

احتمالا با عباراتی مانند طراحی table less و مزیت‌های طراحی با CSS، همانند سرعت بالاتر بارگذاری سایت در مقایسه با نمایش یک جدول که نیازمند دریافت تمام جزئیات آن و سپس رندر نهایی اطلاعات آن توسط مرورگر است، بارها برخورد داشته‌اید. اما ... آیا یکبار سعی کرده‌اید که به صورت دستی همان کارهایی را که پیشتر با HTML table انجام می‌دادید، اینبار توسط CSS پیاده سازی کنید؟

در اکثر اوقات نتیجه کار مایوس کننده، بسیار سخت و نگهداری آن در طول زمان بسیار مشکل خواهد بود؛ به علاوه سازگاری با مرورگرهای مختلف و نکات ریز هر کدام را نیز لحاظ کنید. به همین جهت تعدادی فریم ورک CSS برای شبیه سازی گرید و جدول تهیه شده‌اند که کار طراحی table less را بسیار ساده و لذت بخش کرده‌اند. یکی از این موارد، فریم ورک [Blueprint CSS](http://www.dotnettips.info) نام دارد و در ادامه نحوه استفاده از آن را مرور خواهیم کرد. این مرور هم مستقل است از فناوری سمت سرور مورد استفاده و صرفا مباحث HTML و CSS آن بررسی خواهند شد.

دریافت Blueprint CSS

این فریم ورک سورس باز را از مخزن کدهای آن در GitHub می‌توانید دریافت کنید: ([^](#)) البته نگران حجم نزدیک به 4 مگابایتی بسته دریافتی آن نباشید؛ زیرا نهایتا با سه فایل CSS از آن بیشتر کاری نداریم و مابقی مثال‌های آن هستند.

پس از دریافت آن، یک پوشه را به نام blueprint ایجاد کرده و سه فایل ie.css, print.css و screen.css را در آن قرار دهید. به علاوه داخل این پوشه، یک پوشه جدید دیگر را به نام src ایجاد کرده و فایل grid.png موجود در این بسته را نیز در آن کپی کنید.

ساختار ابتدایی یک صفحه مبتنی بر Blueprint CSS

پس از ایجاد پوشه blueprint و src به نحوی که توضیح داده شد، ابتدایی‌ترین ساختار یک صفحه تشکیل شده با blueprint css به نحو زیر است:

```
<!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>Blueprint test page</title>

<!-- Framework CSS -->
<link rel="stylesheet" href="blueprint/screen.css" type="text/css" media="screen, projection">
<link rel="stylesheet" href="blueprint/print.css" type="text/css" media="print">
<!--[if lt IE 8]><link rel="stylesheet" href="blueprint/ie.css" type="text/css" media="screen,
projection"><![endif]>-->
</head>
<body>
<div class="container showgrid">
test
<hr class="space" />
<hr class="space" />
<hr class="space" />
<hr class="space" />
<hr class="space" />
test
</div>
</body>
</html>
```

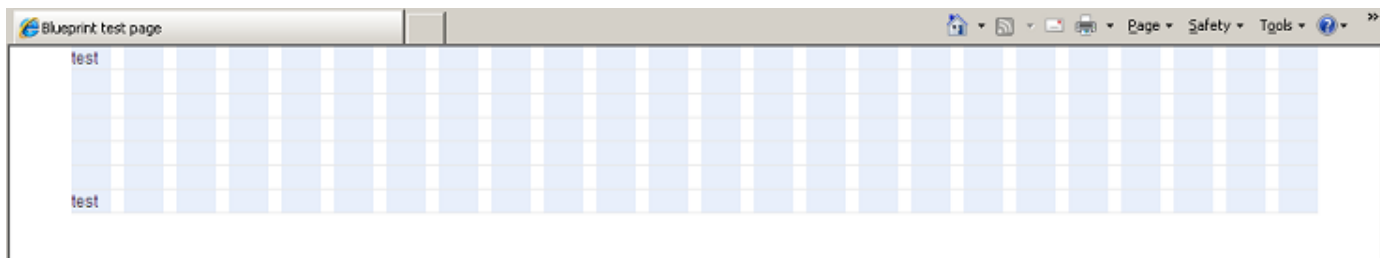
توضیحات:

پس از مشخص سازی DocType (مهم)، سه فایل CSS یاد شده به header صفحه اضافه خواهند شد. همانطور که ملاحظه می‌کنید، سازگاری با IE نیز مدنظر آن بوده است.

