

۷ کارنیل، بزرگترین شبکه موفقیت ایرانیان می باشد، که افرادی زیادی توانسته اند با آن به موفقیت برسند، فاطمه رتبه ۱۱ کنکور کارشناسی، محمد حسین رتبه ۶۸ کنکور کارشناسی، سپیده رتبه ۳ کنکور ارشد، مریم و همسرش راه اندازی تولیدی مانتو، امیر راه اندازی فروشگاه اینترنتی، کیوان پیوستن به تیم تراکتور سازی تبریز، میلاد پیوستن به تیم صبا، مهسا تحصیل در ایتالیا، و.... این موارد گوشه از افرادی بودند که با کارنیل به موفقیت رسیده اند، شما هم می توانید موفقیت خود را با کارنیل شروع کنید.

برای پیوستن به تیم کارنیلی های موفق روی لینک زیر کلیک کنید.

www.karnil.com

همچنین برای ورود به کانال تلگرام کارنیل روی لینک زیر کلیک کنید.

<https://telegram.me/karnil>

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

هر کس به زبانی صفت حمد تو گوید

بلبل به غزل خوانی و قمری به ترانه

آموزش ویژگی (property) های Css3

(بخش اول)

Border-radius
Text-shadow
Box-shadow
Multiple background images
Opacity
RGBA
Rotate
Scale
Skew
Animation! (BEGINNER)

نویسنده:

رضا شیخله

Rezaonline.net

RezaOnline.net/blog

info@Rezaonline.net

آبان 1390

منبع:

CSS3 FOR
WEB DESIGNERS
Dan Cederholm

- 7.....css3 شروع کار با ❖
- 7پشتیبانی مرورگرها..... ❖
- 8پیشوند اختصاصی مرورگرها ❖
- 9 border-radius گوشه اجزا بوسیله ❖
- 11 text-shadow سایه متنی برای نوشته ها ❖
- 12 box-shadow (table,div,span,...) سایه برای سایر اجزا ❖
- 14 (background) چند جزئی ❖
- 14 opacity سفید کردن (تار کردن یا کمرنگ کردن) تصاویر با ❖
- 16..... rgba! رنگ شیشه ای! ❖
- 21..... چرخش اجزا ❖
- 22 بزرگنمایی عکس ❖
- 22.....skew کجی و انحراف ❖
- 24..... انیمیشن ساده ❖
- 29 سخن پایانی ❖

شروع کار با CSS3

در این کتاب به هیچ عنوان CSS آموزش داده نمیشود! هدف معرفی و نحوه استفاده از چند ویژگی جدید CSS3 که طراحان وب با این ویژگی های جدید آشنا شوند و بتوانند از آن استفاده کنند و با استفاده از آن کارهای خود را جذابتر و حرفه ای تر کنند. لازم درک کدها و توضیحات این کتابچه ، آشنایی و تسلط به HTML و CSS2 است. در این کتاب فقط به معرفی چند ویژگی جدید CSS3 میپردازیم که فعلاً در تمام مرورگرهای به روز قابل نمایش است. لازم به ذکر است که CSS3 محدود به این چند ویژگی نیست ، برای مثال ویژگی های جدید `text-underline` , `text-overline` , `target` , `transform-orig` , `line-stacking` , `line-grid` , `image-resolution` هنوز بطور کامل توسط مرورگرها پشتیبانی نمیشوند! پس فعلاً از آموزش و بررسی آنها خودداری کنیم! (چون عالم بی عمل به هیچ دردی نمیخوره !)

پشتیبانی مرورگرها

نحوه پشتیبانی مرورگرها از این چند ویژگی جدید در جدول زیر کاملاً مشخصه.

safari	chrome	mozilla	opera	IE	ویژگی / مرورگر
+3	+3	+1	+10.5	9 بتا	Border-radius
+1.1	+2	+3.1	+9.5	---	Text-shadow
+3	+3	+3.5	+10.5	9 بتا	box-shadow
+1.3	+2	+3.6	+10.5	9 بتا	Multiple background images
+1.2	+1	+1.5	+9	9بتا	opacity
+3.2	+3	+3	+10	9بتا	RGBA

البته لازمه بگم `border-radius` های جزئی مثل `border-top-right-radius` (گرد کردن گوشه بالا سمت راست) در مرورگر فایرفاکس پائینتر از 4 نشان داده نمیشود!

پیشوند اختصاصی مروگرها

در CSS3 بیشتر روی حالت نمایشی تاکید شده و بر طبق همین روال مرورگرهای مختلف ، تعبیری متفاوت از یک ویژگی را ارائه میدهند . بطور مثال مروگر سافاری از کد

```
div.className{
  box-shadow: 1px 1px 1px 1px #000000 inset;
}
```

پشتیبانی نمیکند ، در واقع ما انتظار داریم که این کد (ویژگی) سایه ای را برای div مورد نظر ایجاد کند اما اینطور نیست. راه حل استفاده از پیشوند اختصاصی است . بطور مثال کد بالا برای اینکه در مروگر سافاری کار کند باید بصورت

```
div.className{
  box-shadow: 1px 1px 1px 1px #000000 inset;
  -webkit-box-shadow: 1px 1px 1px 1px #000000 inset;
}
```

تغییر یابد . در واقع پیشوند **webkit** مخصوص مروگر سافاری است . پیشوند مروگرهای مختلف در جدول زیر هست .

 WebKit	-webkit-
 Mozilla	-moz-
 Opera	-o-
 Konqueror	-khtml-
 Microsoft	-ms-
 Chrome	-chrome-

* طبق تجربیات بنده مروگر گوگل کروم نیز با افزودن پیشوند **-webkit-** ویژگی مورد نظر را تاثیر میدهد!

گرد کردن گوشه اجزا بوسیله border-radius

دیگر زمان آن منسوخ شده که همه ما برای یک گرد کردن ساده از انواع روشها استفاده میکردیم و تقریباً همه ی آنها شامل افزودن عکسهای متعدد به گوشه ها میشد! حال با استفاده از این ویژگی به راحتی میتوان گوشه های اجزا را گرد کرد.

