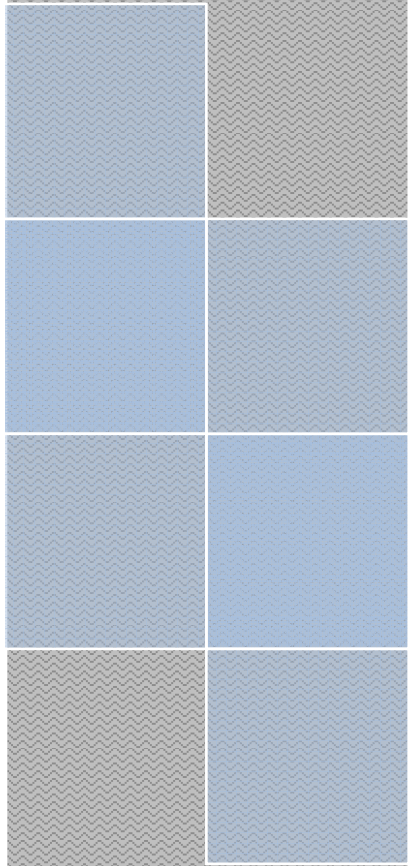
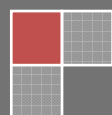


CSS

Cascading Style Sheet



علیرضا حسین زاده
۱۳۹۰



بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

Cascading Style Sheet

گردآوری :

علیرضا حسین زاده

Alireza.h68@hotmail.com

پاییز ۱۳۹۰

فصل اول

CSS چیست؟

CSS لیستی از دستورات در قالب متن است که برای تغییر ظاهر اسناد نوشته شده با زبانهای نشانه گذاری مثل HTML استفاده می شود. با استفاده از CSS می توان رنگ و فونت متن، فضای خالی بین پاراگراف ها، رنگ و عکس پس زمینه و بسیاری دیگر از خصوصیات بصری یک سند HTML را کنترل کرد.

روشهای استفاده از CSS در اسناد HTML

۱. قرار دادن دستورات CSS در داخل فایل HTML (Embedded style sheet)

```
<style type="text/css">.....</style>
```

کد فوق باید در بالای سند HTML و در داخل تگ <head/> قرار بگیرد.

۲. قرار دادن دستورات CSS در یک فایل متنی جداگانه با پسوند CSS. و استفاده از تگ <link/> برای ارتباط دادن این فایل به فایل HTML

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

در کد بالا یک فایل CSS خارجی با نام style.css به صفحه لینک شده است. کد فوق باید در بالای سند HTML و در داخل تگ <head/> قرار بگیرد.

۳. دستورات CSS می توانند به طور مستقیم در داخل عناصر HTML و با استفاده از صفت عمومی style در تگ های HTML استفاده شوند.

```
<body style="font:12px sans-serif;">
</body>
```

۴. با استفاده از دستور @import می توان دستورات CSS را به هر دو صورت embedded و linked در یک فایل وارد کرد:

```
<style type="text/css">
```

```
@import url(example1.css)
```

```
</style>
```

در استفاده از CSS به صورت linked (روش دوم) ممکن است این فایل CSS توسط مرورگر ذخیره شده و با refresh کردن update نشود.

نکته: نحوه نوشتن comment در CSS شبیه به زبان C++ بوده و به صورت زیر است:

```
/* this is comment */
```

گزینشگرها (Selectors)

Type selector

همان نام تگ های HTML می باشد:

```
table
{
border:1px solid #000;
}
```

style تعریف شده در این حالت بر روی تمامی عناصر از این نوع اعمال خواهد شد .

Class selector

Class selector می تواند برای تمامی عناصر HTML که صفت class آنها برابر با نام class مورد نظر تعیین شده استفاده شود. نام یک class selector با یک نقطه شروع شده و به دنبال آن نام class می آید. نام یک class میتواند شامل حروف، اعداد، - ، _ باشد، باید بدون space بوده و با حرف شروع شود. علامت نقطه نمایانگر یک class است مثل planet. علامت نقطه نباید در مقدار صفت class از تگ های HTML نوشته شود .

محدود کردن یک class به یک تگ خاص :

```
div.planet
{
}
```

در مثال فوق کلاس planet فقط روی تگ div اعمال می شود و محدود به این تگ میباشد؛ یعنی این خصوصیات CSS فقط بر روی تگ divی که از این class استفاده می کنند اعمال می شود.

در حالت فوق نباید بین نام تگ و نام class فضای خالی وجود داشته باشد، وجود فضای خالی باعث ایجاد نوع دیگری از selectorها بنام descendant selector می شود.

در صفت class از تگ های HTML می توان نام بیش از یک class را استفاده کرد؛ در این صورت خصوصیات هر دو class برای آن تگ پذیرفته می شود. در این حالت باید نام class ها با space از هم جدا شوند:

```
<div class="planet jupiter"></div>
```

نام class ها ممکن است به صورت زنجیروار (chained) به یکدیگر متصل شوند :

```
.planet.jupiter
{
}
```

در این حالت این قوانین فقط به عناصری اختصاص داده می شوند که در صفت class خود نام هر دو class را استفاده کرده باشند.

امکان فوق در IE6 وجود ندارد. در IE6 فقط نام class آخر در زنجیره به رسمیت شناخته می شود.

در IE6 planet.jupiter. به صورت jupiter. تفسیر می شود؛ یعنی تمامی عناصری که صفت class آنها برابر با jupiter میباشد از قوانین تعریف شده پیروی می کنند.

ID Selector

برای تعریف یک ID Selector شبیه به class عمل می کنیم و در ابتدای نام id از علامت # استفاده می کنیم. یک ID Selector فقط برای یک عنصر منحصر به فرد در سند به کار می رود :

```
#jupiter
{
}
```

```
<div class="planet" id="jupiter"></div>
```

اگر یک ID Selector برای بیش از یک عنصر استفاده شود فقط عنصر اول در نظر گرفته می شود .

Classها به طور مشترک می توانند به چندین عنصر اختصاص داده شوند اما ID Selectorها فقط می توانند به یک عنصر تخصیص داده شوند .

ID Selector فقط برای یک عنصر HTML منحصر به فرد در صفحه که نام ID برابر با مقدار صفت ID در آن تگ HTML میباشد، استفاده میشود. با استفاده نام class یا ID یک عنصر می توان به آن عنصر HTML در یک زبان اسکریپتی مثل JavaScript دسترسی داشت.

Universal Selector

گزینه‌های عمومی با علامت * مشخص می شوند. هنگامیکه از universal selector استفاده می شود به مفسر CSS گفته می شود که این قوانین را برای تمامی عناصر در سند اعمال کند:

```
*{
font-family:arial,Helvetica,sans-serif;
}
```

خصوصیات تعریف شده در universal selector برای همه چیز پذیرفته می شود از جمله form, input fields, table.

Descendant Selector

در descendant, CSS به این معنی است که یک عنصر فرزند یک عنصر دیگر باشد. style استفاده شده برای یک عنصر پدر بر روی تمامی عناصر فرزند اعمال می شود.

مثال:

```
<div class="planet" id="jupiter">
  <p>
  This is some text in a paragraph
  </p>
  <table> ..... </table>/
  <a href="#"> ..... </a>
</div>
```

در مثال فوق تگ div پدر تگ های <a>, <p> و <table> میباشد و این تگ ها فرزندان تگ div بوده و style خود را از تگ div به ارث می برند.

مثال:

```
div.jupiter h2
{
font-size:18px;
margin-top:0;
}
```

style فوق بر روی عنصر h2 ای که داخل یک عنصر div با کلاس jupiter قرار دارد اعمال می شود. این نوع selectorها محدود به دو عنصر نیستند و می توانند بیش از دو عنصر را شامل شوند. selectorها در زنجیره ی descendant selector باید بوسیله ی یک space از یکدیگر جدا شوند:

```
div.planet table td
{
padding:0 10px 0 0 ;
text-align:left;
}
```

در حقیقت یک سطح بندی کامل از عنصر ارشد (بزرگتر یا پدر) تا کوچکترین عنصری که می خواهید انتخاب کنید، می تواند در زنجیره ی descendant selector قرار بگیرد. (با شروع از عنصر <html> الی عنصر مورد نظر)

کلاس های کاذب (Pseudo classes)

این selectorها برای نمایش رویدادهای پویا مورد استفاده قرار می گیرند؛ مثل تغییر وضعیت. یا برای نمایش شرایط عمومی تری که به آسانی قابل

انجام با دیگر وسیله ها و امکانات موجود نیستند. این رویدادها میتوانند مثل حرکت mouse یا کلیک بر روی یک عنصر باشند. pseudo class به طراح آزادی عمل می دهد که چگونگی رفتار یک عنصر را در شرایط مختلف به آن عنصر دیکته کند.

بر خلاف کلاس های معمولی ، کلاس های کاذب یک علامت : (single colon) قبل از خصوصیت کلاس کاذب دارند. کلاس های زیر به عنوان dynamic pseudo class شناخته می شوند . این کلاس ها برای عناصر خاصی دسته بندی شده اند که این عناصر با انجام دادن یا انجام ندادن عمل خاصی توسط کاربر نمایش داده می شوند :

Pseudo classes	توضیحات
:link	اشاره به لینک های بازدید نشده دارد
:visited	لینک های بازدید شده را نشان می دهد
:hover	اشاره به عناصری دارد که اشاره گر موس کاربر در حال حاضر روی آن عنصر شناور می باشد
:focus	عنصری را که هم اکنون focus روی آن عنصر است را مشخص می کند
:active	عنصر جاری که کاربر روی آن کلیک کرده را مشخص می کند

برای کلاس های link و visited باید از style های جداگانه ای استفاده شود . بدیهی است که کلاسهای کاذب link و visited فقط برای عناصر <a/> قابل استفاده می باشند.

اگر کلاس کاذب link پس از کلاس کاذب focus تعریف شود ، کلاس link اولویت بالاتری برای اجرا خواهد داشت . (کلاسی که آخر از همه تعریف شود اولویت بالاتری خواهد داشت)

تعریف کلاس کاذب link به ترتیب فوق ، style تعریف شده برای کلاس focus را باطل می کند . یک روش برای به خاطر سپردن ترتیب کلاس کاذب استفاده از کلمه یادمان Love Hate می باشد . (ترتیب حروف بزرگ کلمه ی Love Hate) باید به خاطر داشته باشید focus همواره پس از hover می آید .

Style تعریف شده برای کلاس hover هنگامی اعمال می شود که اشاره گر موس روی عنصر مورد نظر قرار بگیرد و با ترک عنصر مورد نظر style به حالت قبلی باز می گردد . کلاس focus به طور مشابه کار می کند اما برای focus کیبورد نه موس .

در IE6 کلاس کاذب :hover فقط برای لینک ها پذیرفته می شود که این برخلاف CSS2 است . مرورگرهای دیگر بر طبق تعاریف CSS2 کلاس کاذب :hover را بر تمامی عناصر اعمال می کنند . مشکل فوق در IE7 و نسخه های بعد از آن حل شده است .

کلاس کاذب active اشاره به عناصری دارد که کاربر در حال حاضر روی آنها کلیک کرده و دکمه ی موس را پایین نگه داشته است ، تا زمانیکه کاربر دکمه ی موس را رها نکند style به حالت قبل باز نمی گردد . در IE6 و IE7 کلاس active فقط برای لینک ها پذیرفته می شود و سایر مرورگر ها این کلاس را برای تمامی عناصر می پذیرند .

اولویت و تقدم

لغت cascading برای نشان دادن اولویت و تقدم استفاده می شود . از آنجاییکه یک عنصر ممکن است با چندین تعریف مختلف CSS مطابقت داشته باشد ، خصوصیات CSS شامل یک مجموعه از رهنمودهای تعریف می باشد که اولویت را در مورد آنها مشخص می کنند و در مورد چگونگی آنها تصمیم گیری می کنند .

یک style sheet از سه منبع سرچشمه می گیرد :

۱. تنظیمات پیش فرض مرورگرها مثل فاصله ی بین خطوط و حروف ، فونت ، رنگ متن و ...
۲. Style تعریف شده توسط کاربر
۳. Style تعریف شده توسط طراح وب سایت

از آنجاییکه یک style از چندین منبع مختلف سرچشمه می گیرد ، این ضروری است که یک ترتیب از اولویت ها را مشخص کنیم تا این ترتیب style پذیرفته شده برای نمایش به کاربر نهایی را مشخص کند .

اولویت منابع style sheet به شکل زیر است :

۱. Style مشخص شده توسط مرورگر کمترین اولویت را دارد .
 ۲. Style تعریف شده توسط کاربر اولویتی بالا تر از مرورگر دارد .
 ۳. Style تعریف شده توسط طراح بیشترین اولویت را دارد .
 ۴. Style تعریف شده در داخل صفت style از تگ های HTML دارای بالاترین اولویت می باشد .
- از آنجاییکه style تعریف شده در صفت style سایر style های تعریف شده را لغو می کند ، استفاده از آنها توصیه نمی شود .

اولویت selector ها به این شکل مشخص می شود که یک selector تا چه اندازه صریح (specific) می باشد . برای مثال یک ID Selector صریحتر از یک universal selector می باشد . بین selector های مختلف و ترکیب های مختلف از آنها میزان صراحت (specificity) با استفاده از روش های زیر محاسبه می شود :

۱. اگر style در صفت style از تگ های HTML تعریف شده باشد مقدار ۱ و در غیر این صورت مقدار صفر را برای متغیر a در نظر بگیرید .
 ۲. تعداد ID Selector های استفاده شده در selector ها را در متغیر b قرار دهید .
 ۳. تعداد صفت ها (attribute) ، کلاس های کاذب و class selector ها را در مجموع در متغیر C قرار دهید .
 ۴. تعداد نام های عناصر (type selector) موجود در یک selector را در متغیر d قرار دهید .
 ۵. عناصر کاذب (pseudo element) را نادیده بگیرید .
- چهار مقدار (متغیر) فوق را به همراه هم در یک گروه چهار تایی قرار می دهیم :

selector	selector type	specificity
*	universal selector	0,0,0,0 (a=0,b=0,c=0,d=0)
li	element name	0,0,0,1
ul li	element name (2)	0,0,0,2
div h1+p	Element name(3)	0,0,0,3
input[type="text"]	Element name + attribute	0,0,1,1
.someclass	classname	0,0,1,0
div.someclass	elementname + classname	0,0,1,1
div.someclass.otherclass	elementname + classname(2)	0,0,2,1
#someid	id name	0,1,0,0
div#someid	elementname + id name	0,1,0,1
style(attribute)	style(attribute)	1,0,0,0

صفت style دارای معیار 1,0,0,0 است ، بنابراین دارای بیشترین اولویت در یک style می باشد . selectorی که دارای بیشترین مقدار عددی باشد اولویت بیشتری دارد .

در روش فوق Selector ی که از نظر مقدار عددی بزرگتر باشد پذیرفته خواهد شد. اگر دو selector دارای مقدار عددی یکسان باشند ، آنگاه آنها به ترتیب ظاهر شدن (نوشته شدن) پذیرفته می شوند .

مثال:

```
div.planet{
    background: none;
}
div.planet{
    margin: 10px 0;
    padding: 20px 20px 20px 200px;
    border: 1px solid #FFF;
    background-image: none;
    background-position: 20px 20px;
    background-repeat: no-repeat;
}
```

در مثال فوق هر دو selector دارای اولویت یکسان می باشند اما selector دوم که در انتها آمده پذیرفته خواهد شد .

```
#jupiter{
    background-image: url(jupiter.jpg);    0100
}
```

در id selector فوق دارای اولویت ۱۰۰ می باشد که اولویت بالاتری نسبت به ۱۱ می باشد ، در نتیجه خصوصیت background-image موجود در id selector پذیرفته خواهد شد و باقی خصوصیات div.palnet که مغایرتی با #jupiter ندارند نیز پذیرفته خواهند شد .

قاعده ی !important

موازی با نیار به cascade در یک css style نیاز به override در آن را نیز داریم و این جایی اتفاق می افتد که قاعده ی !important می آید . قاعده ی !important درون یک تعریف بعد از property value و قبل از semicolon می آید و یک تعریف را منسوخ می کند . این قاعده از دو قسمت تشکیل شده است :

۱. علامت تعجب (Exclamation mark) ! ، که به عنوان حایل (delimiter) استفاده می شود .

۲. کلمه ی کلیدی important

در اینجا علامت تعجب انتهای تعریف را مشخص می کند .

مثال :

```
body{
    background:#000 !important;
}
```

تعریفی شامل قاعده ی !important شبیه به مثال قبل اولویتی بالاتر از سایر تعاریف حتی تعاریف انجام شده در صفت style از تگ HTML را به تعریف می دهد .

اگر از بیش از یک قاعده ی !important در یک style sheet استفاده شده باشد و اگر هر دو style دارای منشا و سرچشمه یکسان باشند – هر دو قاعده توسط طراح یا هر دو توسط user تعریف شده باشند – آخرین قاعده ی تعریف شده بر تمامی تعاریف پیشین غلبه خواهد کرد . به این دلیل اکیدا توصیه می شود که هیچ گاه در هر کجای کد خود از این قاعده استفاده نکنید . فقط یک جا که ممکن است نیز به صراحت بیشتری داشته باشد مثل خصوصیت background باید از این قاعده استفاده کرد . استفاده از قاعده ی !important تعریف style توسط کاربر را دشوار می کند و با حفاظت بیشتری از style تعریف شده توسط طراح می شود .

توارث (inheritance)

تگ های فرزند می توانند برخی از مقادیر خصوصیات خود را (property value) از تگ های پدر به ارث ببرند . تمامی خصوصیات موروثی نیستند اما بسیاری از آنها این خاصیت را دارند .

برای بیشتر قسمت ها دو خصوصیت font و text موروثی می باشند . در جاهاییکه از توارث پشتیبانی می کنند بی خبر و بدون مقدمه این خصوصیات در style اعمال می شوند .

مثال :

```
body{
    width:650px;
    margin:0 auto;
    background:#000;
    color:#fff;
    font:12px sans-serif;
}
```


تمام عناصر داخل صفحه در داخل تگ **body** قرار می گیرند و بنابراین خصوصیات فوق را از این تگ به ارث می برند مگر اینکه این خصوصیات به طور جداگانه برای هر یک از آنها به شکل دیگری تعریف شده باشد .

با این وجود تمام **style** ها موروثی نیستند ؛ هیچ یک از عناصر صفحه به غیر از **body** دارای پهنای **650px (width)** نبوده و یا هیچ یک از عناصر دارای حاشیه ی **(margin)** برابر با **body** یا تگ پدر خود نیستند . این خصوصیات فقط لرای عناصری که برای آنها تعریف شده اند استفاده می شوند و نه فرزندان آنها ، این چیز خوبی است در غیر این صورت مجبور بودید خصوصیات **width** و **margin** را برای تمامی عناصر دیگر از نو بنویسید .

فصل دوم

Applying the Font Faces

font family

می توان نوع فونت مورد استفاده برای متن را با استفاده از خاصیت font-family تعیین کرد . فونتهای اصلی تعریف شده در CSS عبارتند از :

Generic fonts	مثال	
Serif	Times,times new roman	serif
Sans-serif	Helvetica,arial	sans-serif
Cursive	Zapf-chancery	monospace
Fantasy	Western	fantasy
Monospace	Courier,courier new	cursive

قاعده ی کلی :

font-family:<family name>;

- ✓ نام فونتها در خاصیت font-family حساس به حالت نبوده (case insensitive) و حروف بزرگ و کوچک تفاوتی ندارند .
- ✓ اگر نام فونت شامل فضای خالی (space) باشد آنگاه این نام باید با دبل کوتیشن محصور گردد .

مثال:

```
p{
font-family:times,"times new roman",serif;
}
```

می توان از چندین فونت در خاصیت font-family استفاده کرد ، این فونت ها باید با علامت (,) از یکدیگر جدا شوند . در این حالت اگر فونت اول بر روی سیستم کاربر موجود نباشد دومین فونت اعمال خواهد شد و اگر فونت دوم نیز بر روی سیستم کاربر موجود نباشد سومین فونت اعمال خواهد شد والی آخر .

style تعریف شده در مثال فوق فونت تمامی پارگراف های موجود را به times تغییر می دهد و اگر فونت times بر روی سیستم کاربر موجود نباشد از فونت times new roman استفاده خواهد شد و
به دومین فونت (فونت بعدی) در لیست مقادیر font-family ، fallback font گفته می شود .

از فونت sans-serif برای خوانا شدن متن در صفحه نمایش استفاده می شود .
از فونت serif بیشتر برای پرینت و چاپ کتاب استفاده می شود ؛ فونت serif در وب برای کلمات درشت تر مثل تیترا (heading) استفاده می شود.

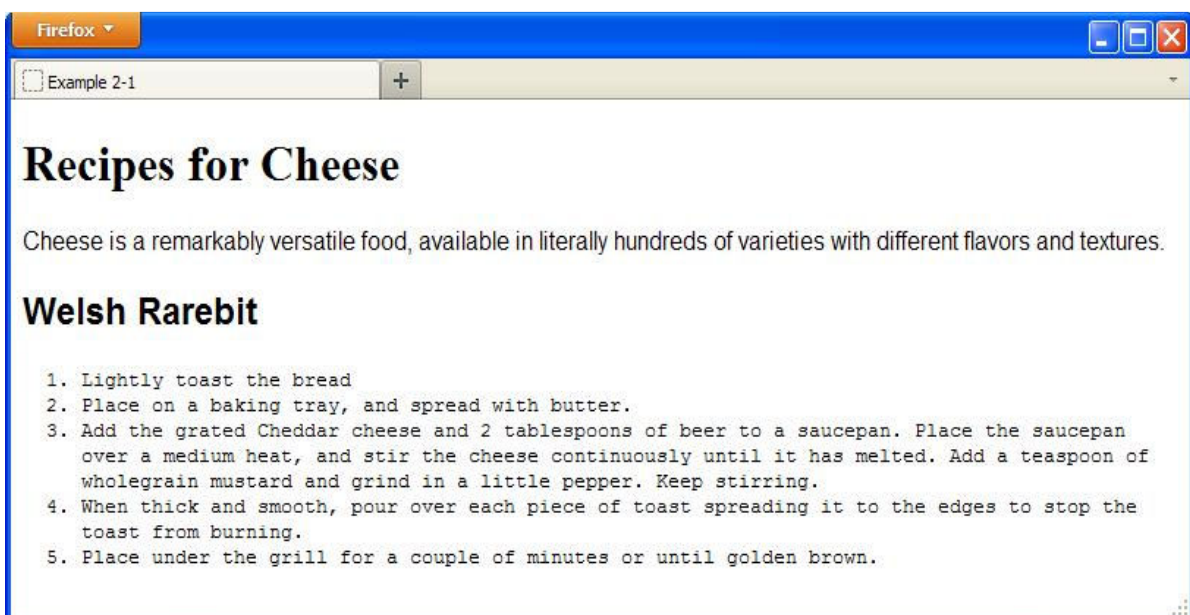
مثال : استفاده از خاصیت font-family

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```

<title>Example 2-1</title>
<style type="text/css">
  body {
    font-family: arial, helvetica, sans-serif;
  }
  h1 {
    font-family: "Times New Roman", Georgia, Serif;
  }
  ol {
    font-family: monospace;
  }
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p>Cheese is a remarkably versatile food, available in literally hundreds of
varieties with different flavors and textures.</p>
<h2>Welsh Rarebit</h2>
<ol>
  <li>Lightly toast the bread</li>
  <li>Place on a baking tray, and spread with butter.</li>
  <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
Place the saucepan over a medium heat, and stir the cheese continuously until
it has melted. Add a teaspoon of wholegrain mustard and grind in a little
pepper. Keep stirring.</li>
  <li>When thick and smooth, pour over each piece of toast spreading it to the edges to stop the
toast from burning.</li>
  <li>Place under the grill for a couple of minutes or until golden brown.</li>
</ol>
</body>
</html>

```



font style

با استفاده از این خاصیت (property) می توان متن را به صورت **italic** یا مورب نمایش داد .
قاعده ی کلی :

font-style:normal | italic | oblique

مقدار پیش فرض : normal

مقادیر ممکن برای این خاصیت عبارتند از :

✓ normal : مقدار پیش فرض و حالت عادی فونت می باشد .

✓ italic : متن را به صورت مورب نمایش می دهد .

✓ oblique : متن را به صورت مورب نمایش می دهد و در اکثر فونت ها تفاوت ظاهری چندانی با italic ندارد .

تمامی فونت ها شکل italic ندارند .

مثال :

```
h1{
    font-style:oblique;
}
p{
    font-style:normal;
}
```

font variant

این خاصیت حروف را به صورت **uppercase** نمایش می دهد اما در این حالت حروف کمی کوچکتر از حالت **capitalized** نمایش داده خواهند شد .

قاعده ی کلی :

font-variant: normal | small-caps

مقدار پیش فرض : normal

مقادیر ممکن برای این خاصیت عبارتند از :

✓ normal : مقدار پیش فرض و حالت عادی حروف

✓ small-caps : نمایش حروف به صورت uppercase

: font weight

با استفاده از این خاصیت می توان مقدار ضخامت متن را مشخص کرد .

قاعده ی کلی :

font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

مقدار پیش فرض : normal

در عمل فقط دو مقدار **normal** و **bold** در اکثر مرورگرها به درستی کار می کنند .

به طور معمول مقدار **normal** معادل مقدار ۴۰۰ و مقدار **bold** معادل مقدار ۷۰۰ می باشد .

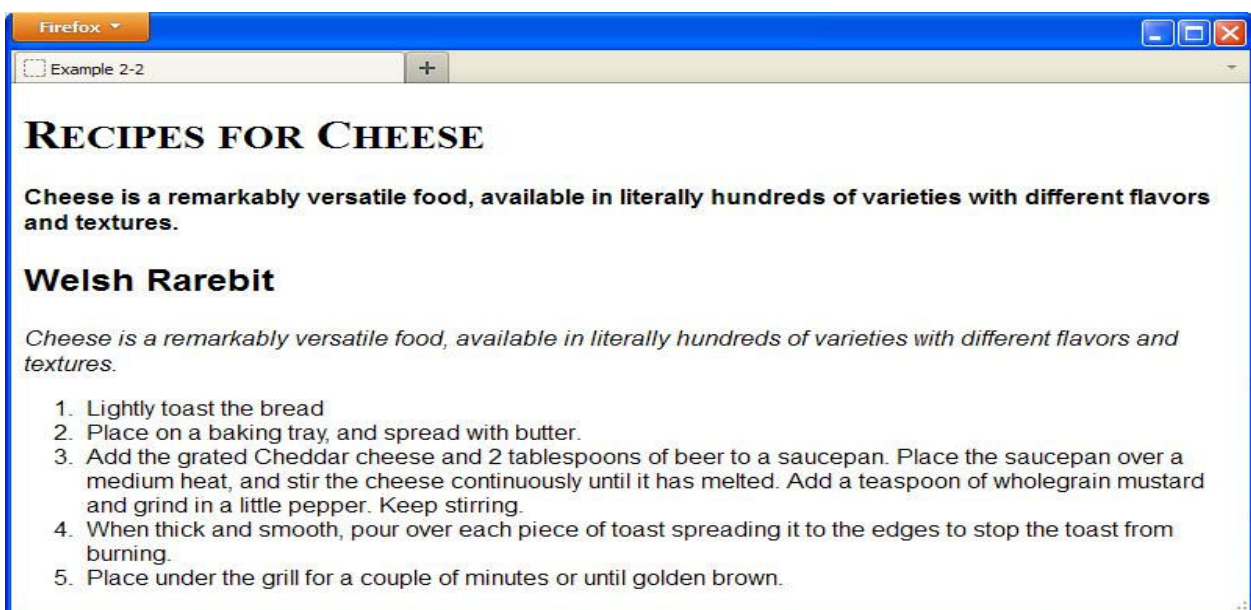
مثال : استفاده از property های font-style و font-variant و font-weight

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 2-2</title>
<style type="text/css">
    body {
        font-family: arial, helvetica, sans-serif;
    }
    h1 {
```

```

    font-family: "Times New Roman", Georgia, Serif;
    font-variant: small-caps;
}
.intro {
    font-weight: bold;
}
.recipe .intro {
    font-weight: normal;
    font-style: italic;
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p class="intro">Cheese is a remarkably versatile food, available in literally
    hundreds of varieties with different flavors and textures.</p>
<div class="recipe">
    <h2>Welsh Rarebit</h2>
    <p class="intro">Cheese is a remarkably versatile food, available in literally
        hundreds of varieties with different flavors and textures.</p>
    <ol>
        <li>Lightly toast the bread</li>
        <li>Place on a baking tray, and spread with butter.</li>
        <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
            Place the saucepan over a medium heat, and stir the cheese continuously until
            it has melted. Add a teaspoon of wholegrain mustard and grind in a little pepper. Keep stirring.</li>
        <li>When thick and smooth, pour over each piece of toast spreading it to
            the edges to stop the toast from burning.</li>
        <li>Place under the grill for a couple of minutes or until golden
            brown.</li>
    </ol>
</div>
</body>
</html>

```



font size

با استفاده از خاصیت font-size می توان اندازه یفونت متون را کنترل کرد .
قاعده ی کلی :

font-size: <absolute-size> | <relative-size> | <percentage> | <length>

مقادیر ممکن برای این خاصیت عبارتند از :
✓ کلمات کلیدی

absolute-size: xx-small | x-small | small | medium | large | x-large | xx-large

مقدار پیش فرض : medium
معادل های کلمات کلیدی برای font-size در HTML عبارتند از :

کلمات کلیدی	x-small	small	medium	large	x-large	xx-large
HTML	<h6>	<h5>	<h4>	<h3>	<h2>	<h1>Heading

در کلمات کلیدی هر مقدار ۱.۲ برابر مقدار قبلی خود می باشد ، به طور مثال اگر medium برابر 16px باشد آنگاه large برابر $16 * 1.2 = 19.2$ یا تقریباً 20px خواهد بود .

