
طراحی صفحات وب به زبان ساده

تألیف:

مهندس نوراله کریم تبار احمدچالی



فن آوری نوین

سرشناسه	: کریم تبار احمد چالی، نوراله، ۱۳۶۶-
عنوان و نام پدیدآور	: طراحی صفحات وب به زبان ساده / تألیف نوراله (صحیح: نورالله) کریم تبار احمد چالی
مشخصات نشر	: بابل: فناوری نوین، ۱۳۹۸.
مشخصات ظاهری	: ۲۳۲ ص. :مصور
شابک	: ۴۵۰۰۰۰ ریال 5-30-7272-600-978:
وضعیت فهرست نویسی	: فیبا
یادداشت	: کتاب نامه
موضوع	: وبگاه ها - طراحی
موضوع	: Web sites -- Design
رده بندی کنگره	: ۱۳۹۸ ت ۴۴ ک / ۵۱۰۵ / TK۸۸۸
رده بندی دیویی	: ۰۰۵/۷۲
شماره کتابشناسی ملی	: ۵۶۳۲۰۲۸

تلفن: ۰۱۱-۳۲۲۰۶۶۸۷

www.fanavarienovin.net
بابل، کد پستی ۷۳۴۴۸-۷۳۱۶۷



طراحی صفحات وب به زبان ساده
تألیف: نوراله کریم تبار احمد چالی
نوبت چاپ: چاپ اول
سال چاپ: بهار ۹۸
شمارگان: ۱۰۰
قیمت: ۴۵۰۰۰ تومان
نام چاپخانه و صحافی:
شابک: ۷-۳۰-۷۲۷۲-۶۰۰-۹۷۸

نشانی ناشر: بابل، چهارراه نواب، کاظم بیگی، جنب مسجد منصور کاظم بیگی، طبقه اول
طراح جلد: کانون آگهی و تبلیغات آبان (احمد فرجی)

تهران، خ اردیبهشت، نبش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

فهرست مطالب

بخش اول - آموزش HTML

- فصل اول - شروع کار با HTML ۸
- فصل دوم - تگ‌های Head و پیوند ۱۸
- فصل سوم - جدول، لیست، فریم، فرم و بلوک‌ها ۲۹
- فصل چهارم - پس‌زمینه، رنگ‌ها، فونت‌ها، چندرسانه‌ای و طرح‌بندی ۴۰

بخش دوم - آموزش CSS

- فصل پنجم - نحوه نوشتن دستورات CSS ۵۵
- فصل ششم - رنگ‌ها، معیار اندازه‌گیری و فونت‌ها ۶۰
- فصل هفتم - پس‌زمینه، مرز و حاشیه ۶۶
- فصل هشتم - لینک، لیست و جدول ۷۸
- فصل نهم - چیدمان عناصر ۸۷
- فصل دهم - خواصی دیگر در CSS ۱۰۱

بخش سوم - آموزش JavaScript

- فصل یازدهم - شروع کار با جاوا اسکریپت ۱۰۷
- فصل دوازدهم - توابع ۱۱۶
- فصل سیزدهم - کار با مقادیر رشته‌ای، عددی و تاریخ ۱۲۱
- فصل چهاردهم - کار با آرایه، دستورات مقایسه‌ای و دستورات تکرار ۱۳۲
- فصل پانزدهم - مدیریت خطا و اعتبارسنجی فرم ۱۴۸

بخش چهارم - آموزش ASP.NET

- فصل شانزدهم - ساخت پروژه ASP.NET و آشنایی با ابزارهای ASP.NET ۱۵۹
- فصل هفدهم - مدیریت صفحات و اعتبارسنجی در ASP.NET ۱۷۴
- فصل هجدهم - دستورات و متدهای مهم در برنامه‌نویسی تحت وب سی شارپ ۱۸۲
- فصل نوزدهم - کار با پایگاه داده در ASP.NET ۱۸۸
- فصل بیستم - Ajax در ASP.NET ۲۰۵

بخش پنجم - نمونه‌ای از انجام یک پروژه وبسایت

- فصل بیست و یکم - گام‌های طراحی یک وبسایت ۲۱۲
- منابع ۲۲۸

مقدمه

از آنجائی که صفحات وب قابل اجرا در تمام پلتفرم‌های سخت‌افزاری و نرم‌افزاری می‌باشد، برنامه‌نویسی وب می‌تواند بسیار جذاب و فراگیر باشد. کتاب پیش‌رو، به زبان ساده و گویا و با بهره‌گیری از مثال‌های متنوع، سه زبان تحت وب شامل HTML، CSS و JavaScript و همچنین چارچوب توسعه تحت وب ASP را پوشش می‌دهد به طوری که نیاز به هیچ پیش‌زمینه‌ی برنامه‌نویسی نمی‌باشد.

بخش اول این کتاب شامل آموزش زبان HTML می‌باشد که اسکلت‌بندی یک صفحه وب را طراحی می‌کند. در بخش دوم زبان CSS که برای سبک دادن به عناصر HTML است، به طور کامل آموزش داده شده است. بخش سوم با آموزش JavaScript همراه است، زبانی که جز زبان‌های اسکریپتی سمت کاربر می‌باشد. در بخش چهارم با آموزش چارچوب توسعه تحت وب ASP با زبان C# (سی شارپ) و ارتباط آن با پایگاه داده SQL Server، نحوه ایجاد سایت پویا، آموزش داده شده است. در انتهای بخش چهارم فناوری AJAX و ابزارهای مربوط به آن معرفی شده است. برای تمام مطالب این کتاب، مثالی ارائه شده است و برای اکثر مثال‌ها، خروجی مربوط به آن مثال به صورت تصویری قرار داده شده است. در نهایت در بخش پنجم این کتاب، گام به گام مراحل ایجاد یک نمونه سایت پویا، شرح داده شده است.

تألیف کتاب دو سال به طول انجامید و تمام تلاش بنده این بوده است تا با زبانی ساده و گویا، طراحی صفحات وب آموزش داده شود. این کتاب مناسب برای تمام دانشجویان رشته کامپیوتر و فناوری اطلاعات و همچنین تمام عزیزانی است که مشتاق به یادگیری طراحی صفحات وب هستند. در پایان از تمام دوستان و عزیزانی که این کتاب را مطالعه کرده‌اند درخواست دارم انتقادات و پیشنهادات سازنده خود را در جهت بهبود این کتاب از طریق پست الکترونیکی بنده دریغ نفرمایند.

