای الکترونیکی مثل موبایل و کامپیوتر های جیبی از این تکنولوژی بهره زیادی می برند .

بر اساس بررسیهای بعمل آمده اکثر صفحات وب از کدهای HTML خوبی برخوردار نیستند. یعنی از استانداردهای روز بهره نبرده اند ، لذا ممکن است به خوبی نمایش داده نشوند یا بر اثر گذر زمان و تغییر و بروزآوری مرورگرها با اشکالاتی مواجه می شوند .

XML یک زبان برچسبی و علامت دار است که درون آن باید tag ها بدرستی بکار روند که در نتیجه سند و فایل به حالت بهینه در خواهد آمد . با توجه به استفاده گسترده اینترنت در ابزارهایی چون

کامپیوترهای خانگی ، موبایل و یا دستگاههای جیبی ، ترکیب HTML و XML می تواند بسیار کارآمد باشد و برچسبهای HTML را به فرم بدون نقص تبدیل نماید در واقع صفحاتی که براساس XHTML طراحی می شوند و در هر دستگاهی که از XML پشتیبانی کند قابل استفاده است و در اینصورت مطمئن خواهید بود که صفحه طراحی شده شما در اکثر مرورگرها و دستگاهها قابل مشاهده است .

تفاوتهای ما بین HTML و XHTML :

بطور کلی XHTML تفاوت زیادی با قواعد HTML 4.1 ندارد . منتهی اگر بخواهیم کد HTML قابل تعمیم به XHTML را بنویسیم بایستی حتما دقت نماییم که در برچسبها و دستورات از حروف کوچک انگلیسی استفاده گردد و نیز حتما قواعد دستوری را رعایت نماییم . مثلا اگر در HTML یک برچسب `<p>` را برای پایان پاراگراف نگذاریم ، مشکلی ایجاد نمی شود ، ولی در XHTML اینگونه نیست .

مهمترین تفاوتهای XHTML و HTML عبارت از :

✓ **Element ها در XHTML باید کاملا تو در تو باشند .** به مثال زیر توجه کنید .

```
<b> <i> Some text </b> </i>
```

مشاهده میکنید در HTML اینگونه می توان عمل نمود اما در XHTML باید به شکل زیر عمل نماییم .

```
<b> <i> Some text </i> </b>
```

همانطور که مشاهده می کنید در XHTML برچسب `<i>` را که درون `` آمده باید قبل از `` قرار گیرد تا درون بدنه آن باشد .

✓ اسناد HTML X باید در بهترین حالت طراحی شوند (well formed)

- بهترین حالت برای XHTML این است که tag های HTML همگی باید درون `<html>` و `</html>` قرار گیرند و نیز هربرچسب sub element درونی یک برچسب `<head>` یا `<body>` است. اگر چنین حالتی را رعایت کنیم و tag ها را بدرستی درون هم قرار دهیم قطعاً که ما در بهترین حالت well formed قرار گرفته و برای XHTML مناسب است.

✓ Tag ها در XHTML با حروف کوچک انگلیسی نوشته می شوند :

- در HTML شما می توانید از حروف بزرگ انگلیسی یا حروف کوچک بدون محدودیت استفاده کنید در حالی که در XHTML حروف بزرگ انگلیسی کارآمد نخواهد بود.

✓ کلیه element های غیر خالی XHTML باید بسته باشند .

منظور از بسته بودن یک element بکار بردن tag انتخابی است. به مثال زیر توجه کنید :

```
<p> Some text
<p> other text
```

دو tag فوق در HTML شناسایی می شود ، اما در XHTML باید `</p>` در انتهای آن حتما قرار گیرد تا element مسدود گردد .

بنابراین کلیه element های غیر خالی (non – empty element) را که درون آنها مطلب قرار می گیرد باید tag انتهایی داشته باشند .