✓ مقادیر نسبی

relative-size: larger | smaller

مقدار larger در relative-size فونت را ۱.۲ برابر فونت قبلی می کند .
مقادیر زیر 100% معادل smaller و مقادیر بالای 100% معادل larger می باشند .

✓ مقادیر درصدی

مقادیر درصدی برای font size وابسته به عنصر پدر می باشد .
مثال :

```
p{
font-size: 80%;
}
```

✓ length

مقادیر length عبارتند از :

length**توضیحات**

in (inches ; 1 in = 2.54 cm)

cm (centimeter)

mm (millimeters)

pt (points ; 1 pt = 1/72 in)

pc (pica ; 1 pc = 12 pt)

px (pixel)

em (ems)

ex (x-height)

مقادیر absolute مناسب برای print

مقادیر absolute مناسب برای print

مقادیر absolute مناسب برای print

مقادیر absolute مناسب برای print

به مقدار resolution صفحه نمایش بستگی دارد (مناسب برای صفحه نمایش)

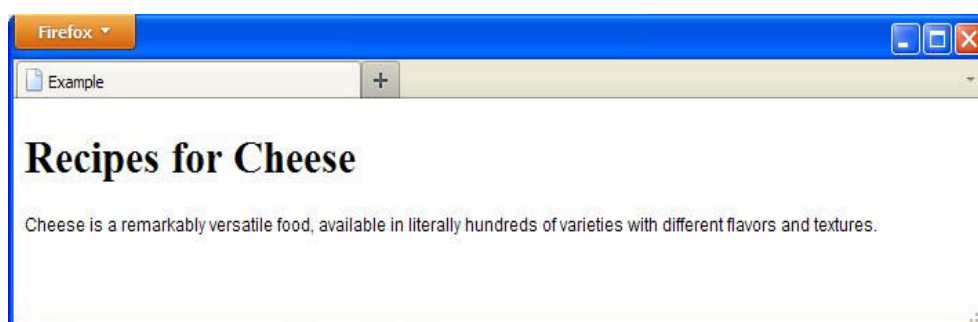
مناسب و سازگار برای font size بوده و به ارتفاع فونت عنصر وابسته می باشد (مناسب برای صفحه نمایش)

به ارتفاع حرف x کوچک وابسته می باشد (مناسب برای صفحه نمایش)

اگر اندازه ی فونت با واحد px مشخص شده باشد IE توانایی تغییر اندازه ی فونت (resize) را نخواهد داشت .
اگر font size با واحد ems یا percentage مشخص شده باشد تمامی مرورگرها از جمله IE توانایی resize کردن متن را خواهند داشت .
مثال:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Example</title>
  <style type="text/css">

    body {
      font-family: arial, helvetica, sans-serif;
      font-size: 12px;
    }
    h1 {
      font-family: "Times New Roman", Georgia, Serif;
      font-size: 30px;
    }
  </style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p>Cheese is a remarkably versatile food, available in literally hundreds
of varieties with different flavors and textures.</p>
</body>
</html>
```



کوتاه نویسی font property

قاعده کلی :

```
font: [<font-style> || <font-variant> || <font-eeight> || ] ?
      || <font-size>
      || [</line-height>] ?
      || <font-family>
```

- | : به معنی OR (یا)؛ یعنی فقط یکی از خاصیت ها باید استفاده شود .
- || : یکی یا دو تا یا . . . همه ی خاصیت ها می توانند همزمان در یک دستور استفاده شوند .
- ? : یعنی دستور اختیاری است و می تواند استفاده نشود .
- [] : استفاده از خاصیت ها به صورت گروهی

نکته: خاصیت line-height فاصله ی عمودی دو خط را تعیین می کند .

line-height: normal | <number> | <length> | <percentage>

در حالت کوتاه نویسی مقادیر خاصیت ها باید با یک space از یکدیگر جدا شوند .

مثال :

```
p{
    font: italic bold 12pt/14pt times,serif;
}
```

مثال فوق پاراگرافی به شکل italic و bold با اندازه فونت 12 point و فاصله ی بین خطوط 14 piont و فونتی به نام times (یا serif) ایجاد می کند .

در حالت کوتاه نویسی فونت (shortand font) حداقل باید مقادیر font-size و font-family تعیین شوند و باقی گزینه ها اختیاری است .

مثال :

```
<style type="text/css">
body{
    font-weight:bold;
}
p{
    font: 100% arial,Helvetica,sans-serif;
}
</style>
```

در مثال فوق font-weight:fvhd برای body به صورت bold در نظر گرفته شده ، اما مقدار bold برای عنصر p به ارث نمی رسد زیرا برای عنصر p شکل کوتاه نویسی شده ی فونت در نظر گرفته شده است .

در مثال فوق هر چند font-weight برای عنصر p در نظر گرفته نشده اما باز هم مقدار خود را از body به ارث نخواهد برد و به صورت normal باقی خواهد ماند .

قانون فوق در مورد تمام shortand property ها (کوتاه نویسی خاصیت ها) صدق می کند .

فصل سوم

Mainpulating the Display of the Text

line height

با استفاده از این خاصیت می توان فاصله ی عمودی بین خطوط یک متن را کنترل کرد .
قاعه ی کلی :

line-height: normal | <number> | <length>| <percentage>

مقدار پیش فرض : normal

number ✓

این مقدار شامل مقادیر صحیح (مثل 0,1,2,3,...) و اعشاری (مثل 0.5, 0.75,1.5,...) می باشد .
مقادیر منفی برای این خاصیت مجاز نمی باشد .

بهترین راه برای تخصیص مقدار به این خاصیت استفاده از number value می باشد .
مقدار یک برای این خصوصیت برابر با 1em یا 10px (برای فونتی با سایز 10px) و... می باشد .

اگر شما line-height را در عنصر body برای فونتی با سایز 10px برابر با 1.5em تعریف کنید ، این مقدار برای تمامی عناصر داخل عنصر body برابر با 15px خواهد بود حتی برای عناصری که فونت آنها اندازه های غیر از 10px مثلا 15px دارد ؛ یعنی مقدار موروثی با توجه به نوع value استفاده شده متفاوت خواهد بود و این دلیلی است که استفاده از number بهترین شکل ممکن می باشد .

مثال :

```
p{
  line-height: 2 ;
}
```

استفاده از مقدار line-height: 3 باعث می شود فاصله یعمودی بین دو خط از متن ۳ برابر فونت استفاده شده شود . مثلا اگر font-size:13px باشد و line-height:3; باشد آنگاه فاصله یعمودی دو خط $13*3=39px$ خواهد بود .
با تخصیص مقدار به line-height مقادیر به دو قسمت تقسیم شده و نصف آن به فضای خالی بالای خط و نصف دیگر به فضای خالی پایین خط اضافه می شود .

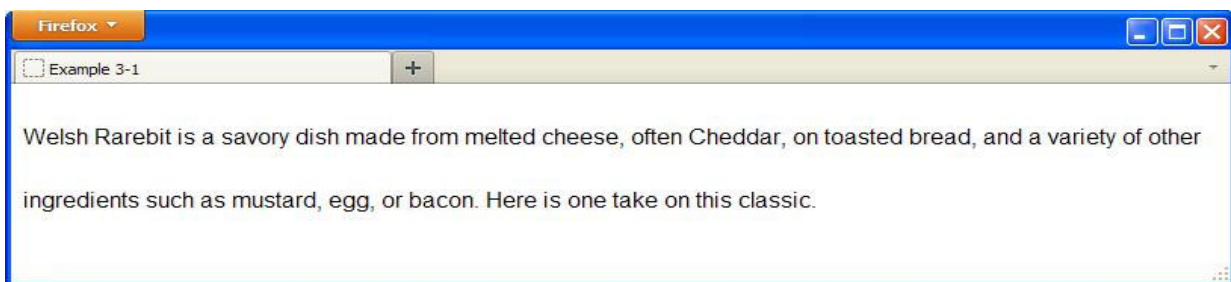
مثال : استفاده از خاصیت line-height

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <title>Example 3-1</title>
```

```

<style type="text/css">
  .intro {
    line-height: 3;
  }
</style>
</head>
<body>
<p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
</body>
</html>

```



letter spacing

خاصیت letter-spacing یک فضای خالی اضافی بین کاراکترها ایجاد می کند .
قاعده ی کلی :

letter-spacing: normal | <length>

مقدار پیش فرض : normal

مقدار پیش فرض normal بوده و توسط فونت استفاده شده تعیین می شود که این مقدار برابر با مقدار صفر (zero length) می باشد .
برای این خاصیت هر دو مقدار مثبت و منفی مجاز می باشد که استفاده از مقادیر منفی باعث تو رفتگی و به هم ریختگی حروف می شود .

word spacing

خاصیت word-spacing یک مقدار فضای خالی اضافی برای فاصله ی بین لغات تعیین می کند . این مقدار باید در یکی از فرمت های length باشد؛
مقدار منفی برای این خاصیت مجاز می باشد .
قاعده ی کلی :

word-spacing: normal | <length>

مقدار پیش فرض : normal

مثال :

```

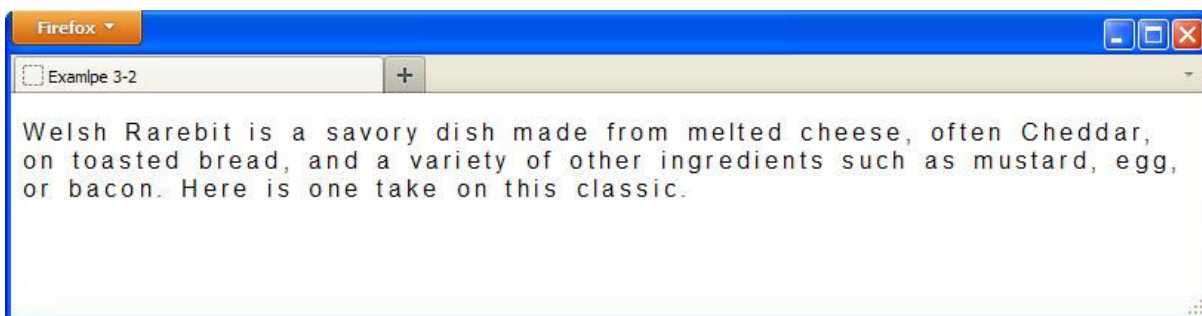
p{
  letter-spacing:0.2em ;
  word-spacing: 0.4em ;
}

```

```
h1{
    letter-spacing: -0.1em ;
    word-spacing: -0.3em ;
}
```

مثال : استفاده از خاصیت های letter-spacing و word-spacing

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Examlpe 3-2</title>
  <style type="text/css">
    .intro {
      letter-spacing: 0.2em;
      word-spacing: 0.2em;
    }
  </style>
</head>
<body>
<p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
</body>
</html>
```



text indent

این خاصیت مقدار تو رفتگی اولین خط از عنصر را تعریف می کند و به طور متداول برای ایجاد تو رفتگی در اولین خط یک پاراگراف استفاده می شود .
قاعده ی کلی :

text-indent: <length> | <percentage>

مقدار پیش فرض : صفر

تخصیص مقدار percentage برای این خاصیت به عرض یا پهنای عنصر پدر وابسته می باشد .

مثال :

```
p{
    text-indent: 2em;
}
```

text align

این خاصیت مشابه صفت align در تگ های HTML می باشد و برای ترازبندی افقی متن استفاده می شود .
قاعده ی کلی :

text-align: left | right | center | justify

مقدار پیش فرض : left

- left : متن را از چپ تراز می کند . ✓
- right : متن را از سمت راست تراز می کند . ✓
- center : متن را از وسط تراز می کند . ✓
- justify : متن را از دو طرف تراز می کند . ✓

استفاده از مقدار justify برای وب ممکن است خواننده متن را کمی مشکل کند ، لذا بهتر است از این مقدار استفاده نشود .

مثال :

```
h1{
    text-align: center ;
}
```

مثال : استفاده از خاصیت های text-indent و text-align

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Examlpe 3-3</title>
    <style type="text/css">
        h2 {
            text-align: center;
        }
        .intro {
            text-indent: 2em;
            text-align: justify;
        }
    </style>
</head>
<body>
<h2>Welsh Rarebit</h2>
<p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.Welsh Rarebit is a savory dish made from melted
```

cheese, often

Cheddar, on toasted bread, and a variety of other ingredients such as mustard, egg, or bacon. Here is one take on this classic.</p>

</body>

</html>



text decoration

قاعده ی کلی :

text-align: none | [underline | | overline | | line-through | | blink]

مقدار پیش فرض : none (هیچ)

- ✓ underline : یک خط زیر متن می کشد .
- ✓ overline : یک خط بالای متن می کشد .
- ✓ line-through : یک خط روی متن می کشد .
- ✓ blink : متن را به صورت چشمک زن نمایش می دهد و معادل تگ <blink> در HTML می باشد .

امکان استفاده از بیش از یک مقدار به طور همزمان در این خاصیت وجود دارد . هر چهار مقدار داخل براکت را می توان به طور همزمان استفاده کرد . مقدار blink در این خاصیت تقریباً منسوخ شده و استفاده زیاد از این مقدار متن را ناخوانا می کند .

مثال :

```
a:link,a:visited,a:active {
    text-decoration: none ;
}
```

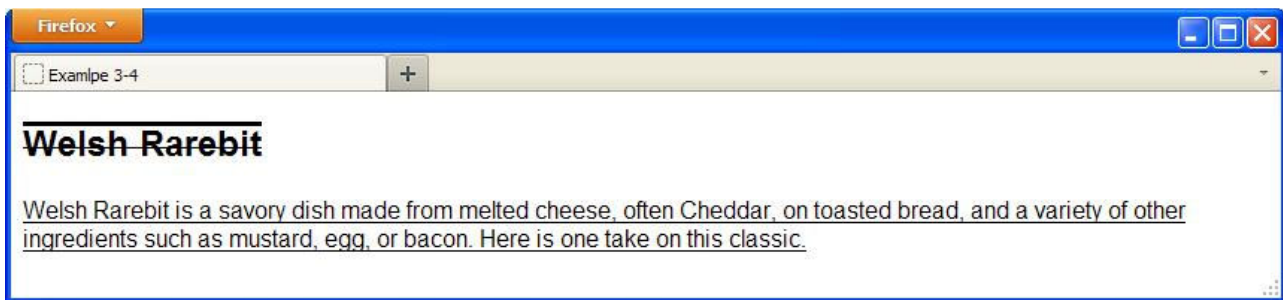
مثال : استفاده از خاصیت text-decoration

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Examlpe 3-4</title>
<style type="text/css">
h2 {
    text-decoration: line-through overline;
```

```

    }
    .intro {
    text-decoration: underline;
    }
</style>
</head>
<body>
<h2>Welsh Rarebit</h2>
<p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
</body>
</html>

```



text transform

قاعده ی کلی :

text-tarnsform: none | Capitalize | UPPERCASE | lowercase

مقدار پیش فرض : none (هیچ)

- ✓ capitalize : اولین کاراکتر از هر کلمه را با حرف بزرگ نمایش می دهد .
- ✓ uppercase : تمامی کاراکتر های یک کلمه را با حروف بزرگ نمایش می دهد .
- ✓ lowercase : تمامی کاراکتر های یک کلمه را با حروف کوچک نمایش می دهد .
- ✓ none : مقدار پیش فرض را برای کاراکترها در نظر می گیرد .

مثال : استفاده از خاصیت text-transform

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Examlpe 3-5</title>
<style type="text/css">

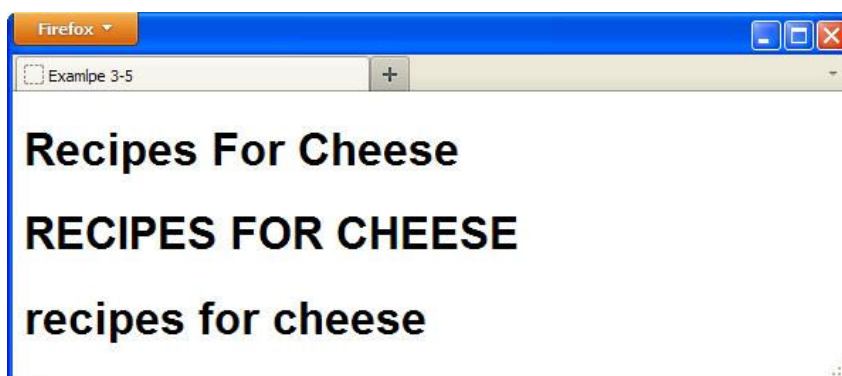
```

```
h1.capitalize {
```

```

    text-transform: capitalize;
  }
h1.uppercase{
  text-transform: uppercase;
}
h1.lowercase{
  text-transform: lowercase;
}
</style>
</head>
<body>
<h1 class="capitalize">Recipes for Cheese</h1>
<h1 class="uppercase">Recipes for Cheese</h1>
<h1 class="lowercase">Recipes for Cheese</h1>
</body>
</html>

```



white space

قاعده ی کلی :

white-space: normal | pre | nowrap

مقدار پیش فرض : normal

✓ **pre** : در این خاصیت بیشتر از مقدار **pre** استفاده می شود و از این مقدار برای نمایش متن به همان شکلی که در عنصر مورد نظر نوشته شده با در نظر گرفتن تمامی فضاهای خالی و شکست های خط استفاده می شود . از مقدار **pre** معمولا برای عنصر **<code>** استفاده می شود . مقدار **pre** معادل تگ **pre** در HTML می باشد .

✓ **nowrap** : استفاده از مقدار **nowrap** از شکست خط در متن جلوگیری می کند مگر اینکه از تگ **
** استفاده شده باشد . این مقدار تقریبا هیچ وقت استفاده نمی شود زیرا استفاده از این مقدار آرایش صفحه را به هم می زند .

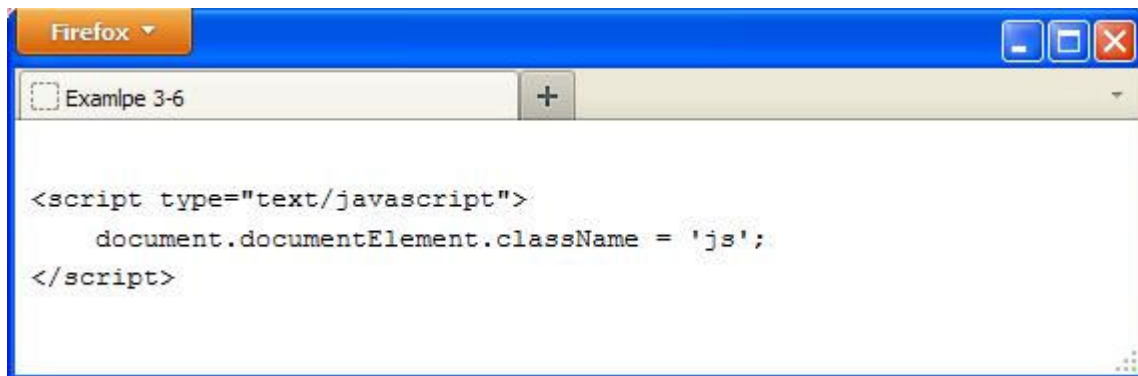
مثال : استفاده از خاصیت **white-space**

```

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

```

```
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Examlpe 3-6</title>
  <style type="text/css">
    code {
      white-space: pre;
    }
  </style>
</head>
<body>
<code>
&lt;script type="text/javascript"&gt;
  document.documentElement.className = 'js';
&lt;/script&gt;
</code>
</body>
</html>
```



فصل چهارم

Background Colors and Images

Color

با استفاده از این خاصیت می توان رنگ متن عناصر را تغییر داد .
قاعده ی کلی :

color: <color>

✓ به چهار روش می توان color را مقدار دهی کرد :

۱. استفاده از کلمات کلیدی : می توان از ۱۶ رنگ استاندارد برای مقدار دهی به color استفاده کرد که عبارتند از :

aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow.

۲. استفاده از تابع rgb : تابع rgb به ترتیب سه رنگ قرمز (red) ، سبز (green) و آبی (blue) را به عنوان مقدار می پذیرد . این مقادیر یا اعداد صحیح هستند (از ۰ تا ۲۵۵) و یا به صورت درصد می باشند (از ۰ تا ۱۰۰٪) . به طور مثال rgb(0,160,255) یا rgb(0%,63%,100%) .

۳. استفاده از مبنای هگزادسیمال یا ۱۶ تایی رنگ ها : رنگ ها را می توان بر مبنای عبارت متناظر آنها در مبنای ۱۶ به کار برد . در این عبارت می توان سه عدد در مبنای ۱۶ را نوشت . عدد اول معرف رنگ قرمز ، عدد دوم معرف رنگ سبز و عدد سوم معرف رنگ آبی است . این اعداد می توانند از دو حرف تشکیل شده باشند زیرا اعداد در مبنای ۱۶ از a تا f نیز می توانند تشکیل شوند . فرم کلی این عبارت ها به شکل #rrggbb است . برای مثال #000000 معرف رنگ سیاه ، #ffffff معرف رنگ سفید و #0000ff معرف رنگ آبی است .

۴. استفاده از کوتاه نویسی هگزادسیمال : در این روش زوج مقادیر تکراری را می توان به صورت تکی نوشت .
برای مثال #fff معادل #ffffff ، #000 معادل #000000 ، #ccddaa معادل #cdaa... می باشد .

مثال :

```
h1{
  color:blue;
}
h2{
  color:#000000;
}
h3{
  color:#0c0;
}
```

background color

با استفاده از این خاصیت می توان رنگ پس زمینه عناصر را تغییر داد .
قاعده ی کلی :

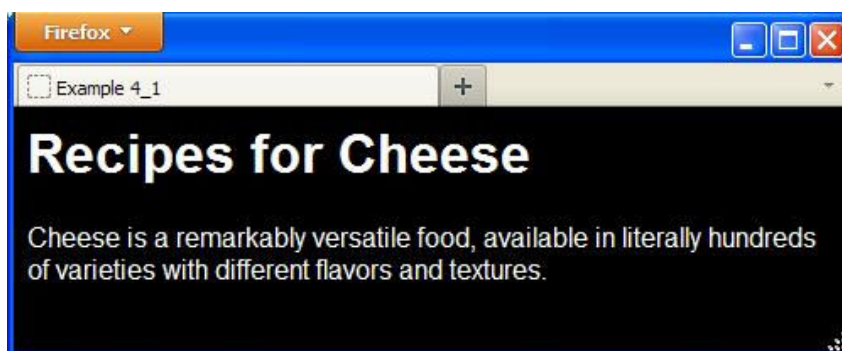
background-color: <color> | transparent

مقدار پیش فرض : transparent (شفاف ، بی رنگ)

مقدار transparent مشخص می کند که هیچ رنگی نباید استفاده شود .

مثال : استفاده از خاصیت های color و background-color

```
<html>
<head>
<title>Example 4_1</title>
<style type="text/css">
  body {
    background-color: #000;
    color: #FFF;
  }
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
</body>
</html>
```



background image

با استفاده از این خاصیت می توان برای عناصر تصویر پس زمینه تعریف کرد . تصویر استفاده شده محتوا و padding عنصر مورد نظر را پوشش می دهد .

قاعده ی کلی :

background-image: <url> | none

مقدار پیش فرض : none

- ✓ url: یک آدرس نسبی یا مطلق را به عنوان مقدار می پذیرد به طور مثال `url(/images/sky.gif)`.
- ✓ none: پس زمینه بدون تصویر خواهد بود.

مثال:

```
body{
  background-image: url(/images/foo.jpg);
}
```

background repeat

با استفاده از این خاصیت می توان نحوه تکرار عکس در پس زمینه را کنترل کرد .
قاعده ی کلی :

background-repeat: repeat | repeat-x | repeat-y | no-repeat

مقدار پیش فرض : repeat

- ✓ repeat: تکرار عکس در دو جهت عمودی و افقی
- ✓ repeat-x: تکرار عکس در جهت افقی (horizontal)
- ✓ repeat-y: تکرار عکس در جهت عمودی (vertical)
- ✓ no-repeat: عدم تکرار عکس

مثال: استفاده از خاصیت های `background-repeat` و `background-image`

```
<html>
<head>
<title>Example 4_2</title>
<style type="text/css">
body {
  background-image: url(fish.jpg);
  background-repeat: repeat-x;
  color:#f0f
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<div class="recipe">
  <h2>Welsh Rarebit</h2>
  <p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
  <ol>
    <li>Lightly toast the bread</li>
    <li>Place on a baking tray, and spread with butter.</li>
  </ol>
</div>
```

```

<li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
Place the saucepan over a medium heat, and stir the cheese continuously...</li>
</ol>
</div>
</body>
</html>

```



background attachment

با استفاده از این خاصیت می توان مشخص کرد که با scroll شدن محتوای یک عنصر عکس پس زمینه ی آن نیز scroll شود یا اینکه ثابت بماند .
قاعده ی کلی :

background-attachment: scroll | fixed

مقدار پیش فرض : scroll

✓ scroll : با scroll شدن محتوا عکس پس زمینه نیز scroll خواهد شد .

✓ fixed : با scroll شدن محتوا عکس پس زمینه در جای خود ثابت باقی خواهد ماند .

در IE6 از مقدار fixed فقط در عنصر body پشتیبانی می شود اما در سایر مرورگرها از کلمه ی کلیدی fixed در تمام عناصر پشتیبانی می شود .

مثال :

```

body{
background-color: blue;
background-image: url(/images/foo.jpg);
background-repeat: repeat-x;

```

```
background-attachment: fixed;
}
```

background position

با استفاده از این خاصیت می توان موقعیت قرار گرفتن عکس در پس زمینه ی عنصر را مشخص کرد .
قاعده ی کلی :

```
background-position: [<percentage> | <length> ] |
[[ top | center | bottom ] || [ left | center | right ]]
```

آسانترین روش برای تخصیص مقدار به این خاصیت استفاده از کلمات کلیدی است :

مقادیر افقی : left , center , right

مقادیر عمودی : top , center , bottom

مثال :

```
background-position: left center;
```

هنگام استفاده از مقادیر percentage و length باید ابتدا موقعیت افقی و سپس موقعیت عمودی ذکر شود .
مثال :

```
background-position: 20% 65%;
```

دستور فوق مشخص می کند که تصویر پس زمینه باید ۲۰٪ به سمت داخل در جهت افقی (به سمت راست) و ۶۵٪ به سمت داخل در جهت عمودی (به سمت پایین) در داخل عنصر مربوطه قرار گیرد .

مثال :

```
background-position:5px 10px;
```

دستور فوق مشخص می کند که تصویر پس زمینه باید از گوشه ی بالا و سمت چپ 5pixel به طرف راست و 10pixel به طرف پایین قرار بگیرد .

اگر در خاصیت background-image فقط مقدار افقی ذکر شود مقدار عمودی ۵۰٪ در نظر گرفته می شود .