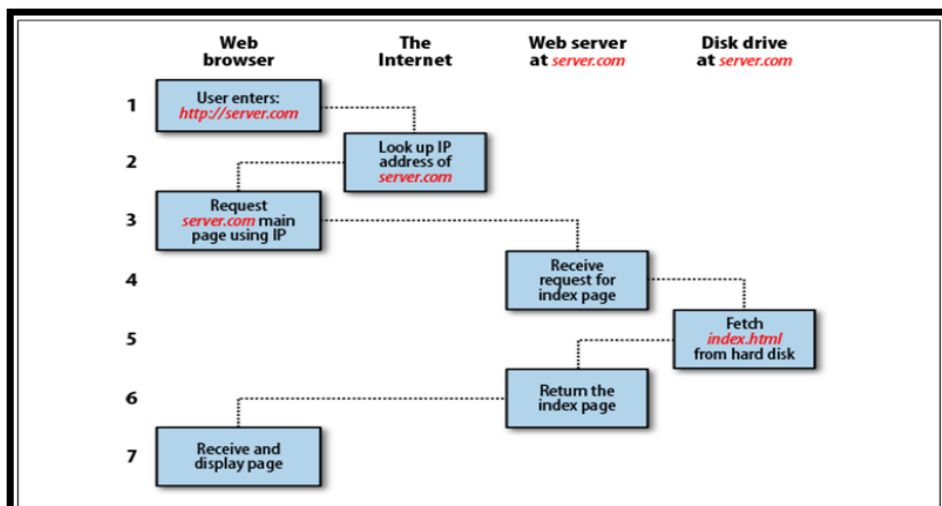
آدرس پست الکترونیکی:

saeidkarim.kk66@gmail.com

آموزش HTML

مقدمه

وقتی می‌خواهید یک سایتی را باز کنید، باید از طریق مرورگر، درخواست خود را به یک فراهم‌کننده سرویس اینترنت^۱ ارسال کنید. برای این منظور، نام دامنه یا آدرس وب‌سایت را در مرورگر تایپ کرده و اجرا می‌کنید. سپس طی انجام یکسری عملیات، مرورگر به سرویس‌دهنده وب^۲ متصل می‌شود. در نهایت سرویس‌دهنده وب، صفحه وب مربوطه را به مرورگر برمی‌گرداند. مراحل اجراشدن یک صفحه وب در شکل ۱-۱ نشان داده شده است.



شکل ۱-۱: مراحل اجرای یک صفحه وب (ترتیب درخواست یا پاسخ)

به‌منظور ایجاد یک صفحه وب، از زبان‌های برنامه‌نویسی مبتنی بر وب استفاده می‌شود. زبان‌های برنامه‌نویسی وب به‌طور کلی به دودسته زبان‌های برنامه‌نویسی سمت کاربر^۳ و زبان‌های برنامه‌نویسی سمت سرویس‌دهنده^۴ تقسیم می‌شوند.

1- Internet Service Provider یا ISP

2- Web Service

۳- Client-Side

۴ - Server-Side

در زبان‌های سمت کاربر، عملیات پردازش یک عمل خاص در سیستم کاربر انجام می‌شود اما در زبان‌های سمت سرورس‌دهنده، درخواست اجرای یک عمل خاص سمت سرورس‌دهنده ارسال می‌شود و در آنجا پردازش شده و پاسخ به سمت کاربر برگشت داده می‌شود. زبان HTML، به همراه زبان CSS و جاوا اسکریپت جز زبان‌های سمت کاربر به شمار می‌آیند.

پایه و اساس صفحات وب زبان تحت وب HTML می‌باشد. HTML یک زبان نمادگذاری است و نحوه نمایش صفحات وب با استفاده از آن مشخص می‌شود. HTML عمل نمادگذاری را با استفاده از تگ‌های^۱ مختلف انجام می‌دهد. باید توجه داشت که دستورات HTML با نام تگ شناخته می‌شوند. اولین نسخه زبان HTML در سال ۱۹۹۱ عرضه شده و با گذشت زمان نسخه‌های متعددی از این زبان ارائه شده است. جدیدترین آن در سال ۲۰۱۲ با نام HTML5 به توسعه‌دهندگان وب عرضه شده است. در نسخه ۵ امکانات زیر اضافه شده است:

- اضافه شدن تگ‌هایی مانند <header>، <footer>، <article> و <section>.
- اضافه شدن صفاتی مانند date، time، number و range.
- اضافه شدن دستورات مربوط به کار با گرافیک مانند <svg> و <canvas>.
- اضافه شدن دستورات مربوط به مدیریت چندرسانه‌ای^۲ مانند <audio> و <video>.

نرم افزارهای برنامه‌نویسی تحت وب

قبل از آنکه به شروع آموزش برنامه‌نویسی وب پردازیم، محیط‌های مناسب برای نوشتن کدهای تحت وب معرفی می‌شود.

- استفاده از یک ویرایشگر ساده مانند Notepad: برنامه‌های تحت وب را می‌تواند در ویرایشگرهای ساده از جمله Notepad نوشت و سپس با پسوند html یا htm (یا js، css و...) ذخیره کرد.
- نرم‌افزار dreamweaver: یکی از نرم‌افزارهای قدرتمند جهت طراحی وب می‌باشد و توسط شرکت Adobe انتشار داده شده است و دارای امکانات مناسب به منظور تسهیل نمودن برنامه‌نویسی تحت وب می‌باشد. این نرم‌افزار از HTML5، CSS3، ASP.NET و PHP پشتیبانی می‌کند.
- نرم‌افزار Visual Studio: یکی از نرم‌افزارهای حرفه‌ای شرکت مایکروسافت می‌باشد و علاوه بر وجود امکاناتی برای برنامه‌نویسی تحت وب، شامل محیط‌های برنامه‌نویسی سی شارپ (C#)، ++C، و visual basic و چارچوب توسعه تحت وب ASP می‌باشد.

^۱ -Tag

^۲ -Multimedia

آموزش HTML ۷

به دلیل آنکه در بخش چهارم این کتاب، برنامه‌نویسی ASP.NET گفته خواهد شد، تمام برنامه‌های این کتاب در محیط نرم‌افزار Visual Studio نوشته خواهد شد.

فصل
۱

شروع کار با HTML

همان طور که گفته شد، HTML یک زبان نماد گذاری می باشد و دارای ویژگی های زیر می باشد:

- HTML یک زبان برنامه نویسی نیست بلکه یک زبان نماد گذاری است.
- زبان های نماد گذاری شامل مجموعه ای از تگ های نماد گذاری می باشند
- HTML از تگ های نماد گذاری برای نمایش محتویات صفحه وب استفاده می کند.
- یک فایل HTML دارای پسوند htm یا html می باشد.

۱-۱ تگ ها در HTML

نحوه نوشتن یک تگ به صورت < نام تگ > می باشد. به جز تعدادی کمی از تگ ها، بیشتر تگ ها شامل یک تگ باز و یک تگ بسته می باشند. به عنوان مثال، تگ <html> دارای تگ بسته ی </html> می باشد. به این تگ ها اصطلاحاً تگ container گفته می شود.

📄 مثال : برنامه ای که نحوه نوشتن یک کد ساده HTML و استفاده از تگ ها را نشان می دهد.

```
<!DOCTYPE html>
<html>
<body>
<h1>Heading</h1>
<p>paragraph.</p>
</body>
</html>
```

📄 خروجی

Heading

paragraph.

به طور کلی ساختار یک برنامه HTML در شکل ۱-۲ نشان داده شده است.



شکل ۱-۲ ساختار کلی یک برنامه HTML

برنامه HTML با تگ <html> شروع می‌شود و با </html> پایان می‌یابد. بعد از تگ <html>، تگ <head> قرار می‌گیرد. این قسمت که بین تگ <html> و تگ <body> قرار می‌گیرد، حاوی آبر داده^۱ می‌باشد. آبر داده شامل اطلاعاتی در مورد سند HTML می‌باشد که در بدنه اصلی صفحه وب نمایش داده نمی‌شود. آبر داده معمولاً شامل عنوان صفحه، مجموعه کاراکترها، سبک‌ها^۲، پیوندها^۳، اسکریپت‌ها و سایر اطلاعات دیگر می‌باشد.

