توسعه برنامه های موبایل

جلسه ششم مجازی

بخش دوم

سحر صادقی

مروری بر Sprites های Css

من مطمئن هستم که کسی را با این که بخوام بگم که Css sprite ها از سال ۲۰۰۳ چندیست که دور وبر ما هستند و حضور بحث برانگیزی هم دارند، متعجب نخواهم کرد.البته هنوز هم CSS sprite ها همینطور که باید راه خودشون رو بین ابزار های توسعه دهندگان وب باز نکرده اند.در حالیکه تئوری ای که در این ها می باشد قابل فهم و همینطور مفید هم هستند،

اجرا و پیاده سازی آن ها زمانی که شما ددلاین کمی داشته باشید یه کم آزار دهنده است. ما برای شما راه رو اسون می کنیم که این چیز ها سد ما نشن که شما بتونین از عالی بودن Css sprite ها به راحتی استفاده کنید.من خیلی نمی خوام تو حالتی بمونم که همش اعتبار CSS sprite ها رو بخوام بگم در حقیقت هدف این مقاله اینه که بفهمیم چرا هنوز مردم استفاده کردن از CSS sprite ها رو سخت می دونند همینطور هم یه سری قانون ها و راه های قابل توجهی رو برای بهتر کردن تکنیک های حال یاد می گیریم. ما تمام مهارت ها از جمله photo shop و Sass و LESS رو به دست میگیریم مخلوط می کنیم با هم تا یه چیز خوب ازشون در بیاریم

مشکل استفاده از CSS Sprites

زمانیکه photoshop داره کارشو میکنه یه دیقه صب کنید و در نظر داشته باشید که چرا CSS sprites ها براشون مشکله که به صورت همگانی مورد پذیرش قرار بگیرن.من همیشه با این مسئله که نقطه آغاز کارم برای هر عکس در Sprite ها باشه کجاست مشکل داشتم. من همیشه با این که خب مختصات ها رو حفظ کنم تا بخوام در صفحه ی استایل ها بنویسم مشکل داشتم. به خصوص وقتی عدد ها در هم و بر همه.

من همیشه هی میرم بین صفحه ها دوباره چک میکنم تا اینکه مطمئن بشم عددی که دارم می نویسم خب درسته! و در هر نقل مکانی که در بین صفحه ها دارم بیشتر اذیت می شم. در اصل می دونم نرم افزار هایی هست که خب باعث میشه به من تو این کار کمک کنه ولی خب من به نرم افزاری احتیاج دارم که بتونم همون طوری مختصات ها رو ازش کپی کنم ودر صفحه ی استایل خودم وارد کنم.

یه مسئله ی مهم دیگه اینه که CSS sprites ها به عنوان یک قسمتی از بهینه سازی به شمار می آیند. بسیاری از مردمی که که وبلاگ هایی که در ارتباط با CSS sprite ها می نویسند به این مسئله اعتماد دارند که باعث بهینه سازی می شود. ولی تمرین هایی که می دهند بیشتر از این مسئله که چقدر باید تلاش بشه تا به اون چیزی که می خواهند برسند، دور میشن.

این مسئله اونقدرا اذیت کننده نیست اگر شما نیاز باشه که به مشکل خودتون به عنوان یک مسئله ی بهینه سازی نگاه کنید. ولی وقتی شما یک ددلاین مشخصی داشته باشید و باید سایت رو از ابتدا بسازید این یک مشکل طاقت فرسا میشه. اگر شما وقت کافی داشته باشین که بتونین روی یک CSS sprite ها کار کنید کار سختی نیستند. ولی وقتی نیاز باشه که در آن واحد به ۵۰ تا اولویت دیگه هم برسین خیلی کاری سختی برای خیلی از توسعه دهندگان میشه. با در نظر داشتن این ۲ مورد ما شروع می کنیم که هدف قرار دادن تصاویر رو راحت تر کنیم. بعضی وقتا هم باید کلن در کل بیخیال بهینه سازی بشیم که توسعه دادن ما راحت بشه.

