

تعریف و توضیح float

ویژگی float از آن دسته از ویژگی هایی است که مکان عنصر را مشخص می کند. این ویژگی باعث می شود تا عنصر به سمت چپ یا راست عنصر نگهدارنده اش هول داده شده یا شناور شود.

این ویژگی می تواند مقادیر زیر را داشته باشد:

```
float: right;
float: left;
float: none; /* مقدار پیشفرض */
float: inherit;
```

زمانی که یک عنصر به سمت راست هول داده می شود (float: right) محتوای بعد از آن دور لبه چپ عنصر پیچیده می شود. و زمانی که عنصر به سمت چپ هول داده شود (float: left) محتوای بعد از آن دور لبه راست عنصر جمع می شود.

ویژگی float کارایی بالایی در طراحی صفحات وب دارد. مثلاً فرض کنید گالری از تصاویر دارید که هر تصویر یک متن توضیح نیز دارد. html آن بصورت زیر خواهد بود:

```
<div class="gallery-container">
  <figure>
    
    <figcaption>
      image caption
    </figcaption>
  </figure>
  <figure>
    
    <figcaption>
      image caption
    </figcaption>
  </figure>
```

```
</figure>
<figure>
  
  <figcaption>
    image caption
  </figcaption>
</figure>
</div>
```

چون عنصر `<figure>` از نوع عناصر بلاک می باشد پس اجازه نمی دهد تا عناصر بعد از آن در کنار آن قرار بگیرند و همچنین این عنصر کل فضای افقی محیط خود را اشغال می کند:



image caption



image caption



image caption

کافی است تا به این عناصر float داده شود:

```
figure {  
  float: left;  
  margin: 10px;  
}
```



زمانی که یک عنصر بلاک به یک سمت float, یا همان شناور می شود عرض آن عنصر به حداقل ممکن یعنی عرض محتوای داخل آن می رسد.

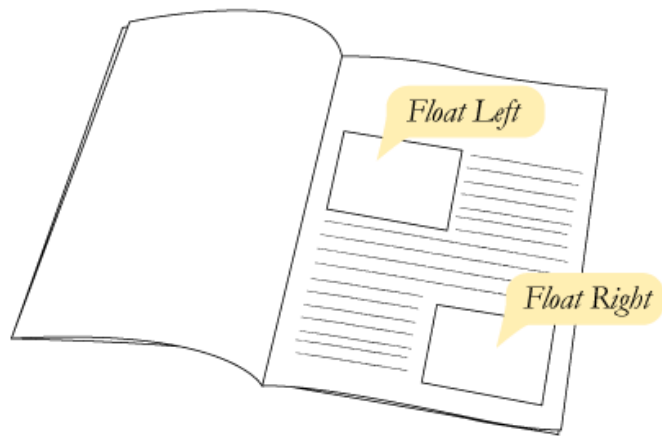
در مثال بالا بعد از شناور کردن عناصر به سمت چپ, عرض عناصر به اندازه تصویر و متن داخلشان کاهش پیدا کرد.

صفحه آرایي با استفاده از float

قبل از اینکه ویژگی float معرفی شود, طراحان با استفاده از عنصر table و اجزا آن عناصر صفحه را در کنار هم می چیدند یا به اصطلاح صفحه آرایي می کردند. که کار جالبی نبود چرا که عنصر table برای ساخت جدول می باشد نه صفحه آرایي.

در نتیجه با ورود float به دنیای وب جایگزین خوبی برای روش قبلی صفحه آرایي شد. کافی است به عناصر بلاک که نقش نگهدارنده های اصلی صفحه را دارند float داده شود تا بتوانیم ستون های مختلف برای محتوای صفحه را بسازیم.