قسمت بعدی یک برنامه HTML، قسمت مربوط به بدنه اصلی برنامه می‌باشد که با تگ <body> شروع و با تگ </body> پایان می‌یابد. در این قسمت تگ‌هایی قرار می‌گیرد که در صفحه مربوط به HTML نشان داده می‌شود. در ادامه به معرفی بعضی از این تگ‌ها خواهیم پرداخت.

تگ <!DOCTYPE html>

دستور DOCTYPE یک نوع راهنما برای مرورگرهای وب است تا باخیر شوند که نسخه استفاده شده در سند HTML چه می‌باشد. نسخه جاری HTML، نسخه ۵ می‌باشد که به صورت <!DOCTYPE html> نوشته می‌شود. برای استفاده از نسخه‌های دیگر HTML از کدهای زیر استفاده می‌شود:

- HTML 4.01:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

```

- XHTML 1.0:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```


¹ -Metadata

² -Styles


³ -Links

تگ Heading

هر سند با یک عنوانی آغاز می‌شود. برای عنوان می‌توان از اندازه‌های مختلفی استفاده کرد. در HTML شش سطح عنوان وجود دارد که شامل <h1> تا <h6> می‌باشد.

 مثال: برنامه‌ای که نحوه نمایش تگ‌های Heading را نشان می‌دهد. اندازه نوشته‌های این تگ‌ها به ترتیب از <h1> تا <h6> کاهش می‌یابد.

```
<!DOCTYPE html>
<html>
<head>
<title>انواع تیترها</title>
</head>
<body>
<h1>تیتر اول</h1>
<h2> تیتر دوم</h2>
<h3> تیتر سوم</h3>
<h4> تیتر چهارم</h4>
<h5> تیتر پنجم</h5>
<h6> تیتر ششم</h6>
</body>
</html>
```

 خروجی

تیتر اول

تیتر دوم

تیتر سوم

تیتر چهارم

تیتر پنجم


تیتر ششم

تگ <p>

این تگ برای ایجاد یک پاراگراف می‌باشد و مانند تگ Heading، باعث ایجاد یک خط جدید می‌شود.

مثال : برنامه‌ای که شامل چند پاراگراف می‌باشد.

```
<!DOCTYPE html>
<html>
<head>
<title>پاراگراف</title>
</head>
<body>
<p>اولین پاراگراف</p>
<p>دومین پاراگراف</p>
<p>سلام<br/>
متشکرم<br/>
خدانگهدار</p>
</body>
</html>
```

خروجی 

اولین پاراگراف
دومین پاراگراف
سلام
متشکرم
خدانگهدار

تگ

مثال قبل از تگ
 استفاده شده است. این تگ برای ایجاد یک خط جدید استفاده می‌شود. تگ فوق نمونه‌ای از یک تگ تکی یا خالی است و نیازی به تگ باز و بسته ندارد و همچنین دستوری در داخل تگ نوشته نمی‌شود.

تگ <center>

به منظور وسط چین کردن عناصر از تگ <center> استفاده می‌شود.

تگ <hr/>

این تگ برای ایجاد خط افقی است. این تگ نیز مانند تگ
 جز تگ‌های تکی می‌باشد.

مثال : برنامه‌ای که پاراگرافی را وسط چین کرده و همچنین بین دو پاراگراف را با خط افقی (تگ <hr>) جدا می‌کند.

```
<!DOCTYPE html>
<html>
<body>
<center>
<p>پاراگراف وسط چین</p>
```

```

</center>
<hr />
<p>پاراگراف پایین</p>
</body>
</html>

```

تگ‌هایی که در قسمت <head> نوشته می‌شود شامل <title>، <style>، <meta>، <link>، <script>، <noscript> و <base> می‌باشد.

تگ <pre> (حفظ قالب‌بندی)

اگر در کدهای HTML از کلیدهایی چون enter یا space به منظور قالب‌بندی متون استفاده کنید تأثیری در خروجی نخواهد داشت و متون پشت سرهم قرار می‌گیرند. اما اگر بخواهید شکل قرارگیری نوشته‌ها به همان صورتی باشد که در کد برنامه است، باید از تگ <pre> استفاده کنید. همچنین برای ایجاد یک فضای خالی از کد استفاده می‌شود.

مثال: برنامه‌ای که شامل دو قسمت است. در قسمت اول قالب‌بندی متون حفظ می‌شود ولی در قسمت دوم قالب‌بندی متون حفظ نمی‌شود.

```

<!DOCTYPE html>
<html>
<head>
</head>
<body>
<pre>
function testFunction( strText ){
alert (strText)
}
</pre>
<p>
function testFunction( strText ){
alert (strText)
}
</p>
<p>"12&nbsp;Angry&nbsp;Men."</p>
</body>
</html>

```

خروجی 

```
function testFunction( strText ){
alert (strText)
}

function testFunction( strText ){ alert (strText) }

"12 Angry Men."
```

۲-۱ تگ‌های فرمت بندی

جدول ۱-۱ لیست تگ‌های مربوط به فرمت بندی نوشته‌ها را معرفی و توضیح می‌دهد.

توضیح	تگ
برای توپر کردن نوشته‌ها بکار می‌رود. تگ مشابه با آن می‌باشد	
برای مورب کردن نوشته‌ها استفاده می‌شود. تگ‌های مشابه با این عمل و <cite> می‌باشد	<i>
برای کوچک‌تر کردن متن به اندازه یک واحد استفاده می‌شود	<small>
برای بزرگ‌تر کردن متن به اندازه یک واحد استفاده می‌شود	<big>
برای زیرنویس دار (اندیس) کردن متن بکار می‌رود.	<sub>
برای بالانویس (توان) کردن استفاده می‌شود.	<sup>
برای زیرخط دار کردن متن استفاده می‌شود. تگ معادل آن <u> می‌باشد.	<ins>
باعث خط کشیدن روی متن می‌شود.	
برای های لایت کردن متن به کار می‌رود.	<mark>
برای تعیین جهت قرارگیری قسمتی از نوشته‌ها بکار می‌رود. این تگ همراه با دستور dir بکار می‌رود. dir دو مقدار ltr (برای نوشتن چپ به راست) و rtl (برای نوشتن راست به چپ) می‌پذیرد.	<bdo dir=" " >
برای توپر کردن نوشته‌ها بکار می‌رود. تگ مشابه با آن می‌باشد	<big>

مثال: برنامه‌ای که استفاده از تگ‌های معرفی شده در جدول ۱-۱ را نشان می‌دهد.

