[htmlandcssguidebook.com](https://htmlandcssguidebook.com/css/flexbox/)

**Flexbox**

**فلکس باکس**

|  |  |  |  |
| --- | --- | --- | --- |
| **انعطاف پذیر** | **خم شدن** |  |  |
|  |  |  |  |

5–7 minutes

Flexbox is a powerful, flexible CSS system for arranging items in a **single line** / along a single axis.

Flexbox یک سیستم CSS قدرتمند و منعطف برای مرتب کردن موارد در یک خط / در امتداد یک محور است.

This article only covers the very basics of flexbox.

این مقاله فقط اصول اولیه flexbox را پوشش می دهد.

You may want to research more advanced properties and techniques on your own.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ممکن است بخواهید خودتان درباره خواص و تکنیک های پیشرفته تر تحقیق کنید.

**On This Page**

**در این صفحه**

* **یک سیستم CSS قدرتمند و منعطف برای مرتب کردن المانها در یک خط یا امتداد یک محور.**

**Activating Flexbox**

**فعال کردن فلکس باکس**

Flexbox is activated for a group of elements by applying display: flex; to the *direct* parent of the elements to be placed in a row.

Flexbox برای گروهی از عناصر با اعمال نمایش فعال می شود: flex; به والد مستقیم عناصری که در یک ردیف قرار می گیرند.

* داره میگه اینکه با display:flex شناخته میشه و مستقیما در المان پرنت میاد.

A good mnemonic for remembering that flex properties are applied to the direct parent is:

یک یادگاری خوب برای به خاطر سپردن اینکه ویژگی های flex برای والد مستقیم اعمال می شود:

Parents keep *their* **children** in line (pun intended). They **aren't** responsible for grandchildren or other decendants.

والدین فرزندان خود را در صف نگه می دارند (جناسی در نظر گرفته شده). آنها مسئولیتی در قبال نوه ها یا فرزندان دیگر ندارند.

So let's look at some code. If we start with a series of elements within a shared parent:

پس بیایید به کدی نگاه کنیم. اگر با یک سری عناصر در یک والد مشترک شروع کنیم:

We can apply display: flex; to the parent div, and its direct children are automatically arranged in a left-justified horizontal line, according to flexbox defaults:

ما می توانیم نمایش را اعمال کنیم: flex; به div والد، و فرزندان مستقیم آن به طور خودکار در یک خط افقی توجیه شده سمت چپ، مطابق پیش فرض های flexbox مرتب می شوند:

*Notice that the list items still stack on top of one another.*

توجه داشته باشید که آیتم های لیست همچنان روی هم قرار می گیرند.

*This is because they are not direct children of the div that has been set to flex.*

این به این دلیل است که آنها فرزندان مستقیم div نیستند که برای انعطاف پذیری تنظیم شده است.

**Flex Direction**

As mentioned above, by default, flex-children are arranged in a horizontal row, but we can use the flex-direction property on the flex-parent to change the direction of the line. The possible values are:

جهت فلکس

همانطور که در بالا ذکر شد، به طور پیش فرض، flex-children در یک ردیف افقی قرار می گیرند، اما می توانیم از ویژگی flex-direction در flex-parent برای تغییر جهت خط استفاده کنیم. مقادیر ممکن عبارتند از:

* row: The default value, which places the items in a horizontal line.
* row-reverse: Reverses the order of elements and places them in a horizontal line.
* column: Places the items in a vertical line.
* column-reverse: Reverses the order of the items and places them in a vertical line.
* • row: مقدار پیش فرض که موارد را در یک خط افقی قرار می دهد.
* • row-reverse: ترتیب عناصر را معکوس می کند و آنها را در یک خط افقی قرار می دهد.
* • ستون: موارد را در یک خط عمودی قرار می دهد.
* • ستون- معکوس: ترتیب موارد را معکوس می کند و آنها را در یک خط عمودی قرار می دهد.

The direction of specified here is referred to as the **main axis**.

جهت مشخص شده در اینجا به عنوان محور اصلی نامیده می شود.

Here, the line with the arrow represents the direction of the main axis.

در اینجا، خط با فلش، جهت محور اصلی را نشان می دهد.

And here is a similar example with code:

و در اینجا یک مثال مشابه با کد وجود دارد:

In the example above, try the different possible values for flex-direction for the .row-parent selector.

