

<<Flexbox>>

یکی از مژول های CSS 3 که برای طراحی های تک بعدی و نمایش عناصر موجود در صفحه ی وب هست برای نصب آن به وابستگی خاصی نیاز نیست و کارایی زیادی در طراحی های واکنش گرا دارد چون با چیدن آیتم ها درون یک سطر یا ستون در قالب نگهدارنده، بطور خودکار عناصر طراحی را با اندازه ی صفحه منطبق می کند و مشکلاتی مثل Float & WhiteSpace را بر طرف می کند پس با استفاده از آن می توان طراحی چند سطحی یا چند ستونی را با زحمت کتری انجام داد و بهتر آن را مدیریت کرد، به عبارت دیگر ابزاری است که با تک بعدی دین المان ها کمک می کند تا آیتم های یک صفحه و برا به آسانی و با سرعت بالا و بصورت واکنش گرا طراحی کنیم با آن می توان المان ها را به راحتی در حالت عمودی یا افقی، پشت سر هم قرار بدیم یا چینش المان ها را معکوس کنیم و یا فاصله بین المان ها ایجاد کنیم.

Flex Container

همان طور که در طراحی صفحه با CSS Grid، المان ها در یک Grid مادر با نام flex قرار می گیرند، در FlexBox نیز یک المان نگهدارنده با عنوان Container داریم: container

display:flex | inline-flex

برای اینکه بتوانیم خواص flex box را روی المان ها اجرا کنیم، باید از آن استفاده کنیم. و اگه این خاصیت رو به المان والد بدهیم، تمام المان هایی که داخل آن قرار می گیرند از خواص flex box تبعیت می کنند.

<<Flexbox>>

flex-direction

یه خاصیت مهم دیگه به نام **flex-direction** هست . با استفاده از این خاصیت میتوانیم بگیم المان ها ، افقی کنار هم ، یا این که عمودی زیر هم قرار بگیرن .

چهار تا مقدار میگیره که برآتون مختصرا توضیح میدیم .

: row : در این حالت المان ها به صورت افقی به ترتیب از چپ به راست کنار هم قرار میگرین (با این فرض که direction: ltr باشه) . در ضمن این حالت به صورت پیش فرض روی المان ها اجرا میشه .

: row-reverse : با این مقدار میتوانیم جهت چینش المان ها رو معکوس کنیم . مثلا اگر المان ها از چپ به راست چیده شده بودن ، حالا از راست به چپ چیده میشن .

: column : با این مقدار المان ها به صورت عمودی به ترتیب از بالا به پایین زیر هم قرار میگیرن .

: column-reverse : جهت چینش المان ها رو در حالت عمودی عوض میکنه (جهت چینش میشه از پایین به بالا)

justify-content

وقتی که المان های فرزند کل فضای المان والد رو نگرفتباشن ، میشه از این خاصیت برای مرتب کردنشون استفاده کرد . پنج تا مقدار میگیره که مختصرا برآتون توضیح میدیم .

: flex-start : المان های فرزند در نقطه شروع المان والد قرار میگیرن . مثلا اگر direction: ltr باشه ، المان های فرزند از سمت چپ المان والد کنار هم چیده میشن . (این حالت به صورت پیش فرض اجرا میشه)

: flex-end : در این حالت المان های فرزند از انتهای المان والد کنار هم دیگه چیشده میشن .

: center : المان های فرزند در وسط المان والد قرار میگیرن .

: space-between : المان ها با فاصله بینشون کنار هم قرار میگرین .

: space-around : المان ها با فاصله بین ، قبل و بعدشون کنار هم قرار میگرین .

align-items

یکی دیگه از خاصیت های **Flexbox** ، خاصیت **align-item** **Flexbox** هست . با این خاصیت میتوانیم تراز المان های فرزند رو انجام بدیم . پنج تا مقدار هم میگیره .

المان ها فرزند اون قدر کش میان که اندازشون با المان والد متناسب بشه .

المان های فرزند در وسط المان والد قرار میگیرن .

المان های فرزند در ابتدای المان والد قرار میگیرن .

المان های فرزند در انتهای المان والد قرار میگیرن .

المان های فرزند در خط اول المان والد قرار میگیرن .

نکته مربوط به مقدار Stretch

اگر بخواهیم مقدار Stretch عمل کرده و المان مورد نظر تمامی فضای Cross axis را اشغال کند، دو راه داریم:

- از ویژگی height برای المان های مورد نظر استفاده نکنیم.
- در صورت استفاده از height، مقدارش را برابر با auto قرار بدهیم. توجه داشته باشید که اگر ویژگی height را برای المان مورد نظر تعریف کنیم، مقدار stretch را بازنویسی (override) خواهد کرد و در این صورت المان تمامی فضا را اشغال نمیکند.