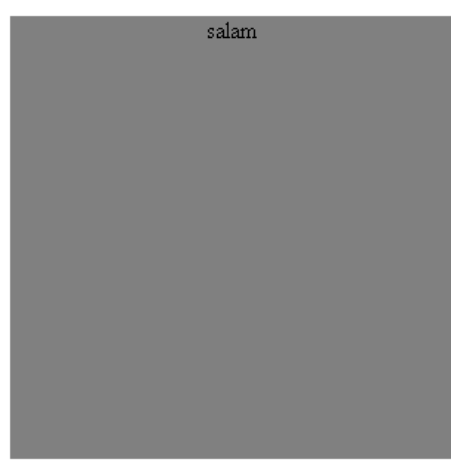
فرض کنید ما در صفحه html خود چنین داریم

```
<div class="use-border-radius">
salam
</div>
```

و در استایل خود چنین

```
div.use-border-radius{
width : 300px;
height : 300px;
background : gray;
}
```

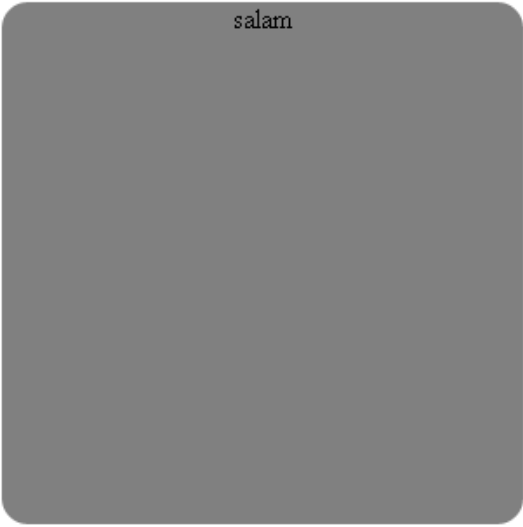
پس در مرورگر چنین چیزی مشاهده خواهیم کرد.



اکنون برای گرد کردن تمامی گوشه های این مربع فقط کافی است ویژگی border-radius را اضافه کنیم.

```
div.use-border-radius{
width : 300px;
height : 300px;
background : gray;
border-radius:15px
}
```

و نتیجه به این صورت (در مرورگر کروم) نشان داده میشود.



همانطور که میبیند ویژگی `border-radius` بدون نیاز به پیشوند `webkit` یا `chrome` در گوگل کروم گوشه ها را گرد میکند و اثر میدهد.

اگر در مروگری این ویژگی نشان داده نشده پیشوند مشخصه آن مروگر را در ابتدای ویژگی اضافه کنید. پیشنهاد میشود استایل را بصورت زیر بنویسید تا خیالتان از نمایش ندادن در مروگری خاص ، راحت کند.

```
div.use-border-radius{
width : 300px;
height : 300px;
background : gray;
border-radius:15px;
-moz-border-radius:15px; /* mozilla */
-o-border-radius:15px; /* opera */
-webkit-border-radius:15px; /* chrome and safari */
}
```

* نکته جالب دیگه اینکه اگر مقدار `border-radius` را زیاد انتخاب کنید ، `div` مورد نظر (اگر مربع باشد) تبدیل به دایره میشود !

اکنون شاید شما نیاز داشته باشید که فقط گوشه ای خاص را گرد کنید . `border-radius` قابل تجزیه شدن به ویژگی های کوچکتری است که میتوان نیاز شما را برآورده کند.

```
border-top-left-radius:15px; /*بالا چپ*/
border-top-right-radius:10px ; /*بالا راست*/
border-bottom-left-radius:8px ; /*پائین چپ*/
border-bottom-right-radius:20 px; /*پائین راست*/
```

نکته قابل توجه اینکه شما میتوانید هر چهار ویژگی جزئی را ، یا بصورت تک تک و یا باهم بکار ببرید. نیاز به یادآوری نیست در مورد این ویژگی هاهم اگر در مرورگری نشان داده نشد از پیشوند اختصاصی آن مرورگر استفاده کنید!
(قول میدم دیگه نگم !)

سایه متنی برای نوشته ها `text-shadow`

Text-shadow ساخت سایه برای نوشته ، متن ، لینک و ...

" البته پیشبینی میشود در آینده نزدیک خیلی از دوستانی که خواننده سایتهایی هستن که از این ویژگی بطور غلط استفاده کردن ، دچار کوری ، چشم زخم ، کوررنگی ، ضعف چشم و خلاصه چشمشون مشکل پیدا میکنه !
 چون خیلی از سایتهای ایرانی که از این ویژگی توی استایلشون استفاده کردن بطور غیر استاندارد رنگهایی رو بکار بردن که چشم رو اذیت میکنه ! مثلاً رنگ سیاه با سایه قرمز دیگه پدر چشم منو در میاره!
 پیشنهاد میکنم از این ویژگی برای نوشته های ثابت استفاده نکنین و فقط برای لینک ها اونم در حالت `a:active` یا `a: hover` استفاده کنین! از رنگ های ملایم هم استفاده کنین مثلاً نارنجی با سایه زرد!"

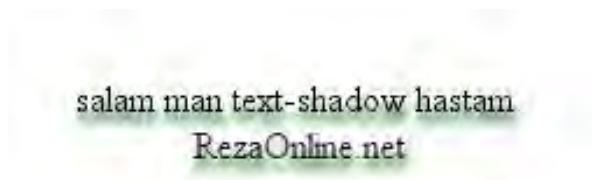
فرض کنید در صفحه `html` چنین داریم

```
<div id="use-text-shadow">
salam man text-shadow hastam <br>
RezaOnline.net
</div>
```

و استایل رو بصورت زیر داریم

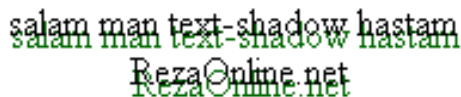
```
#use-text-shadow{
text-shadow:1px 5px 7px #006600;
}
```

نتیجه در مروگر اینگونه نشان داده میشود.



ویژگی `text-shadow` سه مقدار عددی و یک کد رنگ میگیرد. که به ترتیب به شکل زیر است
1px فاصله از متن اصلی از چپ (یک پیکسل به راست)
5px فاصله از متن اصلی از بالا (5 پیکسل به پائین)
7px میزان تار کردن سایه ساخته شده
#006600; کد رنگ سایه ساخته شده.

اگر ما میزان تار کردن سایه را 0 در نظر بگیریم در مروگر چنین چیزی خواهیم داشت.



پیشنهاد میکنم برای زیبایی کار همیشه صفر پیکسل از بالا و چپ فاصله بگیرید و سایه را تار کنید!
 نتیجه زیبایی خواهد داشت.