ترکیب length و percentage و همچنین استفاده از مقادیر منفی مجاز است . length و percentage نمی توانند با کلمات کلیدی ترکیب شوند .

```
background-position: 10% -2cm;
```

اگر تصویر پس زمینه به صورت fixed تعریف شده باشد ، background position به جای آنکه نسبت به موقعیت عنصر تغییر کند نسبت به زمینه تغییر خواهد کرد .

هنگامیکه عکس پس زمینه در هر دو محور X و Y تکرار می شود موقعیت تعیین شده برای عکس بستگی به نقطه ی شروع تکرار دارد .

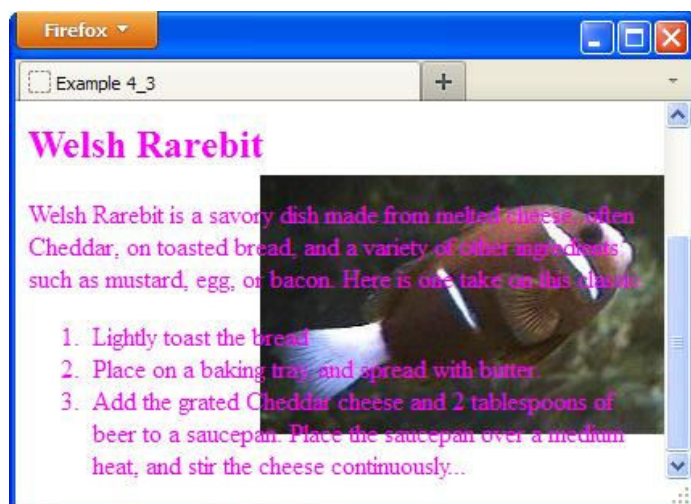
مثال : استفاده از خاصیت های background-position و background-attachment

```
<html>
<head>
<title>Example 4_3</title>
<style type="text/css">
body {
```

```

background-image: url(fish.jpg);
background-repeat: no-repeat;
background-position: left bottom;
color:#f0f
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<div class="recipe">
  <h2>Welsh Rarebit</h2>
  <p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
  <ol>
    <li>Lightly toast the bread</li>
    <li>Place on a baking tray, and spread with butter.</li>
    <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
Place the saucepan over a medium heat, and stir the cheese continuously...</li>
  </ol>
</div>
</body>

```



کوتاه نویسی (shortand background) background

پنج خاصیت فوق (مربوط به background) را می توان به صورت کوتاه در یک دستور ادغام کرد .
قاعده ی کلی :

```
background: <background-color> || <background-image> || <background-repeat> ||  
          <background-attachment> || <background-position>
```

هر پنج مقدار ممکن برای background را می توان به طور همزمان در این خاصیت استفاده کرد . در این کوتاه نویسی ترتیب خاصیت های مورد استفاده مهم نمی باشد .

اگر مقداری برای یک خاصیت ذکر نشود ، مقدار پیش فرض آن استفاده خواهد شد .

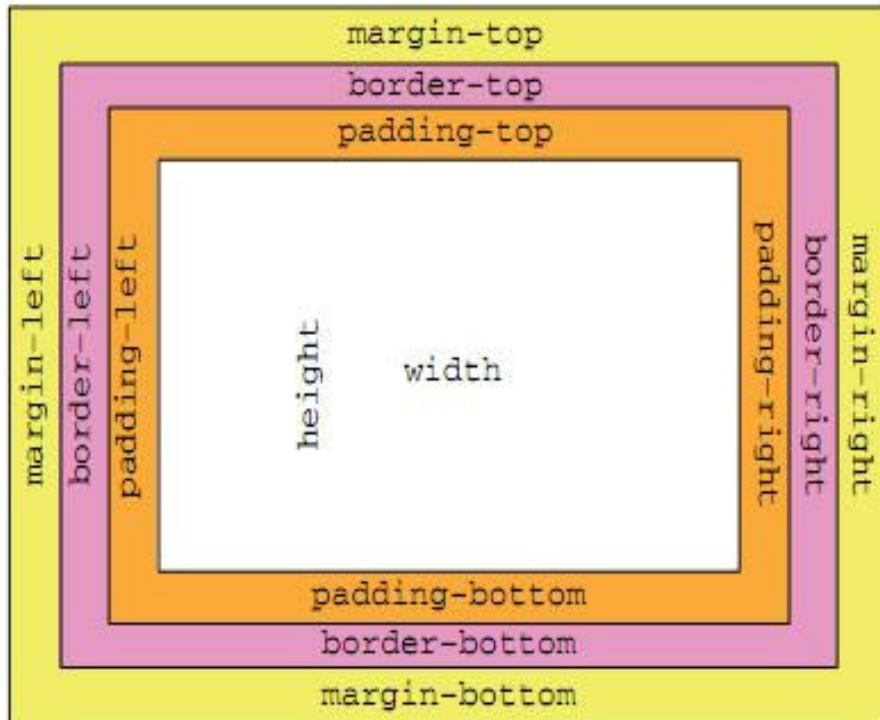
مثال :

```
table{  
    background: #0c0 url(./images/sky.png) no-repeat fixed bottom right ;  
}
```

فصل پنجم

The Box Model :

Controlling Margins,Borders,Padding,Width and Height



margin

این خاصیت فضای خالی بیرونی عناصر را مشخص می کند؛ یعنی فضای خالی بین عنصر و پنجره ی مرورگر و یا فضای خالی بین دو عنصر را مشخص می کند.
قاعده ی کلی:

```
margin      : [ <length> | <percentage> | auto ] {1,4}
margin-top  : <length> | <percentage> | auto
margin-right
margin-bottom
margin-left
```

- ✓ margin-top : حاشیه ی بالای عنصر را مشخص می کند .
- ✓ margin-right : حاشیه ی سمت راست عنصر را مشخص می کند .
- ✓ margin-bottom : حاشیه ی پایینی عنصر را مشخص می کند .
- ✓ margin-left : حاشیه ی سمت چپ عنصر را مشخص می کند .

مقدار پیش فرض : auto

استفاده از مقادیر منفی برای margin مجاز است .

مثال :


```
body {
    margin-top:0;
}
```

مثال فوق حاشیه بالای سند را حذف می کند .

مثال :

```
p.narrow{
    margin-right: 50%;
    margin-bottom: 30px;
    margin-left: 2em;
}
```

با استفاده از کوتاه نویسی می توان چهار دستور فوق را در یک دستور ترکیب کرد . در حالت کوتاه نویسی می توان از یک تا چهار مقدار استفاده کرد .

مثال :

```
body{
    margin:5em;
}
```

دستور فوق تمامی حاشیه های عنصر `body (top , right , bottom ,left)` را برابر `5em` تعریف می کند . در صورت استفاده از یک مقدار در کوتاه نویسی این مقدار به هر چهار حاشیه تخصیص خواهد یافت .

مثال :

```
p{
    margin: 2em 5em;
}
```

دستور فوق حاشیه ی بالا و پایین (`top , bottom`) را برابر `2em` و حاشیه ی چپ و راست (`right , left`) را برابر `5em` تعریف می کند . هنگام استفاده از دو مقدار برای کوتاه نویسی مقدار اول `top , bottom` و مقدار دوم `right , left` را مشخص می کند .

در صورت استفاده از سه مقدار در کوتاه نویسی مقدار اول `top` ، مقدار دوم `right , left` و مقدار سوم `bottom` را مشخص می کند .

مثال :

```
p{
    margin: 70px 100px 50px;
}
```

دستور فوق `top` را برابر `70px` ، `left` و `right` را برابر `100px` و `bottom` را برابر `50px` تعیین می کند .

در حالت کوتاه نویسی خصوصیات (`margin , padding , border`) `box model` چهار مقدار در جهت عقربه های ساعت (`clockwise`) تخصیص می یابند ؛ یعنی به ترتیب `top` و `right` و `bottom` و `left` .

مثال :

```
body{
    margin:70px 100px 50px 20px;
}
```

دستور فوق top را برابر 70px ، right را برابر 100px ، bottom را برابر 50px و left را برابر 20px قرار می دهد .

معمولا در حالت کوتاه نویسی margin از دو یا چهار مقدار استفاده می شود و استفاده از سه مقدار چون گیج کننده است توصیه نمی شود .

در حاشیه های عمودی (vertical margin) امکان فروپاشی یا تداخل حاشیه ها (margin collapsing) وجود دارد ؛ این تداخل زمانی اتفاق می افتد که margin-top یا margin-bottom از یک عنصر با margin-bottom یا margin-top از عنصر دیگر برخورد (concat) داشته باشند . اگر تداخل حاشیه ها اتفاق بیفتد فاصله بین دو عنصر بسیار بیشتر از چیزی می شود که انتظار داشتیم ، هنگامی که چنین اتفاقی می افتد فقط حاشیه های عنصر با مقدار margin بزرگتر اعمال خواهند شد و حاشیه ی عنصر با margin کوچکتر در نظر گرفته نشده و حذف می شود .

اگر عنصری فرزند عنصر دیگری باشد و به هر دو عنصر margin-top یا margin-bottom اختصاص داده باشیم در این صورت نیز تداخل حاشیه ها پیش خواهد آمد و margin عنصر پدر صرف نظر از کوچکتر یا بزرگتر بودن اعمال خواهد شد و margin عنصر فرزند در نظر گرفته نخواهد شد و حذف می شود .

می توان با تعریف padding یا border متناسب با محل تداخل برای عنصر پدر از برخورد دو margin در عناصر پدر و فرزند جلوگیری کرد .

استفاده از مقدار auto برای margin افقی باعث وسط چین شدن (centering) عنصر می شود .

مثال :

```
margin: 0 auto 0 0;      left-align
margin: 0 auto;         center-align
margin: 0 0 0 auto;    right-align
```

در مثال فوق دستور اول باعث چپ چین شدن عنصر ، دستور دوم باعث وسط چین شدن عنصر و دستور سوم باعث راست چین شدن عنصر می شود .

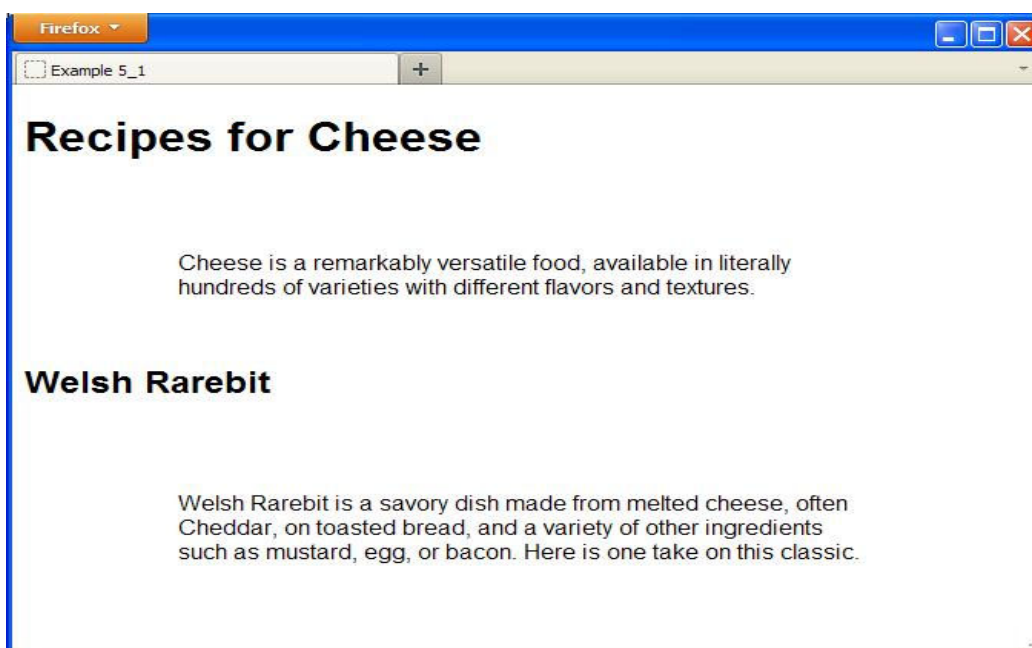
مثال : استفاده از خاصیت margin

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Example 5_1</title>
  <style type="text/css">
    .intro {
      margin-top: 70px;
      margin-left: 100px;
      margin-bottom: 50px;
      margin-right: 100px;
    }
  </style>
</head>
```

```

</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<h2>Welsh Rarebit</h2>
<p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
</body>
</html>

```



border

border در box model بین margin و padding قرار گرفته و یک خط به دور عنصر مورد نظر می کشد. استفاد از این خصوصیت مشاهده می سایر خصوصیات box model را آسان تر می کند.

border width

خاصیت border-width برای مشخص کردن پهنا یا عرض کادر (border) یک عنصر استفاده می شود. قاعده کلی:

```

border-top-width: thin | medium | thick | <length>
border-right-width:
border-bottom-width:
border-left-width:

```

مقدار پیش فرض : medium

مقادیر منفی برای length مجاز نمی باشد .

کوتاه نویسی خاصیت border-width

`border-width: [thin | medium | thick | <length>]{1,4}`

مقادیر percentage برای border قابل استفاده نیست کلمات کلیدی نیز به ندرت استفاده می شوند . معمولا از واحد pixel برای مقدار دهی به این خاصیت استفاده می شود .
همانند margin در کوتاه نویسی فوق نیز می توان از یک تا چهار مقدار استفاده کرد و قواعد گفته شده در کوتاه نویسی margin در اینجا نیز صدق می کنند .

border color

خاصیت border-color برای مشخص کردن رنگ border عناصر استفاده می شود .
قاعده کلی :

`border-color: <color> {1,4} | transparent`

`border-top-color: <color> | transparent`

`border-right-color:`

`border-bottom-color:`

`border-left-color:`

مقدار پیش فرض برای border-color برابر مقدار تعیین شده در خاصیت color از عنصر مورد نظر می باشد .
در حالت کوتاه نویسی می توان چهار طرف border را در یک خط و به طور جداگانه مقدار دهی کرد .
در IE6 و IE7 کلمه کلیدی transparent برای border-color پذیرفته نمی شود . در IE7 مقدار transparent به صورت black تفسیر می شود .

border style

قاعده کلی :

`border-style: none | dotted | dashed | solid | double | hidden | groove | ridge | inset | outset {1,4}`

`border-top-style:`

`border-right-style:`

`border-bottom-style:`

`border-left-style:`

مقدار پیش فرض : none

این خاصیت برای یک border باید تخصیص یابد تا border به نمایش درآید . یک تا چهار کلمه کلیدی می توان به این خاصیت تخصیص داد .
در حالت کوتاه نویسی می توان چهار طرف border را در یک خط و به طور جداگانه مقدار دهی کرد .
مثال :

```
p{
    border-style: solid dotted dashed double;
}
```

border top

قاعده کلی :

border-top: <border-top-width> || <border-style> || <color>

border-right:

border-bottom:

border:left:

مقدار پیش فرض ندارد .

این خاصیت مقدار کوتاه نویسی شده برای سه خاصیت width ، style و color از border می باشد . در این حالت برای border-style فقط یک مقدار می توان مشخص کرد .

مثال : استفاده از خاصیت های border-width , border-color , border-style

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Example 5_2</title>
<style type="text/css">
body {
    border-width: 3px 4px 5px 7px;
    border-color: red green blue #000;
    border-style: ridge dashed dotted double;
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p>Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
</body>
</html>
```



کوتاه نویسی خاصیت border

قاعده کلی :

```
border: <border-width> || <border-style> || <color>
```

مقدار پیش فرض ندارد .

خاصیت border خلاصه ای از تنظیم خصوصیات width ، style و color برای border یک عنصر می باشد .

مثال :

```
a:link{
    border:1px solid #cdf;
}
a:visited{
    border: 2px dashed red;
}
a:hover{
    border: medium double rgb(200,220,25);
}
a:focus{
    border: thick dotted rgb(10%,25%,50%);
}
a:active{
    border: thick double #ff0000;
}
```

padding

خصوصیت padding مقدار فضای خالی بین محتوای یک عنصر و border آن عنصر را تعیین می کند .

قاعده ی کلی :

```
padding: [ <length> | <percentage> ] {1,4}
padding-top: <length> | <percentage>
padding-right:
padding-bottom:
padding-left:
```

استفاده از مقادیر منفی برای padding مجاز نمی باشد .

مثال :

```
p{
    padding: 1em 4em 3em 2em;
}
```

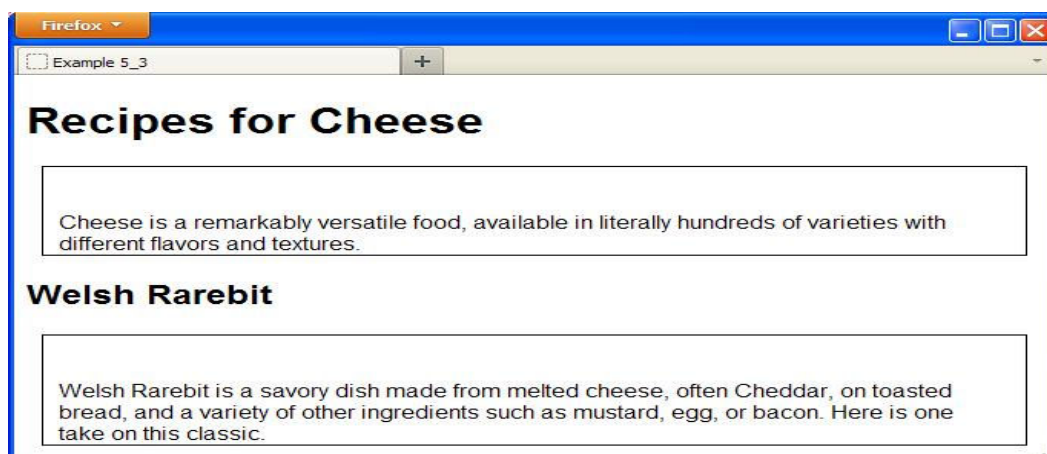
دستور فوق برای تمامی عناصر <p> موجود در صفحه padding-top: 1em; ، padding-right:4em; ، padding-bottom: 3em; و padding-left:2em; در نظر می گیرد .

تفاوت های بین padding و margin

۱. padding فضای خالی بین محتوا و border یک عنصر می باشد .
۲. کلمه کلیدی auto هیچ تاثیری روی خاصیت padding ندارد .
۳. خاصیت margin می تواند مقادیر منفی (negative) بگیرد اما خاصیت padding نمی تواند .
۴. در خاصیت padding تداخل وجود ندارد ، فقط در margin این اتفاق می افتد .

مثال : استفاده از خاصیت padding

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 5_3</title>
<style type="text/css">
.intro {
margin: 10px;
border: 1px solid #000;
padding: 40px 30px 0px 10px;
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<h2>Welsh Rarebit</h2>
<p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
</body>
</html>
```



width

خاصیت width عرض محتوا (content) یک عنصر را تعیین می کند . در نهایت عرض کلی یک عنصر برابر با عرض محتوا (تعیین شده توسط width) بعلاوه ی عرض padding و عرض border آن عنصر می باشد .
قاعده ی کلی :

width: <length> | <percentage> | auto

مقدار پیش فرض : auto مقدار عرض واقعی و مورد نیاز یک عنصر مقادیر منفی برای این خاصیت مجاز نیستند .

از خاصیت width می توان برای تخصیص یک مقدار کلی برای برخی از عناصر input مثل دکمه های reset و submit استفاده کرد .
مثال :

```
input.button{
    width: 10em;
}
```

دستور فوق برای تمامی عناصر input با کلاس button عرض 10em را در نظر می گیرد .

height

خاصیت height مشابه خاصیت width و قابل استفاده برای تمامی عناصر می باشد . این خاصیت پهنای عنصر را در بین padding-top و padding-bottom تعیین می کند . استفاده از خاصیت height برای عناصر چندان متداول نمی باشد و صفحات وب معمولاً با scroll عمودی طراحی می شوند . استفاده از این خاصیت ممکن است باعث ایجاد تاثیرات نا آشنا در صفحه گردد .
قاعده ی کلی :

height: <length> | <percentage> | auto

مقدار پیش فرض : auto

همانند خاصیت width از خاصیت height می توان برای مشخص کردن ابعاد یک تصویر استفاده کرد .
مثال :

```
img.foo{
    width: 40px;
    height:40px;
}
```

اغلب توصیه می شود که مقیاس یک تصویر در برنامه های مخصوص این کار تغییر کرده و سپس در صفحه وب استفاده شود و کمتر از خصوصیات فوق در مورد تصاویر استفاده شود . زیرا استفاده از خصوصیات فوق باعث کاهش کیفیت تصویر می شود و باعث می شود که کاربر یک فایل غیر ضروری با حجم بالا را download کند . با این حال خصوصیات width و height خصوصیات مناسبی برای غلبه بر مشکلات بصری در طراحی سبک صفحات وب می باشد .

ابعاد max و min برای width و height

طراحی با عرض (width) متغیر هنگامی استفاده می شود که می خواهید محتویات (content) را برای مطابقت با resolution های مختلف صفحه نمایش طراحی کنید . همچنین هنگامی مفید می باشد که می خواهید عرض عنصر منبسط یا منقبض نشود .

(stop stretching , stop contracting)

هنگامی که شما صفحه ای را اصولاً برای نمایش در resolution , 800*600 یا 1024*768 طراحی می کنید ، اگر کاربری سایت شما را با resolution 1600*1200 مشاهده کند صفحه را به شکلی متفاوت نه چندان زیبا خواهد دید ، یا هنگام استفاده از کلمه کلیدی auto یا مقادیر percentage ممکن است مشاهده با resolution های مختلف نتایج متفاوتی داشته باشد . در این مواقع استفاده از مقادیر min و max برای width و height می تواند مانع از تغییر ظاهر صفحه در resolution های مختلف شود .

قاعده ی کلی برای min :

min-width: <length> | <percentage>

min-height: <length> | <percentage>

مقدار پیش فرض : صفر

خصوصیات min-width و min-height محدودیت کوچکترین اندازه را برای یک عنصر تعریف می کنند .

IE6 و قبل از آن از این خصوصیات پشتیبانی نمی کنند .

خصوصیات فوق هنگامی تعریف می شوند که می خواهیم از منقبض شدن (shrinking) یک عنصر برای متناسب شدن با پنجرهی کاربر یا محتویات عنصر جلوگیری کنیم .

مثال :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Exmample 5_4</title>
<style type="text/css">
body {
min-width: 1000px;
min-height: 250px;
border: 1px solid #666;
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
</body>
</html>
```



در مثال فوق تعریف `min-width: 1000px;` برای عنصر `body` مانع از منقبض شدن محتوای صفحه شده و صفحه `scroll` افقی پیدا می کند .
قاعده ی کلی برای `max` :

`max-width: <length> | <percentage> | auto`
`max-height: <length> | <percentage> | auto`

مقدار پیش فرض : `auto`

خصوصیات `max` محدودیت بالاتر و حد اکثر اندازه را برای یک عنصر مشخص می کنند و هنگامی تعریف می شوند که ابعاد یک عنصر ممکن است افزایش یابند که با استفاده از این خصوصیات می توان مانع از منبسط شدن و افزایش ابعاد عناصر شد .

overflow

این خصوصیت هنگامی استفاده می شود که محتویات یک عنصر به دلیل محدودیت در ابعاد تعریف شده برای آن امکان سرریز دارند . خاصیت `overflow` نحوه ی برخورد با سرریز را مشخص می کند .
قاعده ی کلی :

`overflow: visible | hidden | scroll | auto`

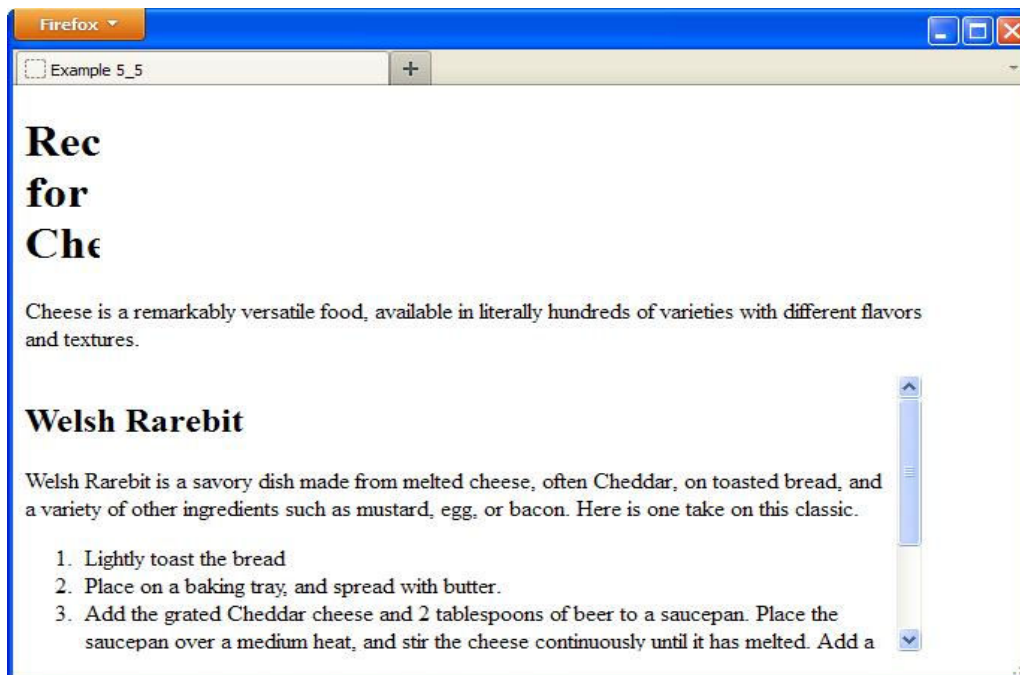
مقدار پیش فرض : `visible`

- ✓ `hidden` : اگر محتویات از ابعاد عنصر بزرگتر باشند مقدار اضافی مخفی خواهد شد .
- ✓ `visible` : اگر محتویات از ابعاد عنصر بزرگتر باشند مقدار اضافی به شکل سرریز نمایش داده خواهد شد .
- ✓ `scroll` و `auto` : مشابه یکدیگر می باشند با تفاوت که استفاده از مقدار `scroll` باعث می شود حتی زمانیکه محتوا کمتر از ابعاد عنصر است scrollbar نمایش داده شود و مقدار `auto` باعث می شود که scrollbar فقط در مواقع ضروری نمایش داده شود .

