

# Introduction to Flexbox

Theory

Practice

0% completed, 0 problems solved

## Theory

🕒 17 minutes reading

Verify to skip

Start practicing

For a long time, there were only several simple CSS properties like `float`, `margin`, `position`, `display` and other similar tools for setting up a proper layout. Those still work and you may still use them, but in some ways, it's frustrating, time-consuming, and most importantly, very limiting.

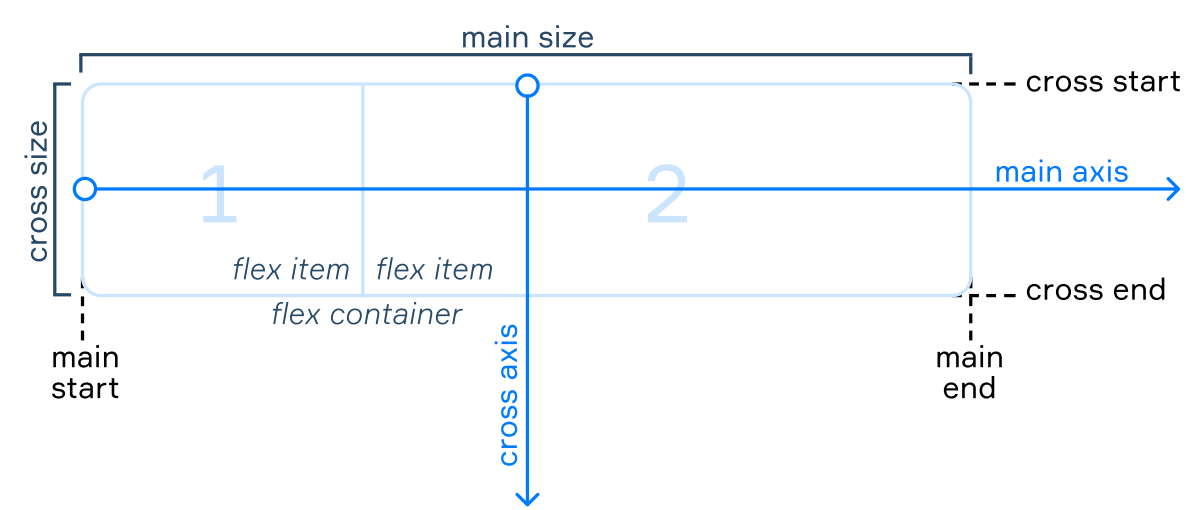
Everything becomes much easier with **flexbox**! **Flexbox** is short for **Flexible Box Layout Module**. It is a one-dimensional layout method for laying out items in rows or columns. Items flex to fill additional space and shrink to fit into smaller spaces. It allows us to align elements vertically and horizontally or change their display direction and order using only several properties.

### §1. Basic terms

Flexbox consists of a **flex container** and **flex items**. Flex container is a wrapping for flex items. Flex items are its children and they can line up in a row or a column. The remaining free space is distributed among the items in a way determined by the corresponding property.

With a flexbox you can line up elements in four different directions: left to right, right to left, top to bottom, and bottom to top. You can also override the elements' display order, automatically set up their sizes so that they fit in the available space, change the positioning of items inside the container, and finally solve the problem with horizontal and vertical alignment that all the programmers are dealing with.

Now let's go over some important terms that will help you understand flexbox better:



- **Main axis** is the axis that determines the direction in which all the elements are going to be placed by default.
- **Cross axis** is the axis perpendicular to the main axis.
- **Main start** and **main end** are borders that determine where the flex container begins and ends.
- **Cross start** and **cross end** are borders that determine where the cross axis begins and ends.
- **Main size** is the size of the main axis.
- **Cross size** is the size of the cross axis.

Depending on how you want to align flex items, the dimensions may shuffle, meaning that the main axis will become the cross axis and the cross axis will become the main axis. This affects the way items are arranged: vertically or horizontally.