در واقع ویژگی float نیز برای صفحه آرایي معرفی نشده است بلکه همانطور که اول مطلب گفته شد کار اصلی آن این است که محتوای بعد از آن دور عنصر جمع شود. یعنی همان حالتی که در مجلات و روزنامه ها وجود دارد:



چندین قانون در مورد **float** وجود دارد که نحوه رفتار این ویژگی را شرح می دهد:

۱. زمانی که یک عنصر **float** می گیرد و به یک سمت شناور می شود، بیرونی ترین لبه عنصر نمی تواند از عنصر نگهدارنده اش خارج شود. (اگر عنصر **margin** داشته باشد، **margin** بیرونی ترین لبه محسوب می شود و اگر نباشد بعد از آن **border** منظور این قانون است)



همانطور که در تصویر بالا مشاهده می کنید عنصر دارای فاصله بیرونی ۵۰ پیکسل از سمت چپ است (**margin-left: 50px**) اما آن فاصله در زمانی که عنصر به سمت چپ شناور شده است (**float: left**) همچنان در عنصر نگهدارنده اش حفظ شده و خارج نشده است.

۲. وقتی یک عنصر به سمتی شناور می شود باید تا جایی که امکان دارد به سمت منظور هول داده شود. مثلاً اگر به سمت چپ شناور می شود باید تا جایی که می تواند به سمت چپ عنصر نگهدارنده اش تغییر مکان دهد.

۳. یک عنصر شناور تا جایی به سمت منظور هول داده می شود تا بیرونی ترین لبه آن، لبه عنصر نگذارنده اش یا لبه عنصر شناور دیگری را لمس کند.

۴. فاصله بیرونی سمت راست (margin-right) یک عنصری که به سمت چپ شناور شده است در سمت راست فاصله بیرونی سمت چپ یک عنصری که به سمت راست شناور شده است و در سمت راست عنصر نیز است، قرار نمی گیرد. برای سمت چپ هم این قانون برقرار است. (منظور این قانون عدم تداخل فواصل بیرونی عناصر شناور است)

۵. اگر عرض عنصر شناور از فضای موجودی که در اطرافش هست بیشتر شود عنصر به خط بعد سقوط می کند.

۶. اگر یک عنصر inline را شناور کنیم مهم نیست که مقدار display آن چه باشد، در هر صورت عنصر به نوع block تبدیل می شود. البته به غیر از حالت none

۷. عنصر ریشه (html) نمی تواند شناور شود.

۸. عناصری که مقدار position آنها absolute یا fixed است نمی توانند شناور شوند.

۹. عنصر شناور از flow صفحه خارج نمی شود، این یعنی حرکت آن بر عناصر همسایه آن تاثیر می گذارد.

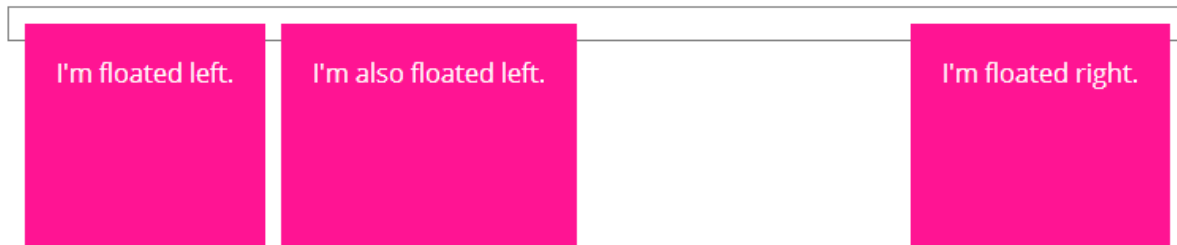
۱۰. علاوه بر نکات و قانون های گفته شده دانستن تاثیر عناصر شناور بر رفتار نگهدارنده شان یا همان عنصر پدرشان حائز اهمیت است:

عناصر شناور تاثیری بر روی ارتفاع نگهدارنده شان نمی گذارند.

