

ژنرال وب دیزاین

جلسه هفدهم



ایجاد اسلایدر دوم با ناویگیتور کناری

- ۱- میبینیم که یک فاصله بین قیمت جدید و قدیم وجود دارد پس به **div** قیمت جدید یک مارجین راست میدیم.
- ۱۲- برای رنگ پس زمینه سفید قیمت جدید هم یک مثلث جدید ایجاد میکنیم با رنگ سفید
- ۱۳- مثلث قبلی هم به والد **position** میدیم هم به خودش
- ۱۴- اگه مثلث قیمت قدیمی زیر افتاده بود بهش یک **z-index** میدیم

```
# price-data-new {  
  ...  
  margin-right: 3px; //۱۱  
  
  position: relative; //۱۳  
}
```

```
# price-data-new::before { //۱۲  
  content: " ";  
  width: 0;  
  height: 0;  
  border-style: solid;  
  border-width: 8px 12px 8px 0;  
  border-color: transparent #fff  
  transparent transparent ;  
  position: absolute; // ۱۳  
  right: 0;  
  top: 8px  
}
```

```
# price-data-old::after { //۱۴  
  ...  
  z-index: 2;  
}
```



ایجاد اسلایدر دوم با ناویگیتور کناری

۱۵- در داخل price ها قیمت ها رو وارد میکنیم

۱۶- برای نوشتن اطلاعات محصول، زیر قیمت جدید با چندتا تگ p این اطلاعات رو وارد میکنیم و استایل میدیم

```
<div id="price-data">  
  <div id="price-data-old"> ۲</div> //۱۵  
  <div id="price-data-new"> ۱,۵ میلیون تومان</div>  
</div>
```

```
# price-info-new {  
  ...  
}  
<p class="fontsm"> //۱۶  
  وزن: اکیلوگرم  
</p>  
<p class="fontsm">  
  رنگ: نامعلوم  
</p>
```

```
# price-data p { //۱۶  
  float: right;  
  width: 100%;  
  margin: 2px;  
}
```



استایل transform

۱۷- این دستور برای تغییر شکل یک تگ به کار میره مثلا سایز رو چند برابر میکنه یا باعث چرخش آن میشه
مثال:

```
<div id="test"> //۱۷  
</div>
```

```
#test{ //۱۷  
width: 200px;  
height: 200px;  
background: green;  
margin: auto;  
transform: scale(2,2);  
}
```

این دستور اندازه این **div** رو در راستای **X** و **Y** دوبرابر میکنه. برای چرخش هم از **rotate** استفاده میکنیم و میگیریم چند درجه بچرخه برای چرخش در خلاف عقربه های ساعت هم درجه منفی میدیم:

```
#test{ //۱۷  
...  
transform: rotate(30deg);  
}
```

```
#test{ //۱۷  
...  
transform: rotate(-30deg);  
}
```



ادامه اسلایدر دوم

۱۸- برای ایجاد خط روی قیمت قدیمی باید از **after** و **before** استفاده کنیم. ازونجایی که قبلا از **after** استفاده کردیم الان از **before** استفاده میکنیم

```
# price-data-old::before { //۱۸
  content: " ";
  position: absolute;
  right: 0;
  top: 10px;
  width: 95%;
  height: 0;
  border-bottom: 2px solid #000;
  transform: rotate(-45deg);
}
```



استفاده از پلاگین (ایجاد تایمر تخفیف ها)

۱۹- برای تایمر این اسلایدر از یک پلاگینی به اسم flip timer استفاده میکنیم.

میشه از روش کدنویسی با JS هم استفاده کرد که کارمون سخت تر میشه به همین دلیل برای اینکه روش استفاده از پلاگین رو هم یاد بگیرید از روش پلاگین استفاده می کنیم
این پلاگین یک فایل CSS و JS است که به پروژه مون اضافه میکنیم
ولی برای تست کردن این پلاگین اول فایل ها رو در یک پروژه دیگر اجرا میکنیم تا خروجی رو ببینیم.
یادمون باشه که فایل JQuery رو ابتدا باید به پروژه جدید اضافه کنیم

نکته بسیار مهم :