مثال : استفاده از خاصیت `overflow`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
```

```
<title>Example 4_7</title>
<style type="text/css">
  body {
    width: 600px;
  }
  h1 {
    width: 50px;
    overflow: hidden;
  }
  .recipe {
    height: 200px;
    overflow: auto;
  }
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p>Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<div class="recipe">
  <h2>Welsh Rarebit</h2>
  <p>Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
  <ol>
    <li>Lightly toast the bread</li>
    <li>Place on a baking tray, and spread with butter.</li>
    <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
Place the saucepan over a medium heat, and stir the cheese continuously
until it has melted. Add a teaspoon of wholegrain mustard and grind in
a little pepper. Keep stirring.</li>
    <li>When thick and smooth, pour over each piece of toast spreading it to
the edges to stop the toast from burning.</li>
    <li>Place under the grill for a couple of minutes or until golden brown.</li>
  </ol>
</div>
</body>
</html>
```



overflow-x , overflow-y

این خصوصیات مشابه overflow می باشند و اجازه ی کنترل overflow را با scrollbar افقی یا عمودی به طور جداگانه در اختیار می گذارد .

قاعده ی کلی :

overflow-x: visible | hidden | scroll | auto

overflow-y: visible | hidden | scroll | auto

مقدار پیش فرض : visible

فصل ششم

Floating and Vertical Alignment

float

از خاصیت float برای قرار دادن محتوا یا عناصر در کنار یکدیگر استفاده می شود. تاثیر float فی نفسه بر روی ccs box model است. پس از اینکه float برای یک عنصر پذیرفته شد صرف نظر از نوع آن عنصر آن عنصر شبیه به عناصر block رفتار خواهد کرد. (عناصر block عناصری هستند که ابعاد آنها با استفاده از width , height , padding , border , margin تعریف شده باشد).
قاعده ی کلی :

float: left | right | none

مقدار پیش فرض : none (هیچ)

✓ left : عنصری که این خاصیت برای آن تخصیص یابد در سمت چپ صفحه و یا در سمت چپ عنصر پدر خود قرار خواهد گرفت و عناصر پس از آن نیز در کنار این عنصر شناور خواهند شد.

✓ right : عنصری که این خاصیت برای آن تخصیص یابد در سمت راست صفحه و یا در سمت راست عنصر پدر خود قرار خواهد گرفت و عناصر پس از آن نیز در کنار این عنصر شناور خواهند شد.

floating box model

- در حاشیه (margin) عناصر شناور (float) امکان تداخل (فروپاشی) وجود ندارد.
- در عناصر هم جوار عنصر floated فقط محتوای این عناصر (content) تحت تاثیر خاصیت float عنصر floated قرار می گیرند.
- یک floated element همیشه مانند یک block element رفتار می کند.

نکته : منظور از عنصر floated عنصری است که خاصیت float برای آن تعریف شده است و منظور از عناصر همجوار (یا عناصر float یا شناور) عناصری است که در ترتیب دستورات HTML بعد از عنصر floated قرار می گیرند.

خاصیت float برای ایجاد و تنظیم بخش navigation سایت مناسب می باشد.

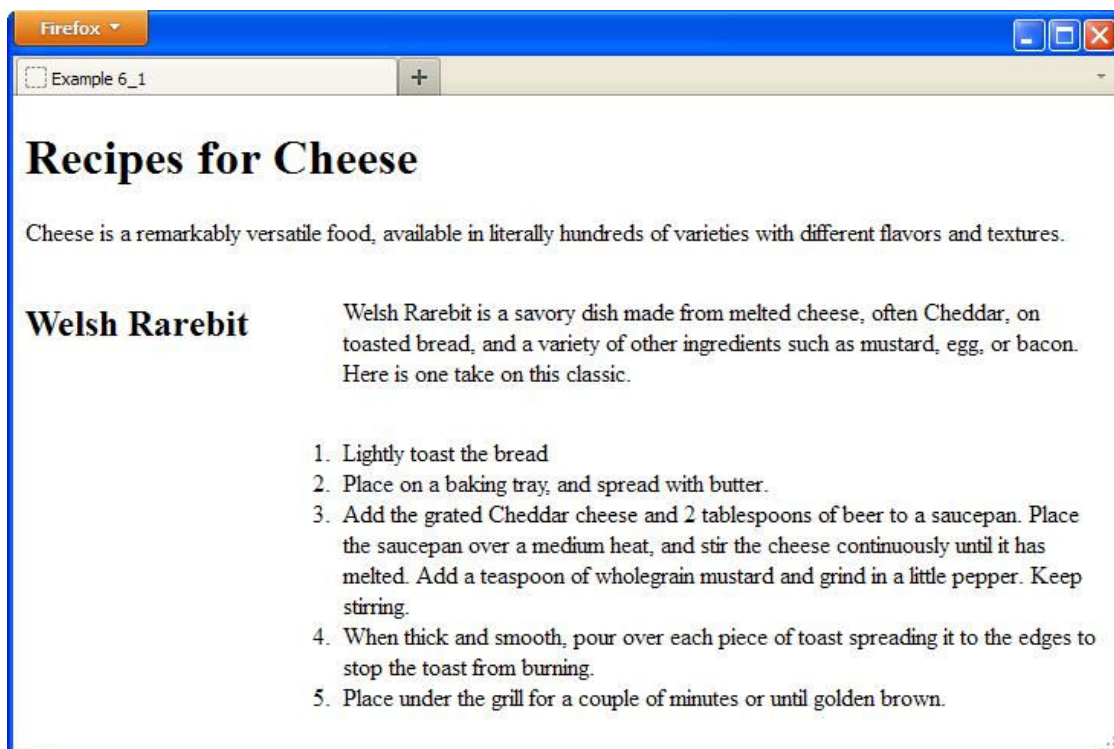
مثال : استفاده از خاصیت float

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Example 6_1</title>
<style type="text/css">
.recipe h2 {
  float: left;
}
.recipe .intro,
.recipe ol {
  float: right;
  width: 500px;
```

```

}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<div class="recipe">
  <h2>Welsh Rarebit</h2>
  <p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
  <ol>
    <li>Lightly toast the bread</li>
    <li>Place on a baking tray, and spread with butter.</li>
    <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
Place the saucepan over a medium heat, and stir the cheese continuously
until it has melted. Add a teaspoon of wholegrain mustard and grind in
a little pepper. Keep stirring.</li>
    <li>When thick and smooth, pour over each piece of toast spreading it to
the edges to stop the toast from burning.</li>
    <li>Place under the grill for a couple of minutes or until golden brown.</li>
  </ol>
</div>
</body>
</html>

```



مثال : ایجاد بخش نوبری بالای سایت با استفاده از float

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Example 6_2</title>
<style type="text/css">
  body {
    width: 600px;
    margin: 1em auto;
  }
  h1 {
    float: left;
    margin-top: 0;
  }
  .navigation {
    float: right;
    margin: 0;
    list-style: none;
  }
  .navigation li {
    float: left;
  }
  .navigation a {
    display: block;
    margin-left: 0.5em;
    padding: 0.5em;
    border: 1px solid #CCC;
    color: #233;
    text-decoration: none;
  }
  .navigation a:focus,
  .navigation a:hover {
    background: #aaa;
    color: #FFF;
  }
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<ul class="navigation">
  <li><a href="#">Home</a></li>
```

```

<li><a href="#">Recipes</a></li>
<li><a href="#">Suggestions</a></li>
</ul>
</body>
</html>

```



clear

خاصیت clear وابسته به خاصیت float بوده و برای کنترل خاصیت float به کار می رود .
قاعده ی کلی :

clear: none | left | right | both

مقدار پیش فرض : none

- ✓ left : تاثیر خاصیت float: left; را از بین می برد ؛ یعنی باعث می شود که تاثیر left floated element بر روی عناصر دیگر از بین برود .
- ✓ right : تاثیر خاصیت float: right; را از بین می برد .
- ✓ both : تاثیر هر دو خاصیت float: left; و float: right; را از بین می برد .

یک توضیح ساده برای خاصیت clear این است که این خاصیت برای cancel کردن تاثیر یک یا تعداد بیشتری از عناصر floated بر روی دیگر عناصر استفاده می شود . تاثیر یک عنصر floated بر روی دیگر عناصر این است که عناصر همجوار این عنصر به صورت شناور در کنار این عنصر قرار خواهند گرفت . به منظور از بین بردن تاثیر عنصر floated بر روی عناصر همجوار از خاصیت clear استفاده می کنند . اگر خاصیت clear برای عنصری تعریف شود این عنصر در کنار عنصر floated شناور نخواهد شد .

اگر خاصیت float برای یک عنصر تعریف شده باشد و ما بخواهیم که عناصر پیرو (یا پی آمد follow) این عنصر که این عنصر floated را احاطه کرده اند شکسته (wrap) نشوند از تعریف خاصیت clear استفاده می کنیم .
مثال :

```

...
<style type="text/css" >
img{
float: right;
}
p{

```



```

clear: right;
}
</style>

```

...

```


<p> this is some text in a paragraph . . . </p>

```

...

در مثال فوق اگر از خاصیت clear برای عنصر p استفاده نکنیم محتوای عنصر p در کنار عکس مورد نظر قرار خواهند گرفت اما در صورت استفاده از خاصیت clear محتوای عنصر p در زیر عکس قرار خواهند گرفت .

مثال : استفاده از خاصیت clear

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Example 6_3</title>
<style type="text/css">
body {
width: 600px;
margin: 1em auto;
h1 {
float: left;
margin-top: 0;
}
.navigation {
float: right;
margin: 0;
list-style: none;
}
.navigation li {
float: left;
}
.navigation a {
display: block;
margin-left: 0.5em;
padding: 0.5em;
border: 1px solid #CCC;
color: #233;
text-decoration: none;
}
.navigation a:focus,
.navigation a:hover {

```

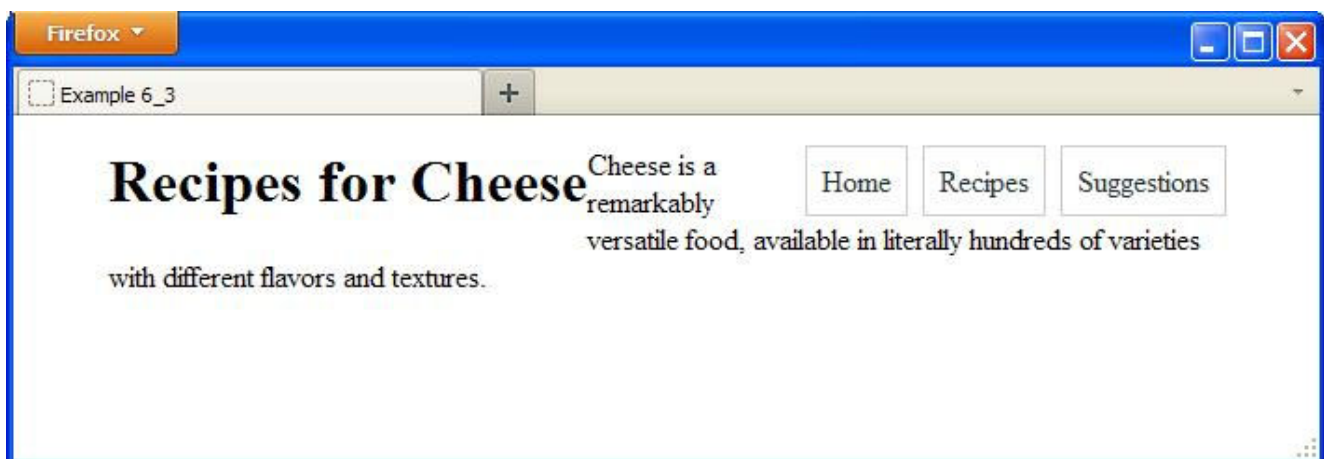
```

background: #aaa;
color: #FFF;
}
.intro{
clear:both;
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<ul class="navigation">
<li><a href="#">Home</a></li>
<li><a href="#">Recipes</a></li>
<li><a href="#">Suggestions</a></li>
</ul>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
</body>
</html>

```



در مثال فوق اگر از خاصیت clear استفاده نکنیم صفحه به هم ریخته خواهد شد . همانند شکل زیر :



vertical align

این خاصیت به منظور تراز عمودی عناصر inline مثل `<a>` , `` , `` و ... استفاده می شود . اگر از این خاصیت برای تراز کردن سلول های یک table استفاده شود معنای دیگری خواهد داشت .

این خاصیت به عنصر والد یا عناصر خطی بستگی دارد .

- عناصر block : عناصری هستند که قبل و بعد از آنها یک خط خالی وجود دارد ؛ مثل `<p>` و `<div>` و ... در HTML .
- عناصر inline (خطی) : عناصر خطی عناصری هستند که قبل و بعد از آنها خط خالی وجود ندارد ؛ مثل `<a>` و `` در HTML .

قاعده ی کلی :

vertical-align: baseline | _{sub} | ^{super} | top | text-top | middle | bottom | text-bottom |
<percentage> | <length>

کلمات کلیدی زیر موقعیت عمودی را با توجه به عنصر والد مشخص می کنند :

- ✓ baseline : baseline خطی است فرضی که همتراز با قسمت پایینی حروف یک کلمه یا جمله می باشد . مثلا خط فرضی که کلمه ی sit روی آن می نشیند .
- ✓ middle : تراز با میانه ی خط
- ✓ sub : تراز با بالای خط مثل ² ؛ معادل تگ `<sub>` در HTML
- ✓ super : تراز با پایین خط و ایجاد یک اندیس مثل ₂ ؛ معادل تگ `<sup>` در HTML
- ✓ text-top : متن را با بالای بلندترین حروف اطراف تراز می کند .
- ✓ text-bottom : متن را با کوتاه ترین یا پایین ترین حرف تراز می کند ، برای مثال p , y , g , زیر خط اصلی (baseline) قرار می گیرند و متن با پایین ترین قسمت این حروف تراز خواهد شد .

کلمات کلیدی زیر موقعیت را با توجه به خطی که عنصر در آن قرار دارد مشخص می کنند :

- ✓ top : قسمت بالای عنصر را با بلندترین عنصر موجود در خط تراز می کند .
- ✓ bottom : پایین عنصر را با کوتاه ترین عنصر موجود در خط تراز می کند .

استفاده از مقادیر percentage و length بستگی به خاصیت line-height تعریف شده برای عنصر مورد نظر دارد .

مثال : استفاده از vertical-align:50%; متن را در وسط خط تراز می کند .

مثال :

```
p{
    line-height: 10px;
    vertical-align: 5px;
}
```

در مثال فوق line-height برابر 10px می باشد و از آنجاییکه vertical-align برابر 5px و نصف مقدار line-height می باشد متن در وسط خط تراز خواهد شد .

خاصیت vertical-align به طور خاص برای تراز بندی تصاویر مفید است .

مثال :

```
img.middle{
    vertical-align: middle;
}
img{
    vertical-align:50%;
}
.exponent{
    vertical-align: super;
}
```

خاصیت `vertical-align` هنگام استفاده برای سلول های `table` شکل کاملا متفاوتی به خود می گیرد . این خاصیت هنگامی که برای سلول های `table` استفاده می شود فقط مجاز به استفاده از مقادیر زیر می باشد :

baseline , top , middle , bottom

مثال : استفاده از خاصیت `vertical-align`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Example 6_4</title>
<style type="text/css">
    h1 span {
        font-size: 0.5em;
    }
    .top {
        vertical-align: top;
    }
    .middle {
        vertical-align: middle;
    }
    .bottom {
        vertical-align: bottom;
    }
</style>
</head>
<body>
<h1>
<span class="top">Top</span>
<span class="middle">Middle</span> Recipes for
<span class="bottom">Buttom</span> Cheese
</h1>
</body>
</html>
```



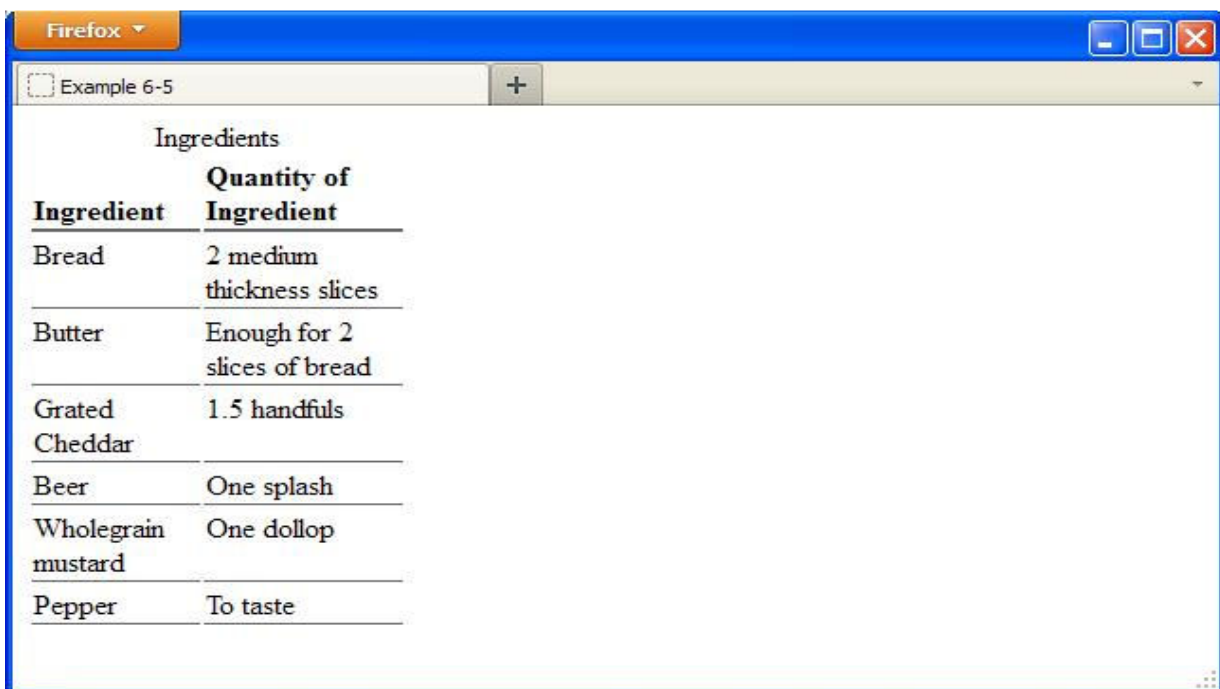
مثال : استفاده از خاصیت vertical-align در جداول

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Example 6-5</title>
<style type="text/css">
table {
width: 200px;
text-align: left;
}
th {
vertical-align: bottom;
border-bottom: 2px solid #666;
}
td {
vertical-align: top;
border-bottom: 1px solid #666;
}
</style>
</head>
<body>
<table>
<caption>Ingredients</caption>
<thead>
<tr>
<th>Ingredient</th>
<th>Quantity of Ingredient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bread</td>
```

```

    <td>2 medium thickness slices</td>
</tr>
<tr>
    <td>Butter</td>
    <td>Enough for 2 slices of bread</td>
</tr>
<tr>
    <td>Grated Cheddar</td>
    <td>1.5 handfuls</td>
</tr>
<tr>
    <td>Beer</td>
    <td>One splash</td>
</tr>
<tr>
    <td>Wholegrain mustard</td>
    <td>One dollop</td>
</tr>
<tr>
    <td>Pepper</td>
    <td>To taste</td>
</tr>
</tbody>
</table>
</body>
</html>

```



The screenshot shows a Firefox browser window with a single tab titled "Example 6-5". The page content is a table with the following structure:

Ingredients	
Ingredient	Quantity of Ingredient
Bread	2 medium thickness slices
Butter	Enough for 2 slices of bread
Grated Cheddar	1.5 handfuls
Beer	One splash
Wholegrain mustard	One dollop
Pepper	To taste

فصل هفتم

Styling Lists

لیست ها عناصری با کاربرد های بسیار متنوع هستند . از جمله برای بخش navigation سایت و برای ایجاد کنترل های tab استفاده می شوند . navigation شامل لیستی از لینک ها است که دارای ویژگی های بصری می باشد و ثابت و خشک نیست .

list style type

با استفاده از این خاصیت می توان برای لیست های مرتب و نامرتب bullet تعریف کرد .
قاعده ی کلی :

list-style-type: disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman
| lower-greek | lower-latin | upper-latin | armenian | georgian | none

مقدار پیش فرض برای لیست های نامرتب (ul) : disc

مقدار پیش فرض برای لیست های مرتب (ol) : decimal

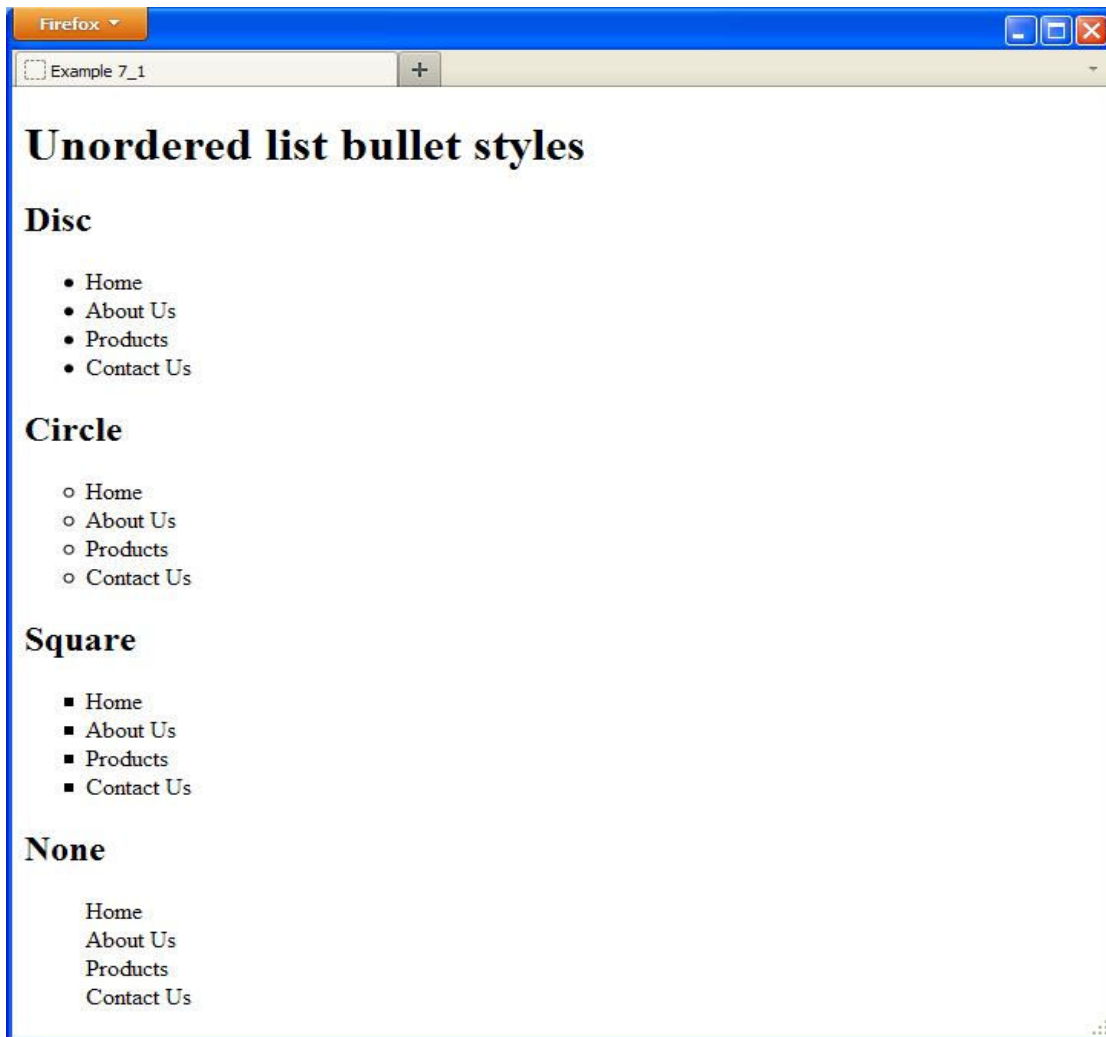
ممکن است در هنگام تعریف خصوصیات فوق bullet لیست های مرتب را برای لیست های نامرتب در نظر بگیریم و برعکس . شدیداً توصیه می شود که این کار انجام نشود و برای هر نوع از لیست ها bullet مخصوص به همان لیست را با توجه به معنای آن در نظر گرفت .

bullet های مخصوص لیست های نامرتب عبارتند از : disc , circle , square و none . مقدار پیش فرض در این نوع لیست ها disc می باشد .

مثال : استفاده از خاصیت list-style-type در لیست های نامرتب

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Example 7_1</title>
<style type="text/css">
.demo-disc {
list-style-type: disc;
}
.demo-circle {
list-style-type: circle;
}
.demo-square {
list-style-type: square;
}
.demo-none {
list-style-type: none;
}
</style>
</head>
```

```
<body>
<h1>Unordered list bullet styles</h1>
<h2>Disc</h2>
<ul class="demo-disc">
  <li>Home</li>
  <li>About Us</li>
  <li>Products</li>
  <li>Contact Us</li>
</ul>
<h2>Circle</h2>
<ul class="demo-circle">
  <li>Home</li>
  <li>About Us</li>
  <li>Products</li>
  <li>Contact Us</li>
</ul>
<h2>Square</h2>
<ul class="demo-square">
  <li>Home</li>
  <li>About Us</li>
  <li>Products</li>
  <li>Contact Us</li>
</ul>
<h2>None</h2>
<ul class="demo-none">
  <li>Home</li>
  <li>About Us</li>
  <li>Products</li>
  <li>Contact Us</li>
</ul>
</body>
</html>
```

bullet های مخصوص لیست های مرتب عبارتند از :

decimal , decimal-leading-zero , lower-greek , lower-latin , upper-latin , armenian , georgian , none

نکته : کلمات کلیدی decimal-leading-zero , lower-greek , lower-latin , upper-latin , armenian , georgian در IE6 و IE7 پشتیبانی نمی شوند .

مقادیر decimal , lower-roman , upper-roman و none توسط تمامی مرورگرها پشتیبانی می شوند .

مثال : استفاده از خاصیت list-style-type در لیست های مرتب

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Example 7_2</title>
<style type="text/css">
.demo-decimal {
list-style-type: decimal;
}
.demo-lower-roman {
list-style-type: lower-roman;
```

```

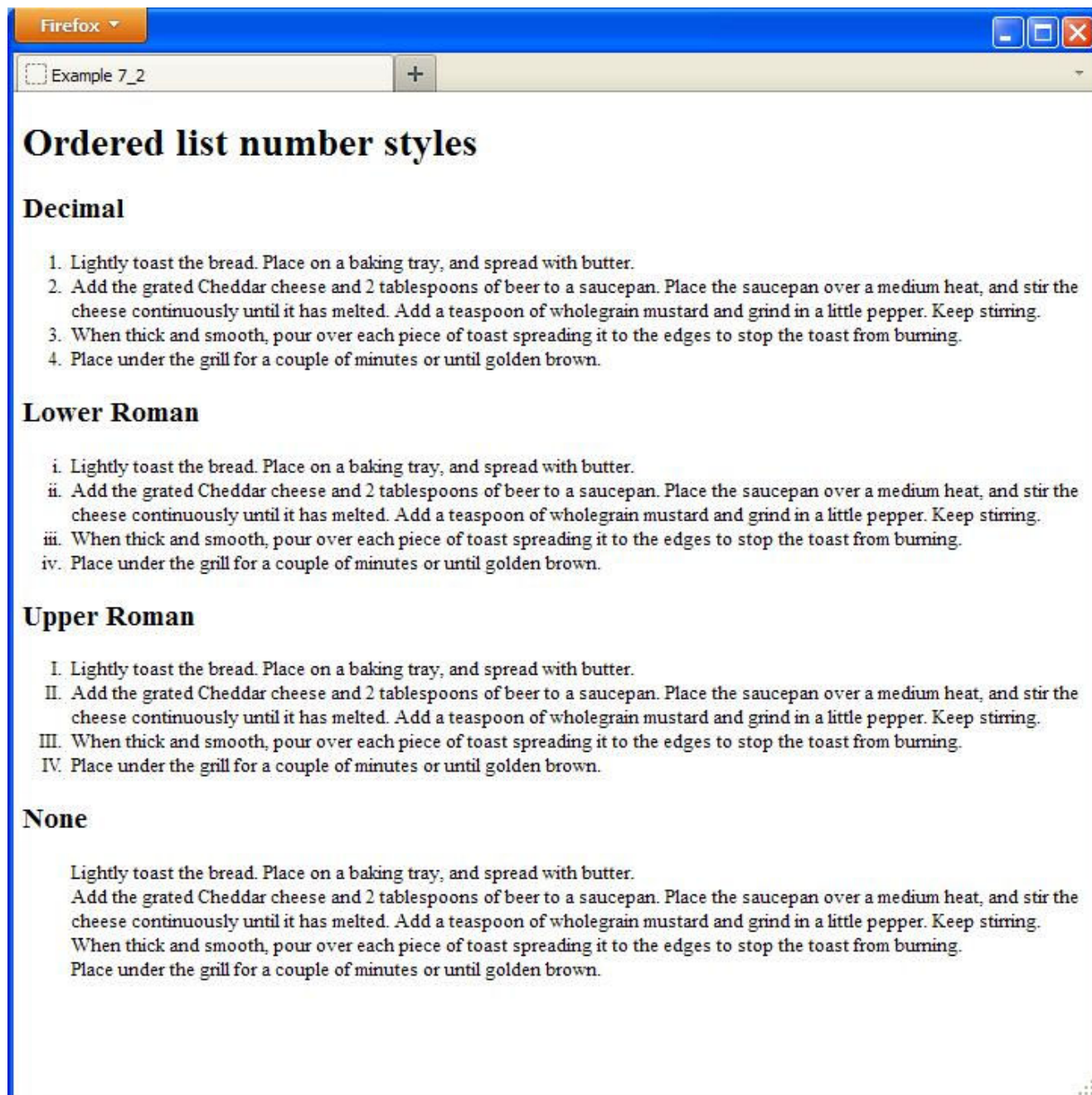
}
.demo-upper-roman {
  list-style-type: upper-roman;
}
.demo-none {
  list-style-type: none;
}
</style>
</head>
<body>
<h1>Ordered list number styles</h1>
<h2>Decimal</h2>
<ol class="demo-decimal">
  <li>Lightly toast the bread. Place on a baking tray, and spread with butter.</li>
  <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
  Place the saucepan over a medium heat, and stir the cheese continuously until it
  has melted. Add a teaspoon of wholegrain mustard and grind in a little pepper. Keep stirring.</li>
  <li>When thick and smooth, pour over each piece of toast spreading it to the edges to stop the
  toast from burning.</li>
  <li>Place under the grill for a couple of minutes or until golden brown.</li>
</ol>
<h2>Lower Roman</h2>
<ol class="demo-lower-roman">
  <li>Lightly toast the bread. Place on a baking tray, and spread with butter.</li>
  <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
  Place the saucepan over a medium heat, and stir the cheese continuously until it
  has melted. Add a teaspoon of wholegrain mustard and grind in a little pepper. Keep stirring.</li>
  <li>When thick and smooth, pour over each piece of toast spreading it to the
  edges to stop the toast from burning.</li>
  <li>Place under the grill for a couple of minutes or until golden brown.</li>
</ol>
<h2>Upper Roman</h2>
<ol class="demo-upper-roman">
  <li>Lightly toast the bread. Place on a baking tray, and spread with butter.</li>
  <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
  Place the saucepan over a medium heat, and stir the cheese continuously until it
  has melted. Add a teaspoon of wholegrain mustard and grind in a little pepper. Keep stirring.</li>
  <li>When thick and smooth, pour over each piece of toast spreading it to the
  edges to stop the toast from burning.</li>
  <li>Place under the grill for a couple of minutes or until golden brown.</li>
</ol>
<h2>None</h2>
<ol class="demo-none">
  <li>Lightly toast the bread. Place on a baking tray, and spread with butter.</li>
  <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.

