

# Flex Sampler

Flexbox CSS rules allow you to dynamically arrange web elements but can be tricky to figure out. Flex Sampler allows you to play around with flexbox rules and see the results in real time.

## Flexbox

Remember to Add "display:flex" to the parent element's css

flex-direction: row

flex-wrap: nowrap

justify-content: flex-start

align-items: flex-start

align-content: flex-start (Only has effect while wrapping)

order: (Anon) 0 (Ronald) 0

flex-grow: (Anon) 0 (Ronald) 0

flex-shrink: (Anon) 0 (Ronald) 0

flex-basis: (Anon) 20em auto (Ronald) 20em auto

align-self: (Anon) auto (Ronald) auto

reset Add Anon Add Ronald (Click to remove)



## Flexbox

Remember to Add "display:flex" to the parent element's css

flex-direction: column

flex-wrap: nowrap

justify-content: flex-start

align-items: center

align-content: flex-start (Only has effect while wrapping)

order: (Anon) 0 (Ronald) -1

flex-grow: (Anon) 0 (Ronald) 0

flex-shrink: (Anon) 0 (Ronald) 0

flex-basis: (Anon) 20em auto (Ronald) 20em auto

align-self: (Anon) auto (Ronald) stretch

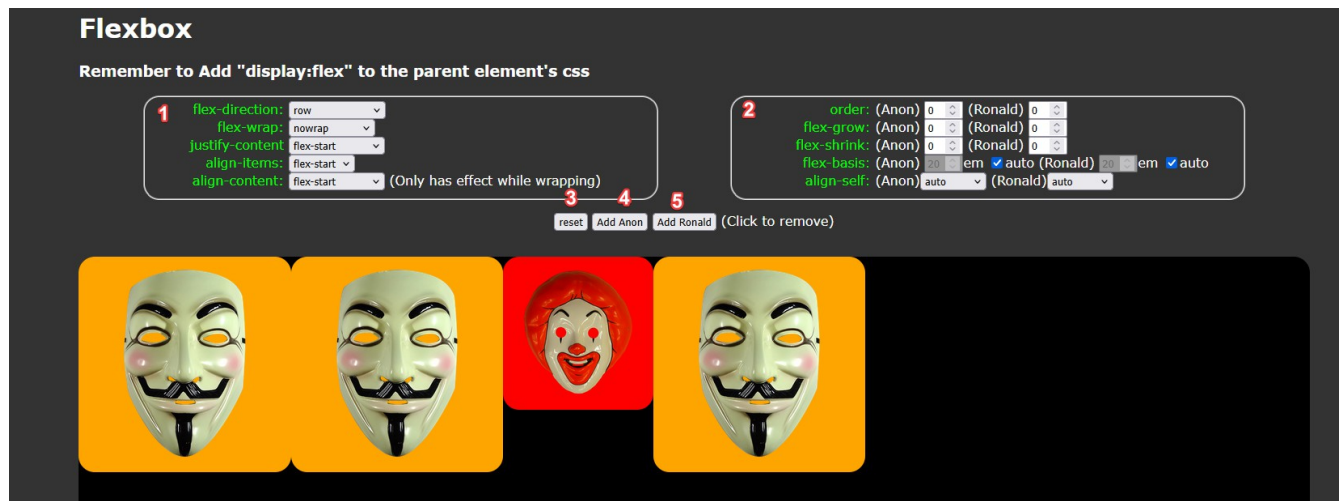
reset Add Anon Add Ronald (Click to remove)



Cyan boxes highlight changed fields.

# How to Use

The controls are as follows



1: CSS rules for all child elements. Play around with these to effect the layout of all elements in the black box.

2: CSS rules targeting specific classes of elements. These target either the Guy Fawkes masks or the Ronald Masks by their class names.

3: resets all flexbox rules back to defaults and changes the contents of the black box back to 3 Guy Fawkes masks and 1 Ronald mask.

4: Adds an additional Guy Fawkes mask (referencing hacker group "Anonymous"). You may click these elements to delete them.

5: Adds an additional Ronald McDonald mask. You may click these elements to delete them.

Play around with these until you get a layout that you want to incorporate into your own front-end web project.

# How to Incorporate

For this example: here is the code for a simple web page.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>