نکته مربوط به مقدار baseline

زمانی که مقدار ویژگی را برابر با baseline قرار می‌دهیم، المان‌ها طوری تنظیم می‌شوند که پایین پاراگراف درون div ها، در یک خط قرار گرفته و تراز شود. اگر درون div ها پاراگرافی وجود نداشته باشد، div ها نسبت به موقعیت پایین خودشان تراز خواهند شد. مانند شکل زیر:

flex-wrap

چیکار میتوانیم بکنیم ؟ می تونیم مشخص کنیم که المان های دارای خاصیت Flex دارای **Wrap** باشن یا نه . چند تا مقدار میگیره که برآتون توضیح میدیم :

now-wrap : اگر مقدار nowrap بگیرن بدون اینکه وابسته به المان مادر باشن المان های فرزند در کنار هم قرار میگیرن اما اگر مقداری غیر از nowrap بگیرن وابسته به المان مادر ، اگر توی عرض المانی جا نشه ، میره پایین قرار میگیره .

wrap : المان ها در صورت ضرورت مقدار wrap رو میگیرن .

المان ها در صورت ضرورت مقدار wrap رو به طور **wrap-reverse** برعکس میگیرن .

order , flex-flow

flex-direction و **flex-flow** : این خاصیت کد کوتاه برای دو تا خاصیت **flex-wrap** هست . طرز نوشتن کد هم به شکل زیر هست :

flex-flow: flex-direction flex-wrap

order : این همون خاصیت جالبی هست که گفتم . با این خاصیت میشه ترتیب چینش المان ها رو به شکل دلخواه خودمون تغییر داد . مثلا اگر چهار تا المان پشت سر هم داشته باشیم ، میتوانیم المان دومی رو جای اولی و المان اولی رو جای سومی بذاریم . لینک زیر رو هم به

Align content

این خاصیت هنگامی که flex container دارای فضای بیشتر از flex line هاست، می‌تواند نحوه چینش آن‌ها را تعیین کند. مقادیر خاصیت align-content برابر مقادیر خاصیت justify-content می‌باشد و تنها یک مقدار بیشتر از justify-content دارد. این مقادیر عبارتند از:

- با این مقدار flex line‌ها در ابتدای flex container جمع می‌شوند.
- با این مقدار flex line‌ها در انتهای flex container جمع می‌شوند.
- با این مقدار flex line‌ها در وسط flex container جمع می‌شوند.
- فضای خالی به طور مساوی بین flex line‌ها توزیع می‌شود. خط اولی در ابتدا و خط آخری در انتهای flex container قرار می‌گیرد.
- فضای خالی به طور مساوی بین flex line‌ها تقسیم می‌شود.
- (پیشفرض) flex line‌ها کشیده می‌شوند و فضای خالی را پر می‌کنند.

Flex Item

به المانهایی که درون نگهدارنده و به عنوان فرزند قرار می‌گیرند Flex Item گفته می‌شود. برای Flex شدن فرزندان کافی است مانند کد زیر در تگ HTML آنها را درون یک div نگهدارنده قرار دهیم و نیازی به تعریف کلاس خاصی در استایل نیست:

خواص فرزندان (Flex Items)

order •

flex grow •

flex shrink •

flex basis •

flex •

align self •

Order

در این قسمت آموزش FlexBox به توضیح ویژگی Order در فلکس باکس می‌پردازیم. این مقدار می‌تواند مکان قرارگیری آیتم در کنار دیگر آیتم‌ها را مشخص کند. مقدار پیش‌فرض 0 است. اگر تنها مقدار order یکی از آیتم‌ها تعیین شود، آن آیتم بعد از تمامی آیتم‌ها قرار می‌گیرد؛ ولی اگر این خاصیت برای همهی آیتم‌ها تعریف شود، می‌توان جایگاه قرارگیری آیتم‌ها را مشخص کرد. این خاصیت یکی از کاربردی‌ترین خاصیت‌ها هنگام طراحی واکنش‌گرا (responsive) هست. برای مقداردهی order می‌توان برای هر آیتم در استایل کلاس جداگانه تعریف کرد یا به صورت مستقیم در HTML آن را همانند زیر مشخص نمود:

```
<div class="flex-container">
  <div style="order: 3">1</div>
  <div style="order: 2">2</div>
  <div style="order: 4">3</div>
  <div style="order: 1">4</div>
</div>
```

Flex grow

در این قسمت آموزش FlexBox به توضیح ویژگی Flex grow در فلکس باکس می‌پردازیم. این خاصیت توانایی بزرگ شدن یک آیتم را تعیین می‌کند. flex-grow یک مقدار عددی بدون واحد می‌پذیرد. مقدار پیش‌فرض آن ۰ است؛ یعنی آیتم‌ها توانایی بزرگ شدن را ندارند. در صورتی که flex container یا نگهدارنده فضای خالی داشته باشد و به یک آیتم flex-grow: 1 بدهید، آن آیتم تا جایی بزرگ می‌شود که فضای خالی درون نگهدارنده از بین برود.