در مثال بالا، مقادیر مختلف ممکن برای flex-direction را برای انتخابگر .row-parent امتحان کنید.

Pay attention to how the children change direction and order.

به نحوه تغییر جهت و نظم بچه ها توجه کنید.

**Justify Content**

مطالب را توجیه کنید

By default, flex-children are aligned along the *main axis* at the start of the axis line, kind of like words in a sentence.

به طور پیش‌فرض، کودکان انعطاف‌پذیر در امتداد محور اصلی در ابتدای خط محور قرار می‌گیرند، مانند کلماتی در یک جمله.

But this can be changed using the justify-content property on the flex parent.

اما این را می توان با استفاده از ویژگی justify-content در والد flex تغییر داد.

The possible values are:

مقادیر ممکن عبارتند از:

* flex-start: The default value, which groups the items in order at the start of the line of the main axis.
* • flex-start: مقدار پیش فرض که موارد را به ترتیب در ابتدای خط محور اصلی گروه بندی می کند.
* flex-end: Groups the items at the end of the main axis.
* • : موارد را در انتهای محور اصلی گروه بندی می کند.
* center: Groups the items at the very center.
* • مرکز: موارد را در مرکز گروه بندی می کند.
* space-between: Distributes the items along the main axis, with the remaining space placed evenly *between* the items.
* • فضای اطراف: موارد را در امتداد محور اصلی توزیع می کند و فضای باقیمانده به طور مساوی در دو طرف هر مورد قرار می گیرد.
* space-around: Distributes the items along the main axis, with the remaining space placed evenly *to the sides* of each item.

فاصله بین: موارد را در امتداد محور اصلی توزیع می کند و فضای باقیمانده به طور مساوی بین موارد قرار می گیرد.

* space-evenly: Distributes the items along the main axis, with the remaining space placed evenly*around* the items.
* • فضای یکنواخت: موارد را در امتداد محور اصلی توزیع می‌کند و فضای باقی‌مانده به‌طور مساوی در اطراف آیتم‌ها قرار می‌گیرد.

Justify-content controls how items are **spaced out** from one another in the container.

In this illustration, the darker background box represents the flex-parent.

And a code example:

In the above example, try the different possible values for justify-content for the .flex-start-parent selector. Pay attention to how the children are redistributed across the main axis.

•

Justify-content نحوه فاصله گرفتن آیتم ها از یکدیگر در ظرف را کنترل می کند.

در این تصویر، کادر تیره‌تر پس‌زمینه نشان‌دهنده والد انعطاف‌پذیر است.

و یک مثال کد:

در مثال بالا، مقادیر مختلف ممکن را برای justify-content برای انتخابگر .flex-start-parent امتحان کنید. به نحوه توزیع مجدد کودکان در محور اصلی توجه کنید.

**Margin: Auto**

حاشیه: خودکار

When margin: auto is applied to one side of a flex-child on the main axis, it serves to push that item and its neighbor as far from one another as possible.

وقتی حاشیه: خودکار به یک سمت فلکس-کودک در محور اصلی اعمال می‌شود، این امر باعث می‌شود که آن مورد و همسایه‌اش تا حد امکان از یکدیگر دور شوند.

This allows an additional element of control along the main axis in addition to what justify-content allows.

این اجازه می دهد تا یک عنصر کنترل اضافی در امتداد محور اصلی علاوه بر آنچه که justify-content اجازه می دهد.

Notice in the following example how the main axis is defined as justify-content: space-between, but because the middle item has margin-left: auto, it is pushed all the way to the right and sits right next to the last item:

در مثال زیر توجه کنید که چگونه محور اصلی به‌عنوان justify-content تعریف می‌شود: فاصله-بین، اما چون آیتم میانی دارای حاشیه-چپ: خودکار است، تا آخر به سمت راست فشار داده می‌شود و درست در کنار آخرین مورد قرار می‌گیرد:

Because margin: auto in flexbox is very powerful in the layout, try to only use it if **no other method** is able to achieve the desired result.

In the above example, try changing margin-left to margin-right for the .example div.

از آنجایی که margin: auto در flexbox در چیدمان بسیار قدرتمند است، سعی کنید تنها در صورتی از آن استفاده کنید که هیچ روش دیگری قادر به رسیدن به نتیجه مطلوب نیست.

در مثال بالا، سعی کنید margin-left را به margin-right برای .example div تغییر دهید.

