

Master CSS Flexbox in just 5 minutes

Invite the Administrator to learn the article of mastering CSS Flexbox in just 5 minutes below!

1. Top 5 popular CSS Framework that you should keep in mind
2. 5 free online HTML editing tools that test the best code
3. 13 skills needed to become Frontend Developer

In this article, you will learn the basics of **CSS Flexbox** , which is a must-have skill for website developers and designers in the last few years. Invite the Administrator to learn the article of **mastering CSS Flexbox in just 5 minutes** below!



We will use navbar as an example, because this is a typical case for Flexbox. This will introduce you to the most used properties of the module, while eliminating unimportant features.

First Flexbox layout

The two main components of the Flexbox layout are the container and the **items** .

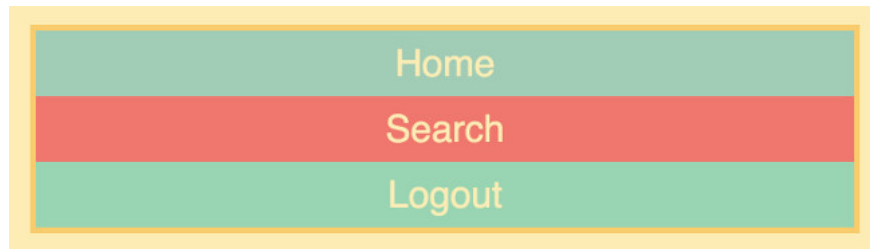
This is the HTML for our example, containing a container with three items:

Home

Search

Logout

Before turning them into Flexbox layouts, the elements will be stacked as follows:



I created a little style for it, but that doesn't affect Flexbox.

To turn it into a Flexbox layout, we just need to supply the following CSS attribute container:

```
.container {  
  display: flex;  
}
```

This will automatically sort items according to the horizontal axis.



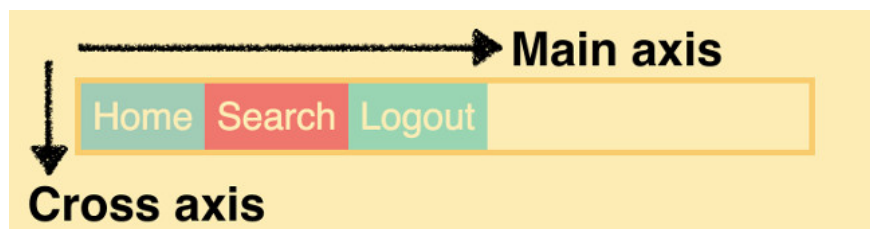
If you want to check the actual code, you can access the Scrimba playground (<https://scrimba.com/c/c3zpnuB>).

Now rearrange these items a bit.

Justify-content and Align-item

Justify-content and **align-item** are two CSS properties that help us distribute items in the container. They control how items are positioned along the **main axis** and **cross axis**.

In this case (not always), the horizontal main axis and vertical cross axis:



In this article, we only look at `justify-content`, as I see it being used more than `align-items`. However, in the upcoming course, I will pass both specific properties.

Focus all the items along the **main axis** using `justify-content`:

```
.container {  
  display: flex;  
  justify-content: center;  
}
```



Or maybe set it to `space-between`, there will be more space between items like:

```
.container {  
  display: flex;  
  justify-content: space-between;  
}
```

Here are the values you can set for `justify-content`:

1. `flex-start` (**default**)
2. `flex-end`
3. `center`
4. `space-between`
5. `space-around`
6. `space-evenly`

You should learn about these values and see how they show up on the page. That will help you understand the concept correctly.

Single item control

We can control individual items. Suppose we want to keep the first two items on the left and move the `logout` button to the right.

To do this, we will use the old technique to set the `auto` margin.

```
.logout {  
  margin-left: auto;  
}
```



If you want both the `search` and `logout` items to be pushed to the right, simply add `margin-left` instead of the `search` entry.

```
.search {  
  margin-left: auto;  
}
```

It will push the `search` item to the right, again you will push the `logout` item as shown:



Flex properties

Up to now, we have only fixed-width entries. But what if we want to display responsive? To do this, we need to have an attribute called `flex`. This makes it much easier than the old way of using percentages.

Simply aim all the items and give them a `flex` value of 1.

```
.container > div {  
  flex: 1;  
}
```



As you can see, it extends the items to fill the entire container.

In many cases, you may want to select one of the items to open more widths and therefore, only put one if they have flexible width. For example, we can make the `search` item have more space:

```
.search {  
  flex: 1;  
}
```



Before the end of the article, I want to tell you that flex properties are an abbreviated feature from three attributes: **flex-grow**, **flex-shrink** and **flex-basis**.

Author: Per Harald Borgen

Refer to some more articles:

1. Beginners of computer programming need to focus on what?
2. 9 reasons you should equip a little knowledge of HTML and CSS
3. Do you know the 15 hottest programming languages ??on this GitHub?

Having fun!

You finished reading the article "**Master CSS Flexbox in just 5 minutes**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