مثلاً استایل زیر

```
#use-text-shadow{
font-family:tahoma;
text-shadow: 0 0 5px #006600;
}
```

در مروگر به صورت

salam man text-shadow hastam
RezaOnline.net

نمایش داده میشود که به نظرم قابل قبولتر است . (اینجوری دیگه کور نمیشیم)

box-shadow

box-shadow هم به نوعی مشابه text-shadow هست با این تفاوت که سایه رو برای سایر اجزا مثل li , span , table , div و ... ایجاد میکند.

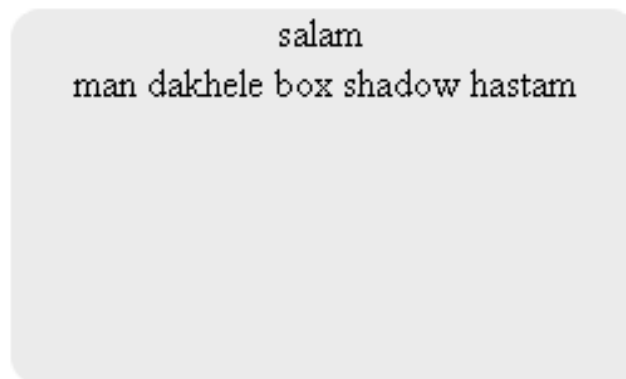
فرض کنید در صفحه html چنین چیزی داریم.

```
<div class="use-box-shadow">
salam <br>
man dakhele box shadow hastam
</div>
```

و در استایل چنین چیزی

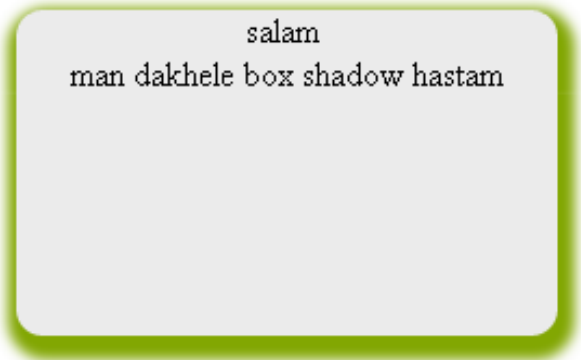
```
.use-box-shadow{
width:250px;
height:150px;
border-radius: 12px;
background: #EBEBEB;
}
```

حال در مروگر چنین نمایشی داریم.



حال اگر استایل را تغییر دهید و ویژگی `box-shadow` را اضافه کنید در مروگر خواهید داشت

```
.use-box-shadow{
width:250px;
height:150px;
border-radius: 12px;
background: #EBEBEB;
box-shadow: 2px 5px 8px 6px #82A701;
}
```



ویژگی `box-shadow` 4 مقدار عددی میگیرد و یک کد رنگ.

`2px` فاصله از چپ (2 پیکس به راست)

`5px` فاصله از بالا (5 پیکسل پائین)

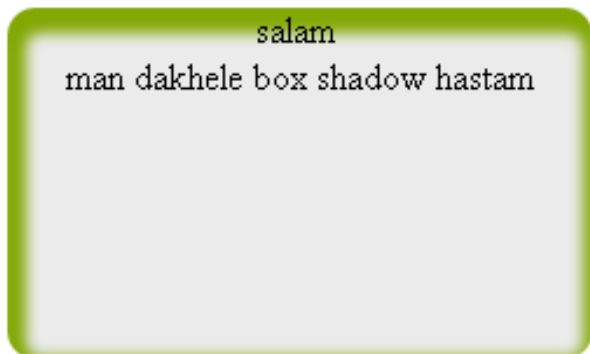
`8px` میزان تار شدن سایه

`6px` میزان وسعت سایه (حجم سایه) - اختیاری

`#82A701` کد رنگ سایه

نکته قابل ذکر در مورد `box-shadow` اینکه اگر در آخر `inset` را درج کنید سایه در داخل `div` تشکیل میشود برای مثال استایل زیر چنین نمایشی دارد.

```
.use-box-shadow{
width:250px;
height:150px;
border-radius: 12px;
background: #EBEBEB;
box-shadow: 2px 5px 8px 6px #82A701 inset;
}
```



پیشنهاد میکنم در استفاده از این کد زیاده روی نکنید و سعی کنید رنگ سایه ، مشابه پس زمینه باشد!

Multiple background images

پس زمینه (یا بک گراوند) چند جزئی

حتماً براتون پیش اومده که پس زمینه کل قالب بالا به عکس ، وسط به عکس ، پائین به عکس باشه !

به روش CSS2 می اومدیم چیکار میکردیم

توی قالب دو تا دیگه div ایجاد میکردیم مثلاً

```
<body>
<div class="bg-top">
  <div class="bg-bottom">
```

شروع محتوا

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

و استایل رو بصورت زیر مینوشتیم

```
body{
  background: url(all.png) ;
}
div.bg-top{
  background: url(top.png) repeat-x top;
}
div.bg-bottom{
  background: url(bot.png) repeat-x bottom;
}
```

حالا با استفاده از CSS3 میتونیم بک گراوند های چند جزئی ایجاد کنیم ، مثلاً استایل body رو بصورت زیر بنویسیم

```
body{
  background: url(all.png) , url(top.png) repeat-x top , url(bot.png) repeat-x bottom;
}
```

البته پیشنهاد من استفاده از روش قدیمی هست چون بک گراوند چیزی نیست که بخوایم نشون ندادنش توسط IE رو نادیده بگیریم!

روشنتر کردن عکس با opacity

خب شما توی خیلی از سایتهای دیدید که وقتی روی عکس میرید روشنتر میشه و وقتی موس رو از روش برمیدارید کمی تیره میشه.

به عکس زیر نگاه کنید



خب فکر کنم الان متوجه منظورم شدید.

اینکار با استفاده از ویژگی opacity قابل انجامه!

فرض کنید توی html چنین چیزی دارید

```
<div id="use-opacity">
  
</div>
```

و در مرورگر این عکس نشان داده میشود



حال میخوایم کاری کنیم که عکس در حالت عادی کمی کمرنگ باشد و وقتی موس روی آن قرار گرفت ، کامل پررنگ شود .