```
<!DOCTYPE html>
<html>
<head>
<title>فرمت بندی متن</title>
</head>
<body>
```

```

<p>The following word uses a <b>bold</b> typeface.</p>
<p>The following word uses a <i>italicized</i> typeface.</p>
<p>The following word uses a <u>underlined</u> typeface.</p>
<p>The following word uses a <sup>superscript</sup> typeface.</p>
<p>The following word uses a <sub>subscript</sub> typeface.</p>
<p>I want to drink <del>cola</del> <ins>wine</ins></p>
<p>The following word uses a <small>small</small> typeface.</p>
<p>The following word uses a <big>big</big> typeface.</p>
<p>The following word has been <mark>marked</mark> with yellow</p>
</body>
</html>

```

عناصر HTML

هر چیزی که بین ابتدا و انتهای یک تگ قرار می‌گیرد، عناصر گفته می‌شود. البته وجود عناصر مختص به تگ‌های باز و بسته است. به‌عنوان مثال، مقادیر یا تگ‌هایی که در داخل یک <p> قرار می‌گیرند، عناصر مربوط به این تگ می‌باشد.

نکته: در بیشتر موارد تگ‌های به‌صورت تودرتو نوشته می‌شوند.

توضیحات^۱ در HTML

توضیح شامل قطعه‌ای از کد است که توسط مرورگر اجرا نمی‌شود. نوشتن توضیح باعث افزایش قابلیت خوانایی برنامه می‌شود. توضیحات در HTML به‌صورت زیر نوشته می‌شوند:

```
<!--Comments Text -->
```

و همچنین نحوه نوشتن چند خط توضیحات به‌صورت زیر است:

```
<!--
```

```
توضیحات چندخطی
```

```
-->
```

۱-۳ تفاوت بین HTML و XHTML

زبان‌های HTML و XHTML^۲ هر دو جز زبان‌های وب به‌شمار می‌آیند و تفاوت آن‌ها در میزان دقت آن‌ها است. XHTML نسبت به HTML دقیق‌تر است. به‌عنوان مثال، در زبان XHTML تگ‌ها و صفات حتماً می‌بایست با حروف کوچک نوشته شوند درحالی‌که در زبان HTML این‌طور نیست. همچنین در XHTML مقادیر صفات باید حتماً در داخل دابل کوتیشن (" ") قرار گیرند. درحالی‌که این کار در HTML اختیاری است.

کد HTML زیر باوجوداینکه قوانین برنامه HTML را رعایت نکرده است، اما در بیشتر مرورگرها اجرا می‌شود:

^۱ -Comment

^۲ -Extensible HyperText Markup Language

```
<html>
<head>
  <title>This is bad HTML</title>
<body>
  <h1>Bad HTML
  <p>This is a paragraph
</body>
```

امروزه مرورگرهای مختلفی وجود دارند. بعضی از این مرورگرها بر روی کامپیوتر اجرا می‌شوند و بعضی‌ها روی تلفن‌های همراه اجرا می‌شوند. مرورگرهایی که روی سیستم‌های کوچک‌تر از جمله تلفن‌های همراه اجرا می‌شوند، فاقد قدرت تفسیر کردن نشانه‌گذاری بد (مانند کد بالا) هستند. بنابراین در این نوع مرورگرها حتماً باید بر اساس قوانین XHTML برنامه وب نوشت چراکه در این زبان نشانه‌گذاری به‌درستی انجام می‌شود. در XHTML قرار دادن تگ‌های <html>، <head>، <title> و <body> اجباری است. همچنین قرار دادن دستور <!DOCTYPE...> در این زبان اجباری است.

تفاوت دیگر این است که در XHTML خاصیت تودرتویی تگ‌ها باید رعایت شود. کد زیر در HTML مشکلی به وجود نمی‌آورد اما در XHTML این عمل اشتباه است:

```
<b><i>This text is bold and italic</i></b></i>
```

در XHTML تگ باز و بسته حتماً باید نوشته شود اما در HTML قرار ندادن تگ بسته مشکلی به وجود نمی‌آورد. به‌عنوان مثال، کد زیر در HTML قابل اجرا است:

```
<p>This is a paragraph
<p>This is another paragraph
```

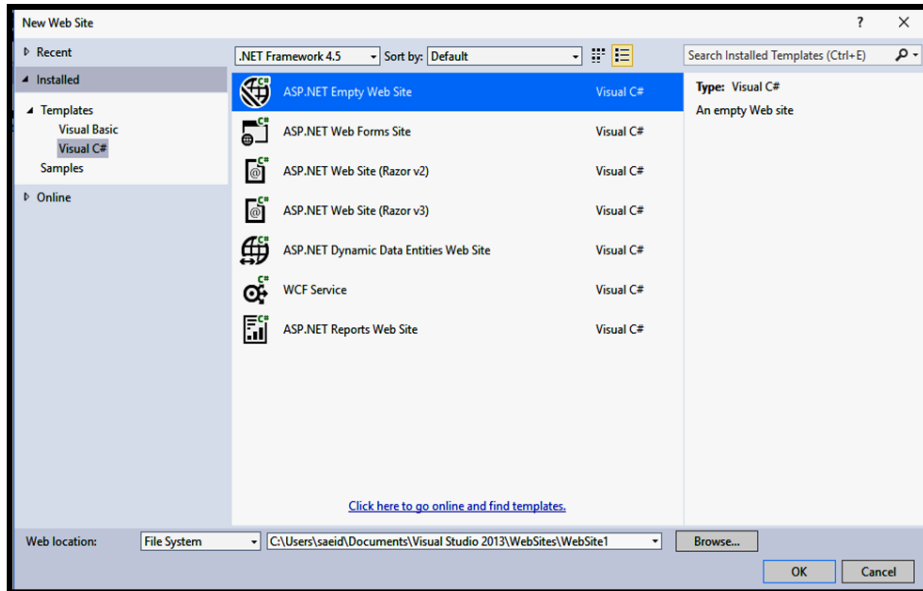
تگ‌های تکی در XHTML حتماً باید به فرم تگ بسته نوشته شود. مثال زیر در XHTML اشتباه است اما در HTML مشکلی به وجود نمی‌آید:

```
A break: <br>
```

```
A horizontal rule: <hr>
```

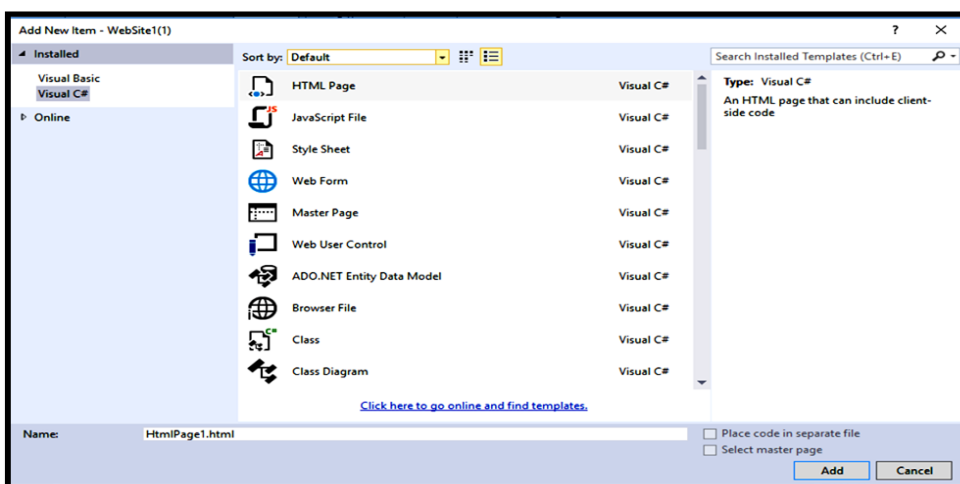
۴-۱ نحوه اجرای یک پروژه تحت وب در Visual Studio

در این کتاب قصد داریم طراحی وب را در محیط برنامه‌نویسی Visual Studio 2013 نشان دهیم. برای ایجاد یک پروژه‌ای از نوع وب، از منوی فایل گزینه‌ی Web Site را انتخاب کنید. با انتخاب این گزینه پنجره‌ی New Web Site باز می‌شود. شکل ۱-۳ این پنجره را نشان می‌دهد.



شکل ۱-۳ پنجره مربوط به ایجاد پروژه وب در ویژوال استادیو ۲۰۱۳

در این پنجره گزینه ASP.NET Empty Web Site را انتخاب کرده و با مشخص کردن مسیر ذخیره کردن فایل، دکمه OK را انتخاب کنید. بعد از ایجاد پروژه، در شروع کار از طریق منوی Web Site گزینه Add New Item را انتخاب کرده و سپس همان‌طور که در شکل ۱-۴ مشاهده می‌کنید گزینه HTML Page را انتخاب کنید. سپس نام صفحه‌ای html خود را در قسمت name بنویسید و سپس بر روی add کلیک کنید.



شکل ۱-۴ ایجاد یک فایل HTML

آموزش HTML ۱۷

با انجام این کار، فایل HTML ایجاد شده و همه چیز آماده است برای نوشتن یک برنامه HTML و اجرا کردن این برنامه. شکل ۱-۵ نمونه‌ای ساده از یک برنامه html در محیط برنامه‌نویسی ویژوال استادیو ۲۰۱۳ را



