

1. Learn CSS Flexbox with Martin 😊

CSS Flexbox is a powerful layout model designed to make it easy to align, distribute, and space elements in a container. It's particularly useful for creating responsive designs.

2. Key Concepts

Flex Container: The parent element that uses `display: flex`. It controls the layout of its children (the flex items).

Flex Items: The direct children of the flex container, which are aligned and spaced according to the rules set on the container.

Here is a basic example of flexbox layout

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Flexbox Example</title>
</head>
<body>

  <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>

</body>
</html>
```

CSS Flexbox Properties

1. display: flex;

This enables flexbox on the container, making its direct children flex items.

2. flex-direction

Defines the direction of the flex items (row or column).

Example:

```
flex-direction: row;    /* Default, aligns items in a row */
flex-direction: column; /* Aligns items in a column */
```

3. justify-content

Controls how items are aligned horizontally within the flex container.

Example:

```
justify-content: flex-start; /* Aligns items to the start (left) */
justify-content: flex-end;   /* Aligns items to the end (right) */
justify-content: center;     /* Centers the items horizontally */
justify-content: space-between; /* Equal space between items */
justify-content: space-around; /* Equal space around each item */
```

4. align-items

Controls how items are aligned vertically within the flex container.

Example:

```
align-items: flex-start; /* Aligns items at the top */
align-items: flex-end;   /* Aligns items at the bottom */
align-items: center;     /* Centers items vertically */
align-items: stretch;   /* Stretches items to fill the container height */
```

5. flex-grow, flex-shrink, and flex-basis

Control how flex items grow, shrink, and set their base size.

Example:

```
flex-grow: 1; /* Item will grow to fill the available space */
flex-shrink: 1; /* Item will shrink if necessary */
flex-basis: 100px; /* Defines the initial size of the item */
```

Summary

Flexbox Cheat Sheet

`flex-direction`: Defines the direction of flex items (row, column).

`justify-content`: Aligns items horizontally (start, end, center, space-between, space-around).

`align-items`: Aligns items vertically (start, end, center, stretch).

`flex-wrap`: Allows items to wrap onto multiple lines if necessary.

`flex-grow`: Specifies how much an item should grow relative to the rest.

`flex-basis`: Sets the initial size of the item.

I used the above CSS Flexbox properties to create the

project below. Feel free to use it as a learning example

Here is my html code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="flexbox.css">

</head>
<body>
  <nav class="nav">
    

    <div id="nav-list">
      <div id="Features">
        <p>Features</p>
        </div>

        <div id="Company">
          <p>Company</p>
          </div>

          <p>Careers</p>
          <p>About</p>

        </div>
      <div>
        <p>Login</p>
        <button id="register">Register</button>

      </div>
    </nav>

    <main class="main">
      

      <div>
        <p id="h1"><b>Make<br>remote work</b></p>
        <p id="paragraph" style="font-size: large;">Get your team in sync, no matter your locat:
          Sreamline processes, create team rituals and <br>
          watch productivity soar.</p>
      </div>
    </main>
  </body>
</html>
```

```
<button id="learnmore" style="font-size: large;">Learn more</button>
</div>
```

```
<div id="footer">
  
  
  
  
</div>
```

```
</main>
```

```
</body>
```

```
</html>
```

