**Flex BOX**

Table of Contents

\*Getting Started

\*Introduction

\*Flex-container properties

\*Flex-item Properties

\*Switching Flex Direction

\*Responsive Design

\*Conclusion

**What is flexbox?**

**CSS** **Flexbox** is essentially a layout module. There are existing layout modes within **CSS**, and they've been there for a long while. One example of a layout mode is block (e.g. display: block).

**Why it came to existence?**

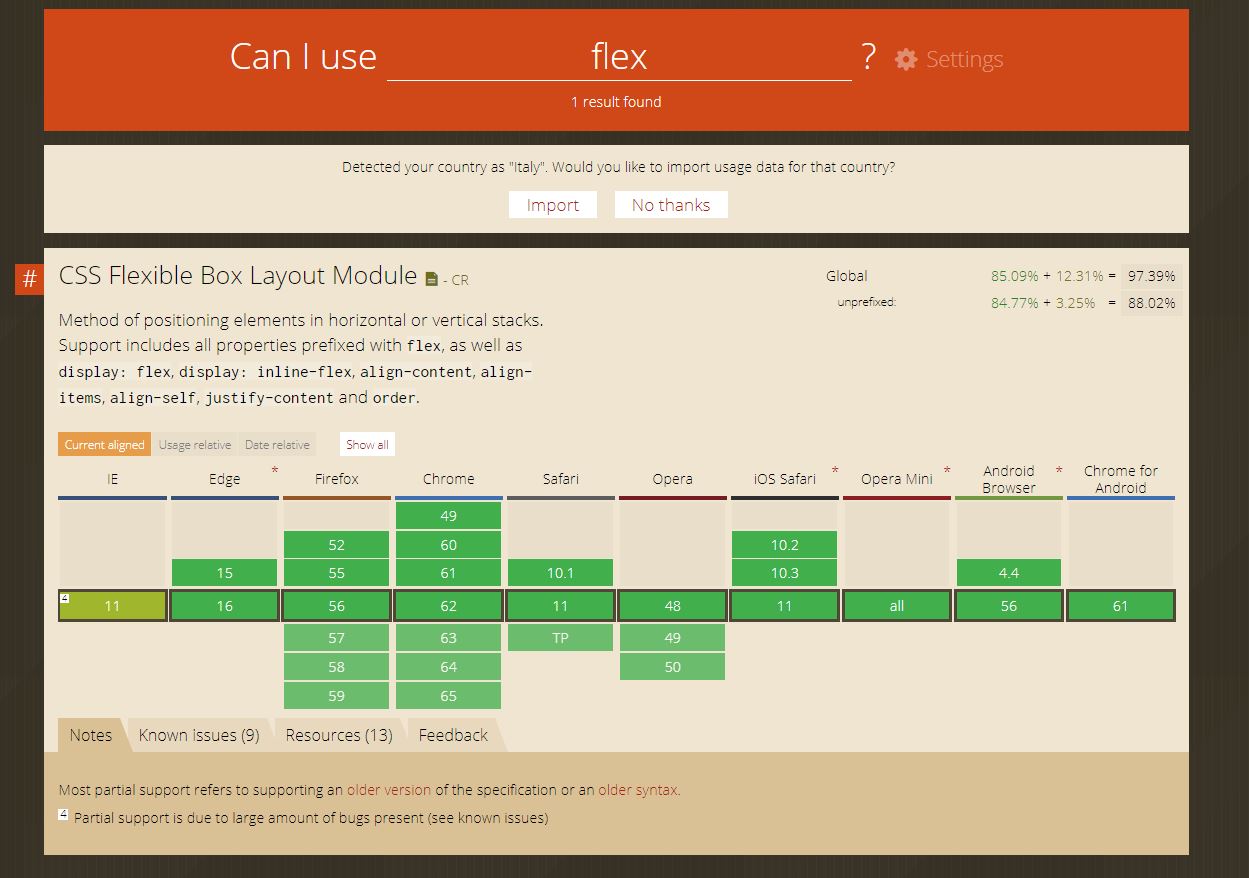
Before the Flexbox Layout module, there were four layout modes:

* Block, for sections in a webpage
* Inline, for text
* Table, for two-dimensional table data
* Positioned, for explicit position of an element
* Use positional properties (absolute)
* Floats and clear fixes
* Fixed heights for columns

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

**Why use flex box?**

Here are 6 reasons you should start using flexbox

1. **It’s supported in all major browsers:** Flexbox is very well supported, at 95.89% global support.****
2. **It can be simple to get started**

3 properties -

* **display** - This sets the element as an inline or block flexbox container element.
* **justify-content** - This controls the horizontal alignment of items within the flex container (If the flex-direction is the default value of row or row-reverse).
* **align-items** - controls the vertical alignment of items within the flex container.

1. **You can finally centre elements:**

The ease in which we can centre elements, both horizontally and vertically.