```
HtmlPage.html*  X
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<p>Hello HTML</p>
<p> Learning HTML</p>
</body>
</html>
```

نشان می‌دهد.

شکل ۱-۵ نمونه‌ای از برنامه html در ویژوال استادیو ۲۰۱۳

برای اجرای برنامه html نوشته‌شده می‌توان در سمت چپ برنامه و در پنجره Solution Explorer روی فایل html موردنظر کلیک راست کرده و گزینه‌ی View In Browser و یا دکمه f5 را انتخاب کنید.

تگ‌های head و پیوند

فصل ۲

۱-۲ صفات در HTML

صفات، ویژگی‌های یک عنصر HTML را مشخص کرده و داخل تگ باز عناصر قرار می‌گیرند. همه صفات از دو قسمت نام و مقدار تشکیل می‌شوند. به‌عنوان مثال، تگ `<p>` دارای صفتی به نام `align` می‌باشد که ترازبندی مربوط به پاراگراف را مشخص می‌کند. مقادیری که برای این صفت قابل قبول است شامل `Left`، `Center` و `Right` می‌باشد.

مثال : برنامه‌ای که شامل سه پاراگراف می‌باشد و هر کدام از این پاراگراف‌ها دارای مقادیر مختلف از صفت `align` است.

```
<!DOCTYPE html>
<html>
<head>
<title>ترازبندی عناصر</title>
</head>
<body>
<p align="left">This is left aligned</p>
<p align="center">This is center aligned</p>
<p align="right">This is right aligned</p>
</body>
</html>
```

خروجی

This is left aligned

This is center aligned

This is right aligned

چهار صفتی که در اکثر تگ‌ها وجود دارد شامل موارد زیر است:

- ✓ **Id:** برای اختصاص نام شناسایی منحصر به فرد به یک عنصر در صفحه HTML بکار می‌رود. در قسمت آموزش CSS در بخش دوم از این صفت به‌طور گسترده استفاده می‌شود.
- ✓ **Title:** عنوان یک عنصر را مشخص کرده و با قرارگیری اشاره‌گر ماوس رو عنصر مورد نظر، متن قرار گرفته شده در این صفت را مشخص می‌کند.
- ✓ **Class:** این صفت تا حدودی شبیه به صفت `Id` می‌باشد با این تفاوت که می‌توان یک نام مشخص را به چند عنصر اختصاص داد. در بخش CSS از `class` زیاد استفاده می‌شود.

✓ Style: این صفت به طراح وب اجازه می‌دهد تا دستورات CSS را در داخل تگ HTML قرار دهد.

📌 مثال: برنامه‌ای که از صفات id و title و style استفاده می‌کند

```
<!DOCTYPE html>
<html>
<head>
<title>صفحات رایج</title>
</head>
<body>
<p id="css">This para explains what is Cascading Style Sheet</p>
<h3 title="Hello HTML!">Titled Heading Tag Example</h3>
<p style="font-family:arial; color:#FF0000;">Some text...</p>
</body>
</html>
```

معرفی صفاتی دیگر

- ✓ در ادامه با صفات مهم‌تری که در اکثر تگ‌ها استفاده می‌شوند آشنا می‌شوید.
- ✓ صفت dir: این تگ برای تغییر مسیر نوشتن (چپ به راست یا راست به چپ) در کل صفحه HTML یا قسمتی از صفحه HTML به کار می‌رود. مقادیر این صفت شامل موارد زیر است:
- ✓ltr: برای چپ به راست کردن مسیر نوشتن به کار می‌رود
- ✓rtl: برای راست به چپ کردن مسیر نوشتن به کار می‌رود
- ✓ صفات عمومی دیگر به همراه توضیحات در جدول ۲-۱ نشان داده شده است.

جدول ۲-۱ معرفی بعضی صفات عمومی

توضیحات	گزینه‌ها	صفت
ترازبندی افقی	right, left, center	Align
ترازبندی عمودی (یا بالا به پایین)	top, middle, bottom	Valign
رنگ زمینه یک عنصر	نام رنگ، به صورت RGB	Bgcolor
قرارگیری یک عکس در زمینه یک عنصر	آدرس یک عکس	Background
پهنای مربوط به یک عکس، جدول، یک خانه از جدول و...	مقدار عددی	Width
ارتفاع مربوط به یک عکس، جدول، یک خانه از جدول و...	مقدار عددی	Height

۲-۲ معرفی تگ‌های head

تمام تگ‌هایی که در قسمت head قرار می‌گیرد به همراه توضیحات در ادامه بیان شده است.

تگ Title

این تگ متن عنوان یک سند html را مشخص می‌کند. متنی که در title قرار می‌گیرد همان متنی است که در قسمت سایت‌های موردعلاقه (Bookmark یا favorite) مربوط به مرورگرها ذخیره می‌شود. همچنین این متن نمایش عنوانی است که در نتایج موتورهای جستجوگر قرار می‌گیرد.

تگ Meta

مفهوم Metadata به معنای اطلاعاتی در مورد یک داده یا سند است. تگ meta اطلاعات مربوط به سند جاری html را مشخص می کند که می تواند توسط موتورهای جستجوگر مورد استفاده قرار گیرد. این تگ چیزی در خروجی نشان نمی دهد و معمولاً شامل اطلاعاتی از قبیل مشخصات کاربر، مشخصات سند، کلمات کلیدی و... است. در جدول ۲-۲ صفات مختلف مربوط به تگ meta را بررسی کنید.

جدول ۲-۲ صفات مربوط به تگ meta	
معرفی صفت به همراه مثال	نوع عملیات
<code><meta name="keywords" content="HTML, Meta Tags, Metadata" /></code>	تعیین کلمات کلیدی صفحه و مورد استفاده در موتورهای جستجوگر
<code><meta name="description" content="Learning about Meta Tags." /></code>	توصیف کوتاه از صفحه و مورد استفاده در موتورهای جستجوگر
<code><meta name="revised" content="HTML learning, 3/7/2014" /></code>	تعیین زمان به روزرسانی صفحه که توسط مرورگرها مورد استفاده قرار می گیرد
<code><meta http-equiv="refresh" content="7" /></code>	Refresh کردن صفحه در یک زمان خاص
<code><meta http-equiv="refresh" content="7; HTMLlearning.com" /></code>	هدایت کردن صفحه به یک صفحه ای دیگر در یک زمان خاص
<code><meta ame="author" content="saeid karimtabar" /></code>	تخصیص دادن نام نویسنده به صفحه
<code><meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /></code>	تعیین مجموعه کاراکترهای مورد استفاده در صفحه (برای نمایش کاراکترهای فارسی از دستور content="text/html; charset=UTF-8" استفاده می شود).

همان طور که در جدول ۲-۲ اشاره شد، برای نمایش کاراکترهای فارسی از دستور content="text/html; charset=UTF-8" استفاده می شود.

یکی از دستورات مهم در تگ meta، دستور مربوط به تنظیمات کوکی ها^۱ می باشد. کوکی ها داده هایی هستند که به صورت فایل های کوچک متنی در کامپیوتر شما ذخیره می شوند و به منظور ره گیری اطلاعات مختلف به منظور هدفی خاص بین مرورگر وب و سرویس دهنده وب ردوبدل می شوند. توسط تگ meta می توان کوکی ها را در سمت کاربر ذخیره کرد و در آینده، این اطلاعات می تواند توسط سرویس دهنده وب مورد استفاده قرار گیرد.

¹ -Cookies

در مثال زیر یک کوکی شامل شماره شناسایی (id) با مقدار xyz، در یک مدت خاصی که در قسمت expires مشخص شده است در کوکی مربوط به مرورگر ذخیره می‌شود. یعنی اینکه اگر در زمان‌های آینده دوباره به این صفحه مراجعه کنید شماره شناسایی ذخیره شده است.