کار با blueprint css همواره داخل div زیر انجام می‌شود:

```
<div class="container">
  page
</div>
```

توسط کلاس container یک گرید به عرض 950px در میانه صفحه برای شما تشکیل خواهد شد. اگر علاقمند باشید که این گرید را مشاهده نمائید و همچنین بتوانید ستون‌های آن را نیز شمارش کنید، تنها کافی است showgrid را به این class تعریف شده اضافه نمائید (همانند ساختار صفحه فوق). به این ترتیب شکل زیر نمایان خواهد شد:



مطابق شکل فوق، در این عرض مشخص، 24 ستون آن در اختیار ما خواهند بود. به علاوه ذکر hr با class=space سبب خواهد شد تا مطابق تنظیمات و فاصله بندی منظم این فریم ورک، یک سطر خالی برای ما ایجاد شود.

طراحی بدون جدول با Blueprint CSS

در ادامه قصد داریم در این صفحه ابتدایی، یک جدول با دو ستون و دو ردیف را ایجاد کنیم:

```
<body>
  <div class="container showgrid">
    <div class="span-12">
      row1-col1
    </div>
    <div class="span-12 last">
      row1-col2
    </div>
    <div class="span-12">
      row2-col1
    </div>
    <div class="span-12 last">
      row2-col2
    </div>
  </div>
</body>
```

که شکل زیر را برای ما ایجاد خواهد کرد:



توضیحات:

ستون‌های گرید نهایی با رنگ آبی مشخص هستند (class=container showgrid). اگر نیاز به 12 ستون داریم، می‌نویسیم span-12 و ... همین! به این ترتیب یک سلول جدول، با 12 ستون در اختیار ما خواهد بود. سلول بعدی هم در اینجا 12 ستونه است. اما یک last را اضافه‌تر دارد. در span-12 last این last به معنای انتهای ردیف جاری است و ذکر آن الزامی است. تا اینجا یک ردیف تمام شد. اکنون در ادامه ردیف دوم را نیز به همین ترتیب با دو div و classهایی که ملاحظه می‌کنید، مشخص خواهیم کرد.

نحوه کار کلی با Blueprint css به همین سادگی است که ملاحظه می‌کنید. تعداد ستون‌های مورد نیاز را با ذکر container showgrid به سادگی می‌توان شمارش کرد. سپس این اعداد شمارش شده و مد نظر را پس از span ذکر کنید. مثلاً اگر یک طرح سه ستونه نیاز دارید به صورت زیر خواهد بود:

```
<body>
  <div class="container showgrid">
    <div class="span-8">
      row1-col1
    </div>
    <div class="span-8">
      row1-col2
    </div>
    <div class="span-8 last">
      row1-col3
    </div>
  </div>
</body>
```

طراحی سلول‌های تو در تو

سؤال: ما پیشتر در یک html table به سادگی می‌توانستیم داخل یک سلول آن حتی یک جدول جدید نیز قرار دهیم، اینجا چطور؟ پاسخ: در اینجا هم بجای td و tr و table، از divهای تو در تو استفاده کنید. بستن ستون آخر را با last یاد شده فراموش نکنید. مثلاً:

```
<body>
  <div class="container showgrid">
    <div class="span-8">
      <div class="span-4">
        row-1, col1 : cell-1
      </div>
      <div class="span-4 last">
        row-1, col1 : cell-2
      </div>
    </div>
    <div class="span-8">
      row1-col2
    </div>
    <div class="span-8 last">
      row1-col3
    </div>
  </div>
</body>
```

در اینجا در اولین div تعریف شده دو div تو در تو اضافه شده‌اند. البته با توجه به اینکه div والد 8 ستونی است، جمع عرض divهای فرزند باید 8 باشد که در اینجا به دو div چهارستونی تقسیم شده است.



تا اینجا با کلیات نحوه طراحی یک جدول به کمک CSS و فریم ورک Blueprint CSS آشنا شدیم (به کمک container و span-n آن). در ادامه مرور سریعی خواهیم داشت بر سایر امکانات این فریم ورک CSS و منظور از این امکانات، کلمات و عبارات مجازی است که می‌توانید داخل classهای divهای تعریف شده اضافه نمایید (CSS selectors تعریف شده در آن):

prepend-n و border:

فرض کنید در divهای تو در تو قسمت قبل، قصد داریم عرض ستون اول را بجای 4 ستون به 3 ستون تبدیل کنیم، اما این div را یک ستون به سمت راست حرکت دهیم:

```
<body>
  <div class="container showgrid">
    <div class="span-8">
      <div class="prepend-1 span-3 border">
        row-1, col1 : cell-1
      </div>
      <div class="span-4 last">
        row-1, col1 : cell-2
      </div>
    </div>
    <div class="span-8">
      row1-col2
    </div>
    <div class="span-8 last">
      row1-col3
    </div>
  </div>
</body>
```

برای این منظور همانطور که ملاحظه می‌کنید از prepend-1 استفاده شده است. border در اینجا سبب خواهد شد تا در سمت راست div یک خط عمودی رسم شود. در مقابل آن colborder هم وجود دارد که سبب ترسیم حاشیه با فاصله بیشتری نسبت به border می‌شود.