در استایل چنین چیزی مینویسیم

```
#use-opacity img{
  opacity : 0.5; /*درصد روشنتر شود50*/
}

#use-opacity img:hover{
  opacity: 1.0 ; /*کامل روشن شود*/
}
```

که در مرورگر در حالت عادی چنین چیزی نشان داده میشود



و وقتی موس روی عکس قرار گیرد بصورت زیر نشان داده میشود



ویژگی **opacity** یک مقدار اعشاری کوچکتر از 1.0 میگیرد.

یعنی 1.0 حالت عادی عکس و به نسبت هر چه کمتر، عکس روشنتر (یا سفید تر) میشود.

نکته قابل توجه اینکه: برای تاثیر دادن **opacity** در مرورگر IE باید از هک زیر استفاده کنید

```
-ms-filter: "progid:DXImageTransform.Microsoft. »
Alpha(Opacity=100)"; /* IE 8 hack */
filter: alpha(opacity = 100); /* IE 5-7 hack */
```

برای مثال برای تاثیر دادن کدهای بالا در IE باید استایل را بصورت زیر نوشت

```
#use-opacity img{
  opacity : 0.5; /*درصد روشنتر شود50*/
  -ms-filter: "progid:DXImageTransform.Microsoft. »
  Alpha(Opacity=50)"; /* IE 8 hack */
  filter: alpha(opacity = 50); /* IE 5-7 hack */
}
```

```
#use-opacity img:hover{
  opacity: 1.0 ; /*کامل روشن شود*/
  -ms-filter: "progid:DXImageTransform.Microsoft. »
  Alpha(Opacity=100)"; /* IE 8 hack */
  filter: alpha(opacity = 100); /* IE 5-7 hack */
}
```

RGBA

بذارید قبلش یه سخنرانی کنم!

همانطور که میدونید (یا شاید نمیدونید) رنگ ها رو به سه صورت میشه داخل استایل نوشت (من که فقط سه روش بلدم!)

یکی استفاده از کد رنگ هگزا برای مثال رنگ سفید #FFFFFF و رنگ سیاه #000000

یکی استفاده از نام آنها (البته تعداد محدودی هستن) برای مثال رنگ سفید white و رنگ سیاه black

و روش آخر استفاده از **rgb** هستش برای مثال سفید **rgb(255,255,255)** و سیاه **rgb(0,0,0)**

الان فقط در مورد روش آخر توضیح میدم که خودم وقتی مبتدی بودم ازش استفاده میکردم (البته چون دوست داشتم آآ ربطی به مبتدی بودن و حرفه ای بودن نداره!)

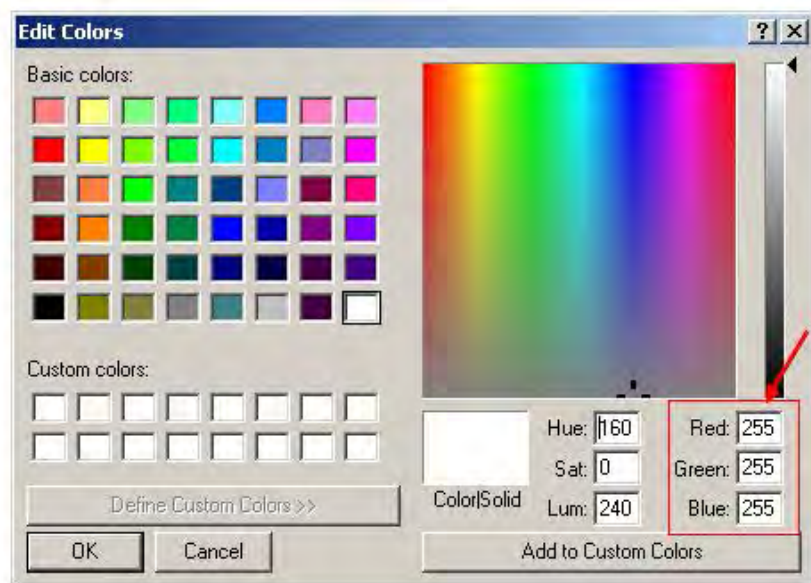
Rgb مخفف سه کلمه **red green blue** هست که همیشه با ترکیب این سه رنگ هر رنگی رو ساخت (کار خداست دیگه! همه رنگا سه رنگن فقط به کمی توی نحوه قروقاتی کردنشون فرق دارن ، درست مثل آدما)! بگذریم فلسفی شد.

خب برای اونایی که نمیدونن میگم اونایی هم که به پا طراح هستن به بزرگی خودشون ببخشن و نادیده بگیرن! برای ساخت رنگ مورد نظر ابتدا وارد برنامه paint شده و اون پائین روی یکی از رنگ ها دابل کلیک کنید.

به پنجره باز میشه **edit colors** اون پائین به دکمه هست که روش نوشته **define custom colors** روی اون کلیک کنید



حالا اون بغل پائین ، تکست باکس های **red green blue** هست!



روی هر رنگی برید ، و هر رنگی رو بسازید ، میتونید با استفاده از سه تا مقدار مقابل **red green blue** اون رنگ رو استفاده کنید . برای مثال برای تغییر رنگ پس زمینه کل صفحه به سیاه باید چنین چیزی توی استایل بنویسید.

```
body{
background: rgb(0,0,0);
}
```

بریم سراغ rgba :

در حقیقت rgba به ویژگی (property) نیست بلکه به نوع رنگ ترکیبی از rgb و opacity هست!

برای مثال رنگ سفید با 75 درصد شفافیت rgba(255,255,255,0.7)

ترجمه تحت الفظی opacity در زبان فارسی "تاری" است .

یعنی متنی که رنگ آن برابر rgba(255,255,255,0.7) باشد بصورت سفید دیده میشود که پشت آن پیداست! به چیزی مثل به

شیشه !

متوجه شدید؟

چند تا مثال میزنیم :

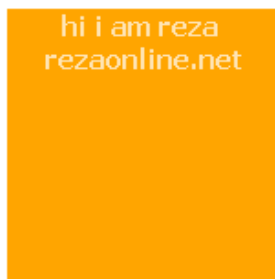
فرض کنید در فایل html چنین چیزی داریم

```
<div class="rgba-example">
  hi i am reza <br>
  rezaonline.net
</div>
```

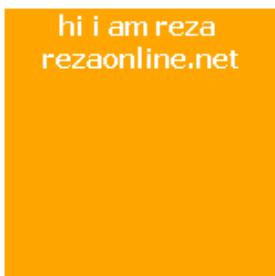
و استایل

```
div.rgba-example{
  background: orange;
  width:150px;
  height:150px;
  font-family: tahoma;
  font-weight: bold;
  font-size:11pt;
  color : rgba(255,255,255,0.7);
}
```

پس در مرورگر چنین چیزی داریم .