```
<meta http-equiv="cookie" content="userid=xyz; expires=Wednesday,
08-Aug-15 23:59:59 GMT;" />
```

اگر دستور expires را حذف کنید، شماره شناسایی با بستن مرورگر از بین خواهد رفت.

تگ base

این تگ قسمت شروع یک آدرس یا URL را مشخص می‌کند. به‌عنوان مثال، فرض کنید مسیر کامل مربوط به عکس‌های سایت در مسیر `http://www.HTMLlearning.com/images/` قرار دارد و در پوشه image، عکس‌ها ذخیره شده است. حال می‌توان با استفاده از تگ base، آدرس شروع را با این مقدار مشخص کرد و در زمان فراخوانی یک عکس، دیگر نیازی نباشد تا مسیر کامل نوشته شود و فقط نام عکس نوشته می‌شود.

📄 مثال: برنامه‌ای که با کمک تگ Base، یک آدرس شروع تصاویر را تعیین می‌کند.

```
<!DOCTYPE html>
<html>
<head>
  <base href="http://www.HTMLlearning.com/">
</head>
<body>

<a href="/html/index.htm" title="HTML Tutorial">HTML Tutorial</a>
</body>
</html>
```

تگ Link

این تگ یک سند جاری را به یک منبع خارجی وصل می‌کند. در کد html زیر یک صفحه وب به یک فایل CSS متصل می‌شود.

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

تگ Style

این تگ برای تعریف یک سبک برای یک صفحه HTML بکار می‌رود.

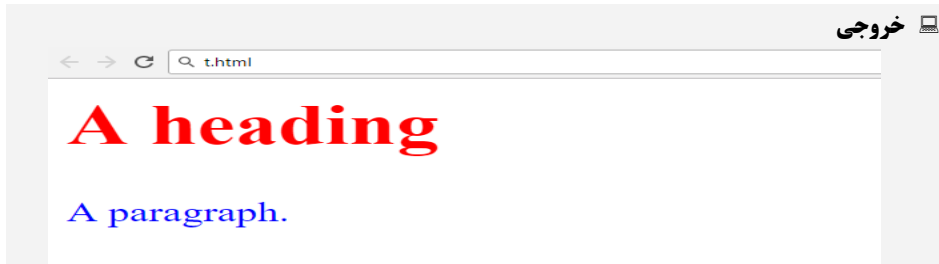
📄 مثال: برنامه‌ای که با کمک تگ style در قسمت head، یک کد ساده روی تگ‌ها اعمال می‌کند.

```
<html>
<head>
<style>
h1 {color : red;}
p {color : blue;}
</style>
```

```

</head>
<body>
<h1>A heading</h1>
<p>A paragraph.</p>
</body>
</html>

```



تگ Script

از این تگ برای تعریف اسکریپت‌های سمت کاربر مانند جاوا اسکریپت استفاده می‌شود.

مثال: برنامه‌ای که با فشردن دکمه، پیغامی را نمایش می‌دهد. این پیغام با استفاده از دستور Alert که مربوط به دستورات JavaScript می‌باشد، نمایش داده می‌شود.

```

<!DOCTYPE html>
<html>
<head>
<title>HTML script Tag Example</title>
<base href="http://www.HTMLlearning.com/" />
<script type="text/JavaScript">
function Hello()
{
alert("Hello, World");
}
</script>
</head>
<body>
<input type="button" onclick="Hello();" name="ok" value="OK" />
</body>
</html>

```

۲-۳ تصاویر در HTML

تصاویر در طراحی صفحات وب بسیار مهم بوده و باعث زیباتر شدن صفحه وب می‌شوند. برای نمایش

یک تصویر خاص در HTML از تگ استفاده می‌شود. شکل کلی این تگ به صورت زیر است:

```

```

همان‌طور که مشاهده می‌کنید، تگ به صورت تکی می‌باشد. در صفت SRC، آدرس عکس

موردنظر مشخص می‌شود.

مثال: برنامه‌ای که یک تصویر را با کمک تگ `` نمایش می‌دهد.

```
<!DOCTYPE html>
<html>
<head>
<title>Using Image in Webpage</title>
</head>
<body>
<p>Simple Image Insert</p>

</body>
</html>
```

پسوند تصاویر در محیط وب می‌توانند PNG، JPEG و GIF باشد. متنی که در صفت alt قرار می‌گیرد زمانی نشان داده می‌شود که تصویر موردنظر به هر دلیلی نشان داده نشود. یعنی به جای عکس، متن داخل alt نوشته می‌شود.

آدرس‌دهی به عکس دارای روش‌های مختلفی است. در مثال بالا فرض کرده‌ایم که عکس در همان جایی قرار دارد که فایل وب قرار دارد. یعنی هر دو در یک مسیر قرار دارند. اما اگر تصویر موردنظر در پوشه‌ی دیگری قرار داشته باشد، آنگاه مسیر موردنظر باید نوشته شود. به عنوان مثال، فرض کنید تصویر در درایو D و در پوشه‌های folder1 و folder2 (به ترتیب) قرار دارد. آنگاه تگ `img` به صورت زیر نوشته می‌شود:

```

```

حال فرض کنید که تصویر موردنظر یک پوشه عقب‌تر از محلی که فایل وب در آنجا ذخیره شده است،

قرار دارد. در این حالت به صورت زیر آدرس‌دهی را انجام دهید:

```

```

همچنین ممکن است تصویر موردنظر در یک فضای دامین باشد، آنگاه به صورت زیر آدرس‌دهی انجام

می‌شود:

```

```

پهنا، ارتفاع و مرز و ترازبندی تصویر

برای تنظیم کردن پهنا و ارتفاع تصویر از صفت `width` و `height` در تگ `` استفاده می‌شود. مقادیر این صفات می‌تواند به صورت پیکسل یا درصد باشد. همچنین برای تعیین میزان ضخامت مرز تصویر از صفت `Border` و برای ترازبندی افقی از صفت `align` استفاده می‌شود.

مثال: برنامه‌ای که مقادیر خواص مربوط به تگ `` شامل `width`، `height`، `Border` و `align` را مشخص می‌کند.