```

Place the saucepan over a medium heat, and stir the cheese continuously until it has melted. Add a teaspoon of wholegrain mustard and grind in a little pepper. Keep stirring.

When thick and smooth, pour over each piece of toast spreading it to the edges to stop the toast from burning.

Place under the grill for a couple of minutes or until golden brown.



list style imgsge

این خاصیت یک عکس را برای استفاده به عنوان نشانه (bullet) در کنار عناصر لیست تعریف می کند. این عکس با دشنانه تعریف شده در خاصیت list-style-type جایگزین می شود.

قاعده ی کلی :

`list-style-image: <url> | none`

مقدار پیش فرض : none

مثال :

```
ul.check{
    list-style-image: url(./images/bullet1.gif|);
}
ul.li.x{
    list-style-image: url(./images/bullet2.gif);
}
```

list style position

این خاصیت موقعیت و نحوه ی قرار گرفتن **bullet** در کنار آیتم های لیست را مشخص می کند .

قاعده ی کلی :

`list-style-position: inside | outside`

مقدار پیش فرض : outside

- ✓ **inside** : مشخص می کند که **bullet** لیست باید در داخل آیتم های لیست قرار گیرد .
- ✓ **outside** : مشخص می کند که **bullet** لیست باید در خارج از آیتم های لیست قرار گیرد .

مثال :

List Style Position

- The markers for these list items are on the inside.
 - One
 - Two
- The markers for these list items are on the outside.
 - One
 - Two

list style

این خاصیت مقدار کوتاه نویسی شده ی سه خاصیت قبل می باشد .

قاعده ی کلی :

`list-style: <list-style-type> | <list-style-position> | <url>`

از یک تا سه مقدار می توان در این خاصیت استفاده کرد .

مثال :

```
li.square{
    list-style: square inside;
}
```

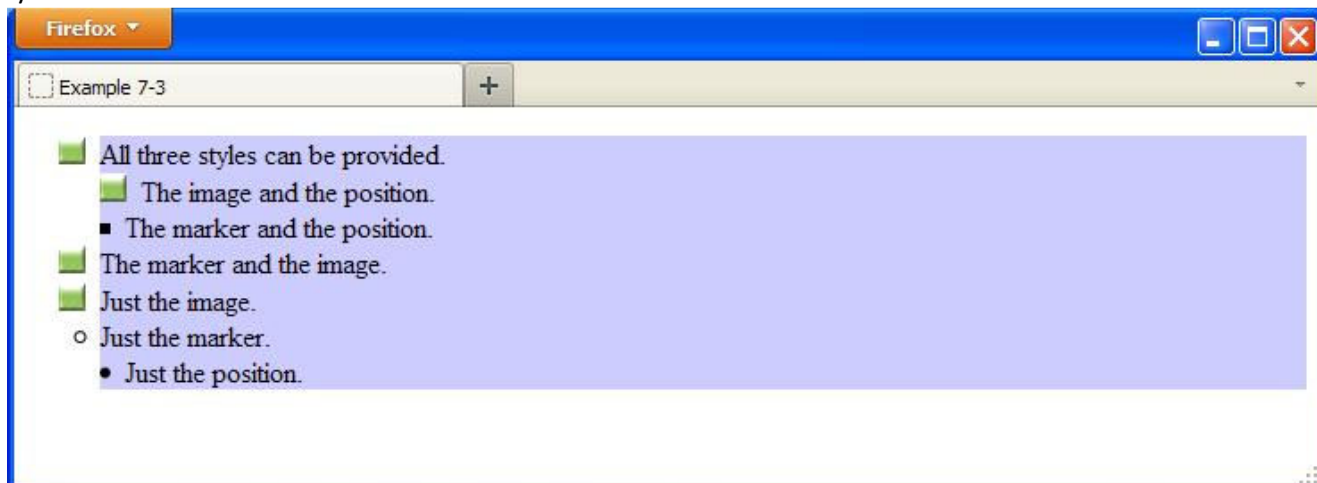
```
ul.plain{
    list-style: none;
}
ul.check{
    list-style: url(/images/bullet3.png) circle;
}
```

در دستور سوم از مثال فوق هر دو خاصیت `list-style-type` و `list-style-image` برای لیست تعریف شده اند ، در این حالت اگر عکس تعریف شده در خاصیت `list-style-image` به هر دلیلی `load` نشود ، مقدار تعریف شده در خاصیت `list-style-type` نمایش داده خواهد شد .

مثال : استفاده از خاصیت `list-style`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Example 7-3</title>
<style type="text/css">
li {
    background: #CCF;
}
.arrow {
    list-style: square url(118546463.bmp) outside;
}
.arrow-inside {
    list-style: url(118546463.bmp) inside;
}
.marker-inside {
    list-style: square inside;
}
.marker-image {
    list-style: square url(118546463.bmp);
}
.arrow-only {
    list-style: url(118546463.bmp);
}
.marker {
    list-style: circle;
}
.position {
    list-style: inside;
}
</style>
</head>
<body>
<ul>
<li class="arrow">All three styles can be provided.</li>
<li class="arrow-inside">The image and the position.</li>
<li class="marker-inside">The marker and the position.</li>
<li class="marker-image">The marker and the image.</li>
<li class="arrow-only">Just the image.</li>
<li class="marker">Just the marker.</li>
<li class="position">Just the position.</li>
```

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



display

قاعده ی کلی :

display: block | inline | list-item

✓ **block** : یک عنصر block ایجاد می کند .

اگر این خاصیت برای یک عنصر inline تعریف شود این عنصر به صورت block در صفحه نمایش داده خواهد شد ؛ یعنی یک خط خالی قبل و بعد از این عنصر ایجاد خواهد شد .

✓ **inline** : یک عنصر inline ایجاد میکند .

اگر این خاصیت برای یک عنصر block تعریف این عنصر به صورت inline در صفحه نمایش داده خواهد شد ؛ یعنی خط خالی قبل و بعد از عنصر block حذف خواهد شد .

- عناصر block : عناصری هستند که قبل و بعد از آنها یک خط خالی وجود دارد ؛ مثل <p> و <div> و و ... در HTML .
- عناصر inline (خطی) : عناصر خطی عناصری هستند که قبل و بعد از آنها خط خالی وجود ندارد ؛ مثل <a> و در HTML .

فصل هشتم

Positioning

positioning به شما این امکان را می دهد که موقعیت هر عنصر را به دقت در یک سند مشخص کنید . می توان مشخص کرد که موقعیت یک عنصر به عنصر پدر خود وابسته باشد یا به پنجره ی مرورگر . با استفاده از positioning می توان یک عنصر را بر روی عنصر دیگر لایه بندی کرد . positioning شامل خاصیت position و چهار خاصیت top , right , bottom , left می باشد که مختصات محل قرارگیری عنصر یا offset عنصر را مشخص می کنند .

قاعده ی کلی :

position: static | relative | absolute | fixed

مقدار پیش فرض static : position

- ✓ static : پیش فرض مرورگر می باشد و همان حالت و ترتیب عادی قرار گرفتن عناصر در ترتیب سند HTML می باشد .
- ✓ absolute : اجازه می دهد که عنصر در یک محل خاص از سند قرار گیرد .
- ✓ relative : بسیار شبیه به static می باشد ، عناصری که relative positioning را پذیرفته اند جریان عادی سند را ترک نمی کنند.
- ✓ fixed : عناصر fixed همواره در محل خود باقی می مانند حتی اگر سند scroll شود .

قاعده ی کلی :

top: <length> | <percentage> | auto

right:

bottom:

left:

مقدار پیش فرض auto : top , right , bottom , left

به چهار خاصیت top , right , bottom , left در اصطلاح offset property گفته می شود . این چهار خاصیت برای کنترل موقعیت عناصر با مقادیر absolute , relative و fixed به کار می روند .

static , absolute

اگر از مقدار static برای خاصیت position که مقدار default نیز می باشد استفاده کنیم و در صورت استفاده از لیست های تو در تو برای قسمت navigation سایت لیست داخلی شکسته خواهد شد و صفحه آرابی ما شکسته شده و قسمت navigation ظاهر خوبی نخواهد داشت . مطلب فوق مثالی است از اینکه چه زمانی absolute positioning مفید می باشد . استفاده از absolute باعث می شود که فضای خالی عناصر بیش از حد نباشد و موقعیت یک عنصر بر موقعیت دیگر عناصر تاثیری نداشته باشد .

عناصر absolute به طور پیش فرض وابسته به نمایی از صفحه که در دید قرار دارد موقعیت دهی می شوند . عناصر absolute , relative و fixed را می توان با استفاده از offset property ها موقعیت دهی کرد . چهار ترکیب مجاز و قابل استفاده از offset property ها عبارتند از :

- top , left ✓
- top , right ✓
- bottom , left ✓
- bottom , right ✓

ترکیب هایی به غیر از ترکیب های فوق توسط مرورگر نادیده گرفته خواهند شد .

top ,left

در این حالت لبه ی بالایی عنصر وابسته به نمای دید بالایی (top viewport) و لبه ی چپ عنصر وابسته به نمای دید چپ (left viewport) می باشد .
سه ترکیب ممکن دیگر از offset property ها نیز مشابه top , left می باشند .

مثال : استفاده از absolute positioning

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

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 8-1</title>
<style type="text/css">
body {
width: 600px;
margin: 1em auto;
}

h1 {
float: left;
margin-top: 0.2em;
}

.navigation {
float: right;
}

.navigation li {
display: inline;
}

.navigation a {
margin-left: 0.5em;
padding: 0.5em;
border: 1px solid #CCC;
}

.navigation ul {
position: absolute;
top: 20px;
left: 40px;
}

.navigation ul a {
display: block;
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<ul class="navigation">
<li><a href="#">Home</a></li>
<li>
```



```

<a href="#">Recipes</a>
<ul>
  <li><a href="#">Snacks</a></li>
  <li><a href="#">Meals</a></li>
  <li><a href="#">Desserts</a></li>
</ul>
</li>
<li><a href="#">Suggestions</a></li>
</ul>
</body>
</html>

```



relative positioning

بسیار شبیه به static می باشد ، عناصری که relative positioning را پذیرفته اند جریان عادی سند را ترک نمی کنند . سه تفاوت بین عناصر static و relative وجود دارد :

۱. عناصر relative می توانند یک نقطه ارجاع (point of reference) برای عناصر فرزند خود که به صورت absolute تعریف شده اند باشند .
۲. عناصر relative را می توان با استفاده از offset property موقعیت دهی کرد اما offset property ها در عناصر static کاربردی ندارند .
۳. عناصر relative می توانند یک موقعیت در محور Z (z-axis) داشته باشند (با استفاده از خاصیت z-index) .

در قسمت navigation (ناوبری) سایت موقعیت یک sub list می تواند به جای وابسته بودن به viewport به عنصر پدری که این sub list در آن قرار دارد وابسته باشد ؛ یعنی اگر عنصر li را به صورت relative تعریف کنیم و یک عنصر ul به صورت sub list درون این li قرار گیرد و این ul به صورت absolute تعریف شده باشد ، در این صورت موقعیت ul وابسته به عنصری است که درون آن قرار گرفته (عنصر پدر) و با توجه به آن موقعیت دهی می شود .

مثال : استفاده از relative positioning

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 8-2</title>
<style type="text/css">
  body {
    width: 600px;
    margin: 1em auto;

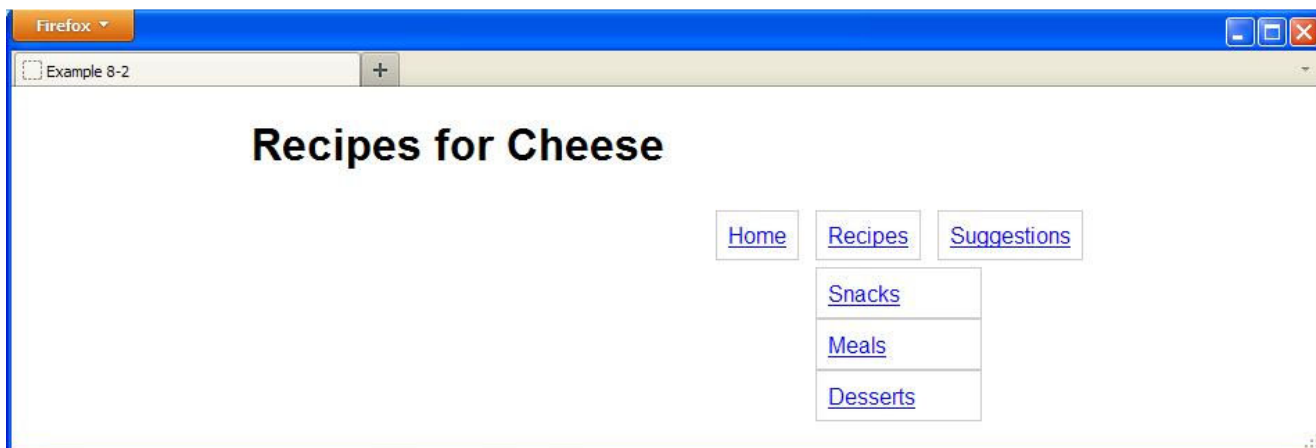
```

```

}

h1 {
  float: left;
  margin-top: 0.2em;
}
.navigation {
  float: right;
}
.navigation li {
  position: relative;
  display: inline;
}
.navigation a {
  margin-left: 0.5em;
  padding: 0.5em;
  border: 1px solid #CCC;
}
        .navigation ul {
  position: absolute;
  top: 2em;
  left: 0;
  width: 8em;
  padding: 0;
}
.navigation ul a {
  display: block;
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<ul class="navigation">
  <li><a href="#">Home</a></li>
  <li>
    <a href="#">Recipes</a>
    <ul>
      <li><a href="#">Snacks</a></li>
      <li><a href="#">Meals</a></li>
      <li><a href="#">Desserts</a></li>
    </ul>
  </li>
  <li><a href="#">Suggestions</a></li>
</ul>
</body>
</html>

```



در این حالت موقعیت عنصر ul دوم (sub list) با توجه به موقعیت عنصر پدر خود یعنی li تعیین می شود (یعنی موقعیت ul وابسته به viewport نیست بلکه وابسته به li است).

زمانیکه هیچ position ای برای عنصر والد (ansector) تعریف نمی شود تمامی عناصر به صورت default وابسته به viewport مربوط به مرورگر تعیین موقعیت می شوند.

اگر یک عنصر دارای absolute, relative, یا fixed position باشد و این عنصر پدر عنصر دیگری باشد و عنصر فرزند به صورت absolute باشد، آنگاه عنصر فرزند از موقعیت عنصر پدر خود به عنوان نقطه ارجاع (point of reference) برای تعیین موقعیت مطلق خود استفاده می کند؛ یعنی عنصر فرزند با توجه به موقعیت عنصر پدر خود تعیین موقعیت می شود.

استفاده از relative position به تنهایی (و بدون استفاده از offset property) باعث تعیین موقعیت عنصر نسبت به مکان پیش فرض آن می شود؛ یعنی در این حالت خود عنصر به صورت عناصر static موقعیت دهی می شود اما عناصر فرزند آن می توانند از این عنصر به عنوان نقطه ارجاع برای موقعیت دهی خود استفاده کنند.

عناصر relative می توانند به عنوان نقطه ارجاع برای عناصر absolute استفاده شوند؛ یعنی عنصر پدر به صورت relative تعریف شده و به عنوان مبدایی برای عنصر فرزند که به صورت absolute می باشد استفاده می شود.

محتوای عناصر relative می توانند دارای پشته ولایه (stacked & layered) در راستای محور Z (z-axis) باشند. (با استفاده از خاصیت z-index)

fixed positioning

در عناصر fixed شبیه به عناصر absolute، عناصر جریان نرمال سند را رها می کنند (محل نرمال تعیین شده برای عناصر را تغییر می دهند). اما بر خلاف عناصر absolute محتوای عناصر fixed نمی توانند توسط عنصر پدر خود تغییر کنند. عنصر fixed همواره وابسته به نمای دید مرورگر (browser viewport) موقعیت دهی می شوند و این عناصر همواره در محل خود باقی می مانند حتی اگر صفحه scroll شود. fixed positioning برای نگه داشتن یک عنصر در یک مکان ثابت از صفحه نمایش استفاده می شود به گونه ای که حتی با scroll کردن صفحه نیز مکان عنصر تغییر نکند.

عناصر fixed همواره وابسته به viewport و در یک مکان ثابت موقعیت دهی می شوند صرف نظر از اینکه درون یک عنصر relative یا absolute قرار داشته باشند.

برای استفاده مفید از این تکنیک (fixed) باید یک فضای خالی مخصوص در طول صفحه به محتویات عنصر fixed اختصاص داد تا با scroll شدن صفحه این عنصر با عناصر دیگر تداخل پیدا نکند. استفاده از واحد em برای رزرو کردن این فضای خالی باعث می شود در صورت بزرگ شدن فونت فضای خالی نیز به همان نسبت بزرگ شده و مشکلی در صفحه آرایی بوجود نیاید.

از IE6 fixed positioning پشتیبانی نمی کند.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 8_3 </title>
<style type="text/css">
body {
  width: 600px;
  margin: 1em auto;
}

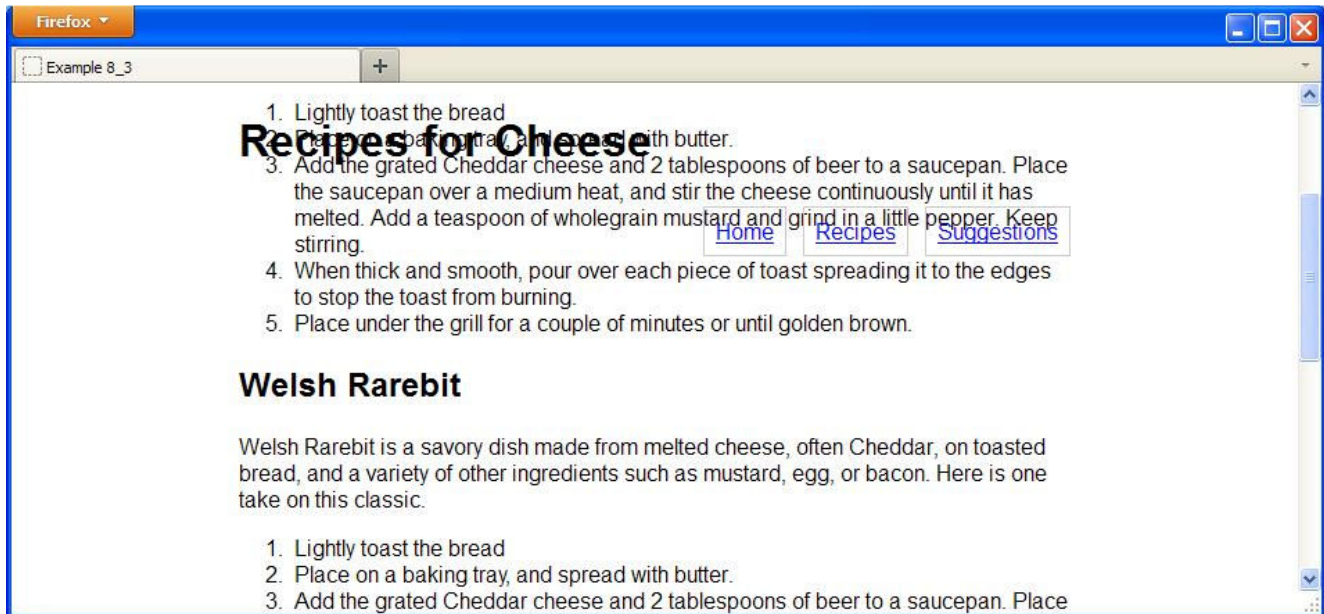
#header {
  position: fixed;
  width: 600px;
  margin: 0 auto;
}
h1 {
  float: left;
  margin-top: 0.2em;
}
.navigation {
  float: right;
}
.navigation li {
  display: inline;
}
.navigation a {
  margin-left: 0.5em;
  padding: 0.5em;
  border: 1px solid #CCC;
}
</style>
</head>
<body>
<div id="header">
  <h1>Recipes for Cheese</h1>
  <ul class="navigation">
    <li><a href="#">Home</a></li>
    <li><a href="#">Recipes</a></li>
    <li><a href="#">Suggestions</a></li>
  </ul>
</div>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<div class="recipe">
  <h2>Welsh Rarebit</h2>
  <p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>

```

```

<ol>
  <li>Lightly toast the bread</li>
  <li>Place on a baking tray, and spread with butter.</li>
  <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
  Place the saucepan over a medium heat, and stir the cheese continuously
  until it has melted. Add a teaspoon of wholegrain mustard and grind in
  a little pepper. Keep stirring.</li>
  <li>When thick and smooth, pour over each piece of toast spreading it to
  the edges to stop the toast from burning.</li>
  <li>Place under the grill for a couple of minutes or until golden
  brown.</li>
</ol>
</div>
<div class="recipe">
  <h2>Welsh Rarebit</h2>
  <p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
  Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
  egg, or bacon. Here is one take on this classic.</p>
  <ol>
    <li>Lightly toast the bread</li>
    <li>Place on a baking tray, and spread with butter.</li>
    <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
    Place the saucepan over a medium heat, and stir the cheese continuously
    until it has melted. Add a teaspoon of wholegrain mustard and grind in
    a little pepper. Keep stirring.</li>
    <li>When thick and smooth, pour over each piece of toast spreading it to
    the edges to stop the toast from burning.</li>
    <li>Place under the grill for a couple of minutes or until golden
    brown.</li>
  </ol>
</div>
<div class="recipe">
  <h2>Welsh Rarebit</h2>
  <p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
  Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
  egg, or bacon. Here is one take on this classic.</p>
  <ol>
    <li>Lightly toast the bread</li>
    <li>Place on a baking tray, and spread with butter.</li>
    <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
    Place the saucepan over a medium heat, and stir the cheese continuously
    until it has melted. Add a teaspoon of wholegrain mustard and grind in
    a little pepper. Keep stirring.</li>
    <li>When thick and smooth, pour over each piece of toast spreading it to
    the edges to stop the toast from burning.</li>
    <li>Place under the grill for a couple of minutes or until golden
    brown.</li>
  </ol>
</div>
</body>
</html>

```



مثال : استفاده از fixed positioning

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 8_4 </title>
<style type="text/css">
  body {
    width: 600px;
    margin: 1em auto;
    padding-top: 3em;
  }

  #header-wrapper {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    padding-bottom: 0.5em;
    background: #FFF;
  }

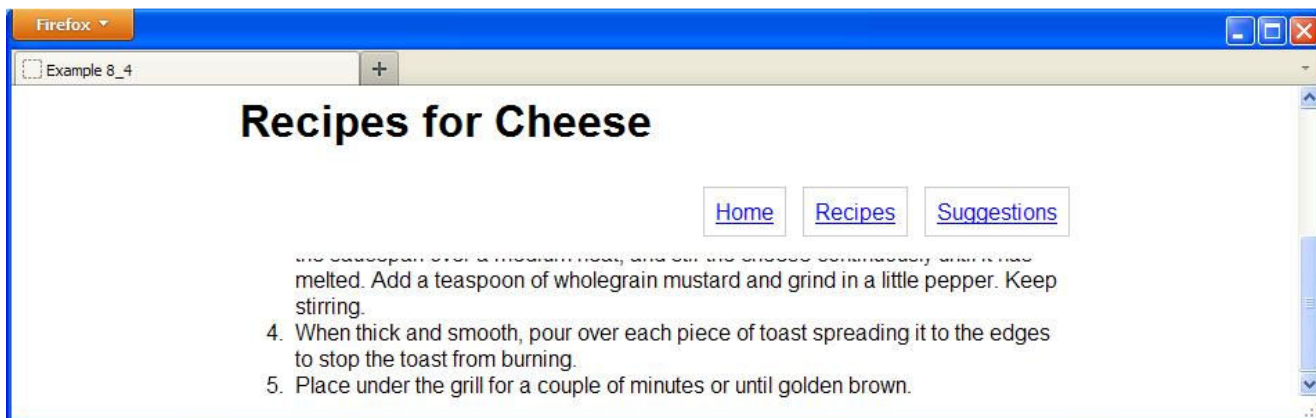
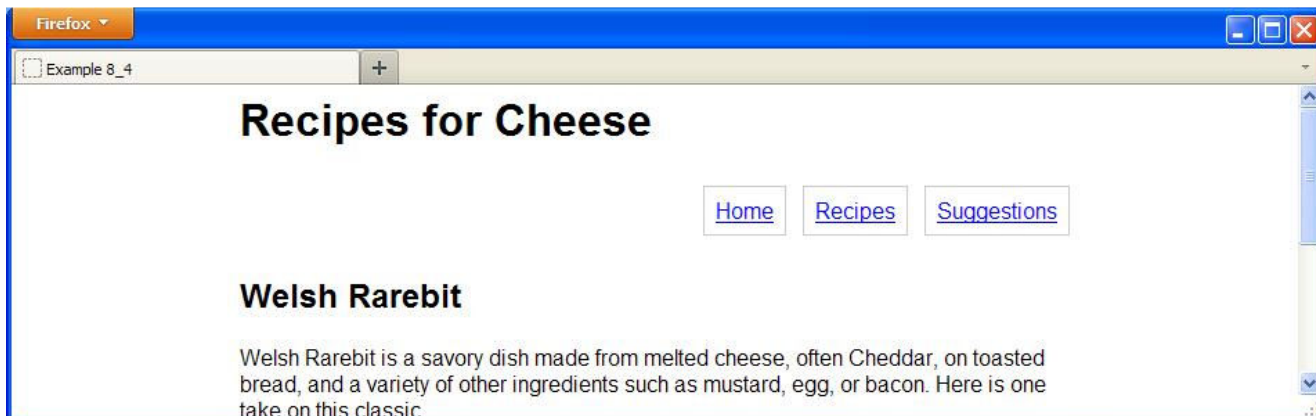
  #header {
    width: 600px;
    margin: 0 auto;
  }
  h1 {
    float: left;
    margin-top: 0.2em;
  }
  .navigation {
    float: right;
  }
</style>
</head>
<body>
  <h1>Recipes for Cheese</h1>
  <ol>
    <li>1. Lightly toast the bread
      <small>Place on a baking tray, and spread with butter.</small>
    </li>
    <li>3. Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan. Place
      the saucepan over a medium heat, and stir the cheese continuously until it has
      melted. Add a teaspoon of wholegrain mustard and grind in a little pepper. Keep
      stirring.
    </li>
    <li>4. When thick and smooth, pour over each piece of toast spreading it to the edges
      to stop the toast from burning.
    </li>
    <li>5. Place under the grill for a couple of minutes or until golden brown.
    </li>
  </ol>
  <div class="navigation">
    <a href="#">Home
    <a href="#">Recipes
    <a href="#">Suggestions
  </div>
  <h2>Welsh Rarebit</h2>
  <p>Welsh Rarebit is a savory dish made from melted cheese, often Cheddar, on toasted
  bread, and a variety of other ingredients such as mustard, egg, or bacon. Here is one
  take on this classic.</p>
  <ol>
    <li>1. Lightly toast the bread
    </li>
    <li>2. Place on a baking tray, and spread with butter.
    </li>
    <li>3. Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan. Place
    </li>
  </ol>
</body>
</html>
```