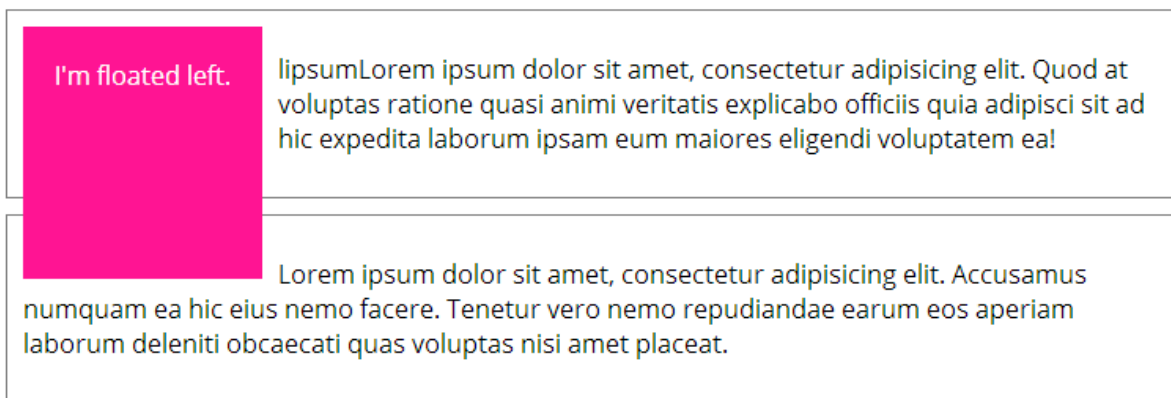
در حالت عادی عنصر ارتفاع محتوا یا فرزندانش را می گیرد (مگر اینکه ارتفاع (height) خاصی برای آن مشخص شود). اما اگر فرزندان آن شناور (float) باشند، این فرزندان تاثیری بر ارتفاع عنصر پدر نخواهند گذاشت.

I'm floated left.

lipsumLorem ipsum dolor sit amet, consectetur adipiscing elit. Quod at voluptas ratione quasi animi veritatis explicabo officii quia adipisci sit ad hic expedita laborum ipsam eum maiores eligendi voluptatem ea!



از این وحشتناک تر اینکه عنصر شناور روی عناصر بعدی در نگهدارنده های دیگر هم قرار می گیرد و حتی بر روی محتوای آنها تاثیر می گذارد (متن عنصر دوم به علت وجود عنصر صورتی رنگ که شناور است به جلو حرکت کرده است):



این اتفاق اکثر مواقع دیده نمی شود چون عناصر گاهی حاشیه یا رنگ خاصی ندارند تا طراح متوجه این اتفاق باشد. در نتیجه با مشکلات عجیب و غریب صفحه آرای روبرو خواهد شد.

برای جلوگیری از این اتفاق باید عملیاتی را انجام دهید که به آن **Clearing floats** می گویند. چندین روش برای این کار وجود دارد که در ادامه معرفی شده اند و در نهایت بهترین و محبوب ترین روش معرفی می شود:

استفاده از یک عنصر خالی در نگهدارنده

در این روش از یک عنصر خالی (ترجیحا **div**) با اعمال ویژگی **clear** بر روی آن استفاده می شود. کار این ویژگی این است که اطراف عنصر شناور را پاکسازی کند تا عناصر دیگر به پایین عنصر شناور سقوط کنند. برای بررسی دقیقتر این ویژگی به مطلب مربوط به آن مراجعه کنید.