در صورتیکه بخواهیم به جای rgba(255,255,255,0.7) از rgb(255,255,255) استفاده کنیم چنین نمایشی داریم



باز من مثال بد زدم الان دچار اشتباه میشد!
 میبینید که در حالت اول (rgba) رنگ متن که سفید بود بصورت 75 درصد در آمده که یعنی 25 درصد از پشتش نمایانه که البته توی این مثال یه کمی اشتباه ایجاد میکنه!
 مثلاً یکی میتونه بگه ما اگه رنگ سفید رو که انتخاب کردیم به جاش از `rgb(255,228,179)` استفاده کنیم همین رنگ برای متن ساخته میشه و نیازی به تداخل پشتش نیست که مثلاً کمی نارنجی قاطیش بشه .
 در جواب هیچی ندارم بگم چون طرف راست میگه دیگه (البته توی این مثال!)

مثال بعد :

بیائیم پس زمینه (background) رو از حالت یکنواختی در بیاریم و به اون یه پترن بدیم!
 برای مثال من به پس زمینه پترن زیر رو میدم!



حالا کد استایل رو تغییر میدیم

```
div.rgba-example{
background: url(bg.png);
width:150px;
height:150px;
font-family: tahoma;
font-weight: bold;
font-size:11pt;
color : rgba(255,255,255,0.7);
}
```

در مرورگر چنین نمایشی داریم



میبینید که هر قسمت از نوشته متناسب با پس زمینه پشتش 25 درصد تداخل میده به سفید!
 که اینو به غیر از rgba با هیچ rgb نمیشه ساخت!

یه مثال دیگه میزنیم :
فرض کنید همان استایل بالا

```
div.rgba-example{
  background: url(bg.png);
  width:150px;
  height:150px;
  font-family: tahoma;
  font-weight: bold;
  font-size:11pt;
  color : rgba(255,255,255,0.7);
}
```

در html داریم :

```
<div class="rgba-example">

  <div id="child">
    man bache hastam
  </div>

</div>
```

یک division جدید میسازیم و بک گراوند رو بصورت rgba میذاریم

```
#child{
  background: rgba(255,255,255,0.7) ;
  width:100px;
  height:125px;
  margin:0 auto;
  color:RoyalBlue;
}
```

حالا در مرورگر چنین نمایشی داریم



کاملاً مشخص است که `div#child` رنگ سفید دارد و حالت شیشه ای که پشتش مشخص است!
ذکر یک نکته ضروری است : مرورگرها اگر نتوانند خطی از استایل را تشخیص بدهند یعنی برایشان تعریف نشده باشد آنرا نادیده میگیرند!

برای مثال اگر این مثال را در IE6 ببینید علنا خط `background: rgba(255,255,255,0.7)` پردازش نمیشود پس استایل شما بدون `background` خواهد بود پس برای حل این مشکل میتونید کد بالا را بصورت زیر بنویسید!

```
#child{
background: rgb(255,255,255) ;
background: rgba(255,255,255,0.7) ;
width:100px;
height:125px;
margin:0 auto;
color:RoyalBlue;
}
```

یعنی دو تا پس زمینه تعریف کنید! از نظر منطقی اشتباه است ولی کار آمد!
نتیجه این چنین است ابتدا پس زمینه سفید تعیین میشود ، در خط بعد اگر مرورگر از CSS3 و نهایتاً rgba پشتیبانی کند ، بک گراوند جدید پذیرفته میشود و قبلی نادیده گرفته میشود اما اگر از CSS3 پشتیبانی نکند مثل IE7 , IE6 و... ابتدا پس زمینه سفید تعیین میشود . خط بعد که مقدار ناصحیحی دارد (از نظر مرورگر) نادیده گرفته میشود پس در نهایت بک گراوند سفید تعیین میشود!
این قانون را در تمامی ویژگیهایی که استفاده میکنید رعایت کنید قبل از هر rgb ، rgba را بنویسید تا بعضی مرورگرهای ...

چرخش اجزا

شما میتوانید با استفاده از درجه هر چیزی را به چرخانید(در آخر این کتاب یاد میگیرم که چگونه با استفاده از این خاصیت انیمیشنی بسازیم)

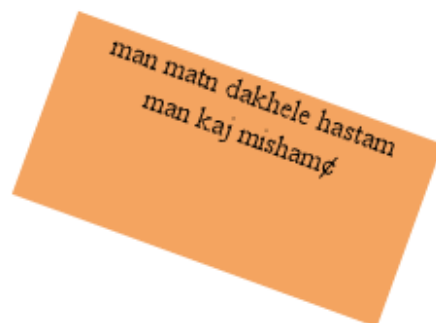
فرض کنید ما در html چنین کدی داریم

```
<div class="rezaonline">
man matn dakhele hastam<br />
man kaj misham&cent;<br />
RezaOnline.net
</div>
```

استایل را به صورت زیر وارد میکنیم

```
div.rezaonline{
width:200px;
height:100px;
background: SandyBrown;
transform: rotate(20deg);
-moz-transform: rotate(20deg);
-o-transform: rotate(20deg);
-webkit-transform: rotate(20deg);
}
```

در مرورگر چنین نمایشی داریم



شما میتوانید با درجه منفی در جهت پادساعتگرد بچرخانید کنید.

بزرگنمایی عکس

فرض کنید در html چنین داریم

```
<div id="reza">

</div>
```

براحتی با استفاده از ویژگی زیر میتوانیم عکس مورد نظر را بزرگ کنیم

```
#reza img{
-webkit-transform: scale(1.5);
-moz-transform: scale(1.5);
-o-transform: scale(1.5);
transform: scale(1.5);
}
```

من معمولاً برای حالت ها `img: hover` عکس موجود در مطلب رو کمی بزرگ میکنم به نظرم اینجوری خوبتره ولی خب سلیقه ایه!