```

.navigation li {
  display: inline;
}
        .navigation a {
  margin-left: 0.5em;
  padding: 0.5em;
  border: 1px solid #CCC;
}
</style>
</head>
<body>
<div id="header-wrapper">
  <div id="header">
    <h1>Recipes for Cheese</h1>
    <ul class="navigation">
      <li><a href="#">Home</a></li>
      <li><a href="#">Recipes</a></li>
      <li><a href="#">Suggestions</a></li>
    </ul>
  </div>
</div>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<div class="recipe">
  <h2>Welsh Rarebit</h2>
  <p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
  <ol>
    <li>Lightly toast the bread</li>
    <li>Place on a baking tray, and spread with butter.</li>
    <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
Place the saucepan over a medium heat, and stir the cheese continuously
until it has melted. Add a teaspoon of wholegrain mustard and grind in
a little pepper. Keep stirring.</li>
    <li>When thick and smooth, pour over each piece of toast spreading it to
the edges to stop the toast from burning.</li>
    <li>Place under the grill for a couple of minutes or until golden
brown.</li>
  </ol>
</div>

</body>
</html>

```



z index

خاصیت z-index موقعیت عناصر را در طول محور نامرئی Z (invisible z-axis) کنترل می کند مشروط به اینکه عناصر مذکور به صورت relative , absolute و یا fixed موقعیت دهی شده باشند. (ایجاد پشته و لایه در راستای محور Z)

قاعده ی کلی :

z-index: auto | <integer>

مقدار پیش فرض : auto

integer : اعداد | 0,1,2,3,4,...,n ✓

در حالت auto اگر چند عنصر با یکدیگر تداخل داشته باشند ، عنصری آخر از همه در ترتیب عناصر فایل HTML وارد شده در صفحه نمایش بالاتر قرار می گیرد (نسبت به محور Z که بر صفحه نمایش عمود می باشد) و باقی عناصر نیز به همین ترتیب گفته شده قرار خواهند گرفت (در حالت auto عنصری که اول از همه وارد شود در زیر سایر عناصر قرار خواهد گرفت) .

می توان با تخصیص مقادیر integer به خاصیت z-index ترتیب قرارگیری عناصر در راستای محور Z را تغییر داد . عنصری که مقدار عددی integer بالاتری داشته باشد بالا (مافوق) سایر عناصر در صفحه نمایش داده خواهد شد و سایر عناصر نیز به ترتیب بزرگی مقدار خود در صفحه ظاهر خواهند شد .

به این حالت که چند عنصر روی هم قرار می گیرند پشته سازی محتوا (atacking context) یا لایه بندی محتوا (layering context) می گویند .

اگر یک عنصر relative پدر عناصر دیگری از نوع absolute باشد آنگاه این عنصر پدر خود یک stacking context ایجاد می کند و فرزندان این عنصر relative خاصیت z-index خود را نسبت به عنصر پدر می سنجند .

از آنجاییکه در حالت static عناصر به ترتیبی که وارد می شوند در صفحه قرار می گیرند و قابل موقعیت دهی نیستند لذا خاصیت z-index در این حالت کاربردی ندارد و تاثیری روی عناصر نخواهد داشت .

مقادیر رایج برای خاصیت z-index مقادیر ۱۰ و کمتر از آن می باشند و مقادیر در محدوده ی ۱۰۰ و ۱۰۰۰ و ... رایج نیستند . استفاده از مقادیر بزرگ برای عنصر اول باعث می شود هر عنصر جدیدی که وارد می شود و دارای مقدار کمتر می باشد در زیر قرار گیرد . توصیه می شود که مقادیر استفاده شده برای خاصیت z-index همیشه کمتر از ۱۰۰ باشند .

عناصر داخل یک stacking context پدر نسبت به عناصر خارج از آن ایزوله می باشند و بنابراین درون یک عنصر پدر می توان مقادیر z-index را مجددا از یک شروع کرد ؛ یعنی مقادیر z-index داخل یک عنصر پدر می توانند مقادیری برابر با z-index خارج از آن داشته باشند بدون اینکه مشکلی پیش بیاید .

IE6 و IE7 از z-index پشتیبانی می کنند اما با کمی اشکال که در IE8 رفع شده است .

عناصر موقعیت دهی شده (absolute , relative , fixed) که مقدار خاصیت z-index برای آنها به صورت auto تعریف شده است با توجه به ترتیب قرارگیری در فایل HTML به خاصیت z-index آنها با شروع از صفر به صورت افزایشی مقداردهی می شود ؛ این مطلب به این معنی است که تمامی عناصر موقعیت دهی شده یک stacking context ایجاد می کنند .

به عنوان مثال اگر به دومین عنصر z-index:1 داده شود این عنصر در تمامی مرورگرها بالاتر از باقی عناصر که مقدار auto دارند قرار خواهد گرفت . اما در IE این عنصر به گونه ای دیگر رفتار کرده و در موقعیت خود قرار می گیرد ؛ z-index بزرگتر از auto می باشد و بالاتر قرار می گیرد .

مثال : استفاده از خاصیت z-index

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 8_5</title>
<style type="text/css">
div {
width: 100px;
height: 100px;
border: 1px solid #000;
font-size: 25px;
text-align: center;
}

.z1 {
background: #CC5;
}

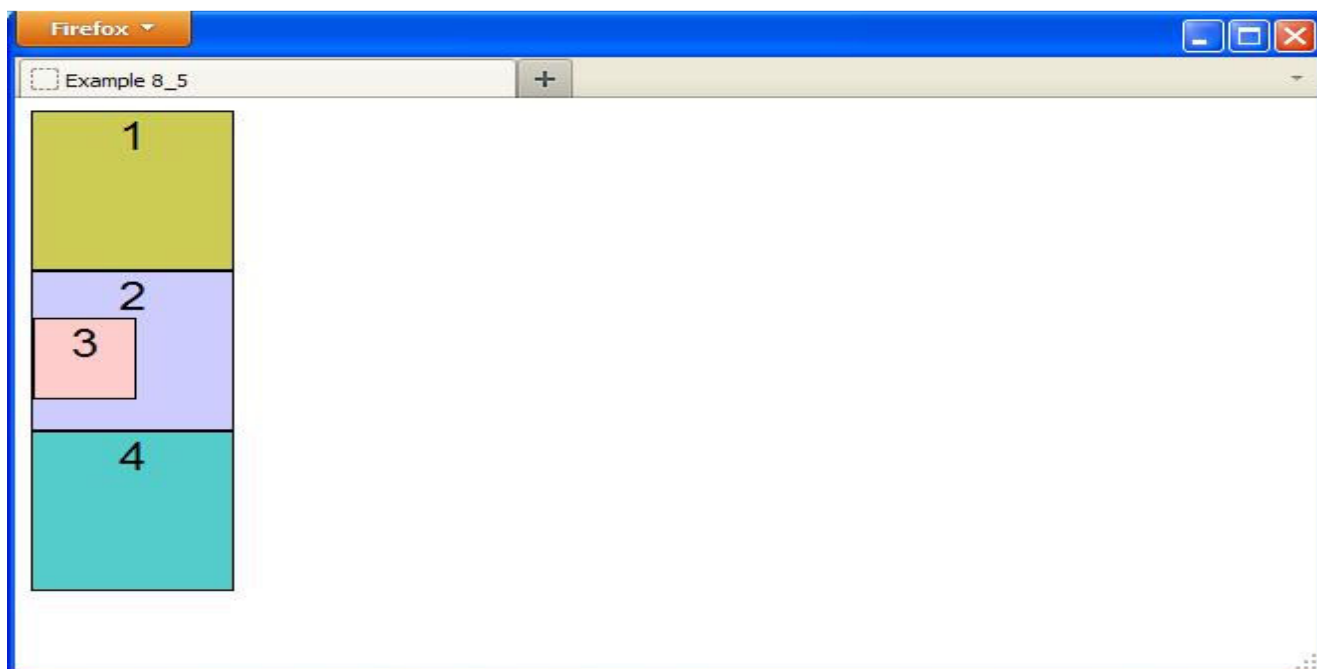
.z2 {
background: #CCF;
}

.z3 {
width: 50px;
height: 50px;
background: #FCC;
}
```

```

.z4 {
  background: #5CC;
}
</style>
</head>
<body>
<div class="z1">1</div>
<div class="z2">
  2
  <div class="z3">
    3
  </div>
</div>
<div class="z4">4</div>
</body>
</html>

```



خلاصه

- ◀ عنصر **fixed position** بدون توجه به موقعیت عنصر پدر خود و با توجه به **viewport** مرورگر موقعیت دهی می شود.
- ◀ استفاده از **absolute** به تنهایی وابسته به **viewport** می باشد.
- ◀ استفاده از **relative positioning** به همراه **offset property** به تنهایی باعث تغییر موقعیت عنصر نسبت به محل تعیین شده ی آن توسط مرورگر (**static positioning**) می شود.
- ◀ عنصر با موقعیت **relative** معمولاً به عنوان پدر یک عنصر با موقعیت **absolute** استفاده می شود و این عنصر **absolute** نسبت به مبدا عنصر **relative** موقعیت دهی می شود.
- ◀ عناصر **fixed position** همواره در موقعیت خود باقی می مانند و با **scroll** شدن صفحه نیز جابجا نمی شوند.
- ◀ به صورت پیش فرض عناصر به ترتیب صعودی روی هم لایه بندی می شوند اما این لایه بندی می تواند توسط خاصیت **z-index** کنترل شود.

فصل نهم

Styling Tables

عنصر `<table>` اطلاعات را به صورت مرتب و سازماندهی شده و به صورت پایدار نمایش می دهد .

caption side

خاصیت `caption-side` محل قرارگیری عنوان جدول را کنترل می کند .
قاعده ی کلی :

`caption-side: top | bottom`

مقدار پیش فرض : `top`

IE6 و IE7 از عنصر `caption` پشتیبانی می کنند اما از خصوصیت `caption-side` پشتیبانی نمی کنند و عنوان جدول را به صورت پیش فرض در بالای آن نمایش می دهند .

عنصر `table` دارای تعدادی از عناصر اختیاری می باشد که عبارتند از :

- `<colgroup><col /></colgroup>`
- `<caption />`
- `<tbody/>`
- `<tfoot/>`
- `<thead/>`

عناصر `<colgroup/>` و `<col/>` اجازه ی کنترل ستون های عمودی جدول را می دهند . این ویژگی برای کنترل پهنای ستون ها ، رنگ پس زمینه ی ستون ها و رنگ متن آنها مناسب می باشد . اگر چه عنصر `<col/>` خود در صفحه قابل مشاهده نیست اما خصوصیات تعریف شده برای آن بر روی ستون های متناظر آن قابل اعمال است .

IE تنها مرورگری است که از تنظیمات `text color` بر روی عنصر `<col/>` پشتیبانی می کند . این یک رفتار غیر استاندارد است بنابراین انتظار مشاهده ی این ویژگی را در سایر مرورگرها در آینده نداشته باشید .

مثال :

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

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 9-1</title>
<style type="text/css">
table{
    caption-side: bottom;
}
.ingredient {
    width: 200px;
    background-color: #CC5;
    color: red;
}
```

```

.quantity {
  width: 400px;
  background-color: #CCF;
}
</style>
</head>
<body>
  <table>
    <caption>Ingredients</caption>
    <colgroup>
      <col class="ingredient">
      <col class="quantity">
    </colgroup>
    <thead>
      <tr>
        <th>Ingredient</th>
        <th>Quantity of Ingredient</th>
      </tr>
    </thead>
    <tfoot>
      <tr>
        <th>Ingredient</th>
        <th>Quantity of Ingredient</th>
      </tr>
    </tfoot>
    <tbody>
      <tr>
        <td>Bread</td>
        <td>2 medium thickness slices</td>
      </tr>
      <tr>
        <td>Butter</td>
        <td>Enough for 2 slices of bread</td>
      </tr>
      <tr>
        <td>Grated Cheddar</td>
        <td>1.5 handfuls</td>
      </tr>
      <tr>
        <td>Beer</td>
        <td>One splash</td>
      </tr>
      <tr>
        <td>Wholegrain mustard</td>
        <td>One dollop</td>
      </tr>
      <tr>
        <td>Pepper</td>
        <td>To taste</td>

```

```

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

```

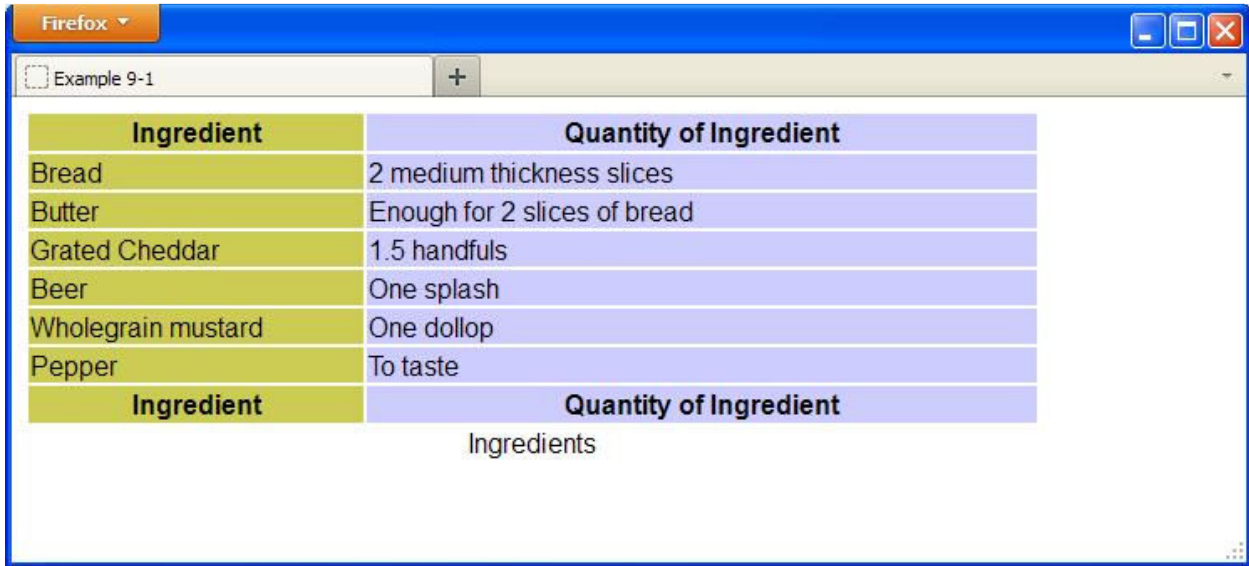


table layout

جداول به صورت پیش فرض برای تطبیق دادن محتوا با اندازه ی جدول گسترش می یابند ، از این نظر جداول به طور ذاتی سیال هستند و به طور پیش فرض به اندازه ی عرض محتوای خود بسط پیدا می کنند . با این وجود گاهی اوقات ضروری است که جدول مجبور به استفاده عرض ثابت برای هر دوی table و cells باشد .

قاعده ی کلی :

table-layout: auto | fixed

مقدار پیش فرض : auto

گاهی اوقات عرض تعریف شده برای جدول کمتر از آن است که محتویات جدول را نمایش دهد و جدول گسترش پیدا می کند و عرضی بیشتر از مقدار تعریف شده برای جدول پیدا می کند . برای جلوگیری از گسترش عرض جدول باید از مقدار fixed برای خاصیت table-layout در جدول مورد نظر استفاده کنیم :

table-layout: fixed;

در این حالت جدول مجبور به پیروی از width تعریف شده می باشد صرف نظر از اینکه چه مقدار داده درون سلول های آن وجود دارد و اگر محتویات جدول بیش از عرض (width) تعریف شده باشد محتویت سرریز خواهند شد . در این حالت با تعریف خاصیت overflow برای عناصر <td/> و <th/> می توان سرریز را کنترل کرد :

overflow: hidden | scroll | auto | visible

مثال :

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 9_2</title>
<style type="text/css">
table {

```

```

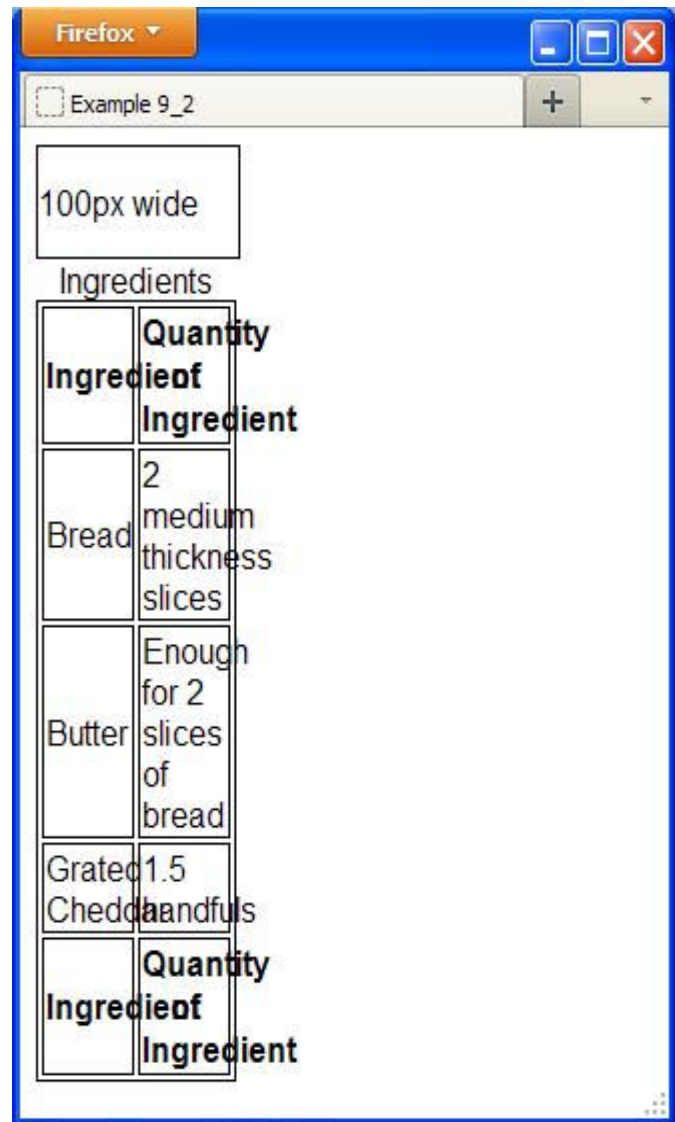
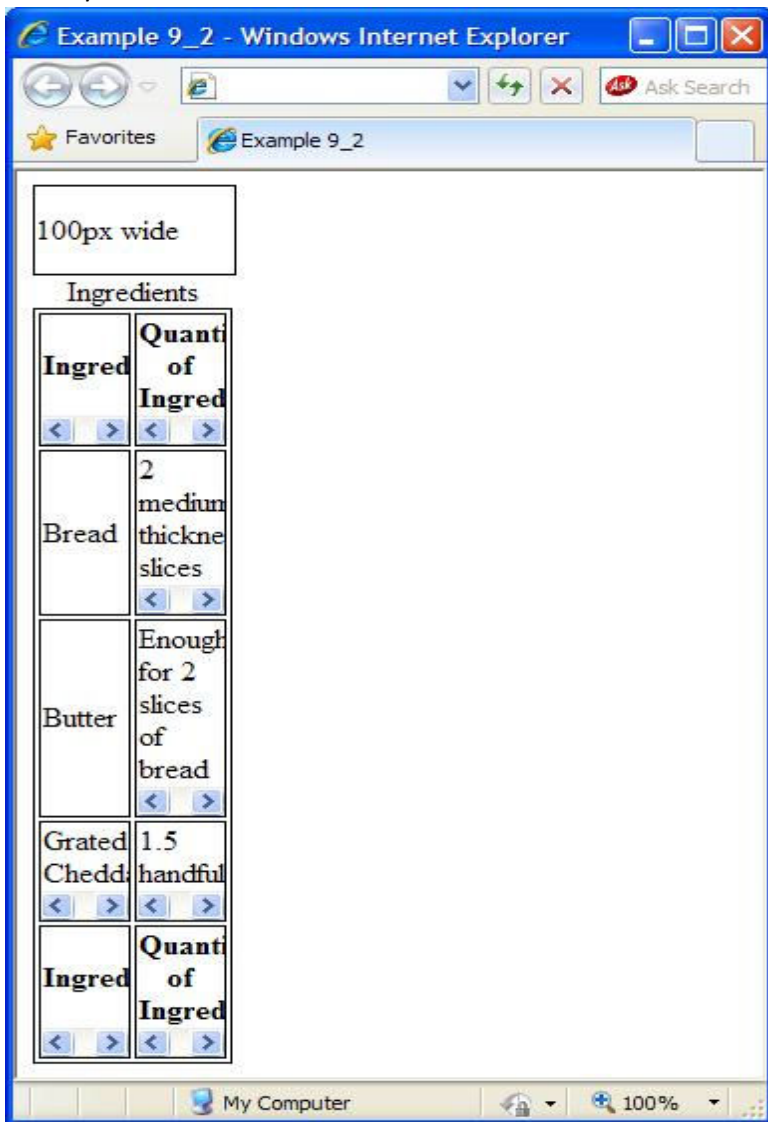
    table-layout: fixed;
}
th,td{
    overflow:auto;
}
table,
.control {
    width: 100px;
}
table,
th,
td,
.control {
    border: 1px solid #000;
}
</style>
</head>
<body>
    <div class="control">
        <p>100px wide</p>
    </div>

<table>
<caption>Ingredients</caption>
<colgroup>
    <col class="ingredient">
    <col class="quantity">
</colgroup>
<thead>
    <tr>
        <th>Ingredient</th>
        <th>Quantity of Ingredient</th>
    </tr>
</thead>
<tfoot>
    <tr>
        <th>Ingredient</th>
        <th>Quantity of Ingredient</th>
    </tr>
</tfoot>
<tbody>
    <tr>
        <td>Bread</td>
        <td>2 medium thickness slices</td>
    </tr>
    <tr>
        <td>Butter</td>
        <td>Enough for 2 slices of bread</td>
    </tr>
    <tr>
        <td>Grated Cheddar</td>

```

```

        <td>1.5 handfuls</td>
    </tr>
    <tr>
        <td>Beer</td>
        <td>One splash</td>
    </tr>
    <tr>
        <td>Wholegrain mustard</td>
        <td>One dollop</td>
    </tr>
    <tr>
        <td>Pepper</td>
        <td>To taste</td>
    </tr>
</tbody>
</table>
</body>
</html>
    
```



border collapse

به طور پیش فرض بین سلول های جدول مقداری فضای خالی وجود دارد که خاصیت `border-collapse` اجازه یحذف کامل این فضای خالی را می دهد و برای کنترل بهتر خاصیت `border-spacing` اجازه ی تعریف این فضای خالی را به میزان دلخواه می دهد .
قاعده ی کلی :

`border-collapse: collapse | separate`

`separate` : مقدار پیش فرض

اگر مقدار این خاصیت برابر `collapse` تعریف شود فضای خالی بین سلول های جدول حذف می شود و تمامی سلول ها به هم فشرده شده و `border` بین سلول هل باهم ترکیب خواهد شد و `border` دور کل جدول نیز از بین خواهد رفت .
مثال :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title> Example 9_3 </title>
<style type="text/css">
table {
border-collapse: collapse;
}
table,
th,
td {
border: 1px solid #000;
}
</style>
</head>
<body>
<table>
<caption>Ingredients</caption>
<colgroup>
<col class="ingredient">
<col class="quantity">
</colgroup>
<thead>
<tr>
<th>Ingredient</th>
<th>Quantity of Ingredient</th>
</tr>
</thead>
<tfoot>
<tr>
<th>Ingredient</th>
<th>Quantity of Ingredient</th>
</tr>
</tfoot>
<tbody>
<tr>
<td>Bread</td>
<td>2 medium thickness slices</td>
</tr>
```



```

<tr>
  <td>Butter</td>
  <td>Enough for 2 slices of bread</td>
</tr>
<tr>
  <td>Grated Cheddar</td>
  <td>1.5 handfuls</td>
</tr>
<tr>
  <td>Beer</td>
  <td>One splash</td>
</tr>
<tr>
  <td>Wholegrain mustard</td>
  <td>One dollop</td>
</tr>
<tr>
  <td>Pepper</td>
  <td>To taste</td>
</tr>
</tbody>
</table>
</body>
</html>

```

The screenshot shows a Firefox browser window with a single tab titled 'example'. The page content is a table with the following structure:

Ingredients	
Ingredient	Quantity of Ingredient
Bread	2 medium thickness slices
Butter	Enough for 2 slices of bread
Grated Cheddar	1.5 handfuls
Beer	One splash
Wholegrain mustard	One dollop
Pepper	To taste
Ingredient	Quantity of Ingredient

border spacing

با استفاده از خاصیت `border-spacing` می توان مقدار فضای خالی افقی و عمودی بین سلول های جدول را مشخص کرد .
قاعده ی کلی :

`border-spacing: <length> <length>?`

مقدار پیش فرض : 0

مقدار `length` اول فضای خالی افقی بین سلول های جدول را مشخص می کند و مقدار `length` دوم فضای خالی عمودی بین سلول های جدول را مشخص می کند . مقدار دوم اختیاری است و اگر ذکر نشود هر دو مقدار افقی و عمودی برابر خواهند بود .

مثال :

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">

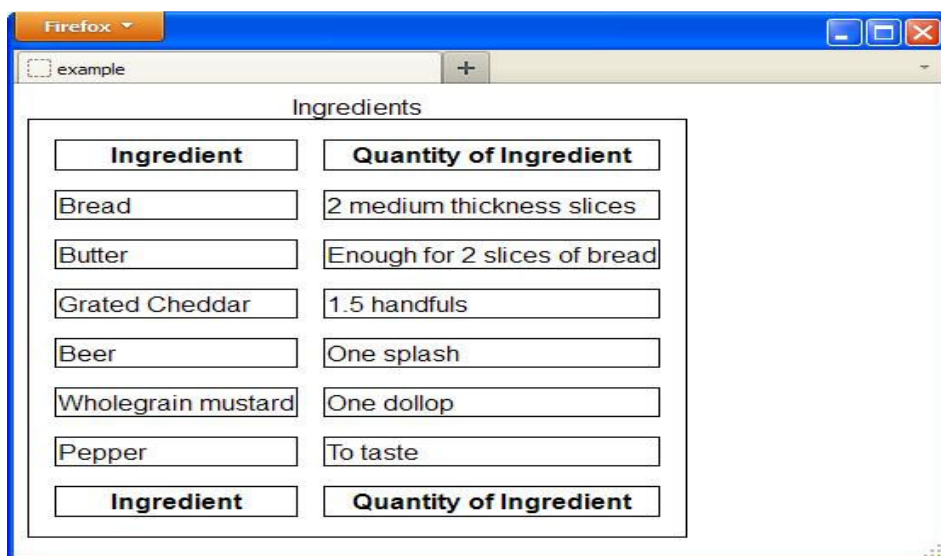
```

```

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Example 9_4</title>
  <style type="text/css">
    table {
      border-spacing: 15px;
    }
    table,
    th,
    td {
      border: 1px solid #000;
    }
  </style>
</head>
<body>
<table>
  <caption>Ingredients</caption>
  <colgroup>
    <col class="ingredient">
    <col class="quantity">
  </colgroup>
  <thead>
    <tr>
      <th>Ingredient</th>
      <th>Quantity of Ingredient</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th>Ingredient</th>
      <th>Quantity of Ingredient</th>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Bread</td>
      <td>2 medium thickness slices</td>
    </tr>
    <tr>
      <td>Butter</td>
      <td>Enough for 2 slices of bread</td>
    </tr>
    <tr>
      <td>Grated Cheddar</td>
      <td>1.5 handfuls</td>
    </tr>
    <tr>
      <td>Beer</td>
      <td>One splash</td>
    </tr>
  </tbody>
</table>

```

```
<td>Wholegrain mustard</td>
<td>One dollop</td>
</tr>
<tr>
<td>Pepper</td>
<td>To taste</td>
</tr>
</tbody>
</table>
</body>
</html>
```



فصل دهم

Customizing The Mouse Cursor

cursor

با استفاده از خاصیت **CURSOR** می توان نوع اشاره گر موس مورد استفاده برای عناصر را تغییر داد .
قاعده ی کلی :

`cursor: [<url>,]* | auto | crosshair | default | pointer | move | e-resize | ne-resize | nw-resize
| n-resize | se-resize | sw-resize | s-resize | w-resize | text | wait | help | progress`

مقدار پیش فرض : `auto`

تعدادی **CURSOR** غیر استاندارد نیز وجود دارند که در برخی از مرورگرها پشتیبانی نمی شوند :

`cursor: hand | all-scroll | col-resize | row-resize | no-drop | not-allowed | vertical-text`

نکته :

مرورگرهای `safari` و `chrom` کلمات کلیدی غیر استاندارد را پشتیبانی نمی کنند . مرورگر `opera` در سیستم عامل `mac` کلمات کلیدی `*-resize` و هم چنین کلمات کلیدی غیر استاندارد را پشتیبانی نمی کند .