البته یادتون باشه برای نوشتن اسلایدر و چیزهای دیگه هم میشه از پلاگین استفاده کرد ما برای یادگیری کدها از روش کدنویسی استفاده کردیم ولی برای پروژه هاتون میتونین منبعه از پلاگین ها استفاده کنین و کد نزنید و یا همین کدی که نوشتیم رو تبدیل به پلاگین کنید و استفاده کنید و یا حتی توی سایت تون بزارید برای فروش، یعنی برای نوشتن هر کدی میتونین بگردید پلاگین یا افزونه شو پیدا کنید و تا جایی میتونید کدهاشو تغییر بدین اما ممکنه بعضی چیزها را نشه تغییر داد بنابراین کدنویسی خودت دیگه همه چیز کاملا سفارشی و شخصی هست و هر جور دلت خواست کد رو اضافه و یا کم کن

```
<script src="js/jquery.js" ></script> //19  
<script src="js/jquery.flipTimer.js" ></script>  
<link href="css/flipTimer.css" />
```



پلاگین ها کدهای آماده ای هستند که توسط یک برنامه نویس نوشته شده و ما از آن استفاده می کنیم . به خاطر اینکه شما کار با پلاگین ها را هم یاد بگیرین ما این قسمت را با پلاگین نوشتیم . هر پلاگین یک راهنمای استفاده داره که معمولا برنامه نویس همراه آن ادامه میده و اگر هم نداد شما به دلیل اینکه JS و یا jquery بلدید میتونید از کدهاش اونها رو استخراج کنید



تایمر تخفیف ها

۲۰- برای ایجاد قسمت html این تایمر کدهای زیر رو مینویسیم و class های از پیش نوشته شده رو قرار میدیم
۲۱- بعد قسمت JS رو مینویسیم و از تابع flipTimer که توی فایل، از قبل نوشته شده استفاده میکنیم.
داخل این تابع آکولاد باز میکنیم تا تنظیمات رو اعمال کنیم از راهنمای این پلاگین استفاده کردم

```
<div class="flipTimer"> //20
  <div class="days"></div>
  <div class="hours"></div>
  <div class="minutes"></div>
  <div class="seconds"></div>
</div>
```



```
<script>
  $('flipTimer').flipTimer({ //21
  })
</script>
```

فعلا تا همینجا برنامه رو اجرا میگیریم و نتیجه رو می بینیم



تایمر تخفیف ها

۲۲- با اجرا کردن برنامه میبینیم که تایمر در حال افزایشه. ولی ما میخواییم بصورت کاهشی باشه. برای این یک موضوع، یک تنظیماتی داریم به اسم **direction** که مقادیر **up** و **down** میپذیره که با تنظیمش به **down** این مشکل حل میشه

نکته مهم: برنامه نویسی که کدهای این تایمر رو نوشته یک آپشن به اسم **direction** داده یعنی یک متغیر هست و به این معنی نیست که همه توابع **Jquery** دارای این آپشن باشن!!



```
<script>
    $('flipTimer').flipTimer({
        direction: 'down', //22
    })
</script>
```

با اجرای برنامه میبینیم که تایمر اعداد صفر رو نشون میدن چون ما هنوز زمان مشخصی ندادیم



تایمر تخفیف ها

۲۳- برای زمان دادن به تایمر از یک آپشن دیگه استفاده میکنیم به اسم **date** که باید مقداری با فرمت زیر بهش بدیم. یعنی به صورت معکوس شروع کن برو تا ساعت و دقیقه و ثانیه X روز Y سال Z
برای تست کردن تاریخ درست به سایت **time.ir** میریم تا تاریخ میلادی رو بدست بیاریم

August 05, 2023 16:20:30

```
<script>
  $('flipTimer').flipTimer({
    direction: 'down',
    date: 'August 05, 2023 16:20:30',
//23
  })
</script>
```



۲۴- برنامه رو اجرا میگیریم و نتیجه رو میبینیم.
برای اینکه بخواییم روز رو نمایش نده آپشن اون رو **false** میکنیم

```
<script>
  $('flipTimer').flipTimer({
    direction: 'down',
    date: 'August 05, 16:20:30',
    days: false , //24
  })
</script>
```



تایمر تخفیف ها

۲۵- برای اینکه وقتی تایمر به پایان برسه اتفاقی بیفته مثلا یک **alert** نمایش داده بشه از یک آپشن به نام **callback** استفاده میکنیم که دراصل این آپشن یک تابع هست. به کدهای زیر توجه کنید

```
<script>
    $('flipTimer').flipTimer({
        direction: 'down',
        date: 'August 05, 16:20:30',
        day: false ,
        callback: function(){ //25
            alert('پایان')
        }
    })
</script>
```



برنامه رو اجرا میگیریم و نتیجه رو میبینیم. اگه تایمر به صفر برسه، پایان نمایش داده میشه



ادامه اسلایدر دوم

۲۶- برمیگردیم به پروژه خودمون و فایل های تایمر رو به پروژه لینک میکنیم و کدهای html رو داخل تگ slider2 قرار میدیم چون نمیخواهیم با تغییر عکس اسلایدر، تایمر هم عوض بشه. دستورات JS هم که نوشتیم داخل تگ script کپی میکنیم

```
<div id="slider2">  
  <div class="flipTimer"> //26  
    <div class="hours"></div>  
    <div class="minutes"></div>  
    <div class="seconds"></div>  
  </div>  
</div>
```



ادامه اسلایدر دوم

۲۷- برنامه رو اجرا میگیریم و نتیجه رو میبینیم.