کجی و انحراف

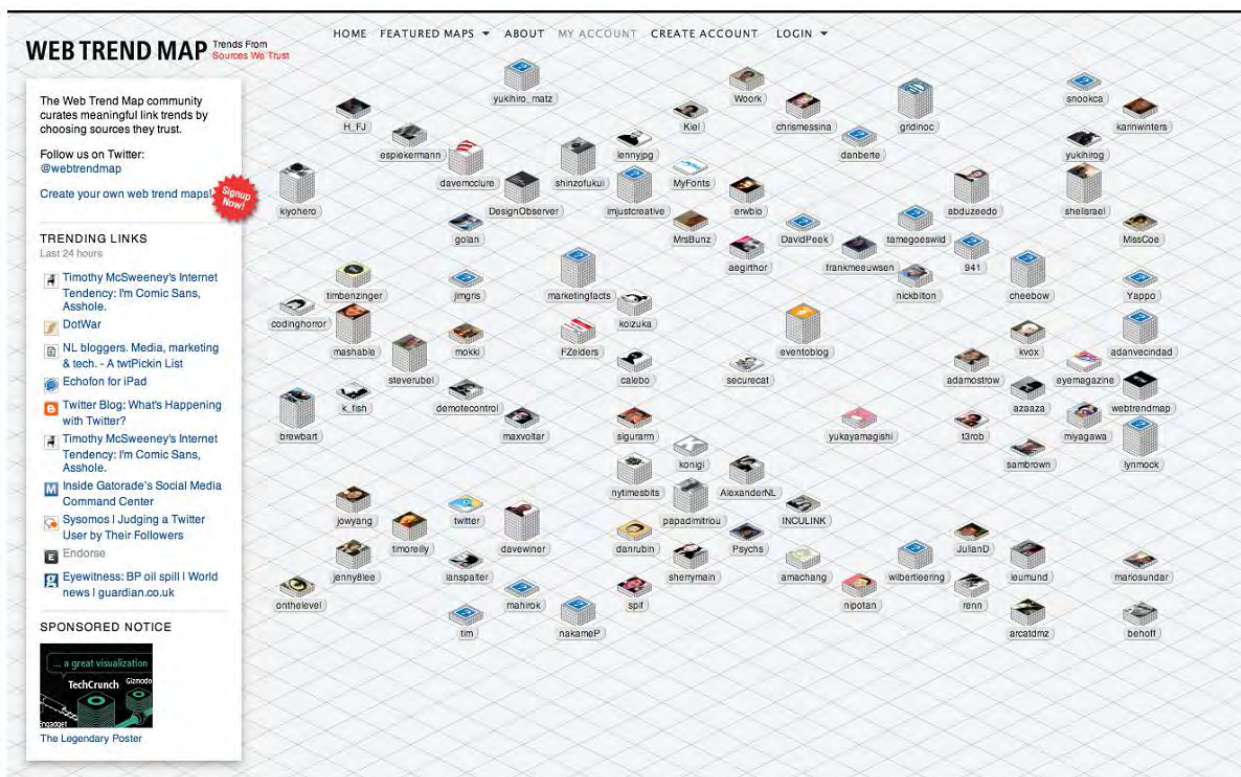
Skew برای کجی و انحراف بکار برده میشود برای مثال مثلاً مستطیل ، متوازی الاضلاع میشود!

این ویژگی با کمی خلاقیت شما میتواند اجزای سه بعدی بسازد

تصاویر زیر همگی skew شده اند

منبع عکس : [Css3 for web designers](#)





فرض کنید در html چنین داریم

```
<div id="reza-sh">

</div>
```

براحتی میتوان با استفاده از استایل زیر عکس مورد نظر را کج کرد.

```
#reza-sh img{
-webkit-transform: skew(30deg);
-moz-transform: skew(30deg);
-o-transform: skew(30deg);
transform: skew(30deg);
}
```

در مرورگر چنین نمایشی داریم



لازم به ذکر است که skew فقط مخصوص عکس نیست سایر اجزا هم مثل table , division و ... هم قابل skew شدن هستند!

میرسیم به یکی از بخشهای جذاب CSS3 .

همه ما کم و بیش حرکات انیمیشنی در وبلاگها و وبسایتهای دیدیم!

قبلنا از Flash استفاده میکردن که البته الان دیگه کمتر استفاده میشه و تقریباً با اومدن کتابخانه های جاوا مثل جی کوئری انیمیشن Flash که غالباً باید در چهارچوب خودش اجرا بشه و نیاز به پلاگین داشت ، منسوخ شده (و خوشحالم که حتی یک روز از عمرم رو برای آموختن Flash هدر ندادم).

انیمیشن در وب به نوعی تعریف دیگه ای داره ، یعنی مثلاً شما نمیتونی انتظار داشته باشی که وقتی یه سایت رو باز میکنی یکی بیاد برات برقصه !

در کل انیمیشن در وب ، (طبق تعریف من البته) به تحرک آنی یا مدت دار با افکت های مناسب گفته میشه .

مثلاً فرض کنید شما موس رو روی یه عکس میبرید و اون عکس میچرخه! این یه نوع انیمیشنه .

امروزه تصور اینکه همه وبسایتهای ، استاتیک باشند و بدون تحرک و خشک چیز مسخره ای شده و غالباً باید از تکنولوژی های روز استفاده کرد .

جاوا اسکریپت و بچه کوچیکش جی کوئری رو که همه میشناسیم . با استفاده از جی کوئری و تعریف افکت مناسب ، میتونیم به اجزای دلخواه موجود در html تحرک و انیمیشن بدیم (چیزی که فلش نمیتونست انجام بده!).

انیمیشن در CSS3 بسیار ساده و رون و کاملاً طبق خلاقیت شما کار میکنه .

به نظر من اگر فقط برای چند حرکت انیمیشن مجبورید حجم 70 (تا 160) کیلو بایت کتابخانه جی کوئری رو تحمل کنید ، دست نگه دارید!

CSS3 انتخاب ساده تر و بهتریه (البته بازم به نظر من!). نکته انجاست که شاید به CSS3 نقدی باشه که روی مرورگرهای قرن بوق درست کار نمیکنه ، در جواب باید بگم اگر کاربری هنوز دارد از مرورگرهایی نظیر IE6 استفاده میکند همان بهتر که صفحات را در هم ریخته و نا مرتب ببیند (خلایق هر چه لایق!).

و اما یه برتری هایی نسبت به جی کوئری داره ، یکی اینکه با سه چهار خط ساده انیمیشن تولید میشه و نیازی نیست اضافه بار کتابخانه جی کوئری رو روی سایت تحمل کنیم و یا ممکنه فردی جاوا رو توی مرورگرش غیرفعال کرده باشه !