شبیه به همین قابلیت، با append-x (افزودن تعدادی ستون به سمت راست)، prepend-top (فاصله‌ای به اندازه 1.5em را به بالای div اضافه می‌کند) و append-bottom (فاصله‌ای به اندازه 1.5em را به پایین div اضافه می‌کند) نیز وجود دارد. در مقابل این‌ها، push-n و pull-n هم وجود دارند. کار append و prepend اضافه کردن چند ستون به بعد و قبل از یک div است. push یک div را به تعداد واحدی که مشخص می‌کنیم به سمت راست حرکت می‌دهد. pull یک div را n ستون به سمت چپ حرکت خواهد داد (بدون تغییری در تعداد ستون‌ها).

[دریافت مرجع سریع Blueprint CSS](#)

نظرات خوانندگان

نویسنده: مجتبی حسینی
تاریخ: ۱۷:۳۰ ۱۳۹۱/۰۶/۰۱

از سرعت پاسخ‌گویی‌تان در شگفتم!
بسیار ممنونم.

نویسنده: na3er-faraji
تاریخ: ۲۰:۳۶ ۱۳۹۱/۰۶/۰۱

خسته نباشید. بسیار عالی و کاربردی بود. sample هاشو دارم می‌بینم چیزهای کاربردی دیگه ای هم توش پیدا میشه.

نویسنده: هوشنگ
تاریخ: ۲۲:۳۹ ۱۳۹۱/۰۶/۰۱

طرز کار همه css framework ها شبیه به هم هست . من از 960gs استفاده میکنم . تنها تفاوت این اضافه داشتن typography و style فرم هست که من خودم اضافه میکردم

نویسنده: میثم
تاریخ: ۱۷:۱۷ ۱۳۹۱/۰۶/۲۳

سلام و تشکر فراوان
به نظر بنده این CSS Framework یک مشکل کوچک دارد و اون محدودیت عرض هستش و رویکرد جدیدی که در طراحی وب سایت‌ها مطرح هستش رو در نظر نگرفتند. سایتهایی که محدودیت عرض ندارند و بنا به رزولوشن صفحه کاربر عرض اونا تغییر میکنه و در حقیقت فقط یک minwidth دارند. یه چیزی شبیه به فریم ورک [YAML](#)
آیا همچون طرح‌هایی شبیه به همین [سایت](#) اشکالی دارند؟ یا این مساله به سلیقه بستگی دارد
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۷:۴۰ ۱۳۹۱/۰۶/۲۳

من چون در سایت جاری از blueprint css استفاده می‌کنم، خواستم نحوه کار با آن را به اشتراک بگذارم.

نویسنده: احسان بوربور
تاریخ: ۱۹:۲۱ ۱۳۹۱/۰۷/۱۳

با سپاس فراوان خیلی کاربردی بود

من هم خواستم برای به اشتراک گذاشتن این مطلب ازتون تشکر کنم.

نویسنده: علی مهدوی
تاریخ: ۱۱:۵۴ ۱۳۹۱/۰۸/۰۹

با bootstrap چه فرقی داره؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۵ ۱۳۹۱/۰۸/۰۹

بسته به نیاز و کاربرد باید ابزار مناسبی رو انتخاب کرد. آیا شما نیاز به تمام عناصر موجود در فریم ورک توئیتر bootstrap دارید؟ من نیازی نداشتم به همین جهت انتخاب انجام شده هم بسیار سبک، کم حجم و متناسب بوده. از توئیتر bootstrap هم فقط دکمه‌های css ایی آن برای من جالب بود. نیازی به مابقی آن نداشتم.

نویسنده: علی

تاریخ: ۱۳۹۲/۰۵/۰۹ ۱۵:۸

سلام و خسته نباشید

ممنونم از بابت معرفی این فریم ورک، ولی آیا از rtl هم پشتیبانی میکنه ؟
و اینکه من موفق به دانلود نشدم، شما چطور دانلود کردین ؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۲/۰۵/۰۹ ۱۵:۲۰

- به آدرس مخزن کد آن در متن [اشاره شده](#) .

- نیازی به نسخه RTL نداره و من از نسخه اصلش استفاده کردم. ولی بله، نسخه RTL [هم داره](#) .