مرورگر `opera` در سیستم عامل `windows` کلمات کلیدی `*-resize` را پشتیبانی می کند اما کلمات کلیدی غیر استاندارد را پشتیبانی نمی کند .
مرورگر `firefox` در `mac` کلمه یکلیدی `all-scroll` را پشتیبانی نمی کند اما در `windows` پشتیبانی می کند .
مرورگر `IE` تمامی گزینه های ممکن را پشتیبانی می کند .

با استفاده از مقادیر `url` می توان یک `custom cursor` را بر اساس یک `img` تعریف کرد . در عمل این روش تقریباً هرگز استفاده نمی شود و برای استفاده از آن نیز توصیه نمی شود و استفاده از `CURSOR` های متداول بهتر می باشد .

در بین `cursor` ها استفاده از مقدار `pointer` متداول می لاشد از جمله برای `input submit` , `button` و اعمال اضافه شده توسط کدهای `javascript` .
`pointer` برای لینک ها (عناصر `<a/>`) مقدار پیش فرض می باشد .

مثال :

```
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>example</title>
  <style type="text/css">
    .crosshair {
      cursor: crosshair;
    }
    .pointer {
      cursor: pointer;
    }
    .move {
      cursor: move;
    }
    .e-resize {
```

```
    cursor: e-resize;
}
.w-resize {
    cursor: w-resize;
}
.ne-resize {
    cursor: ne-resize;
}
.sw-resize {
    cursor: sw-resize;
}
.n-resize {
    cursor: n-resize;
}
.s-resize {
    cursor: s-resize;
}
.nw-resize {
    cursor: nw-resize;
}
.se-resize {
    cursor: se-resize;
}
.text {
    cursor: text;
}
.wait {
    cursor: wait;
}
.help {
    cursor: help;
}
.progress {
    cursor: progress;
}
.hand {
    cursor: hand;
}
.all-scroll {
    cursor: all-scroll;
}
.col-resize {
    cursor: col-resize;
}
.row-resize {
    cursor: row-resize;
}
.no-drop {
    cursor: no-drop;
}
.not-allowed {
```

```

    cursor: not-allowed;
  }
  .vertical-text {
    cursor: vertical-text
  }
</style>
</head>
<body>

```

<h1>Cursor Types</h1>

```

<ul>
  <li class="crosshair">Crosshair</li>
  <li class="pointer">Pointer</li>
  <li class="move">Move</li>
  <li class="e-resize">E-Resize</li>
  <li class="w-resize">W-Resize</li>
  <li class="ne-resize">NE-Resize</li>
  <li class="sw-resize">SW-Resize</li>
  <li class="n-resize">N-Resize</li>
  <li class="s-resize">S-Resize</li>
  <li class="nw-resize">NW-Resize</li>
  <li class="se-resize">SE-Resize</li>
  <li class="text">Text</li>
  <li class="wait">Wait</li>
  <li class="help">Help</li>
  <li class="progress">Progress</li>
  <li class="hand">Hand</li>
  <li class="all-scroll">All-Scroll</li>
  <li class="col-resize">Col-Resize</li>
  <li class="row-resize">Row-Resize</li>
  <li class="no-drop">No-Drop</li>
  <li class="not-allowed">Not-Allowed</li>
  <li class="vertical-text">Vertical-Text</li>
</ul>
</body>
</html>

```

CURSOR	IE 6 WIN XP	IE 7 VISTA	FIREFOX MAC	FIREFOX WIN	SAFARI MAC	OPERA MAC	OPERA WIN
default							
crosshair							
pointer							
move							
e-resize							
w-resize							
ne-resize							
sw-resize							
n-resize							
s-resize							
nw-resize							
se-resize							
text							
wait							
help							
progress							
hand							
all-scroll							
col-resize							
row-resize							
no-drop							
not-allowed							
vertical-text							

فصل یازدهم

Controlling Opacity and Visibility

opacity

با استفاده از خاصیت opacity می توان میزان شفافیت یک عنصر را کنترل کرد .
قاعده ی کلی :

opacity: <number>

مقدار پیش فرض : 1

مقدار یک برای خاصیت opacity مشخص می کند که یک عنصر با کاملا ناشفاف یا تیره (مات ، کدر ، مبهم ، opaqua) باشد و مقدار صفر برای ویژگی opacity تعیین می کند که عنصر باید کاملا نامرئی (invisible) باشد .

مقدار opacity را باید به صورت یک عدد اعشاری و حداکثر تا دو رقم اعشار تعریف کرد . مقادیر بین صفر تا یک و تا دو رقم اعشار برای opacity مجاز می باشند .

تمامی مرورگرهای اصلی به غیر از IE6 , IE7 , IE8 از خاصیت opacity پشتیبانی می کنند . یک روش اختصاصی برای تعیین opacity در IE8 و قبل از آن وجود دارد که عبارت است از :

filter:alpha(opacity=<number>);

مقدار پیش فرض : 100

در این روش برای مقادیر opacity اعداد صحیح بین ۰ تا ۱۰۰ مجاز می باشد .
روش فوق صرفا در مرورگر IE8 - IE8 (و قبل از آن) کار می کند .
IE9 از خاصیت opacity پشتیبانی می کند .

مثال :

```
img{
  opacity:0.5;
  /* opacity for IE8 - */
  filter: alpha(opacity=50);
}
```

در مثال فوق opacity تعریف شده برای img در تمامی مرورگرها به درستی کار می کند .

مثال :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 10_1</title>
<style type="text/css">
body {
width: 600px;
margin: 1em auto;
}

h1 {
float: left;
margin-top: 0;
```



```

}

.header {
  background: #000;
  color: #FFF;
  padding: 1em;
}

.header .intro {
  clear: both;
}
.navigation {
  float: right;
  margin: 0;
  list-style: none;
}
.navigation li {
  float: left;
}
.navigation a {
  display: block;
  margin-left: 0.5em;
  padding: 0.5em;
  border: 1px solid #CCC;
  background: #FFF;
  color: #233;
  text-decoration: none;
}

li {
  opacity: 1;
}
li.current {
  opacity: 0.5;
}

li.last {
  opacity: 0.25;
}
</style>
</head>
<body>
  <div class="header">
    <h1>Recipes for Cheese</h1>

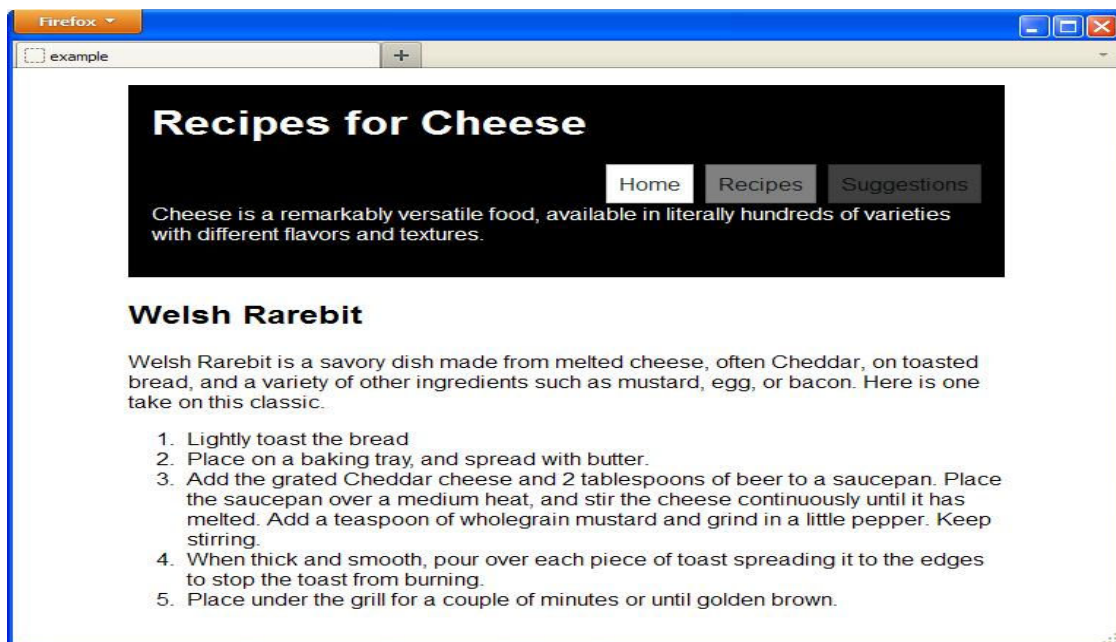
    <ul class="navigation">
      <li><a href="#">Home</a></li>
      <li class="current"><a href="#">Recipes</a></li>
      <li class="last"><a href="#">Suggestions</a></li>
    </ul>

```

```

<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
</div>
<div class="recipe">
  <h2>Welsh Rarebit</h2>
  <p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
  Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
  egg, or bacon. Here is one take on this classic.</p>
  <ol>
    <li>Lightly toast the bread</li>
    <li>Place on a baking tray, and spread with butter.</li>
    <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
    Place the saucepan over a medium heat, and stir the cheese continuously
    until it has melted. Add a teaspoon of wholegrain mustard and grind in a
    little pepper. Keep stirring.</li>
    <li>When thick and smooth, pour over each piece of toast spreading it to
    the edges to stop the toast from burning.</li>
    <li>Place under the grill for a couple of minutes or until golden
    brown.</li>
  </ol>
</div>
</body>
</html>

```



visibility

با استفاده از این خاصیت می توان یک عنصر را در صفحه نمایش داد و یا مخفی کرد .
قاعده ی کلی :

visibility: visible | hidden

مقدار پیش فرض : visible

✓ visible : با اختصاص این مقدار عنصر در صفحه نمایش داده خواهد شد .

✓ hidden : تخصیص مقدار hidden به خاصیت visibility باعث می شود که یک عنصر در صفحه مخفی شده و به جای آن یک فضای خالی نمایش داده شود .

مثال :

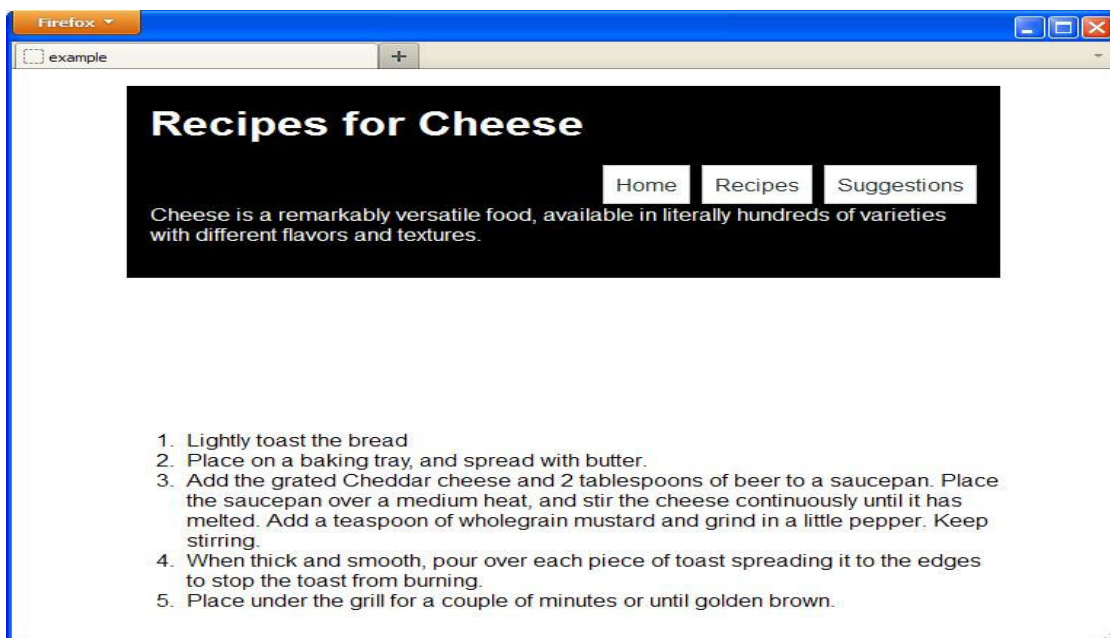
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 10_2</title>
<style type="text/css">
body {
width: 600px;
margin: 1em auto;
}
h1 {
float: left;
margin-top: 0;
}
.header {
background: #000;
color: #FFF;
padding: 1em;
}
.header .intro {
clear: both;
}
.navigation {
float: right;
margin: 0;
list-style: none;
}
.navigation li {
float: left;
}
.navigation a {
display: block;
margin-left: 0.5em;
padding: 0.5em;
border: 1px solid #CCC;
background: #FFF;
color: #233;
text-decoration: none;
}
.recipe {
visibility: hidden;
border: 1px solid #000;
}
.recipe ol {
```

```

    visibility: visible;
}
</style>
</head>
<body>
  <div class="header">
    <h1>Recipes for Cheese</h1>
    <ul class="navigation">
      <li><a href="#">Home</a></li>
      <li class="current"><a href="#">Recipes</a></li>
      <li class="last"><a href="#">Suggestions</a></li>
    </ul>
    <p class="intro">Cheese is a remarkably versatile food, available in literally
    hundreds of varieties with different flavors and textures.</p>
  </div>
  <div class="recipe">
    <h2>Welsh Rarebit</h2>
    <p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
    Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
    egg, or bacon. Here is one take on this classic.</p>
    <ol>
      <li>Lightly toast the bread</li>
      <li>Place on a baking tray, and spread with butter.</li>
      <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan.
      Place the saucepan over a medium heat, and stir the cheese continuously
      until it has melted. Add a teaspoon of wholegrain mustard and grind in a
      little pepper. Keep stirring.</li>
      <li>When thick and smooth, pour over each piece of toast spreading it to
      the edges to stop the toast from burning.</li>
      <li>Place under the grill for a couple of minutes or until golden
      brown.</li>
    </ol>
  </div>
</body>
</html>

```



فصل دوازدهم

Advanced Selectors

direct child selector

direct child selector بسیار شبیه به descendant selector عمل می کند و مانند آنها بر یک رابطه ی اجدادی برای تعیین اینکه style برای چه قسمتی پذیرفته شود استفاده می کند . descendant selectors نامفهوم هستند زیرا style تعریف شده را برای تمامی فرزندان یک عنصر پدر اعمال می کنند . direct child selector ، style را بدون واسطه و مستقیماً بر روی فرزند تعیین شده اعمال می کند . اینکار با تعریف یک قاعده ی جدید برای selector انجام می شود :

```
body > .intro{
    font-weight:bold;
}
```

در دستور فوق اگر عنصر body دو فرزند یا بیشتر با کلاس intro داشته باشد این style فقط روی اولین فرزند عنصر body (فرزند مستقیم عنصر body) که دارای کلاس intro می باشد ، اعمال خواهد شد . توجه کنید که در مثال فوق بین body و علامت > و intro . یک space وجود دارد .

این selector ها نیز همانند descendant selector می توانند به صورت زنجیروار به یکدیگر متصل شوند اما علاوه بر space باید از یک علامت (greater than) > نیز استفاده کنند . تمامی مرورگرهای اصلی به جز IE6 از این نوع selector پشتیبانی می کنند .

next sibling selector

در استاندارد W3C ، selector ، adjacent sibling combinator وجود دارد که next sibling selector شکل ساده شده ی آن می باشد . این selector عنصر هم جوار بعدی را برای اعمال قوانین تعریف شده انتخاب می کند . مثال :

```
h1 + .intro{
    font-weight:bold;
}
```

در مثال فوق اگر عنصری بلافاصله پس از عنصر h1 بیاید و این عنصر از کلاس intro استفاده کرده باشد متن آن به صورت bold نمایش داده خواهد شد . توجه کنید که در مثال فوق بین h1 و علامت > و intro . یک space وجود دارد .

این نوع selector ها با استفاده از علامت + (plus sign) به هم زنجیر می شوند . تمامی مرورگرهای اصلی به جز IE6 از این نوع selector پشتیبانی می کنند .

selector های next sibling و direct child را می توان به گونه ای استفاده کرد که نتیجه ی یکسانی داشته باشند .

attribute selectors

چندین نوع مختلف از attribute selector وجود دارد و CSS توانایی آشکار کردن attribute ها را بر اساس ظوابط زیر دارد :

۱. شخصیت یا جایگاه (presence) یک attribute
۲. مقدار یک attribute
۳. شروع مقدار یک صفت با یک رشته ی خاص (specific string)
۴. پایان مقدار یک صفت با یک رشته ی خاص
۵. مقدار یک صفت شامل یک رشته ی خاص در هر کجای رشته ، خواه در ابتدا ، انتها و یا میانه ی آن

تمامی مرورگرهای اصلی به جز IE6 از این نوع attribute selector پشتیبانی می کنند .

۱- select by presence of an attribue

ساده ترین نوع attribute selector یک style را بر اساس نوع یک صفت یک عنصر می پذیرد .
مثال :

```
input[name]{
    border: 1px solid #000;
}
```

style فوق برای تمامی عناصر input که از صفت name استفاده کرده باشند پذیرفته می شود .

۲- select by attribute value

این نوع selector ، style را بر پایه ی مقدار و نوع صفت تعریف می کند .
مثال :

```
input[name="email"]{
    border: 2px dashed #000;
}
```

style فوق بر روی عنصر input که مقدار صفت name ان برابر با email اعمال خواهد شد .

۳- substring attribute selector

می توان یک attribute selector را بر اساس اینکه مقدار آن attribute با چه رشته ای شروع ، به چه رشته ای ختم می شود و یا شامل چه رشته ای می باشد تعریف کرد .

۳-۱ (۳) صفتی که با رشته ای خاص شروع می شود :

مثال :

```
a[href^="mailto:"]{
    padding-left: 23px;
    background: transparent url(mailto.png) no-repeat center left ;
}
```

کاراکتر (^) پس از مقدار href به این معنی است که style باید بر روی عنصر <a> ی اعمال شود که مقدار صفت href آن با زیر رشته ی mailto: شروع می شود .

۳-۲ (۴) صفتی که با رشته ای خاص پایان می یابد :

مثال :

```
input[id$="name"]{
    border: 2px solid #000;
}
```

کاراکتر (\$) پس از مقدار id به این معنی است که style باید بر روی عنصر <input> ی اعمال شود که مقدار صفت id آن با زیر رشته ی name تمام می شود .

۳-۳ (۵) صفتی که شامل رشته ای خاص می لاشد :

مثال :

```
[name*="recipe"]{
    border:2px solid #aaa;
}
```

کاراکتر (*) پس از مقدار name به این معنی است که style باید بر روی تمامی عناصری اعمال شود که مقدار صفت name آن شامل زیر رشته ی recipe می باشد .

عناصر کاذب (pseudo-elements)

عناصر کاذب جنبه‌هایی از document را که به سادگی قابل تعریف نیست به نمایش در می‌آورند . عناصر کاذب `first-letter` و `first-line`: به ترتیب اشاره به اولین حرف و اولین خط از عنصری که شامل text است دارند . با استفاده از این عناصر کاذب می‌توان خصوصیات اولین حرف و اولین خط از متن را تغییر داد . از جمله این خصوصیات عبارتند از : `background-color` , `background-image` , `color` , `font` , ...

در CSS3 قاعده (syntax) مربوط به pseudo-elements از `single colon (:)` به `double colon (::)` تغییر کرده و این تغییر برای ایجاد تمایز بین عناصر کاذب و کلاس‌های کاذب (pseudo-class) می‌باشد :

`p::first-letter` عنصر کاذب
`a:hover` کلاس کاذب

عناصر کاذب با قاعده `(::)` در IE پشتیبانی نمی‌شود اما عنصر کاذب با قاعده `(:)` در تمامی مرورگرهای اصلی پشتیبانی می‌شود و توصیه می‌شود که از `(:)` به جای `(::)` استفاده شود .

کلاس‌های کاذب ساختاری (structural pseudo-classes)

شبيهه به `direct child selector` و `next sibling selector` که قبلا گفته شد ، کلاس‌های کاذب ساختاری برای اشاره به موقعیت (Position) عناصر در document به کار می‌روند .

`:first-child`

این کلاس کاذب ساختاری فقط هنگامی پذیرفته می‌شود که عنصری اولین فرزند از عنصر دیگری باشد . تمامی مرورگرهای اصلی به جز IE6 از این کلاس کاذب ساختاری پشتیبانی می‌کنند .

`:last-child`

این کلاس کاذب ساختاری فقط هنگامی پذیرفته می‌شود که عنصری آخرین فرزند از عنصر دیگری باشد . IE6,7,8 از این کلاس کاذب پشتیبانی نمی‌کنند . سایر مرورگرهای اصلی از جمله IE9 از این کلاس کاذب پشتیبانی می‌کنند .

`:nth-child(n)`

این کلاس کاذب ساختاری فقط هنگامی پذیرفته می‌شود که عنصری `n` امین فرزند از عنصر دیگری باشد . به عبارت دیگر اگر مقدار داخل پرانتز (`n`) به طور مثال برابر با عدد ۳ باشد ، سومین فرزند عنصر انتخاب می‌شود . IE6,7,8 از این کلاس کاذب پشتیبانی نمی‌کنند . سایر مرورگرهای اصلی از جمله IE9 از این کلاس کاذب پشتیبانی می‌کنند .

مثال :

```
form div:first-child{
    background:#cc5;
}
```

دستور فوق به این معنی است که عنصر `div` ی که اولین فرزند عنصر `form` می‌باشد دارای رنگ پس زمینه `#cc5` باشد و نه اینکه اولین فرزند عنصر `div` دارای این رنگ باشد .

مثال :

```
form div:last-child{
    background:#cc5;
}
```

دستور فوق به این معنی است که عنصر `div` ی که آخرین فرزند عنصر `form` می‌باشد دارای رنگ پس زمینه `#cc5` باشد و نه اینکه آخرین فرزند عنصر `div` دارای این رنگ باشد .

مثال :

```
form div:nth-child(4){  
    background:#cc5;  
}
```

دستور فوق به این معنی است که عنصر `div` ی که چهارمین فرزند عنصر `form` می باشد دارای رنگ پس زمینه ی `#cc5` باشد و نه اینکه چهارمین فرزند عنصر `div` دارای این رنگ باشد .

فصل سیزدهم

Styling for Print

در طراحی یک سند HTML برخی از قسمت های صفحه مخصوص print گرفتن می باشند و باید از style مربوط به print پیروی کنند و بعضی دیگر از قسمت ها از جمله navigation سایت نیازی به print ندارد و می تواند از هر نوع style ای استفاده کند .
مقادیر مطلق (absolute length) مثل inch ، centimeter ، point و سایر مقادیر مطلق برای print مناسب می باشند .

از آنجاییکه print اسناد رنگی هزینه ی بیشتری دارد پس اسنادی که مخصوص print طراحی می شوند نسبت به سایر اسناد ساده تر بوده و از رنگ و background کمتری استفاده می کنند و اکثرا از رنگ های سیاه و سفید استفاده می کنند .

CSS چندین property برای کنترل نحوه ی نمایش سند در رسانه ها ی (media) مختلف فراهم می کند .

برای داشتن یک print درست style ای که برای اسناد مخصوص print و اسناد مخصوص صفحه نمایش (screen) تعریف می شود باید متفاوت باشد .

CSS می تواند اسناد متفاوتی را بپذیرد ؛ نه تنها اسناد HTML و CSS بلکه می تواند استفاده از HTML و CSS را در device های مختلف و رسانه های مختلف پشتیبانی کند .

برای هدف قرار دادن رسانه های مختلف باید از صفت media (attribute) در تگ <link/> یا در تگ <style/> استفاده کرد یا درون قواعد نوشته شده ی CSS می توان از قاعده ی @media برای نشان کردن رسانه های مختلف استفاده کرد .

مثال :

```
<link rel="stylesheet" type="text/css" media="print" href="myprint.css" />
```

```
<style type="text/css" media="print"> ... </style>
```

```
<style type="text/css">
```

```
...
```

```
...
```

```
@media print{
```

```
  p{
```

```
    font: 1em bold tahoma;
```

```
    color:#ff0;
```

```
  }
```

```
  ...
```

```
  ...
```

```
}
```

```
</style>
```

انواع مختلف media عبارتند از :

- ✓ all : مخصوص تمامی device ها
- ✓ braille : استفاده برای device های مخصوص نابینایان
- ✓ embossed : استفاده برای printer های مخصوص خط braille
- ✓ handheld : استفاده برای device های جیبی و قابل حمل مثل tablet ها

- ✓ **print** : این مقدار برای ارائه ی **style** به **printer** استفاده می شود . (در مرورگرها می توان از قسمت **print preview** برای مشاهده ی صفحه در این حالت استفاده کرد) .
- ✓ **projection** : استفاده برای ویدئو پروژکتورها (**projected presentations**)
- ✓ **screen** : مخصوص نمایش در صفحه نمایش کامپیوتر
- ✓ **speech | aural** : استفاده در **device** های مخصوص سخنرانی های ترکیبی (**Intended for presentation to a speech synthesizer**)
در **CSS2** مقدار **aural** و در **CSS2.1** مقدار **speech** معادل یکدیگر می باشند .
- ✓ **tty** : استفاده برای ابزارهایی با کاراکترهای طول ثابت (**fixed-pitch**) مثل ماشین تحریر (**teletypes**) و ابزار های قابل حمل (**portable device**) با قابلیت صفحه نمایش محدود
- ✓ **tv** : استفاده برای تلویزیون (با وضوح پایین ، رنگ پایین ، **scroll** محدود)

مقدار پیش فرض : **all**

فقط مقادیر **screen** ، **print** و **all** در مرورگرها به طور گسترده پشتیبانی می شوند .

برای تعریف نوع **media** در بین کدهای **CSS** از قاعده ی **@media** استفاده می شود . عنصر موردنظر (و خاصیت های آن را) که می خواهیم **media** روی اعمال شود را در بین این دستور محصور می کنیم :

```
@media{
  h1{
    font-size: 50pt;
    text-align: center;
  }
  p{
    text-indent: 25px;
    color: #000;
  }
  ...
}
```

نکته :

فونت **sans-serif** متن را در صفحه نمایش (**screen**) خواناتر کرده و فونت **serif** برای خوانایی بیشتر متن پرینت شده مناسب می باشد . بیشتر وب سایت ها از فونت **sans-serif** برای **body** استفاده می کنند در حالیکه کتاب ها و روزنامه ها با فونت **serif** چاپ می شوند .