✓ کلیه element های خالی هم باید در XHTML بسته باشند :

منظور element خالی (empty element) در واقع tag های یکتا مانند `
` ، `<hr>` و `` است که درون محدوده آنها مطلبی قرار نمی گیرد . این tag ها در XHTML باید به `>` / ختم گردد به مثال زیر توجه کنید :

```
Some text <br>
Here is horizontal line <hr>
<img src = "logo.gif" >
```

این tag ها در HTML شکلی ایجاد نمی کند اما در XHTML باید به صورت زیر بکار رود :

```
Some text <br/>
This is horizontal line <hr/>
<img src = "logo.gif" />
```

در خصوص صفات برچسبها (tag Attribute) نیز باید موارد زیر را حتما رعایت نماییم :
 ✓ نام صفات حتما باید با حروف کوچک انگلیسی نوشته شود .

```
< Table width = "100%" >
```

✓ مقدار صفات حتما باید درون کوتیشن قرار گیرد .

```
< Table width = "100%" >
```

✓ خلاصه نویسی صفات اشتباه است .

```
< input checked = "checked" >
```

65

✓ برای استفاده صفت name در tag ها باید از صفت id استفاده شود .

```
< img src = "picture.gif" id = "picture" >
```

Element های اجباری در XHTML (mandatory element) :

بطور کلی در قالب XHTML وجود برخی element ها اجباری است . یکی از این element ها <!Doctype > است که برای تعریف نوع سند بکار می رود . <html> ، <body> و <head> نیز جزء المنتهای اجباری محسوب می شوند که باید در سند وجود داشته باشد . همچنین <title> بایستی درون <head> موجود باشد . بنابراین فرم کلی یک سند XHTML به صورت زیر است :

```
<!Doctype >
<html>
<head>
<title> _____ </ title>
<body>
```

```
</body>
```

</html>

نکته ای که باید در نظر بگیرید این است که برچسب Doctype جزئی از سند XHTML نیست بنابراین استفاده از حرف بزرگ اشکالی ندارد و نباید Tag انتهایی داشته باشد .

آشنایی با طریقه تعریف اسناد XHTML :

همانطور که قبلا مشاهده شد برچسب <!DOCTYPE > یک tag اجباری (mandatory) و در سند XHTML است . این tag برای تعریف نوع سند به سه صورت که قبلا در HTML نیز توضیح داده شد بکار می آید. اما در اینجا به ذکر یک نمونه آن در سند HTML بسنده می کنیم . در هر سند XHTML باید برچسب <!DOCTYPE > به صورت زیر وجود داشته باشد .

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 strict //EN" >

حالت strict برای معرفی سند حجیم بوده و بجای آن می توان از Transitional و Frameset نیز استفاده نمود .
بنابر آنچه تاکنون مشاهده گردید طریقه تبدیل یک سند HTML به XHTML در موارد ذیل خلاصه می گردد:

✓ ابتدا یک برچسب <!DOCTYPE > به سند اضافه می کنیم (مخصوص XHTML)

✓ نام برچسب و صفات آن را به حروف کوچک انگلیسی تغییر می دهیم .

✓ همه مقادیر صفات برچسبها را درون زوج کوتیشن قرار می دهیم .

✓ برچسبهای خالی empty tag را مسدود می کنیم .

شما می توانید برای اطمینان از صحت سند XHTML خود از ادرس Validator.W3.org استفاده نمایید .

آشنایی با مدل ساختار بندی شده HTML X :

X HTML آسان است اما حجم کد های آن بسیار زیاد می باشد . همچنین یکی از ترفندهای مورد نیاز طراحان سایت استفاده مجدد از کدهایی است که قبلا طراحی کرده اند ، به همین جهت ماژولهای XHTML می تواند بسیار کارآمد باشد .

* با ماژولار کردن XHTML ، طراحان سایت می توانند اعمال زیر را انجام دهند :

- ✓ توسعه X HTML با افزودن کدهای XML بدون شکستن استانداردهای XHTML
- ✓ طراحی ساده فایل های XHTML ویژه ابزارهایی چون کامپیوترهای دستی ، موبایل و غیره
- ✓ توسعه برنامه های کاربردی XHTML با افزودن توابع XML نظیر توابع ریاضی ، منطقی و چند رسانه ای

برخی ماژولهای XHTML عبارتند از :

Applet module : جهت معرفی یک Applet درون XHTML

Forms Module : معرفی کلیه المنتهای قابل استفاده در Form

Object Module : تعریف المنتهای تصویری با استفاده از img

در پایان این بخش شما با تفاوت های XHTML و انعطاف آن نسبت به HTML آشنا شده اید . مواردی که در این بخش اشاره نشده مانند رویدادها و غیره همگی بین XHTML و HTML مشترک هستند و با استفاده از ضمائم این کتاب می توانید به مرجع کاملی از برچسبها و طریقه استفاده از آنها دسترسی پیدا کنید .