آماده سازیSprite

اگر شما به صورت آنلاین به دنبال این CSS Sprite **ها بگیردین میبینین که بیشترشون برای استفاده در مشاور املاک استفاده شدن. فاصله ی بین تصاویر در کمترین میزان است که لود شدن کل این ها در آهسته ترین زمان ممکن باشه. این مسئله زمان لود شدن رو به کمترین زمان ممکن میرسونه ولی خب اون مختصات های وحشتناک رو هم نشون میده با اینکه از همون زمان اول شروع به دانلود شدن میکنن عکس ها. خب چرا ما مسیری که یه کم متفاوت تره رو انتخاب نکنیم؟ بیاین یه مسیر راحت تر انتخاب کنیم برای هدف قرار دادن تصاویر ، و اونم اینه که مطمئن باشیم که لود شدن** Strike **ها از لودشدن مجموع تصویر ما زمان بیشتری نگیرن.**

بیشتر از اینکه Strike **ها رو به کمترین اندازه ممکن برسونیم بیاین از یک طرح دیگه استفاده کنیم. به جای اینکه از شبکه ای تصادفی انتخاب کنیم ، یک شبکه ی تمیز مربعی شکل برای هر کدوم می سازیم.حالا هر کدومشون رو با یه ترفند ساده در سمت چپ بالا آدرس دهی می کنیم که خب خیلی هم راحت تره.**

اندازه ی شبکه ی مربعی بستگی به میانگین ابعاد عکس ها که سعی در گذاشتن داریم دارد. برای وب سایت من مثلن من ۳۲ در ۳۲ پیکسل برای شبکم استفاده کردم ولی سایز شبکه ممکنه بر اساس موقعیت ای که دارید متنوع باشه. بعد چیز دیگه ای که باید در نظر داشته باشین اینه که یه کم padding **بدین بهشون که وقتی دارین زوم می کنین مشکلی براتون ایجاد نکنه. برای بهینه سازی بیشتر میتونین از شبکه ی مستطیلی به جای مربعی استفاده کنید که خب مکان های بیهوده ی کمتری تولید می کنید. فعلن در این مقاله ما شبکه ی مربعی رو پیش میگیریم**

آماده سازی : Sprite با بهره گیری از Photoshop

من طراح نیستم برای همین نمیدونم این قسمت چقدر داره به درستی از Photoshop **استفاده می کنه. ولی هنوز یه سری نکته های با ارزش هست که قبل به سمت کد رفتن بهشون باید توجه بشه.**

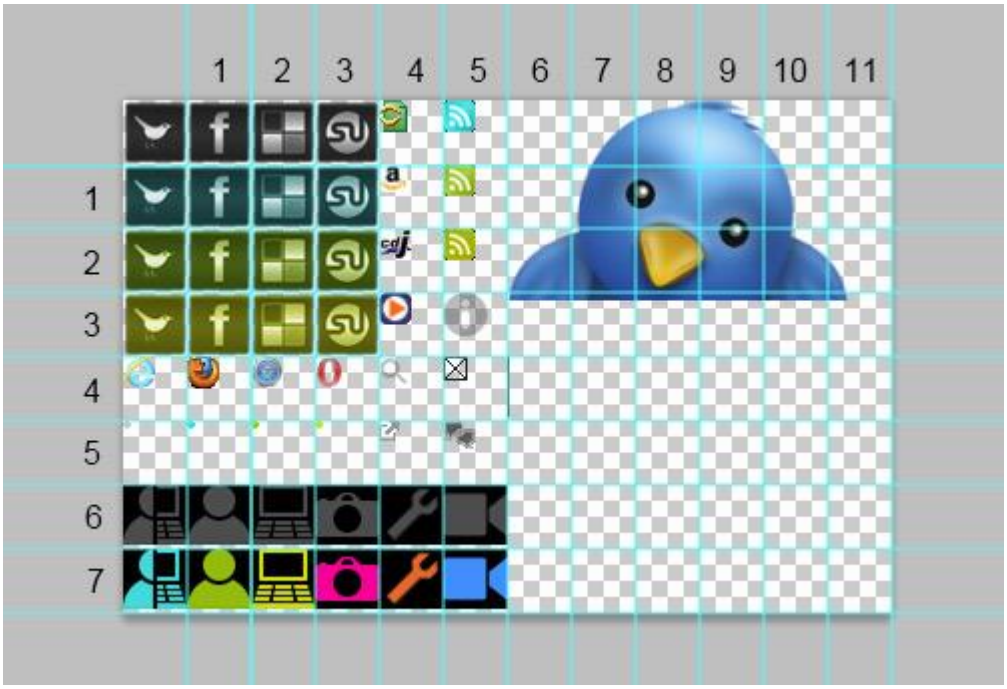
اول از همه به صورت بصری دیدن شبکه خیلی میتونه کمک کنه Photoshop **.راهنمایی هایی در این زمینه داره.البته در اضافه به بهره های دیگه ای که میتونین از** Photoshop **ببرین.**