And below is the styles I used for the project:

```
*{
  margin: 0;
  padding: 0;
  text-decoration: none;
}
body{

  display: flex;
  height: 100vh;
  flex-direction: column;
}
.nav{
  background-color: whitesmoke;
  height: 10%;
  display: flex;
  flex-direction: row;
  align-items: center;
  width: 100w;
}
#nav-list{
  margin-left: 100px;
  display: flex;
  flex-direction: row;
  justify-content: center;
  align-items: center;
  list-style: none;
  gap: 20px;

}

.nav div{
  display: flex;
  gap: 25px;
  margin-left:auto;
  margin-right: 50px;
  font-size: x-large;

}
.logo{
```

```
    height: 30px;
    object-fit: contain;
    margin: 35px;
    margin-right: 20px;
}
#register{
    padding: 3px;
    border-radius: 8px;
    width: 120px;

}
#image{
    height: 600px;
    float: right;
    margin-top: 50px;
    margin-right: 300px;
}
.main div{
    display: block;
    justify-content: flex-start;
    align-items: center;
    margin-left: 200px;
    position: relative;
    margin-top: 150px;
}
#learnmore{
    padding: 3px;
    border-radius: 10px;
    color: white;
    background-color: black;
    margin-top: 20px;
    width: 140px;
    height: 50px;
}
#footer{
    display: flex;
    flex-direction: row;
    align-self: stretch;
}
#f-images{
    margin-right: 25px;
}
#h1{
```



```

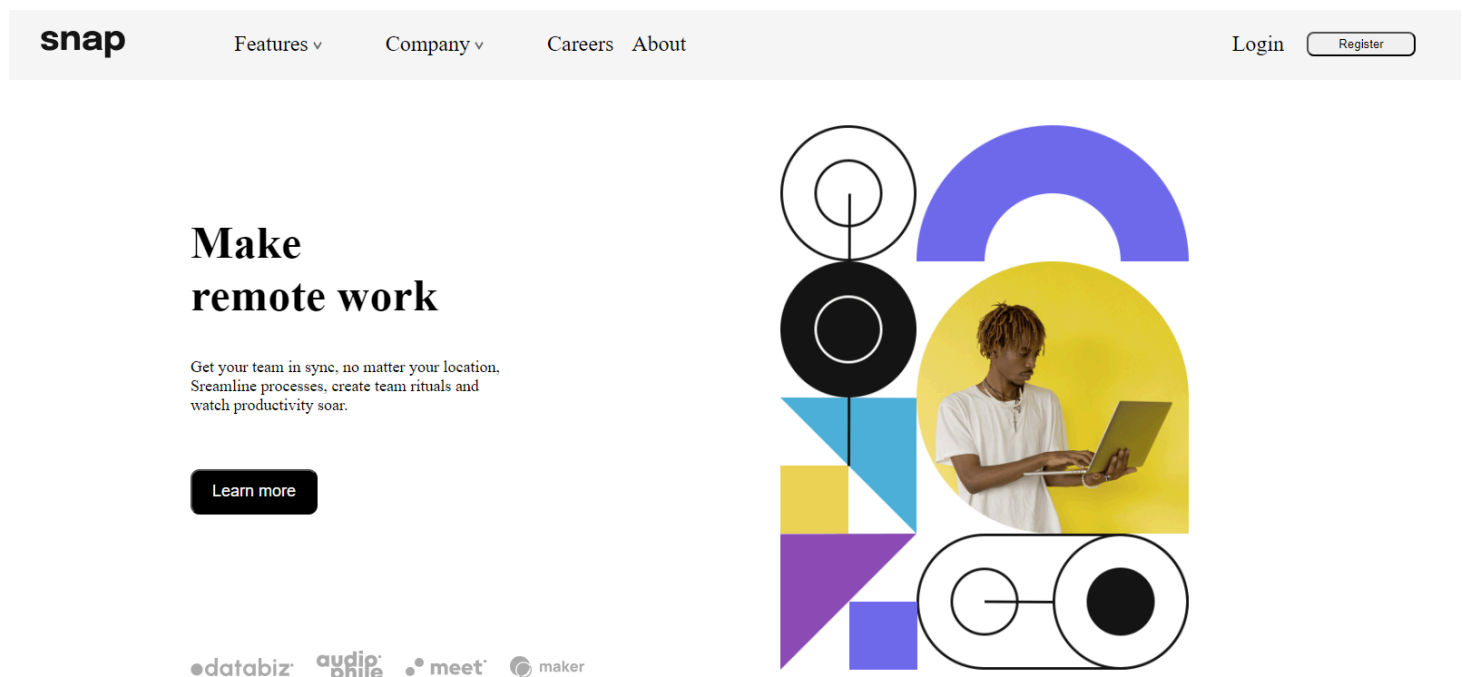
    font-size: 50px;
  }
#paragraph{
    font-size: larger;
    margin: 40px 0px;
  }
#list{
    font-size: x-large;
  }
#dropdown{

    height: 10px;
    width: 10px;
    margin-top: 10px;

  }
#Features,#Company{
    gap: 5px;
  }
}

```

The result for this project was as in the picture below. Go on and try out the code :



Find the resources used for this project here:

<https://www.frontendmentor.io/challenges/intro-section-with-dropdown-navigation-ryaPetHE5>