```
<!DOCTYPE html>
<html>
<head>
<title>ترازبندی تصویر</title>
</head>
<body>
```

```
<p>Setting image Alignment</p>

</body>
</html>
```

۲-۴ پیوندها

یک صفحه وب می‌تواند حاوی انواع پیوندهایی باشد که باعث انتقال به صفحه دیگر و یا حتی پرش به قسمتی از همان صفحه شود. به این پیوندها، ابرپیوند^۱ می‌گویند. با استفاده از متن یا تصویر می‌توان یک پیوند ایجاد کرد. به‌طور کلی پیوندها دو نوع هستند: پیوند داخلی (پیوند به یک قسمتی از همان صفحه) و پیوند خارجی (پیوند به یک صفحه وب دیگر).

برای ایجاد پیوند از تگ <a> استفاده می‌شود. ساختار کلی این تگ به‌صورت زیر می‌باشد:

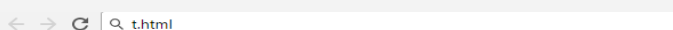
```
<a href="Document URL" ... attributes-list> Link Text </a>
```

در صفت href، آدرس مقصد نوشته می‌شود.

 مثال: برنامه‌ای که یک مثال ساده از ایجاد پیوند را نشان می‌دهد.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="http://www.HTMLlearning.com/test.html">Tutorials Point</a>
</body>
</html>
```

 خروجی

 t.html

Click following link

[Tutorials Point](http://www.HTMLlearning.com/test.html)

به‌طور پیش‌فرض بعد از ایجاد یک پیوند، متن پیوند به‌صورت زیر خط‌دار است و با خط آبی نشان داده می‌شود. در مثال بالا با کلیک بر روی پیوند موردنظر صفحه مربوط به آدرس صفت href باز می‌شود.

صفت Target در تگ <a>

این صفت نحوه باز شدن صفحه مقصد تعیین شده در تگ <a> را مشخص می‌کند. جدول ۲-۳ گزینه‌های مربوط به این صفت به همراه توضیحات را نشان می‌دهد.

¹ -Hyper Link

جدول ۲-۳ مقادیر مربوط به صفت Target	
گزینه	توضیحات
_blank	صفحه در یک پنجره یا زبانه (Tab) جدید باز می‌شود
_self	باز شدن صفحه در همان صفحه قبلی (یعنی صفحه جدیدی باز نمی‌شود)

البته این صفت دارای سه گزینه دیگر شامل _parent، _top و targetframe می‌باشد که در تگ <a> مانند گزینه _self عمل می‌کند اما در تگ <iframe> که در آینده به آن می‌پردازیم، کاربرد متفاوتی دارند.

مثال: برنامه‌ای که مقادیر مختلف صفات Target در تگ <a> را تعیین می‌کند.

```
<html>
<head>
<title>مثالی از لینک‌ها</title>
<base href="http://www.HTMLLearning.com/">
</head>
<body>
<p>Click any of the following links</p>
<a href="/html/index.htm" target="_blank">Opens in New</a>
<a href="/html/index.htm" target="_self">Opens in Self</a>
<a href="/html/index.htm" target="_parent">Opens in Parent</a>
<a href="/html/index.htm" target="_top">Opens in Body</a>
</body>
</html>
```

پیوند به بخشی از صفحه

پیوند به یک قسمت خاصی از صفحه که به آن پیوند داخلی می‌گوییم، دارای دو مرحله می‌باشد. در مرحله اول، آن قسمتی از صفحه که مقصد پیوند می‌باشد را باید با استفاده از تگ <a> و توسط صفت name، نام‌گذاری کنید. ساختار کلی آن به صورت زیر می‌باشد:

```
<h1>HTML Text Links <a name="top"></a></h1>
```

در مرحله بعد پیوند به قسمت موردنظر را به صورت زیر برقرار کنید:

```
<a id="tips">Useful Tips Section</a>
.....
<a href="#tips">Visit the Useful Tips Section</a>
```

یا

```
<a href="http://www.HTML.com/html_links.htm#tips"> Visit the Useful Tips Section</a>
```

همان‌طور که مشاهده می‌کنید، در پیوند داخلی، در ابتدای آدرس پیوند از علامت # قرار داده می‌شود. به منظور تغییر رنگ پیش فرض پیوندها می‌توان در <body> از صفات link (رنگ متن پیوند) و alink (رنگ پیوند زمانی که عمل کلیک روی پیوند انجام می‌شود) و vlink (لینک‌های انتخاب شده) استفاده کرد. اما باید در نظر داشت که صفت vlink در <body> در html5 شناخته شده نیست.

می‌توان پیوندی برای بارگیری^۱ کردن فایل‌های PDF، DOC یا ZIP ایجاد کرد. برای این کار آدرس پیوند برابر با آدرس مکان یکی از این فایل‌ها در نظر می‌گیریم.

مثال: برنامه‌ای که پیوند به یک صفحه‌ای دیگر و پیوند به یک فایل از نوع pdf را ایجاد می‌کند.

```
<!DOCTYPE html>
<html>
<head>
<title>مثالی از لینک‌ها</title>
<base href="http://www.HTMLLearning.com/">
</head>
<body alink="#54A250" link="#040404" vlink="#F40633">
<p>Click following link</p>
<a href="/html/index.htm" target="_blank" >HTML Tutorial</a>
<a href=" http://www.HTMLLearning.com/page.pdf"> Download PDF File</a>
</body>
</html>
```

ایجاد پیوند روی تصویر

برای آنکه بتوانیم به جای یک متن، یک عکس را پیوند بدهیم، فقط کافی است، به جای متن پیوند، یک تصویر قرار دهیم.

مثال: برنامه‌ای که بر روی یک تصویر پیوند ایجاد می‌کند.

```
<!DOCTYPE html>
<html>
<head>
<title> مثالی از لینک‌ها تصویر </title>
</head>
<body>
<p>Image-link: Still a link, but with no borders:
<a href="default.asp"></a></p>
</body>
</html>
```

تصاویر حساس به اشاره‌گر (پیوندهای مختلف به نقاط مختلف یک تصویر)

در html و XHTML دارای دستوراتی است که می‌توان، برای نواحی مختلف یک عکس پیوندهای مختلفی تعریف کرد. در ابتدا نواحی مختلف تعریف شده و سپس به هر ناحیه یک پیوند اختصاص می‌دهیم به طوری که با کلیک روی هر ناحیه، پیوند موردنظر برقرار می‌شود. به این نوع پیوند، تصویر نقشه نیز می‌گویند. با دو روش می‌توان تصویر نقشه ایجاد کرد: روش Server-side image maps و روش Client-side image maps.

¹ - Download

- روش Server-side image maps: در این روش از صفت ismap در تگ استفاده می‌شود و نیاز به دسترسی به سرورس‌دهنده و برنامه پردازش تصویر نقشه است. در اینجا به یک تصویر پیوند ایجاد و از صفت ismap استفاده می‌شود. وقتی کاربر روی قسمتی از تصویر کلیک کرد، مرورگر مختصات آن قسمت را به همراه آدرس پیوند مشخص شده در تگ <a>، به سرورس‌دهنده وب ارسال می‌کند و سرورس‌دهنده مشخص می‌کند که کدام سند بر اساس منطقه کلیک شده، باید باز شود. آدرس فایل مشخص شده در تگ <a> می‌تواند یکی از برنامه‌های مبتنی بر سرورس‌دهنده مانند PHP یا ASP باشد.