دقت داشته باشید که هنگامی که خاصیت flex-direction نگهدارنده برابر با row باشد، آیتم اعمال شده و اگر مقدار آن خاصیت برابر باشد، flex-grow بر عرض (width) آیتم اعمال شده و اگر مقدار آن خاصیت column باشد، flex-grow بر ارتفاع (height) آیتم اعمال می‌شود.

```
<div class="flex-container">
  <div style="flex-grow: 1">1</div>
  <div style="flex-grow: 1">2</div>
  <div style="flex-grow: 8">3</div>
  <div style="flex-grow: 1">4</div>
</div>
```

اگر به همه ی فلکس آیتم‌ها flex-grow: 1 بدهید، همه‌ی آنها به یک اندازه بزرگ می‌شوند. اگر در همین حالت مقدار flex-grow یکی از آیتم‌هارا 2 قرار دهید، آن آیتم 2 برابر دیگر آیتم خواهد شد.

Flex shrink

در این قسمت آموزش Flex shrink به توضیح ویژگی FlexBox در فلکس باکس می‌پردازیم. این خاصیت توانایی کوچک شدن یک آیتم را در صورت لزوم تعیین می‌کند. مقدار پیش‌فرض آن ۱ است، یعنی آیتم‌ها می‌توانند کوچک شوند. اگر خاصیت flex-wrap نگهدارنده برابر nowrap باشد، آیتم‌ها تا زمانی که عرضشان به ۰ برسد کوچک می‌شوند:

حال اگر در این صورت مقدار flex-shrink یکی از آیتم‌ها را برابر ۲ قرار دهیم، آن آیتم ۲ برابر دیگر آیتم‌ها کوچک می‌شود.

Flex basis

در این قسمت آموزش Flex basis به توضیح ویژگی FlexBox در فلکس باکس می‌پردازیم. این خاصیت مقدار پایه‌ی عرض یا ارتفاع آیتم را تعیین می‌کند. یعنی عرض یا ارتفاع (به سطونی یا ردیفی بودن آیتم‌ها بستگی دارد) از این مقدار پایه نمی‌تواند کمتر شود. این خاصیت مانند min-width و min-height عمل می‌کند. مقدار پیش‌فرض این خاصیت auto می‌باشد، یعنی فلکس آیتم‌ها می‌توانند تا جایی کوچک شوند که عرض یا ارتفاع آنها به صفر برسد.

Flex

در این قسمت آموزش FlexBox به توضیح ویژگی Flex در فلکس باکس می‌پردازیم. این خاصیت در واقع یک امکان برای مختصر نویسی برای خاصیت‌های در کامنت مشخص شده است. مقادیر دوم و سوم (flex-basis و flex-shrink) در کامنت مشخص شده است. مقادیر دوم و سوم (flex-grow ، flex-shrink و flex-basis) اختیاری هستند.

```
ntainer">  
  
0 0 200px">3</div> /* flex: flex-grow flex-shrink flex-basis */
```

Align self

در این قسمت آموزش FlexBox به توضیح ویژگی Align self در فلکس باکس می‌پردازیم. به عنوان آخرین خاصیت باقی‌مانده از مبحث FlexBox، ویژگی align-self را معرفی می‌کنیم. این ویژگی موقعیت و ترازبندی یک المان را به صورت جداگانه مشخص کرده و ویژگی align-items را برای المان مورد نظر، بازنویسی می‌کند. مقادیر مورد استفاده برای این ویژگی مشابه با ویژگی align-items مقدار پیش فرض auto می‌باشد که در این حالت موقعیت از ویژگی align-items تبعیت می‌کند. برای نمونه با اعمال این کد، فقط مربع اول در وسط قرار گرفته و بقیه موارد در ابتداء قرار می‌گیرند:

```
#container {
    align-items: flex-start;
}
.square#one {
    align-self: center;
}
```

حال فرض کنید که 4 مربع دارید و برای 2 مورد از ویژگی align-self استفاده کرده و برای سایر موارد ویژگی‌های flex-direction: row و align-items: center را تنظیم می‌کنیم. با این کار دو المان سمت راست ثابت در مرکز قرار می‌گیرند و دو مورد سمت چپ با استفاده از ویژگی align-self تغییر موقعیت می‌دهند. در نتیجه خروجی به صورت زیر خواهد شد:

