

Flexbox Basics > How To Guide

Here is quick overview of -

- How flexbox works

- 4 Steps to follow to setup using flexbox

- Examples of containers and inside elements

- Annotated Example layouts

- Other flexbox properties to explore

Flexbox Basics > flex-direction

Flexbox allows you to group **elements** inside a **container** and position those elements **horizontally** or **vertically** -

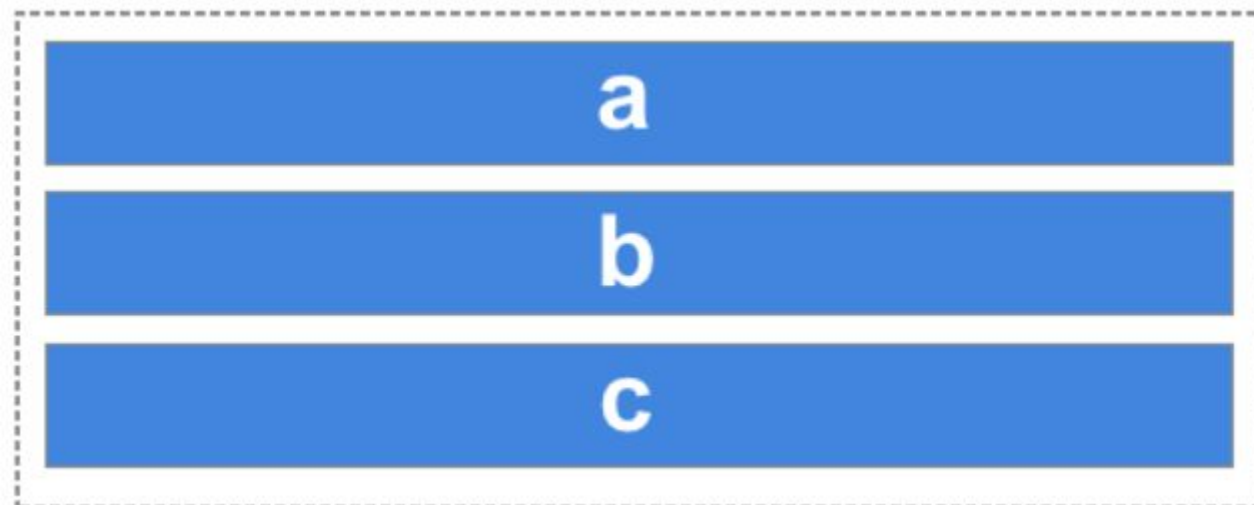
Horizontal -

`flex-direction: row;`



Vertical -

`flex-direction: column;`



F
L
E
X

C
O
N
T
A
I
N
E
R

Flexbox Basics > 4 Setup Steps

To setup flexbox you need to -



1. Define flexbox “container”

HTML: `<div> ...`

CSS: `display: flex;`

2. Set the “direction” of the flexbox “container”

CSS: `flex-direction: row;`

← horizontal

`flex-direction: column;`

← vertical

3. Define the elements “inside” the flexbox container

HTML: `<div> <section> <main> <article> <aside> ...`

4. Specify the dimensions of the “inside” elements

CSS: `width: ____px;`
`____%;`

`flex-grow: ____;`

Flexbox Basics > Setup Flex Container

1 & 2. Set up the flexbox “container”

CSS: set `display` and `flex-direction` properties

```
.flex-container {  
  display: flex;  
  flex-direction: row | column;  
}
```

HTML: usually a `<div>` but can be other elements

```
<div class="flex-container"> ... </div>
```

Flexbox Basics > Setup Inside Elements

3 & 4. Set up the “inside” elements width

CSS: the “inside” elements can be many different elements

.box | section | main | article | aside | ... {

width: 100px; ← fixed width

or

width: 50%; ← fluid width

or

flex-grow: __; ← a # of proportional units

}

Flexbox Basics > Putting it Together

A flex container with 3 generic content boxes:



CSS:

```
.flex-container {  
  display: flex;  
  flex-direction: row;  
}
```

```
.box {  
  width: 100px;  
  height: 100px;  
}
```

HTML:

```
<div class="flex-container">  
  <div class="box">a</div>  
  <div class="box">b</div>  
  <div class="box">c</div>  
</div>
```

Flexbox Basics > Examples

A semantic element as a flex-container with other semantic elements inside

```
<div class="container">
  <article>...</article>
  <article>...</article>
  <article>...</article>
</div>
```

A semantic element as a flex-container with other semantic elements inside

```
<main class="container">
  <article>...</article>
  <aside>...</aside>
</main>
```

A flex-container with semantic inside elements

```
<div class="container">
  <aside>...</aside>
  <main>...</main>
  <aside>...</aside>
</div>
```

A flex-container with semantic inside elements

```
<div class="container">
  <section>...</section>
  <section>...</section>
  <section>...</section>
</div>
```

Flexbox Basics > Example Layout

```
<?> ← <div> flex-direction: row;
```

```
<?> ←  
<aside>  
<section>  
<div>  
width: 33%;  
flex-grow: 1;
```

```
<?> ←  
<main>  
<section>  
<div>  
width: 33%;  
flex-grow: 1;
```

```
<?> ←  
<aside>  
<section>  
<div>  
width: 33%;  
flex-grow: 1;
```

Flexbox Basics > Example Layout

```
<?> ← <div> flex-direction: row;
```

```
<?> ←  
<main>  
<section>  
<div>  
width: 75%;  
flex-grow: 3;
```

```
<?> ←  
<aside>  
<section>  
<div>  
width: 25%;  
flex-grow: 1;
```

Flexbox Basics > Example Layout

```
<?> ← <div> flex-direction: column;
```

```
<?> ← <div> <section> <article>  
width: ____; | flex-grow: ____;
```

```
<?> ← <div> <section> <article>  
width: ____; | flex-grow: ____;
```

```
<?> ← <div> <section> <article>  
width: ____; | flex-grow: ____;
```

Flexbox Basics > Flex Properties

Other **properties** that position elements inside the **.flex-container** -

```
.flex-container {  
    display: flex;  
    flex-direction: row;  
  
    flex-wrap: nowrap | wrap;  
    justify-content: center | space-around |  
                    space-between;  
    align-items: center | stretch;  
    align-content: center;  
}
```

Flexbox > Tips

- Use your browser's **inspector** to troubleshoot CSS issues.
- **Resize** your browser window to verify layout for different sized screens.
- If your layout isn't working, **check** that you've done each flexbox step.
- **The TAs are here to help you learn.**
The TAs will not fix your code for you.
The TAs will help you learn how to fix your own code.
The TAs will ask you about each step when helping you.

Flexbox > Resources

You are Web Programmers. Be sure to be reading the docs!

[MDN Flexbox Overview](#)

[MDN Flexbox Guide](#)

[A Complete Guide to Flexbox](#)

[Flexbox Cheatsheet](#)

[Interactive Guide to Flexbox](#)

[Flexbox30: Learn Flexbox with 30 Code Tibits](#)