افزودن مالتی مدیا با <object> :

برچسب <object> همتراز برچسب <script> برای افزودن موضوعات صوتی یا تصویری خارج از سند HTML است که به صورت پیوسته embeded به سند متصل می گردد . درون object element با استفاده از برچسب <param> پارامترهای لازم برای object مشخص می گردد . معمولا object و صفات و پارامترهای مربوط به آن توسط برنامه های تولید کننده آن پیشنهاد می گردد . یک object element می تواند درون <head> یا <body> قرار گیرد . متنی که درون Object قرار می گیرد ، بعنوان جمله Alternative است که چنانچه موضوع <object> توسط مرورگر قابل پخش نباشد این متن به نمایش در می آید .

```
<object classid = "clsid : F084DF954-8592-11D1-B764-OOCOFO283623" id
="slider1"width="100" height="50">0
<param name ="border style " Value ="1" />
<param name ="Mouse Pointer " Value ="0" />
<param name ="Enabled " Value ="1" />
<param name ="Min " Value ="0" />
<param name ="max " Value ="1" />
</object>
```

اگر کد فوق را به یک سند اضافه کنید می بینید که روی صفحه HTML شما یک Slider یا میله تنظیم ظاهر می شود که توسط param مقدار min آن برابر با 10 تنظیم شده است . در object عبارت classid برای معرفی کد رجیستری ویندوز مربوط به object (مثال بالا) و یا یک آدرس وب سایت است .

کنترل رویدادها در HTML (HTML Events) :

در HTML 4.0 وقایع مختلف از طریق مرورگر قابل کنترل هستند ، مثلا می توان Script خاصی را در زمان Click روی یک موضوع اجرا کرد و با استفاده از ترفندهای این چینی انعطاف پذیری اسناد HTML بیشتر شده و قابل برنامه ریزی می شوند . برای آشنایی با چگونگی Script نویسی در رابطه با رویدادها (events) لازم است به مطالعه Javascript بپردازید . بطور کلی رویدادها به 4 دسته زیر تقسیم می شوند :

✓ رویدادهای پنجره (Window event) که در رابطه با پنجره مرورگر صفحه ای که باز می شود عمل می نماید :

- **Onload** : زمانی رخ می دهد که فایل load می شود می شود . (هنگام باز شدن صفحه)
- **Onunload** : زمانی رخ می دهد که فایل unload می شود (هنگام بسته شدن صفحه)

✓ رویدادهای عناصر فرم (From Element Events) : این رویدادها درون فرمها اتفاق می افتد و عبارتند از :

- **Onchange** : هنگامی رخ می دهد که تعداد درون یک المنت تغییر کند، مثلا متنی در Text box تایپ شود .
- **Onsubmit** : هنگامی رخ می دهد که فرم بخواهد اطلاعات را ارسال کند .
- **Onreset** : هنگامی رخ می دهد که فرم reset می گردد .
- **Onselect** : هنگامی رخ می دهد که یک المنت درون فرم انتخاب می شود .
- **Onblur** : هنگامی رخ می دهد که یک المنت از حالت انتخاب خارج می شود .

- **Onfocus** " هنگامی رخ می دهد که یک المنت در حال عملیات جاری قرار می گیرد .
- ✓ رویدادهای صفحه کلید (keyboard events) : این رویدادها مربوط به رفتار صفحه کلید می شود .
- **Onkeydown** : هنگامی رخ می دهد که کلیدی از صفحه کلید پایین نگه داشته شود .
- **Onkeypress** : هنگامی رخ می دهد که کلیدی از صفحه کلید پایین نگه داشته شود و رها گردد .
- **Onkeyup** : هنگامی رخ می دهد که کلیدی از صفحه کلید رها می گردد
- ✓ رویدادهای ماوس (mouse events) : این رویدادها در اثر کار با ماوس به وجود می آیند .
- **OnClick** : هنگامی رخ می دهد که با ماوس روی یک موضوع کلیک کنیم .
- **Ondblclick** : : هنگامی رخ می دهد که با ماوس روی یک موضوع دابل کلیک کنیم .
- **Onmousedown** : : هنگامی رخ می دهد که کلیدی از ماوس فشرده شود .
- **Onmousemove** : هنگامی رخ می دهد که نشانگر ماوس روی موضوع حرکت کند .
- **Onmouseout** : هنگامی رخ می دهد که که نشانگر ماوس از روی یک المنت خارج شود .
- **Onmouseover** : هنگامی رخ می دهد که نشانگر ماوس روی یک المنت قرار گیرد .
- **Onmouseup** : هنگامی رخ می دهد که دکمه فشرده شده ماوس رها گردد .