### 5 required topics

✓ 

Layout grids

 ▾

✓ 

Clear

 ▾

✓ 

Margin and Padding

 ▾

✓ 

Width and Height

 ▾

✓ 

CSS Selectors

 ▾

### 1 dependent topic

Orientation and Display order

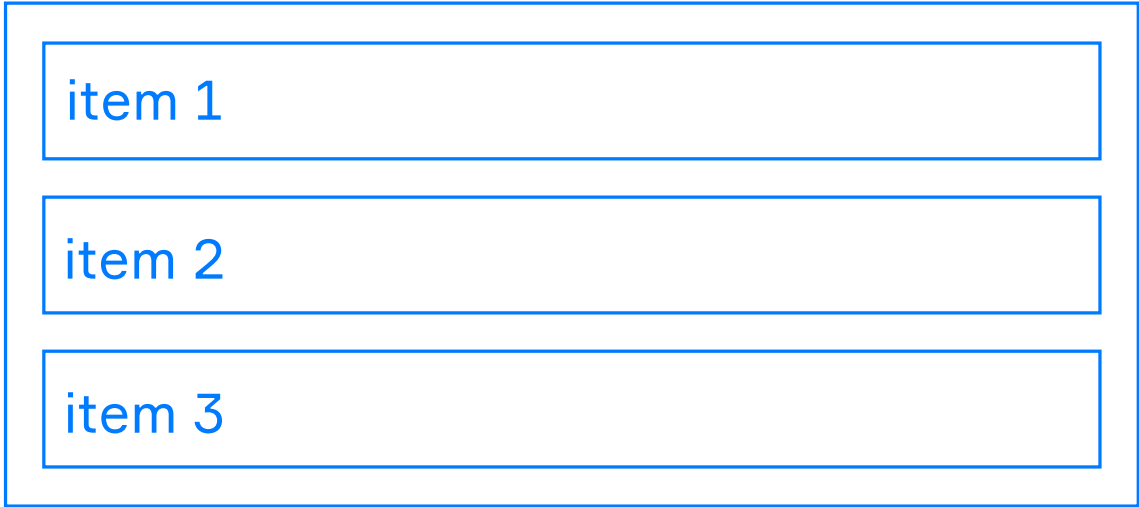
 ▾

## §2. Setting up a flexbox

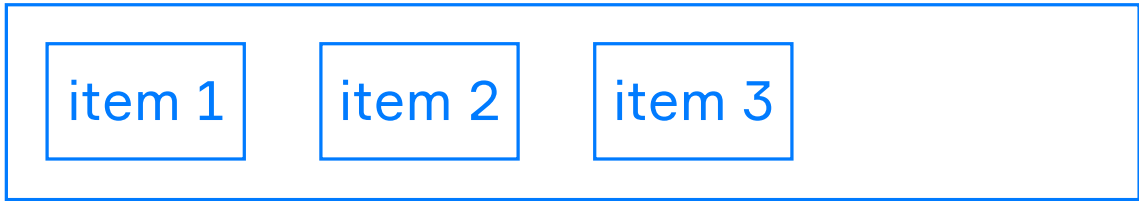
To set up a flexbox, you have to create a block and set its property `display: flex;`. After that, all the child elements of this block will automatically become **flex items** lined up along the direction of the main axis. Here's an example of a flex container with 3 child elements:

```
<div class="flex-container">
  <div class="flex-item">item 1</div>
  <div class="flex-item">item 2</div>
  <div class="flex-item">item 3</div>
</div>
```

This is what it looks like with `display: block;` set to the flex-container:



And this is what happens if `display: flex;` is applied:



If the block has a text or an image not wrapped with `<div></div>`, they become **anonymous flex items**. In this case, the text will be stuck to the top, and the height of the image will be equal to the height of the container. If that sounds undesirable, just don't forget to wrap everything into `<div></div>` !

There are a couple more things to remember about flexboxes:

- Properties `display`, `float`, `clear`, and `vertical-align` are redundant for flexbox, so they will be ignored even if you set them.
- You don't need to set the `display: flex;` property for flexible elements, just apply it to the container that is the parent element.
- If the values of `margin` and `padding` are set as a percentage, they will be calculated based on the inner size of the parental block.
- The default minimum size of the flex item is the minimal size of its content. Basically, it's `min-width: auto;`. The minimum size of the elements with visible overflow is 0 by default.

### Table of contents:

- [↑ Introduction to Flexbox](#)
- [§1. Basic terms](#)
- [§2. Setting up a flexbox](#)
- [§3. Summary](#)
- [Discussion](#)

## §3. Summary

Flexbox module is a blessing for all web developers. It is a simple structure that consists of only two types of blocks: a parent and its children. Child elements line up along the main or cross axis, and you can choose the axis and its direction. There is more to know about flexbox: this will be covered in the upcoming topics.

Report a typo

Thanks for your feedback!

Start practicing

Verify to skip