.flex-container {

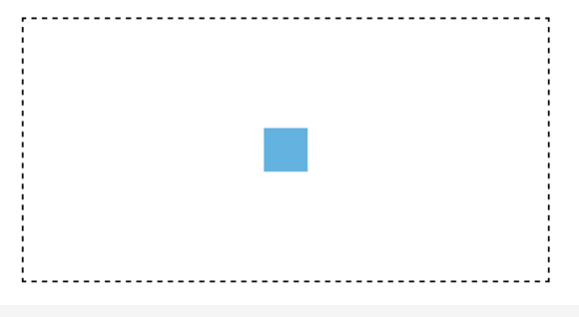
display: flex;

justify-content: center; /\* horizontal centering \*/

align-items: center; /\* vertical centering \*/

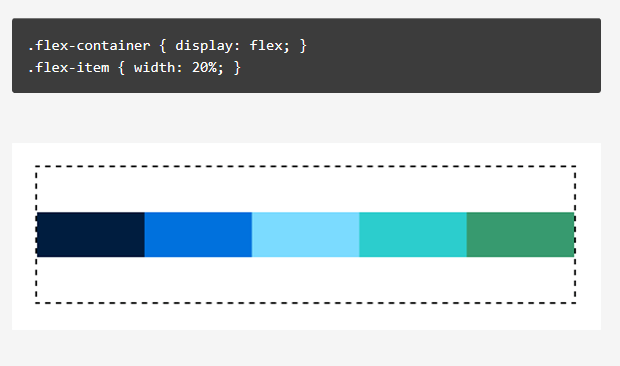
border: 2px dashed #000;

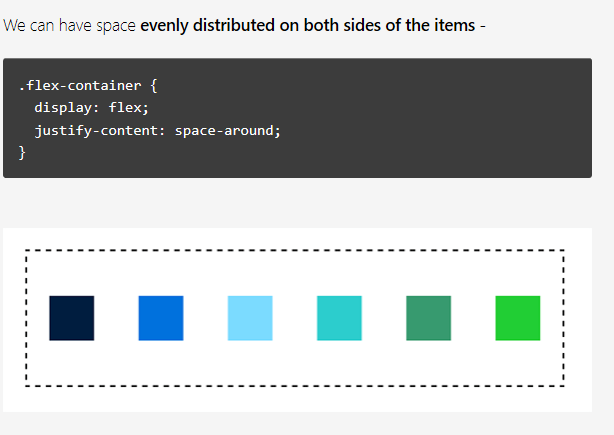
}



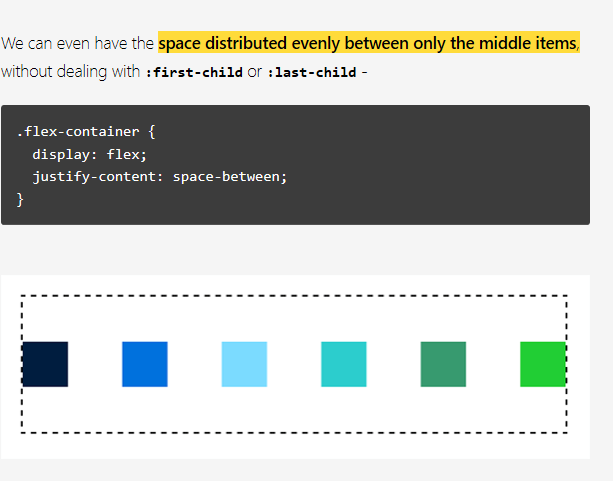
1. **You can manipulate inline elements easily:**

One issue with having inline items positioned beside each other is the [infamous extra 4px margin](https://css-tricks.com/fighting-the-space-between-inline-block-elements/). Although there are ways to get around this, like floating the elements, that comes with its own problems as well.



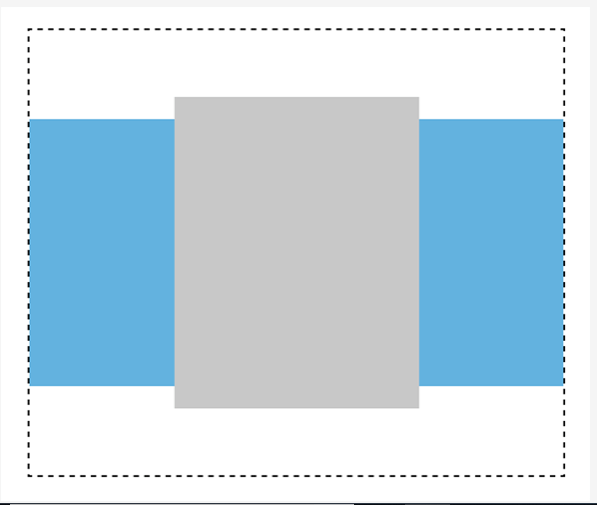


**space distributed evenly between only the middle items**



1. **It simplifies complexity**

The reason flexbox was created in the first place was for this very reason, to allow us to achieve the complex layouts we already create, in as little as one declaration.



How do I start using the Flexbox model?

To start using the Flexbox model, all you need to do is first define a flex-container.

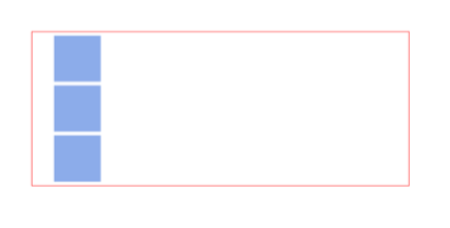
<ul> <!--parent element-->

<li></li> <!--first child element-->

<li></li> <!--second child element-->

<li></li> <!--third child element-->

</ul>



initiating a Flexbox formating context:

To use the Flexbox model, you must make a parent element a flex container (AKA flexible container).

setting display: flex or display: inline-flex for the inline variation

/\*Make parent element a flex container\*/

ul {

display: flex; /\*or inline-flex\*/

}

li {

width: 100px;

height: 100px;

background-color: #8cacea;

margin: 8px;

}



two key words

Flex container : The parent element you’ve set display: flex on.

Flex items : The children elements within a Flex container.