If you want to go further, try moving the margin to other selectors to see what happens.

اگر می خواهید جلوتر بروید، سعی کنید حاشیه را به انتخابگرهای دیگر منتقل کنید تا ببینید چه اتفاقی می افتد.

**Align Items**

By default, flex-children are stretched so that all are equally sized along the *cross axis* (at least the containing box is, so a background-color fills that whole space). The position of the item along the cross axis is controlled using the align-items property on the flex parent. Possible values are:

* stretch: The default value, which grows the item to fill the available space on the cross axis.
* flex-start: Aligns the items to the start of the cross axis line.
* flex-end: Aligns the items to the end of the cross axis line.
* center: Center aligns the items on the cross axis.
* baseline: Aligns the baselines of the first line of text in each item.

Align-items controls how items **line up** with one another.

In this illustration, the darker background box represents the flex-parent.

In the above example, try the different possible values for align-items for the .stretch-parent selector. Pay attention to how the children are realigned on the cross axis.

تراز کردن موارد

به‌طور پیش‌فرض، کودکان انعطاف‌پذیر به‌گونه‌ای کشیده می‌شوند که اندازه همه آنها در امتداد محور متقاطع به یک اندازه باشد (حداقل جعبه حاوی آن است، بنابراین یک رنگ پس‌زمینه کل فضا را پر می‌کند). موقعیت آیتم در امتداد محور متقاطع با استفاده از ویژگی align-item در والد flex کنترل می شود. مقادیر ممکن عبارتند از:

• stretch: مقدار پیش فرض، که مورد را رشد می دهد تا فضای موجود در محور متقاطع را پر کند.

• شروع انعطاف پذیر: موارد را با شروع خط محور متقاطع تراز می کند.

• : موارد را در انتهای خط محور متقاطع تراز می کند.

• مرکز: وسط موارد را روی محور متقاطع تراز می کند.

• خط پایه: خطوط پایه خط اول متن را در هر مورد تراز می کند.

Align-Items نحوه چینش اقلام با یکدیگر را کنترل می کند.

در این تصویر، کادر تیره‌تر پس‌زمینه نشان‌دهنده والد انعطاف‌پذیر است.

در مثال بالا، مقادیر مختلف ممکن را برای align-ites برای انتخابگر .stretch-parent امتحان کنید. به نحوه تراز مجدد کودکان در محور متقاطع توجه کنید.

**Video Lesson**

The following video demo is fully interactive. You can pause it at any time to directly edit the code, and resume playback to pickup where you left off.

**Practice Exercise**

**Resource Links**

* [Flexbox on MDN](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)
* [A Complete Guide to Flexbox](https://css-tricks.com/snippets/css/a-guide-to-flexbox/)
* [An Animated Guide to Flexbox](https://medium.freecodecamp.org/an-animated-guide-to-flexbox-d280cf6afc35)
* [Flexbox Video Course](https://www.youtube.com/playlist?list=PLu8EoSxDXHP7xj_y6NIAhy0wuCd4uVdid)
* [Everything You Need to Know About Flexbox](https://medium.freecodecamp.org/understanding-flexbox-everything-you-need-to-know-b4013d4dc9af)
* [Solved by Flexbox (demos)](https://philipwalton.github.io/solved-by-flexbox/)
* [Flexbox Froggy (a learning game)](https://flexboxfroggy.com/)

1. Previous Chapter:

[30: Transform](https://htmlandcssguidebook.com/css/transforms/)

1. Next Chapter:

[32: Grid](https://htmlandcssguidebook.com/css/grid/)

[htmlandcssguidebook.com](https://htmlandcssguidebook.com/css/flexbox/)

درس تصویری

نسخه ی نمایشی ویدیوی زیر کاملاً تعاملی است. می‌توانید در هر زمان آن را متوقف کنید تا مستقیماً کد را ویرایش کنید، و پخش را از جایی که متوقف کردید ادامه دهید.

تمرین تمرین

پیوندهای منابع

• Flexbox در MDN

• راهنمای کامل Flexbox

• راهنمای متحرک برای Flexbox

• دوره ویدیویی Flexbox

• هر آنچه که باید درباره Flexbox بدانید

• حل شده توسط Flexbox (دمو)

• Flexbox Froggy (یک بازی آموزشی)

1. فصل قبل:

30: تبدیل

2. فصل بعد:

32: شبکه

htmlandcssguidebook.com