پس به نظرم بدک نیست یه ذره روی این موضوع تامل کنید .

بریم سراغ اصل مطلب :

انیمیشن ها رو باید با توجه به نوع فعالیت کاربر ترتیب داد .

احتمالاً با `onmouseover` و `onmouseout` در جاوا آشنا باشید .

CSS3 هم به همین صورت انیمیشن ها را ایجاد میکند .

یک مثال ساده :

فرض کنید ما در html چنین چیزی داریم

```
<div id="test">
<a href="http://rezaonline.net">RezaOnline</a>
</div>
```

```
#test a:link{  
padding-right:0px;  
color:black;  
}
```

```
#test a:hover{  
padding-right:40px;  
color:red;  
}
```

خب به توضیح کوچولو: در حالت `a:link` حالت پیشفرض لینک که همون `RezaOnline` هست نشون داده میشه، فاصله از راست 0 و رنگ سیاه.

در حالت `a:hover` لحظه ای که نشانگر موس روی اون میره (کلیک نمیکنه! اگه کلیک کنه و نگه داره میشه `a:active`) خب در این حالت لینک 40 پیکسل از راست فاصله میگیره و رنگش قرمز میشه! البته این فاصله گرفتن بصورت آنی اتفاق می افته پس چیز زیبایی نیست. برای مبتدی ها: کل کد زیر رو کپی کنید و داخل یک فایل `txt` بریزید سپس فرمت فایل را به `html` تغییر بدید.

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

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title>salam</title>
```

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

```
#test a:link{  
padding-right:0px;  
color:black;  
}
```

```
#test a:hover{  
padding-right:40px;  
color:red;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<br /><center>
```

```
<div id="test">
```

```
<a href="http://rezaonline.net">RezaOnline</a>
```

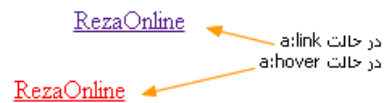
```
</div>
```

```
<br /></center>
```

```
</body>
```

```
</html>
```

در دو حالت `a:link` و `a:hover` در مرورگر چنین نمایشی خواهیم داشت .



همانطور که مشاهده میکنید در حالتی که موس روی لینک قرار میگیره 40 پیکسل به جلو حرکت میکنه (حالا حتماً میپرسید `a:link` باید رنگش سیاه باشه ولی توی عکس بنفشه باید بگم چون من به بار به آدرس `rezaonline.net` رفتم پس اون لینک برای من بعنوان `a:visited` شناخته میشه اما مطمئنم توی مرورگر شما سیاه نشون میده چون تا حالا به وبلاگ ما نیامدید مگر اینکه بعدها مشرف شوید انشالله!)

خب الان اونایی که به پا استادن میگن داداش ما خودمون میدونستیم! (جوابشون رو بعداً میدم) میبینید با حرکت موس کاربر و قرار گرفتن روی لینک به تغییر در استایل داریم پس دو حالت داریم علناً. حالا میخوایم بین این تغییر دو حالت یک انیمیشن بدیم یعنی لینک به جای اینکه آبی بره جلو آروم آروم بره جلو! استایل رو به شکل زیر تغییر بدید

```
#test a:link{
padding-right:0px;
color:black;
-moz-transition-property: all; /*Mozilla*/
-moz-transition-duration: 0.3s; /*Mozilla*/
-moz-transition-timing-function: ease; /*Mozilla*/

-webkit-transition-property: all; /*Chrome safar*/
-webkit-transition-duration: 0.3s; /*Chrome safar*/
-webkit-transition-timing-function: ease; /*Chrome safar*/
}

# test a:hover{
padding-right:40px;
color:red;
-moz-transition-property: all; /*Mozilla*/
-moz-transition-duration: 0.3s; /*Mozilla*/
-moz-transition-timing-function: ease; /*Mozilla*/

-webkit-transition-property: all; /*Chrome safar*/
-webkit-transition-duration: 0.3s; /*Chrome safar*/
-webkit-transition-timing-function: ease; /*Chrome safar*/
}
```

Transition یک ویژگی `Css3` که تغییرات را در طول زمان کش میدهد (خودمم نفهمیدم چی گفتم!) یعنی چی یعنی مثلاً این تغییر `padding-right` که 40 پیکسل لینک رو جلو میاره در عرض 3 صدم ثانیه اتفاق می افته که یعنی به حرکت انیمیشنی رفت و برگشت. (چون `a:link` هم این ویژگی رو داره پس برگشت هم با انیمیشن هست)

خب بیایم کمی در مورد این ویژگی صحبت کنیم

اول از همه اینکه میتونین کل سه خط رو توی یک خط بنویسین!

transition: all 0.3s ease 2s

all: مربوط میشه به ویژگی که در طول تغییرات انیمیشن داده میشه! الان ما گذاشتیم **all** بطور مثال اگر بذاریم **color** میبینیم که فقط تغییر رنگ با انیمیشن انجام میشه و **padding** بصورت آنی اتفاق می افته!

0.3s: مدت زمان اجرای انیمیشن بطور مثال الان 3 میلی ثانیه طول میکشه تا انیمیشن شروع و تمام بشه هر چه زمان بیشتر حرکت آهسته تر!

Ease: تابع و نوع انجام انیمیشن مثلاً **linear** رو میتونید استفاده کنید یا **ease-in-out**

2s: مدت زمان تاخیری قبل از شروع انیمیشن (من در انیمیشن بالا تاثیر ندادم تا بلافاصله با رفتن لینک انیمیشن انجام بشه).

دیدید توی فیلم کارگردانا میگن 3 - 2 - 1 اکشن؟! این همون 1 2 3 اولیه است

یعنی الان باید شما روی لینک دو ثانیه بایستید بعد انیمیشن شروع میشه!

(جون میده واسه ساخت یه اسلاید شو سی اس اسی!)

خب بریم یه مثال دیگه انجام بدیم چون این قسمت نیاز به کار داره :

مثال بعدی میخوام بهتون نشون بدم چطوری وقتی موس رو میبریم روی یه عکس ، اون 3 بار دور خودش بچرخه و وقتی موس رو بر میداریم از روش برگرده سر جاش (3 دور برعکس بچرخه)

فرض کنید توی html یه همچین چیزی دارید یک div که شامل یک یا چند عکس باشه