اینکه بخواین دستی اینا رو اضافه کنین و درست کنین ممکنه براتون یه کم وقت ببره برای همینم دوست خوب من براتون یه سری اسکریپت های Photo shop **آماده کرده که شما نیاز دارین اونارو دانلود کنید و با رفتن به این قسمت بهتون اجازه استفاده رو میده. البته ابتدا اسکریپت رادانلود کنید و در** PresentsScripts **اون رو ذخیره کنید.**

File → Scripts → Sprite Grid **از قسمت منو** Photoshop

در آخر برای اینکه تموم کنید میتونین x **و** y **ای که میخواین رو بهش بدین و به راحتی میتونین به هر نقطه از شبکتون اشاره کنید.**

این عدد ها رو به Sprite **های واقعی اضافه نکنین فقط** Sprite **رو در تصویر جداگونه کپی کنید و اونارو اضافه کنین. این نامه ها رد اصل فقط برای مرجع اند**(:.



اگر شما نمیتونین از زبان های پیش پردازنده ای مثل Sass و LESS استفاده کنید هر کدوم از مربع ها رو جداگونه مقدار بدین.

LESS و Sass برای نجات

با داشتن همه ی تصاویر در یک شبکه ی مربعی ما میتونیم به راحتی قسمت بالا سمت چپ رو اندازه گیری کنیم برای هر عکس . (این مختصات ها درست همون مقدار های background-position برای دوباره موقعیت دادن به Sprite ها هستند) فقط اندازه ی پایه رو بردارین و در تعدادی که در عکس مرحعتون اضافه کردین ضرب کنین . مثلا شما یک عکس در مکان (۵,۳) میخواهید و خب مربع های شما اندازه ی ۳۲ دارند. در اصل میشه (۱۶۰,۹۶). برای قرار دادن این در صفحه ی استایل فقط مقادیر منفی آن ها را قرار دهید:

```
@spriteGrid: 32px; .sprite(@x, @y) { background: url(img/sprite.png) no-repeat; background-position: -(@x*@spriteGrid) -(@y*@spriteGrid); }
```

خیلی چیز جالبی تا اینجا نبوده ما اول از یک متغیر LESS استفاده کردیم و بهش مقدار اندازه شبکه رو دادیم.بعد از اون هم ادامه دادیم کاری که میخواستیم بکنیم رو و مختصات ها رو با محاسباتی که انجام میده به دست میاریم.تاموقعیت پایه ی عکسی که میخواهیم هدف قرار بدیم رو باهاش در بیاریم. خیلی شیک به نظر شاید نیاد ولی استفاده از پایه ی Sprite ها رو همینطوریش آسون کرد.

از الان به بعد فقط میتونید از (sprite1,5) استفاده کنین وهیچ محاسباتی هم نباید انجام بدین.

درسته که کدی که ما بالا نوشتیم فقط در کد ساده ی ما کار میکنه . بعضی از وب سایت ها از چیز های پیچیده تری استفاده میکنن. و همینطور با توجه به نیازشون شاید به چیز های سخت تری نیاز داشته باشن. البته این هم با LESS کار مشکلی نیست

```
.spriteHelper(@image, @x, @y, @spriteX, @spriteY) { background: url("img/@{image}.png") no-repeat; background-position: -(@x*@spriteX) -(@y*@spriteY); } .sprite(@image, @x, @y) when (@image = sprite1), (@image = sprite3){ @spriteX: 32px; @spriteY: 16px; .spriteHelper(@image, @x, @y, @spriteX, @spriteY); } .sprite(@image, @x, @y) when (@image = sprite2){ @spriteX: 64px; @spriteY: 32px; .spriteHelper(@image, @x, @y, @spriteX, @spriteY); }
```

بله قبول دارم یه کم گیج کننده شاید به نظر برسه ولی خب هنوزم خیلی پایه ای هست اگر ۱ دقیقه برای فهم اون وقت بزارین.

مهم تر از اون با نوشتن این قطعه LESS **اون سختی هم محو شده همونطور که میبینین داره از همون ساختار قبلی استفاده می کنه و در مجموع تفاوت چندانی نکرده ما فقط پارامتز عکس رو بهش اضافه کردیم تا بتونیم به راحتی تصمیم بگیریم که با هر صدا کردن از کدوم** sprite **استفاده کنیم.**

Sprites **های مختلف ابعاد شبکه ای مختلفی هم دارند و خب برای هر شبکه ای ما نیاز داریم که کد های** LESS **مختلفی هم بنویسیم.ما هر** Sprite **را به بعد متناسب با خودش متصل می کنیم .و ما** (@spriteX and @spriteY) **را به صورت محلی تعریف می کنیم. بعد از اینکه این مسئله حل شد حالا وقتشه که با** spriteHelper. **محاسبات ما انجام بگیره و به ما همون نتایج قبل رو هم میده.**