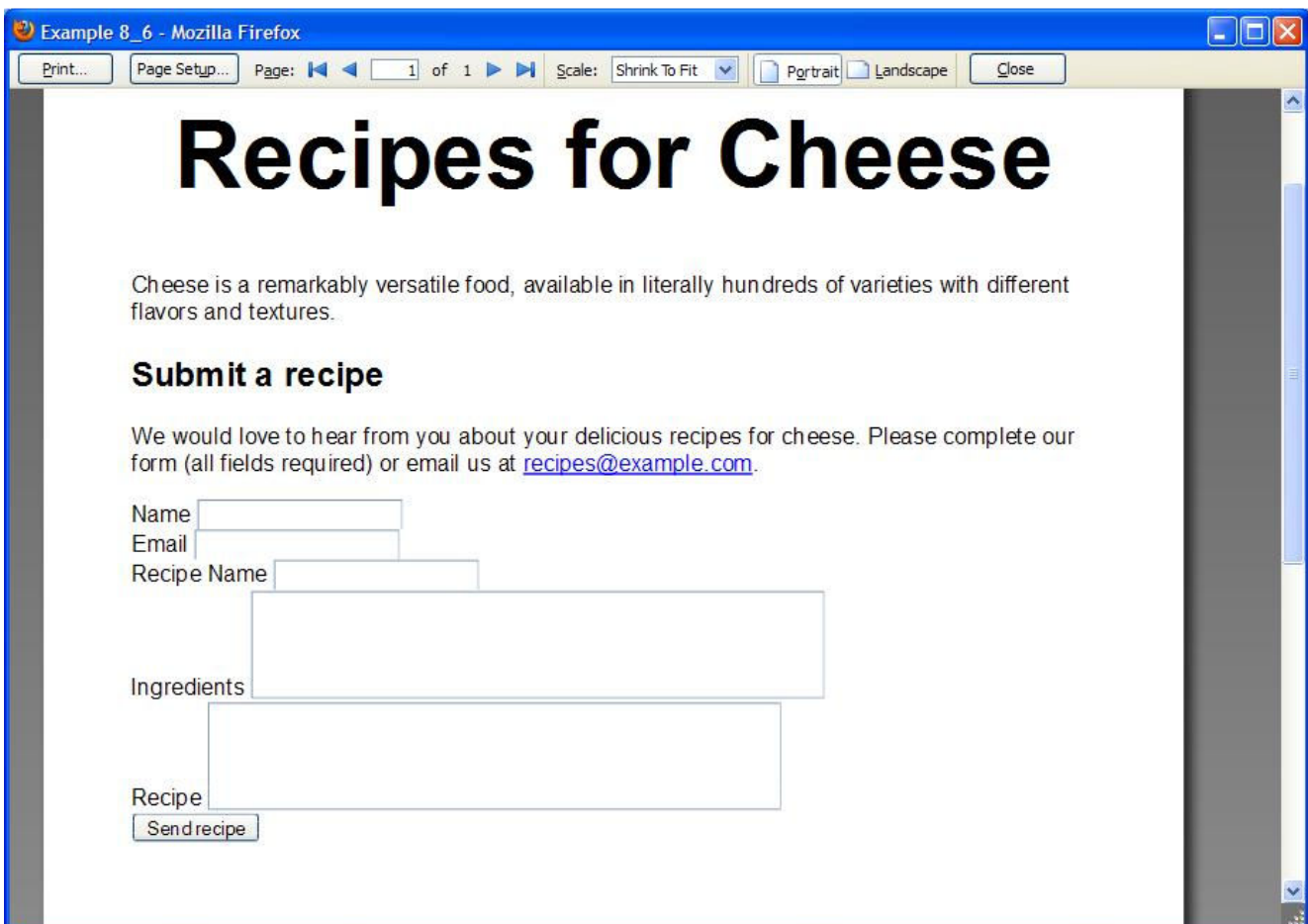
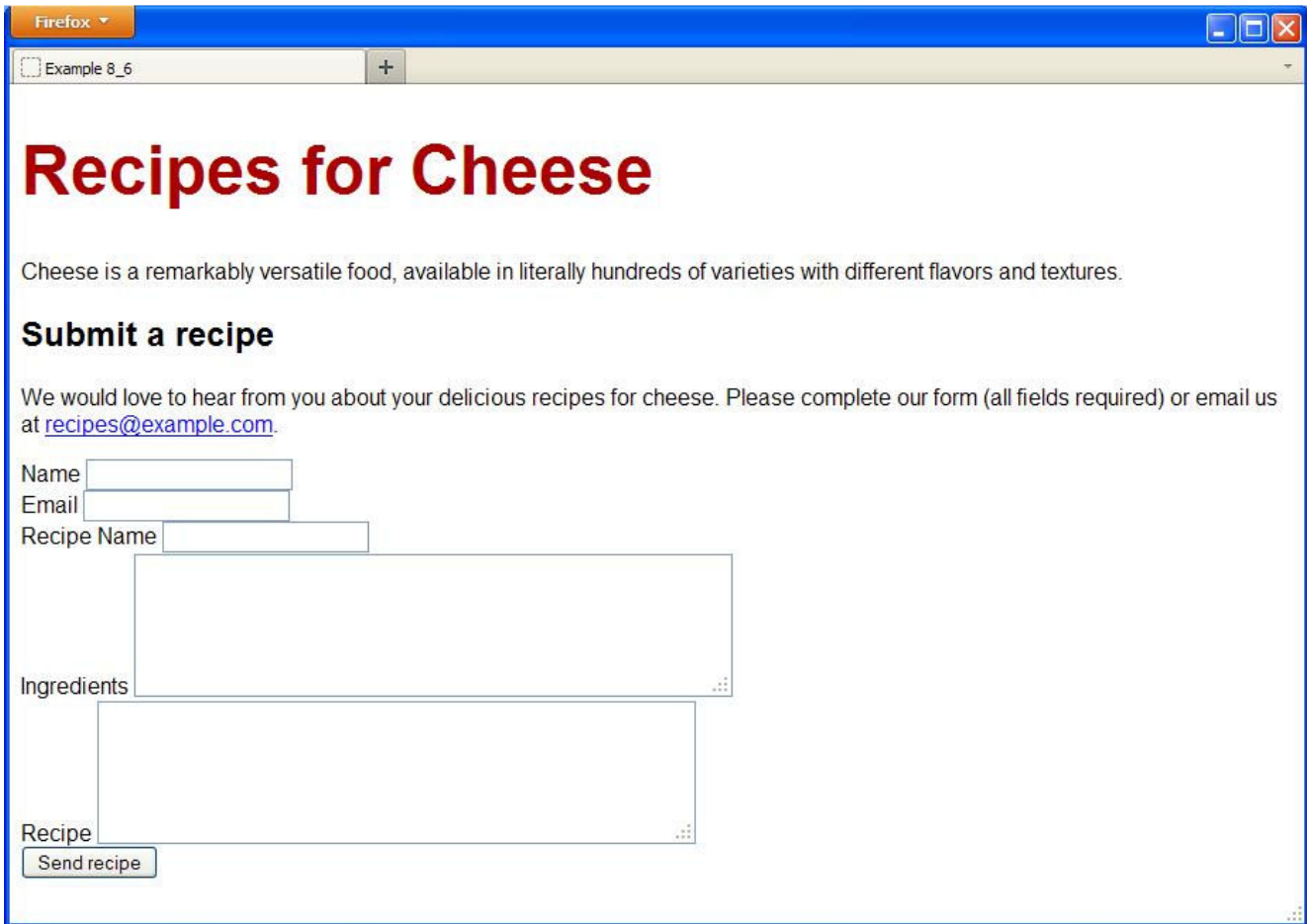
مثال :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 8_6</title>
<style type="text/css">
  @media screen {
    h1 {
      font-size: 3em;
      color: #A00;
    }
  }
  @media print {
    h1 {
```

```

        font-size: 50pt;
        text-align: center;
    }
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<div class="content">
    <h2>Submit a recipe</h2>
    <p class="intro">We would love to hear from you about your delicious recipes
for cheese. Please complete our form (all fields required) or email us at
<a href="mailto:recipes@example.com">recipes@example.com</a>.</p>
    <form method="post" action="">
        <div>
            <label for="submit-name">Name</label>
            <input type="text" name="name" id="submit-name">
        </div>
        <div>
            <label for="submit-email">Email</label>
            <input type="text" name="email" id="submit-email">
        </div>
        <div>
            <label for="submit-recipe-name">Recipe Name</label>
            <input type="text" name="recipe-name" id="submit-recipe-name">
        </div>
        <div>
            <label for="submit-ingredients">Ingredients</label>
            <textarea name="ingredients" id="submit-ingredients" rows="5"
            cols="50"></textarea>
        </div>
        <div>
            <label for="submit-recipe">Recipe</label>
            <textarea name="recipe" id="submit-recipe" rows="5"
            cols="50"></textarea>
        </div>
        <div class="submit">
            <input type="submit" value="Send recipe">
        </div>
    </form>
</div>
</body>
</html>

```



در مثال فوق واحد اندازه گیری فونت هنگام نمایش در screen متفاوت با واحد اندازه گیری فونت در هنگام print خواهد بود. در اصل مثال فوق دو style متفاوت برای screen و برای print تعریف کرده است.

controlling page breaks

دو خاصیت مختص print (یا page media) که در مرورگرها متداول هستند عبارتند از :

page-break-before: auto | always | avoid | left | right

page-break-after:

مقدار پیش فرض : auto

کلمات کلیدی که به طور قابل اعتماد در تمامی مرورگرها پشتیبانی می شوند عبارتند از : auto , always

مثال :

```
h1{
    page-break-after: always;
}
```

دستور فوق باعث می شود که در هنگام print پس از عنصر h1 صفحه شکسته شده و مابقی محتوای صفحه در یک صفحه ی جدید چاپ شود. page-break-before مشابه فوق می باشد و فقط صفحه را مجبور به شکست قبل از عنصر موردنظر می کند.

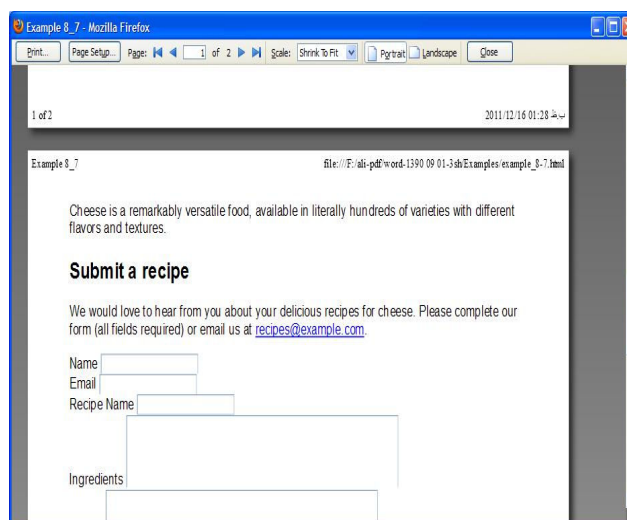
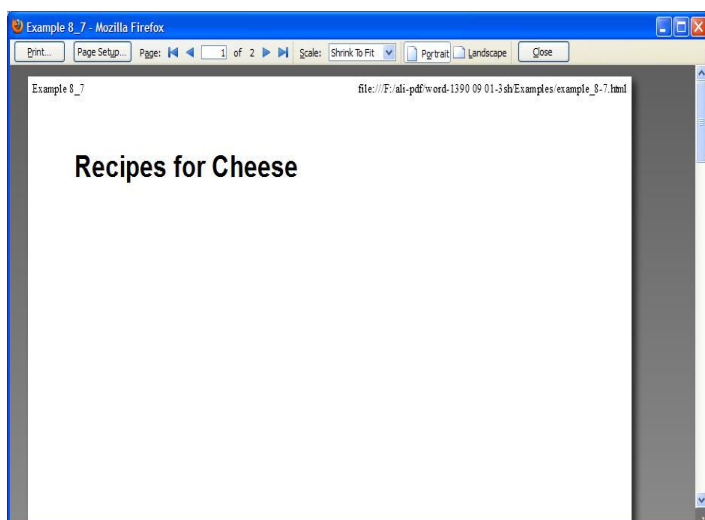
مثال :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 8_7</title>
<style type="text/css" media="print">
h1{
    page-break-after:always;
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<div class="content">
<h2>Submit a recipe</h2>
<p class="intro">We would love to hear from you about your delicious recipes
for cheese. Please complete our form (all fields required) or email us at
<a href="mailto:recipes@example.com">recipes@example.com</a>.</p>
<form method="post" action="">
<div>
<label for="submit-name">Name</label>
<input type="text" name="name" id="submit-name">
</div>
<div>
<label for="submit-email">Email</label>
<input type="text" name="email" id="submit-email">
</div>
</div>
```

```

<label for="submit-recipe-name">Recipe Name</label>
<input type="text" name="recipe-name" id="submit-recipe-name">
</div>
<div>
<label for="submit-ingredients">Ingredients</label>
<textarea name="ingredients" id="submit-ingredients" rows="5"
cols="50"></textarea>
</div>
<div>
<label for="submit-recipe">Recipe</label>
<textarea name="recipe" id="submit-recipe" rows="5"
cols="50"></textarea>
</div>
<div class="submit">
<input type="submit" value="Send recipe">
</div>
</form>
</div>
</body>
</html>

```



content property

خصوصیت `content` راهی است برای ما تا با استفاده از `CSS` محتوایی را درون یک صفحه وب قرار دهیم. این محتوا دقیقا قسمتی از صفحه نیست بنابراین برای برخی از تکنولوژی ها (مثل `screen readers`) قابل دسترسی نیست. تمامی مرورگرهای اصلی به جز `IE6` از خاصیت `content` پشتیبانی می کنند.

یک اشکال عمده در پرینت صفحات وب این است که به صورت پیش فرض `URL` تمامی لینک ها برای ما از دست می رود. از روی `style` تعریف شده برای یک لینک می توان ان را در صفحه تشخیص داد، اما هیچ راهی وجود ندارد که بفهمیم یک لینک به چه آدرسی می رود مگر اینکه روی آن لینک در مرورگر خود کلیک کنیم. با کمی جادوی `CSS` می توان این مسئله را رفع کرد:

```

a:link:after{
  content: "(http://example.com"attr(href)");
}

```

با استفاده از قاعده ی فوق می توان مشاهده کرد که URL یک لینک بلافاصله پس از لینک به نمایش در خواهد آمد . در مثال فوق رشته ی `http://example.com` به عنوان domain سایت ما قبل از `attr(href)` آمده و از تعدادی پرانتز نیز به عنوان حائل بین عنصر و مقدار پس از آن استفاده شده است .
مثال :

```
a:hover:after{
    content: attr(name);
}
```

با استفاده از این قاعده مقدار صفت ذکر شده در داخل پرانتز بلافاصله پس از عنصر مورد نظر به نمایش در خواهد آمد .

کلاس کاذب `after`: مشخص می کند که محتوای ذکر شده در صفت `content` باید بلافاصله پس از عنصر موردنظر به نمایش درآید . در مثال فوق `attr(name)` مشخص می کند که باید مقدار صفت `name` از تگ `<a>` بلافاصله پس از لینک به نمایش درآید .

در خاصیت `content` از کلاس کاذب `before` نیز می توان استفاده کرد .
مثال :

```
p:before{
    content: attr(title);
}
```

با استفاده از دستور فوق مقدار صفت `title` از عنصر `<p>` قبل از آن درون سند درج خواهد شد .

مثال :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 8_8</title>
<style type="text/css">

a:link:after {
    content: " (http://example.com" attr(href) ") ";
}
p:before{
    content: attr(class);
    color:red;
}

</style>
</head>
<body>
<h1><a href="/">Recipes for Cheese</a></h1>
<p class="intro"> &nbsp; Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
<div class="content">
    <h2>Submit a recipe</h2>
```

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



فصل چهاردهم

Mobile Devices

مرورگرهای مخصوص mobile و device های با صفحه نمایش کوچک به دلیل محدودیت در اندازه ی صفحه نمایش چالشی را در طراحی style ایجاد می کنند که همین امر باعث می شود تا style متفاوتی را برای آنها تعریف کنیم . (همانند تعریف دو style متفاوت برای print و screen باید دو style متفاوت نیز برای کامپیوتر و mobile به خاطر تفاوت در اندازه ی screen تعریف کرد .) در موبایل و device های با صفحه نمایش کوچک می توان از media ی handheld استفاده کرد اما در عمل تعداد کمی از موبایل ها از این media پشتیبانی می کنند .

media ی استفاده شده برای mobile :

```
<style type="text/css" media="screen and (max-device-width:480px)"> ... </style>
```

```
<link rel="stylesheet" type="text/css" media=" screen and (max-device-width:480px)"
href="style.css" />
```

```
<style type="text/css">
    @media screen and (max-device-width:480px){
        ...
    }
</style>
```

max-device-width حداکثر اندازه ی صفحه نمایش را مشخص می کند که در mobile های پیشرفته ی امروزی معمولا 480pixel می باشد .

مثال :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 14_1</title>
<style type="text/css">
    body {
        width: 600px;
        margin: 1em auto;
    }

    h1 {
        float: left;
        margin-top: 0.2em;
    }
    .navigation {
        float: right;
    }
    .navigation li {
        display: inline;
    }
    .navigation a {
```

```

margin-left: 0.5em;
padding: 0.5em;
border: 1px solid #CCC;
}

.intro {
clear: both;
}

@media screen and (max-device-width: 480px) {
body {
width: auto;
margin: 1em;
}
}

@media screen and (max-device-width: 480px) and
(orientation: portrait) {
h1,
.navigation {
float: none;
}
}
</style>
</head>
<body>
<h1>Recipes for Cheese</h1>
<ul class="navigation">
<li><a href="#">Home</a></li>
<li><a href="#">Recipes</a></li>
<li><a href="#">Suggestions</a></li>
</ul>
<p class="intro">Cheese is a remarkably versatile food, available in
literally hundreds of varieties with different flavors and textures.</p>
<div class="recipe">
<h2>Welsh Rarebit</h2>
<p class="intro">Welsh Rarebit is a savory dish made from melted
cheese, often Cheddar, on toasted bread, and a variety of other
ingredients such as mustard, egg, or bacon. Here is one take on
this classic.</p>
<ol>
<li>Lightly toast the bread</li>
<li>Place on a baking tray, and spread with butter.</li>
<li>Add the grated Cheddar cheese and 2 tablespoons of beer to
a saucepan. Place the saucepan over a medium heat, and stir the
cheese continuously until it has melted. Add a teaspoon of
wholegrain mustard and grind in a little pepper. Keep stirring.</li>
<li>When thick and smooth, pour over each piece of toast
spreading it to the edges to stop the toast from burning.</li>
<li>Place under the grill for a couple of minutes or until golden brown.</li>
</ol>
</div>

```

</body>

</html>

در مثال فوق در عبارت (orientation: portrait) and (max-device-width: 480px) and @media screen مقدار portrait مشخص می کند که این style تعریف شده برای حالت عمودی صفحه نمایش در موبایل می باشد . تصاویر زیر خروجی مثال فوق را در iPhone simulator نشان می دهند .



style استفاده شده برای حالت افقی همان style اصلی تعریف شده بعلاوه ی style تعریف شده در اولین بلاک @media می باشد . style تعریف شده در بلاک دوم @media فقط در حالت عمودی صفحه نمایش موبایل اعمال خواهد شد .

می توان با تخصیص مقدار landscape به orientation برای حالت افقی صفحه نمایش موبایل نیز یک style جداگانه تعریف کرد .

فصل پانزدهم

Complete Layout

مثال : یک صفحه وب کامل بدون استفاده از CSS

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

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 15-1</title>

<style type="text/css">

</style>

</head>

<body>
<div class="header">
  <h1>Recipes for Cheese</h1>

  <ul class="navigation">
    <li><a href="#">Home</a></li>
    <li><a href="#">Recipes</a></li>
    <li><a href="#">Suggestions</a></li>
  </ul>
  <p class="intro">Cheese is a remarkably versatile food, available in literally
  hundreds of varieties with different flavors and textures.</p>
</div>
<h2>Welsh Rarebit</h2>
<p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
<div class="ingredients">
  <h3>Ingredients</h3>
  <table>
    <colgroup>
      <col class="ingredient">
      <col class="quantity">
    </colgroup>
    <thead>
      <tr>
        <th>Ingredient</th>
        <th>Quantity</th>
      </tr>
```

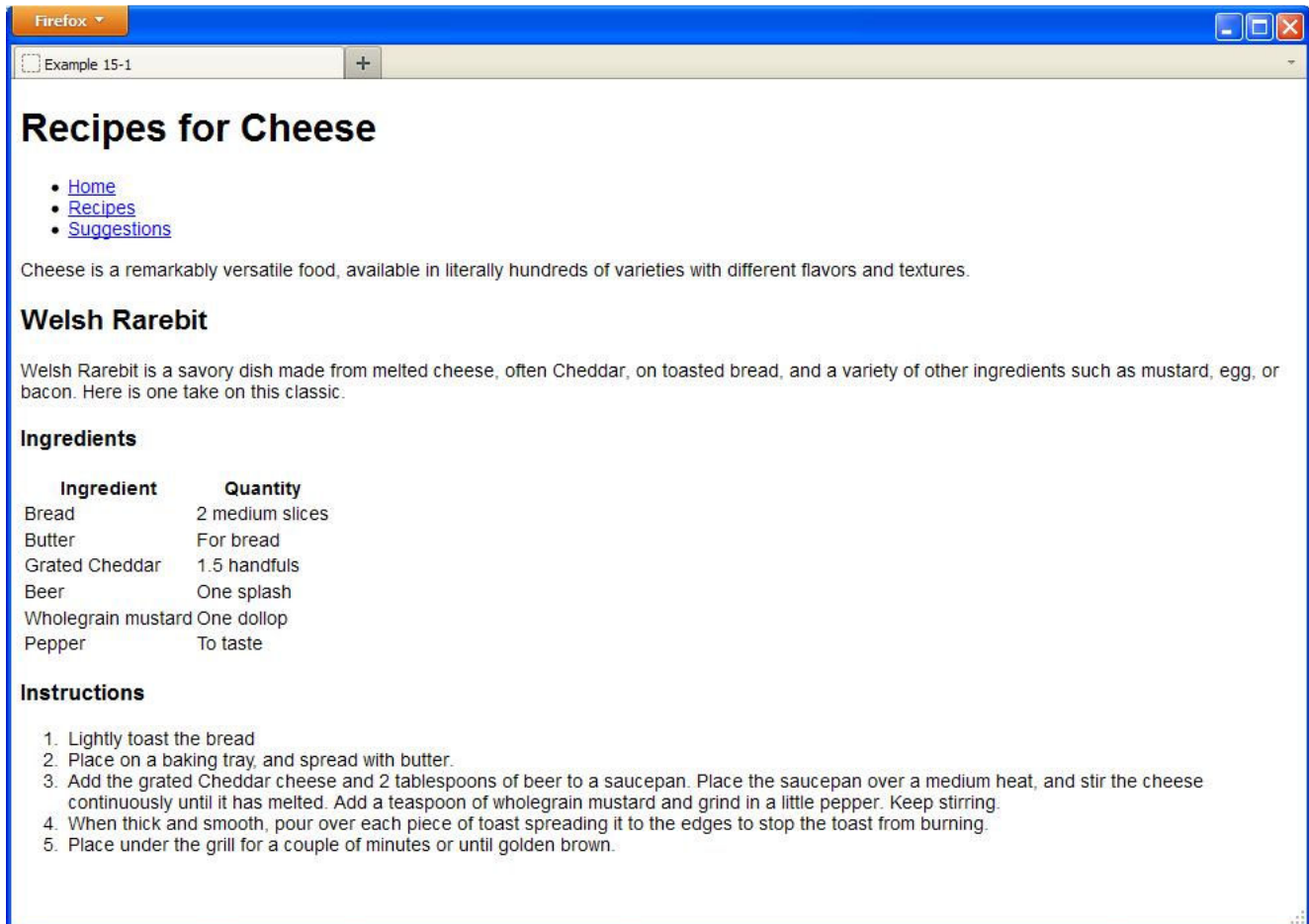
```

</thead>
<tbody>
  <tr>
    <td>Bread</td>
    <td>2 medium slices</td>
  </tr>
  <tr>
    <td>Butter</td>
    <td>For bread</td>
  </tr>
  <tr>
    <td>Grated Cheddar</td>
    <td>1.5 handfuls</td>
  </tr>
  <tr>
    <td>Beer</td>
    <td>One splash</td>
  </tr>
  <tr>
    <td>Wholegrain mustard</td>
    <td>One dollop</td>
  </tr>
  <tr>
    <td>Pepper</td>
    <td>To taste</td>
  </tr>
</tbody>
</table>
</div>
<h3>Instructions</h3>
<ol>
  <li>Lightly toast the bread</li>
  <li>Place on a baking tray, and spread with butter.</li>
  <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan. Place the saucepan over a medium heat, and stir the cheese continuously until it has melted. Add a teaspoon of wholegrain mustard and grind in a little pepper. Keep stirring.</li>
  <li>When thick and smooth, pour over each piece of toast spreading it to the edges to stop the toast from burning.</li>
  <li>Place under the grill for a couple of minutes or until golden brown.</li>
</ol>

</body>

</html>

```



مثال : یک صفحه وب کامل با استفاده از CSS

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

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 15-2</title>

<style type="text/css">

html {
height: 100%;
background-color: #555;
}

body {
width: 700px;
margin: 0 auto;
padding: 29px;
border: 1px solid #233;
border-top: none;
background: #FFF;
```

```
font: 82%/1.4 Verdana, Arial, Helvetica, sans-serif;
}

h1 {
float: left;
margin-top: 0;
font-family: Raleway, Georgia, Times, Times New Roman, serif;
}

h2, h3 {
font-family: Georgia, Times, Times New Roman, serif;
}

.header {
border-bottom: 6px solid #233;
}

.navigation {
float: right;
margin: 0;
list-style: none;
}

.navigation li {
float: left;
}

.navigation a {
display: block;
margin-left: 0.5em;
padding: 0.5em;
border: 1px solid #CCC;
color: #233;
text-decoration: none;
}

.navigation a:focus,
.navigation a:hover {
background: #233;
color: #FFF;
}

.header .intro {
clear: both;
font-weight: bold;
}

.intro {
font-style: italic;
}

.ingredients {
float: right;
margin: 0 0 15px 15px;
padding-left: 15px;
}
```

```

border-left: 1px solid #CCC;
}

.ingredients h3 {
margin-top: 0;
}

.ingredients th {
text-align: left;
}

.ingredients th,
.ingredients td {
padding: 2px 5px;
}

.ingredients .ingredient {
background-color: #CC5;
}

.ingredients .quantity {
background-color: #CCF;
}
</style>
</head>
<body>
<div class="header">
<h1>Recipes for Cheese</h1>

<ul class="navigation">
<li><a href="#">Home</a></li>
<li><a href="#">Recipes</a></li>
<li><a href="#">Suggestions</a></li>
</ul>
<p class="intro">Cheese is a remarkably versatile food, available in literally
hundreds of varieties with different flavors and textures.</p>
</div>
<h2>Welsh Rarebit</h2>
<p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often
Cheddar, on toasted bread, and a variety of other ingredients such as mustard,
egg, or bacon. Here is one take on this classic.</p>
<div class="ingredients">
<h3>Ingredients</h3>
<table>
<colgroup>
<col class="ingredient">
<col class="quantity">
</colgroup>
<thead>
<tr>

```



```

    <th>Ingredient</th>
    <th>Quantity</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>Bread</td>
    <td>2 medium slices</td>
  </tr>
  <tr>
    <td>Butter</td>
    <td>For bread</td>
  </tr>
  <tr>
    <td>Grated Cheddar</td>
    <td>1.5 handfuls</td>
  </tr>
  <tr>
    <td>Beer</td>
    <td>One splash</td>
  </tr>
  <tr>
    <td>Wholegrain mustard</td>
    <td>One dollop</td>
  </tr>
  <tr>
    <td>Pepper</td>
    <td>To taste</td>
  </tr>
</tbody>
</table>
</div>
<h3>Instructions</h3>
<ol>
  <li>Lightly toast the bread</li>
  <li>Place on a baking tray, and spread with butter.</li>
  <li>Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan. Place the saucepan over a medium heat, and stir the cheese continuously until it has melted. Add a teaspoon of wholegrain mustard and grind in a little pepper. Keep stirring.</li>
  <li>When thick and smooth, pour over each piece of toast spreading it to the edges to stop the toast from burning.</li>
  <li>Place under the grill for a couple of minutes or until golden brown.</li>
</ol>

</body>

</html>

```

The screenshot shows a web browser window with the title "Example 15-2". The page content is as follows:

Recipes for Cheese

Home Recipes Suggestions

Cheese is a remarkably versatile food, available in literally hundreds of varieties with different flavors and textures.

Welsh Rarebit

Welsh Rarebit is a savory dish made from melted cheese, often Cheddar, on toasted bread, and a variety of other ingredients such as mustard, egg, or bacon. Here is one take on this classic.

Instructions

1. Lightly toast the bread
2. Place on a baking tray, and spread with butter.
3. Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan. Place the saucepan over a medium heat, and stir the cheese continuously until it has melted. Add a teaspoon of wholegrain mustard and grind in a little pepper. Keep stirring.
4. When thick and smooth, pour over each piece of toast spreading it to the edges to stop the toast from burning.
5. Place under the grill for a couple of minutes or until golden brown.

Ingredients

Ingredient	Quantity
Bread	2 medium slices
Butter	For bread
Grated Cheddar	1.5 handfuls
Beer	One splash
Wholegrain mustard	One dollop
Pepper	To taste