میبینیم که تایمرمون بهم ریخت و اصلا شبیه تایمری که توی پروژه قبل نوشتیم نیست. بنظر تون ایراد کجاست؟ حدس اینکه که ما توی این پروژه کد CSS ی داریم که با کدهای CSS این پلاگین داره تداخل ایجاد میکنه و باید اون رو رفع کنیم. برای این کار ابتدا با **inspect** میبینیم تایمرمون چه استایل هایی گرفته. با کمی دقت میفهمیم که **direction** تایمر روی **rtl** هست. با قرار دادن روی **ltr** مشکل حل میشه البته نکته اینجاست که مانیتونیم بریم **rtl** کل **div** ها را برداریم بلکه باید برای فقط تایمر این تنظیم را انجام بدیم تا سایر تنظیمات بهم نخوره و این نکته مهمی هست

```
.flipTimer , . flipTimer div{ //27
```

```
direction: ltr !important;
```

```
}
```

ما داریم تمام اتفاقاتی که حین کدنویسی رخ میده رو نشون میدیم که شما نحوه خطایابی را هم یاد بگیرین که خیلی ها توی آموزش این کار را نمی کنند



ادامه اسلایدر دوم

۲۸- برای تغییر اندازه تایمر نمیتونیم از **width** و **height** استفاده کنیم چون این تایمر از کلی **div** تشکیل شده. راه حل بهتر استفاده از دستور **transform** هست.

۲۹- برای تغییر موقعیت هم از **position** استفاده میکنیم. به **slider2** میایم **position: relative** میدیم و به خود تایمر **position: absolute**

```
.flipTimer{ //28
  transform: scale(.5);
}
```

```
.slider2{ //29
  ...
  Position: relative;
}

.flipTimer{
  position: absolute;
  top: 180px;
  right: -140px;
}
```



ادامه اسلایدر دوم - تابع CSS

۳۰- برای اینکه بعد به پایان رسیدن تایمر اسلایدر، استایل اسلایدر تغییر کنه ما در **Jquery** برای تغییر استایل از تابع **CSS** استفاده میکنیم. این کار رو در **callback** انجام میدیم.

تابع **CSS** دو تا ورودی دریافت میکنه اولی اون **property** که قراره تغییر بدیم و دومی مقدار اون هست. برای مثال میخواییم **opacity** اسلایدرمون به **0.4** برسه از دستور زیر استفاده میکنیم:

```
<script>
    $('.flipTimer').flipTimer({
        direction: 'down',
        date: 'August 05, 16:20:30',
        day: false ,
        callback: function(){ //30
            $('. slider2-con-right').css('opacity', .4)
            $('. slider2-con-left').css('opacity', .4)
        }
    })
</script>
```



ادامه اسلایدر دوم

۳۱- میخوایم بعد از به پایان رسیدن تایمر یک متن تمام شد هم روی اسلایدر ظاهر شود. برای این کار یک متن با تگ **div** ایجاد میکنیم همراه استایل آن به اندازه خود اسلایدر. **Display** هم **block** قرار میدیم و در **callback** اون رو **fadeIn** میکنیم روی اسلایدر:

```
<div id="slider2">  
  <div class="slider2_finished">  
    تمام شد  
  </div>  
  ....  
</div>
```

```
<script>  
...  
callback: function(){ //30  
$('. slider2-con-right').css('opacity', .4)  
$('. slider2-con-left').css('opacity', .4)  
$('.slider2_finished').fadeIn(100)  
}  
</script>
```

```
.slider2_finished{ //31  
  width: 700px;// اندازه کل کانتنت  
  height: 100%;  
  position: absolute;  
  top:0;  
  right:0;  
  background: rgba(0,0,0,.2);  
  color: red;  
  font-size: 18px  
  text-align: center;  
  line-height: 200px;  
  display: none;  
}
```


پایان جلسه
موفق باشید