"guards"**در** LESS **یک کلمه جدید به حساب میاد پس مطمئن بشین که دارین آخرین ورژن** LESS **رو استفاده می کنید . بهتره خودتون یه کم باهاش کار کنین تا دستتون بیاد وگرنه که اون** Sprite **اس می خواهید رو انتخاب کنید و بزارید که** LASS **ادامه ی کار رو خودش انجام بده.**

CSS Spriteهای متداول از پرونده ها استفاده می کنند

با این کد هایی که در دست داریم ببینیم که چه CSS Sprite هایی از پرونده هایی که ما میتونیم تشخیص بدیم و بهتر بفهمیم که آیا میشه ازین پرونده ها در LESS **ها استفاده کرد یا نه. یه بار دیگه برای اینکه سختی و پیچیدگی هایی که ممکنه بهشون بر بخوریم رو کم کنیم در نظر می گیریم که داریم از شبکه های مربعی استفاده می کنیم. اگر بخواین از ساختار مستطیلی هم استفاده کنین فقط کافیه پارامتر های اونو بدین و بقیش درست میشه.**

برای هر استفاده از پرونده ها (case) **ما ۲ بخش رو مشخص می کنیم. یکی با طول و عرض و دیگری بدون پارامتر های طول و عرض. اگر شما مثل من هستید و طول و عرض رو با هم قاطی میکنین بهتره که از قسمت دوم استفاده کنین. مسولیت اینکه ترتیب درست پارامتر های کدومه رو به عهده خودتون میزاره** LESS.**به شما اجازه میده که این بخش ها رو در یک سند معرفی کنین.**

.1متن جایگزین

این احتمالا راحت ترین پرونده ایه که شما میتونین استفاده کنین که زمانی که ما html **ای داریم در دسترسمون که میتونیم بهش بعد ثابتی بدیم.ما همینطور درون المان** html **نوشته را پنهان می کنیم و اونرو با یک تصویر از** sprite **خودمون تغییر می دیم.**

مثال های معمول این نوع استفاده ها لینک های واکنشی هستند و یا همینطور. headings

```
.hideText{ text-indent: -999em; letter-spacing: -999em; overflow: hidden; }
.spriteReplace(@x, @y) { .sprite(@x, @y); .hideText; } .spriteReplace (@x, @y,
@width, @height) { .sprite(@x, @y); .hideText; width: @width; height: @height; }
```

بخش spriteReplace **، بخش شکل دهنده ی** Sprite **رو پوشش می دهد و همینطور یک بخش کمک کننده ی کوچیک برای پوشاندن نوشته از تصویر اضافه می کند. خیلی پایه ایه ولی خب اینکه هر دفعه ما بخوایم از این پرونده استفاده کنیم یه بخش** hideText. **بخواد اضافه کنه بهتره!**

در بالا ما یه لیستی از گزینه ها اشتراک گذاری داریم برای هر دلیلی بزار بگیم ما میخواستیم به جای html **از** Css background **استفاده کنیم . هیچ ربطی هم به بخشی که قبلا ساخته بودیم نداره . این کدشه:**

- Share [article's title] on Twitter
- ...

```
.sharing .twitter a { .spriteReplace(0, 0, 32px, 32px); display:block; }
```

2.تصاویر درون خطی

برای مورد استفاده ی دوم ما تصاویر درون خطی رو بررسی می کنیم. مشکل اصلی ای که اینجا بهش برخورد می کنیم اینه که ما نمی توانیم ابعاد ثابت براشون در html **اختیار کنیم. چرا که خب ما با محتوایی سرو کار داریم که اندازش متغیره. استفاده ی معمول اون ها هم استفاده از متن ها در کنار آیکون ها می باشد ولی این متد میتونه برای هر آیتمی که نیاز داشته باشه که عکس ها دور نوشته رو پوشش بده استفاده بشه**

```
.spriteInline(@x, @y) { .sprite(@x, @y); display: inline-block; content: ""; }
.spriteInline(@x, @y, @width, @height) { .sprite(@x, @y); display: inline-block;
content: ""; width: @width; height: @height; }
```

ما ممکنه فاقد المانی باشیم که تصویر ما رو بصری کنه ولی ۲۰۱۱ به ما یاد داده بود که شبه عنصر ها بهترین راه برای غلبه بر این مشکل است. به همین دلیله که بخش spriteInline **خصوصا توسعه یافت تا با تگ** a:before **و** :after **استفاده شود.**