مثال: برنامه‌ای که با استفاده از روش server-side، پیوندهای مختلف برای یک تصویر ایجاد می‌کند.

```
<!DOCTYPE html>
<html>
<head>
<title>ISMAP Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="smap.php" target="_self">

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

مختصات محل کلیک شده رو تصویر به فایل smap.php ارسال می‌شود و در آنجا عمل پردازش انجام شده و سند موردنظر باز می‌شود.

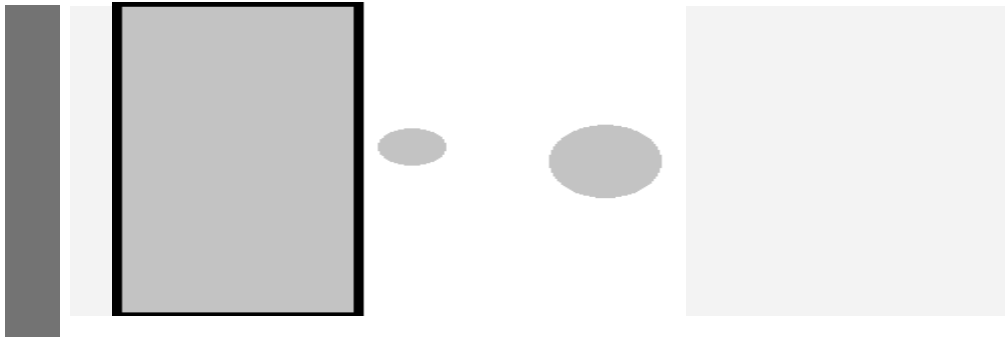
- روش Client-side image maps: در این روش از صفت usemap مربوط به تگ <a> و تگ‌های <map> و <area> استفاده می‌شود. نحوه کار به این صورت است که با استفاده از تگ یک تصویر درج می‌شود و سپس از صفت usemap استفاده کرده و یک نام به آن اختصاص می‌دهیم (در ابتدای نام باید کاراکتر # قرار دهید). سپس تگ <map> را نوشته و در صفت name آن، همان نام مربوط به صفت usemap را قرار می‌دهیم. در داخل تگ <map> از تگ <area> استفاده می‌شود. با استفاده از این تگ، مختصات ناحیه‌ای از متن به همراه پیوند آن قسمت مشخص می‌شود.

مثال: برنامه‌ای که با استفاده از روش client-side، پیوندهای مختلف برای یک تصویر ایجاد می‌کند.

```

<map name="plan-map">
<area shape="rect" coords="0,0,82,126" href="sun.htm" alt="Sun">
<area shape="circle" coords="90,58,3" href="mercur.htm" alt="Mercury"
>
<area shape="circle" coords="124,58,8" href="venus.htm" alt="Venus">
</map>
```

خروجی



نکته: ممکن است در بعضی مثال‌ها به دلیل طولانی بودن کدها، از نوشتن تگ‌های `<html>` و `<head>` خودداری کنیم.

همان‌طور که در مثال بالا مشاهده می‌کنید، صفت `shape`، نوع شکل را مشخص می‌کند و صفت `coords`، مختصات ناحیه موردنظر را مشخص می‌کند. بنابراین نحوه مختصات دادن در صفت `coords` کاملاً وابسته به مقدار صفت `shape` می‌باشد. جدول ۲-۴ نحوه مختصات دادن در انواع `shape` را مشخص می‌کند.

جدول ۲-۴ مقادیر مختلف صفت <code>shape</code> و نحوه مختصات دادن به آن	
گزینه	توضیحات
Rect (مستطیل)	<code>coords="x1,y1,x2,y2"</code> . گوشه سمت چپ و گوشه سمت راست مستطیل را مشخص می‌کند
Circle (دایره)	<code>coords="xc,yc,radius"</code> . نقطه مرکز دایره و شعاع دایره را مشخص می‌کند.
Poly (چندضلعی)	<code>coords="x1 , y1 , x2 , y2 , x3 , y3 , ... xn , yn"</code> . هر <code>x</code> و <code>y</code> یک نقطه یا یک رأس از یک چندضلعی را مشخص می‌کند.

ایجاد پیوند پست الکترونیک

برای ایجاد یک پیوند با آدرس یک پست الکترونیک، از تگ `<a>` استفاده کرده و سپس در صفت `href` پست الکترونیک موردنظر را وارد کنید. در شروع مقاردهی به صفت `href` از کلمه `mailto` استفاده می‌شود. بنابراین شکل کلی این نوع پیوند به صورت زیر می‌باشد:

```
<a href=mailto:someone@example.com?Subject=Hello%20again
&body=Message" >
Send Mail </a>
```

همان‌طور که در مثال بالا مشاهده می‌کنید، بعد از آدرس پست الکترونیک، در قسمت `subject`، عنوان و در قسمت `body`، متن مربوط به پست الکترونیکی مشخص می‌شود.

جدول، لیست، فریم، فرم و بلوک‌ها

۳-۱ جدول

یکی از روش‌ها برای مرتب کردن متن‌ها، تصاویر و دیگر داده‌ها در یک صفحه وب، استفاده از جدول می‌باشد. در HTML برای ایجاد جدول از تگ `<table>` استفاده می‌شود. هر جدول شامل تعدادی سطر و در هر سطر، تعدادی خانه وجود دارد. برای ایجاد یک سطر از تگ `<tr>` و برای ایجاد یک خانه در هر سطر از `<td>` استفاده می‌شود. همچنین برای ایجاد عنوان در یک جدول، از تگ `<th>` استفاده می‌شود.

مثال: برنامه‌ای که ایجاد یک جدول ساده را نشان می‌دهد.

```
<table border="1">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
```

خروجی

Name	Salary
Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2


صفت `border` در تگ `<table>` ضخامت مرز جدول را مشخص می‌کند.

صفات `Cellpadding` و `Cellspacing`

از این دو صفت برای تنظیم فضای خالی بین خانه‌های جدول استفاده می‌شود. صفت `Cellpadding` فاصله محتویات هر خانه با مرز آن خانه از جدول را تعیین می‌کند و صفت `Cellspacing` فاصله بین خانه‌های جدول را مشخص می‌کند.

مثال: برنامه‌ای که در آن از صفات `Cellpadding` و `Cellspacing` استفاده شده است.

```
<table border="1" cellpadding="5" cellspacing="5">
<tr>
    <th>Name</th> <th>Degry</th>
</tr>
<tr>
    <td>امیر علی کریم تبار</td> <td>۲۰</td>
</tr>
<tr>
    <td>علی رضا حسینی</td> <td>۱۸</td>
</tr>
</table>
```

خروجی 

Name	Degry
امیر علی کریم تبار	20
علی رضا حسینی	18

صفات Rowspan و Colspan

از صفت colspan برای ادغام دو یا چند ستون و از صفت rowspan برای ادغام سطرها استفاده می‌شود.

 مثال: برنامه‌ای که در آن از صفات Rowspan و Colspan استفاده شده است.

```
<caption>This is the caption</caption>
<table border="1" bordercolor="green" bgcolor="yellow">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr><td rowspan="2">Row 1 Cell 1</td> <td>Row 1 Cell 2</td><td>Row 1
Cell 3</td> </tr>
<tr><td>Row 2 Cell 2</td>
<td>Row 2 Cell 3</td>
</tr>
<tr><td colspan="3">Row 3 Cell 1</td>
</tr>
</table>
```

خروجی 