```
<div id="test2">
  
</div>
```

استایل رو به شکل زیر قرار میدیم

```
#test2 img{
-moz-transition: 2s; /*Mozilla*/
-webkit-transition:2s ; /*Chrome and safari*/
}
```

```
#test2 img:hover{
-moz-transform: rotate(1080deg); /*Mozilla*/
-moz-transition: all 2s ease-in-out; /*Mozilla*/

-webkit-transform: rotate(1080deg); /*Chrome and safari*/
-webkit-transition: all 2s ease-in-out; /*Chrome and safari*/
}
```

برای مبتدی ها : کدهای زیر را در یک فایل **txt** بریزید سپس فرمت آنرا به **htm** تغییر بدهید و با مرورگر باز کنید

لازم است یک عکس به نام **pic.png** در کنار فایل بگذارید تا لود بشه یا آدرس عکسی وارد کنید .

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>rezaonline.net</title>
<style type="text/css">

#test2 img{
-moz-transition: 2s;
-webkit-transition:2s ;
}

#test2 img:hover{
-moz-transform: rotate(1080deg);
-moz-transition: all 2s ease-in-out;
-webkit-transform: rotate(1080deg);
-webkit-transition: all 2s ease-in-out;
}

</style>
</head>

<body>
<br /><center>
<div id="test2">  </div> <br /></center>
</body>
</html>

```

خب بیایید کمی در موردش بحث کنیم در حالت `img:hover` که موس روی عکس قرار میگیرد دو ویژگی اتفاق می افتد

```

transform: rotate(1080deg);
transition: all 2s ease-in-out;

```

در ویژگی `transform` مقدار `rotate(1080deg)` عکس را 1080 درجه میچرخاند که به عبارتی هر چرخش 360 درجه محسوب میشود پس در کل سه چرخش کامل داریم!
در ویژگی `transition` برای تغییر انیمیشن تعیین میشود.

All بخش قابل تغییر تماماً انتخاب شده البته چون ما فقط یک ویژگی داریم پس همان یک ویژگی اثر دارد بطور مثال اگر ما چند ویژگی برای عکس در نظر گرفته باشیم مثلاً همزمان از `opacity` هم استفاده کنیم با انتخاب `all` تغییرات `opacity` هم انیمیشنی میشود که در این حین اگر بخواهیم فقط چرخش داشته باشیم باید به جای `all` ویژگی چرخش را درج کنیم یعنی `-webkit-transition: -webkit-transform 2s ease-in-out;` (لازم به تذکر است که `moz` , `o` , `webkit` و ... را حتماً درج کنید!)

2s : مدت زمان اجرای انیمیشن

ease-in-out : تابع نحوه اجرای انیمیشن ، خوب کاربرد این تابع به این صورته وقتی که انیمیشن تمام شد و موس را برداشتیم بصورت خودکار انیمیشن برعکس کار میکند یعنی برمیگردد به حالت اول پس ما فقط در

```

#test2 img{
-moz-transition: 2s;
-webkit-transition:2s ;
}

```

مدت زمان حرکت برگشت را ذکر کردیم لازم به ذکر بقیه موارد نیست .

خب ما در اینجا موارد ساده ای را برای انیمیشن های ساده ذکر کردیم تمامی موارد در مرورگر کروم 13 تست شد و به خوبی کار کرد البته انتظار می رود با سایر مرورگرها نظیر mozilla4+ و سافاری نیز به خوبی کار کند.

پیشنهاد میکنم از scale و skew همچنین opacity نیز استفاده کنید و ویژگی transition را برای آنها نیز بررسی کنید . ویژگی های دیگر Css3 نظیر animation که به بررسی دقیقتر انیمیشن میپردازد انشاالله در بخش دوم این کتاب ارائه میشود چون نیاز به زمان و دقت کافی دارد همچنین بعضی مرورگرها بطور کامل از ویژگی های جدید پشتیبانی نمیکند!

سخن پایانی

این کتاب توسط خود بنده تهیه شده و بدون استفاده از هیچ منبع فارسی تهیه شده پس ممکن است در تفهیم و یا تعریف ، دچار خطاهایی باشد . بطور مثال من برای property ، از "ویژگی" استفاده کردم که ممکن است در کتابهای دیگر به نحوی دیگر ترجمه شده باشد. (حقیقتاً من تا حالا کتاب CSS فارسی نخوندم!)

دوستان طراح و مسلط به CSS اگر در جایی اشتباهی رخ داد و یا کدی اشتباه درج شده ممنون میشم بهم اطلاع بدن .

info@rezaonline.net

همچنین در قالب فعلی وبلاگ بنده rezaonline.net/blog در چندین مورد از CSS3 استفاده شده بطور مثال برای لینک نظرات از opacity برای منو از box-shadow و border-radius برای عنوان از text-shadow برای فرم نظرات از transition و box-shadow برای جستجو از transition برای لینک منو موضوعات از transition و در عکسهای مطالب از opacity و scale و border-radius و shadow و transition استفاده شده که به نظرم مثال خوبی برای این چند ویژگی CSS3 هست .



لازمه متذکر بشم :

فقط به دلیل پشتیبانی نکردن مرورگرهای IE6 و... از CSS3 ، از استفاده کردن از CSS3 منصرف نشوید چون به عقیده من وقتی کاربری نمیداند فرق بین IE6 و Chrome چی هست ، پس قطعاً درک درستی از CSS3 و زیبایی وب ندارد و در نهایت شعور استفاده از اینترنت را ندارد ، پس نیازی نیست بخاطر کاربران درجه سه اینترنت ، از استفاده از CSS3 خودداری کنیم!

امید است در آینده ای نزدیک شاهد وبسایتهایی باشیم که از تکنولوژی های **Html5** و **Css3** , **Jquery** , **Ajax** بطور همزمان استفاده کنند .

همچنین امیدوارم **Microsoft** تکونی به خودش بدهد و به **IE** خوب بسازد تا پدر طراحان وب را در نیاورد .

/موفق باشید

رضا شیخله

20 آبان 1390

RezaOnline.Net
RezaOnline.Net/blog
info@rezaonline.net

و آنجا ، در بزرگراه زندگی ، آن چراغ چشمک زن که پایان عمر را هشدار می دهد ، آنجا که جای دور زدن نیست ...
من ... ، سردرگم و خاموش ...
ترس از گذر ، آزارم می دهد...



در کانال تلگرام کارنیل هر روز انگیزه خود را شارژ کنید 😊

<https://telegram.me/karnil>