می توانید از یک کلاس برای این کار استفاده کنید:

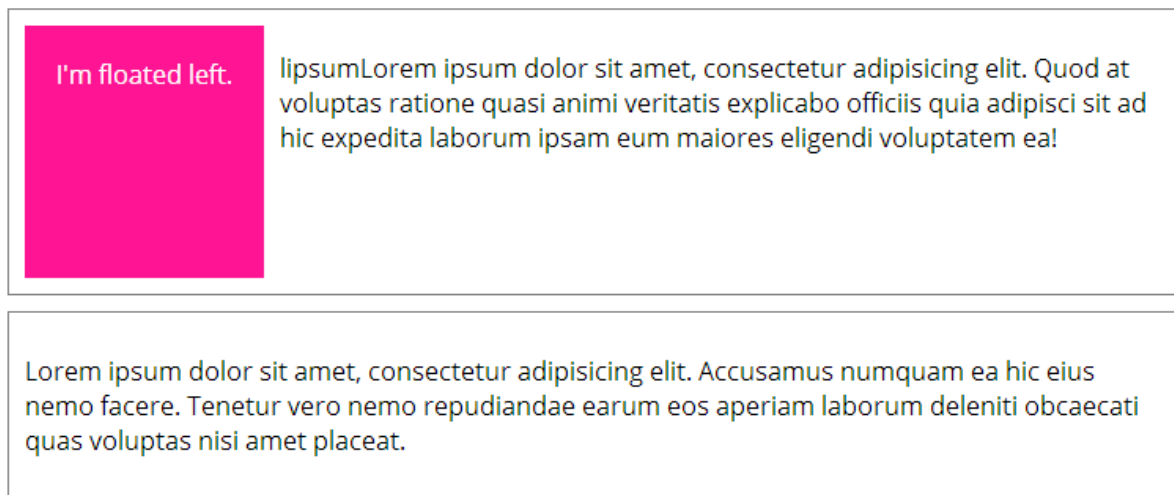
```
<div class="clearfix"></div>

.clearfix {

    clear: both;

}
```

در نهایت نتیجه بصورت زیر خواهد بود و مشکل برطرف می شود:



واضح است که این روش جالب نیست. چراکه یک عنصر بی معنا را به کد اضافه می کند و همچنین هر بار برای رفع این مشکل باید این کار برای عناصر دیگر نیز انجام شود.

استفاده از ویژگی **overflow** برای نگهدارنده:

کافی است ویژگی **overflow** را برای عنصر نگهدارنده بر روی مقدار **hidden** یا **auto** تنظیم کنید.

```
.floats-container {
    overflow: hidden;
    /* یا */
    overflow: auto;}

```

این روش بدلیل سادگی آن زیاد مورد استفاده قرار میگیرد اما مسلماً تعیین **overflow** برای یک عنصر می توان مشکلاتی را به همراه داشته باشد. شاید عنصر نگهدارنده محتوایی داشته باشد که بخواهد بیرون از آن قرار بگیرد، که ما با این کار جلوی این کار را می گیریم.

شناور کردن خود عنصر نگهدارنده

کافی است که به هر سمتی که ترجیح می دهید نگهدارنده را شناور کنید تا ارتفاع فرزندان شناگرش را بگیرد.

```
.floats-container {  
  float: right;  
  /* یا */  
  float: left;  
}
```

اگرچه این روش هم می تواند باعث مشکلاتی در صفحه آرایی شود چرا که شاید شما نخواهید عنصر نگهدارنده به یک سمت هول داده شود و این کار باعث به هم ریختگی در صفحه آرایی شما شود. پس با آگاهی کامل از این روش استفاده کنید.

چند نکته

ویژگی **float** با ویژگی های **display** و **position** تداخل هایی دارد:

اگر **display** برای یک عنصر **none** باشد **float**، روی آن عنصر هیچگونه تاثیری نخواهد داشت.

عناصری که مقدار **position** آنها **absolute** یا **fixed** است نمی توانند شناور شوند و **float** برای آنها **none** در نظر گرفته می شود.

اگر عنصری ذاتاً **inline** باشد یا **display** آن **inline** باشد زمانی که **float** روی آن اعمال می شود تبدیل به **block** می شود.

همینطور عنصری با **display** از نوع **inline-table** به حالت **table** تغییر پیدا می کند.