ما تعریف inline-block **رو استفاده می کنیم در شبه عناصرمون تا اون ها مثل المان های درون خطی عمل کنند در حالیکه بهمون اجازه میدن تا از بعد های ثابت درش استفاده بشه ، و ما خصوصیات محتوا رو بهش اضافه می کنیم. که فقط برای دیده شدن شبه عنصر ها استفاده میشه.**

Buy on amazon.com
Buy on play.com
Buy on yesasia.com
Buy on cdjapan.co.jp

مثال بالا لیستی از لینک های وابسته رو نشون میده. طراحی نشون میده که هر لینک وابسته به یه خط است پس بنابراین ما به راحتی میتونیم از بخش spriteInline. **برای نمایش آیکون جلوی هر لینک استفاده می کنیم.**

- Buy on Amazon.com
- ...

```
.affiliates .amazon a:before { .spriteInline(4, 1, 22px, 16px); }
```

3.تصاویر خالی

سومین و آخرین کاربرد این ها وقتیه که متن ها اجازه ندارن که تصاویر stripe **ها رو دورشون رو پوشش بدن. مثال های متداول هم میتونه لیستی از آیتم ها باشه که چندین خط رو پوشش میده و همینطور همه نوع نمایشی از آیکون های تنها رو نشون بده. هر زمان که شما بخواین یه فضایی رو در المان های چند خطه رزرو کنیم تا مطمئن بشیم به صورت دقیق نوشته های ما کنار عکس ها هستند. این متدیه که شما ازش استفاده خواهید کرد.**

```
.spritePadded(@x, @y) { .sprite(@x, @y); position: absolute; content: ""; }
.spritePadded(@x, @y, @width, @height) { .sprite(@x, @y); position: absolute;
content: ""; width: @width; height: @height; }
```

دوباره ما شانس خودمون رو امتحان می کنیم با شبه عناصر!

این دفعه ما یه سری حیله هایی با موقعیت ها می زنیم.با استفاده از position:absolute **برای شبه عنصرمون. میتونیم درست در فضایی که رزرو کردیم بزاریمش با استفاده از المان پدر! موقعیت درست عکس در بخش** spritePadded **اضافه نشده و و باید در** selector **هایی که استفاده می کنیم برای دقت و تمیزی مشخص کنیم. چون ما داریم از موقعیت** absolute **استفاده می کنیم چه** before: **چه** after: **برای شبه عنصر ما کارش رو انجام خواهد داد. در ذهن داشته باشین که** after: **برای متد های تمیز** CSS **کاندیدای خوبیه. پس برای اینکه از تضاد های آینده صرف نظر کنید چون** after: **برای موقعیت** absolute **کار نمیکنه از** before: **استفاده کنین.**

Translation available in the following language(s): 日本語

بیاین فرض کنیم که ما میخواین نشون بدیم که مقاله ی ما به زبون دیگه هم ترجمه شده ما یک نوتیفیکیشن کوچیکی براش در نظر می گیریم و خب در اون جعبه لیست زبان های دیگه ای که برای مقاله قابل دسترس هست رو توش میاریم . اگر متن ما به ۲ خط شکسته شد ما نمیخواهیم که زیر آیکون ها ریز ما این ایجاد بشه برای همین از بخش spritePadded **استفاده می کنیم.**

Translation available...

```
.translated p { padding-left: 22px; position: relative; } .translated p:before {
.spritePadded(5, 5, 16px, 14px); left: 0; top: 0; }
```

یک Sprite ایده آل

حالا بگین آیا ما با شما به یه چیز جادویی از CSS sprite **رسیدیم؟؟ ۱۰۰٪ که نه!**

اگر شما توجه دقیقی کرده باشین متوجه میشین که که مورد هایی که ما بالا بررسی کردیم هیچ راه حلی پیشنهاد نمیدن برای اضافه کردن پی در پی تصویر پیش زمینه به Sprite . **خب یه سری راه ها هم برای از بین بردن این مشکل وجود داره که هیچکدوم نه سخت هستند و نه حتی ارزش گفتن دارن!**

چیزی که CSS sprite **احتیاج داره اینه که برای بریدن عکس ها راه حلی داشته باشه . این خصوصیت باید جوری تعریف بشه که بشه عکس رو تغییر سایز داد قبل از اینکه بخوایم در مراحل بعدی ازش استفاده کنیم.**

این راه خوبیه که بتونیم عکس رو ببریم و درست حسابی هم بهش موقعیت بدیم و همینطور تا جایی که نیاز داریم این کار روانجام بدیم. بعدش با after: **و** before: **هم بهشون جا میدیم چون دیگه چیزی برای پنهان کردن هم نیست.**