    #squareHolder
    {
      background-color:black;
    }

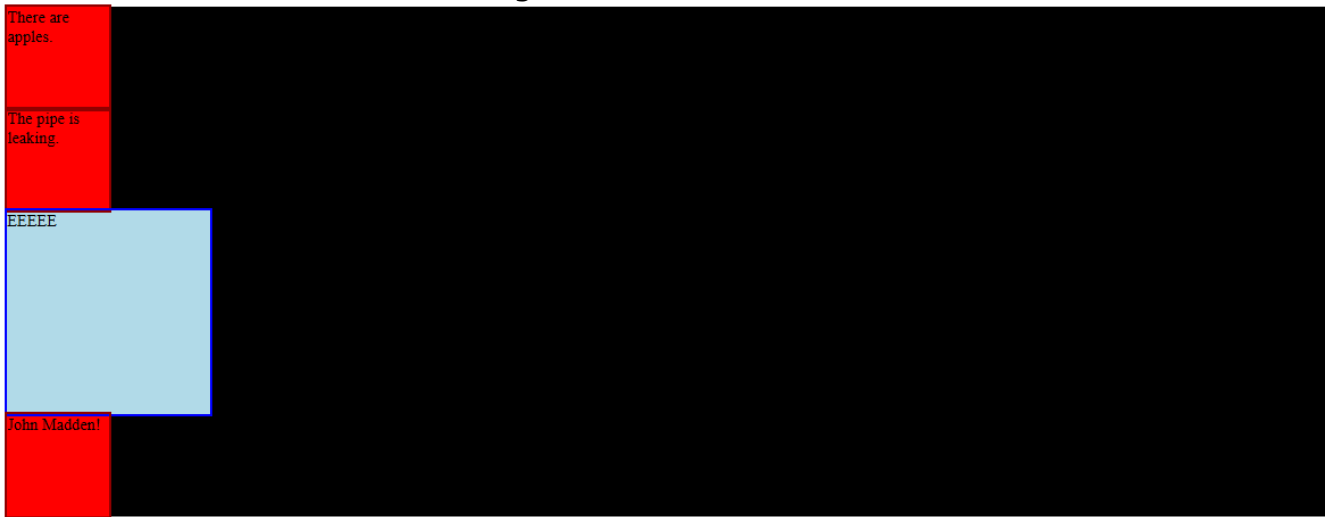
    .redsquare
    {
      width:100px;
      height:100px;
      background-color: red;
      outline:3px solid darkred;
    }

    .bluesquare
    {
      width:200px;
      height:200px;
      background-color: lightblue;
      outline: 3px solid blue;
    }

  </style>
</head>
<body>
  <div id = "squareHolder">
    <div class = 'redsquare'>There are apples.</div>
    <div class = 'redsquare'>The pipe is leaking.</div>
    <div class = 'bluesquare'>EEEEEE</div>
    <div class = 'redsquare'>John Madden!</div>
  </div>
</body>
</html>
```

*Normally you would have the CSS as separate file. CSS is embedded here for simplicity.*

If you copy and paste this into notepad, save it as index.html, and open it in your browser it should look something like this:



*Width may vary depending on your window size.*

To get your web elements ready for flexbox layout add “display:flex;” to the parent element.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>

    #squareHolder
    {
      background-color:black;
      display:flex;
    }

    .redsquare
    {
      width:100px;
      height:100px;
      background-color: red;
      outline:3px solid darkred;
    }

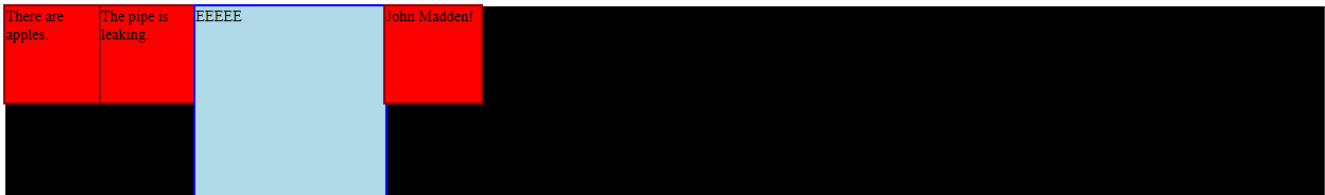
    .bluesquare
    {
      width:200px;
      height:200px;
      background-color: lightblue;
      outline: 3px solid blue;
    }
  </style>
</head>
<body>
  <div id="squareHolder">
    <div class="redsquare">There are apples.</div>
    <div class="redsquare">The pipe is leaking.</div>
    <div class="redsquare">John Madden!</div>
    <div class="bluesquare">EEEE</div>
  </div>
</body>
</html>
```

```

</style>
</head>
<body>
  <div id = "squareHolder"> This is the targeted element.
    <div class = 'redsquare'>There are apples.</div>
    <div class = 'redsquare'>The pipe is leaking.</div>
    <div class = 'bluesquare'>EEEEEE</div>
    <div class = 'redsquare'>John Madden!</div>
  </div>
</body>
</html>

```

If you refresh your page you may notice your elements are already rearranged. Specifically you may find that block elements are now arranged in a row.



Play around with Flex Sampler until you get a layout you like.

## Flexbox

Remember to Add "display:flex" to the parent element's css

flex-direction: column

flex-wrap: nowrap

justify-content: flex-start

align-items: center

align-content: flex-start (Only has effect while wrapping)

order: (Anon) 0 (Ronald) -1

flex-grow: (Anon) 0 (Ronald) 0

flex-shrink: (Anon) 0 (Ronald) 0

flex-basis: (Anon) 200px em auto (Ronald) 200px em auto

align-self: (Anon) auto (Ronald) stretch

reset Add Anon Add Ronald (Click to remove)

Any field you changed in the right box (flex-direction, flex-wrap...) write into the CSS of the parent element. The text in bright green is the exact text you need to write. You do not need to write in elements you didn't change as these are defaults.

## Remember to Add "display:flex" to the parent element's css

flex-direction:	column	▼
flex-wrap:	nowrap	▼
justify-content	flex-start	▼
align-items:	center	▼
align-content:	flex-start	▼

(Only has effect while wrapping)

reset

Add Anon

Add

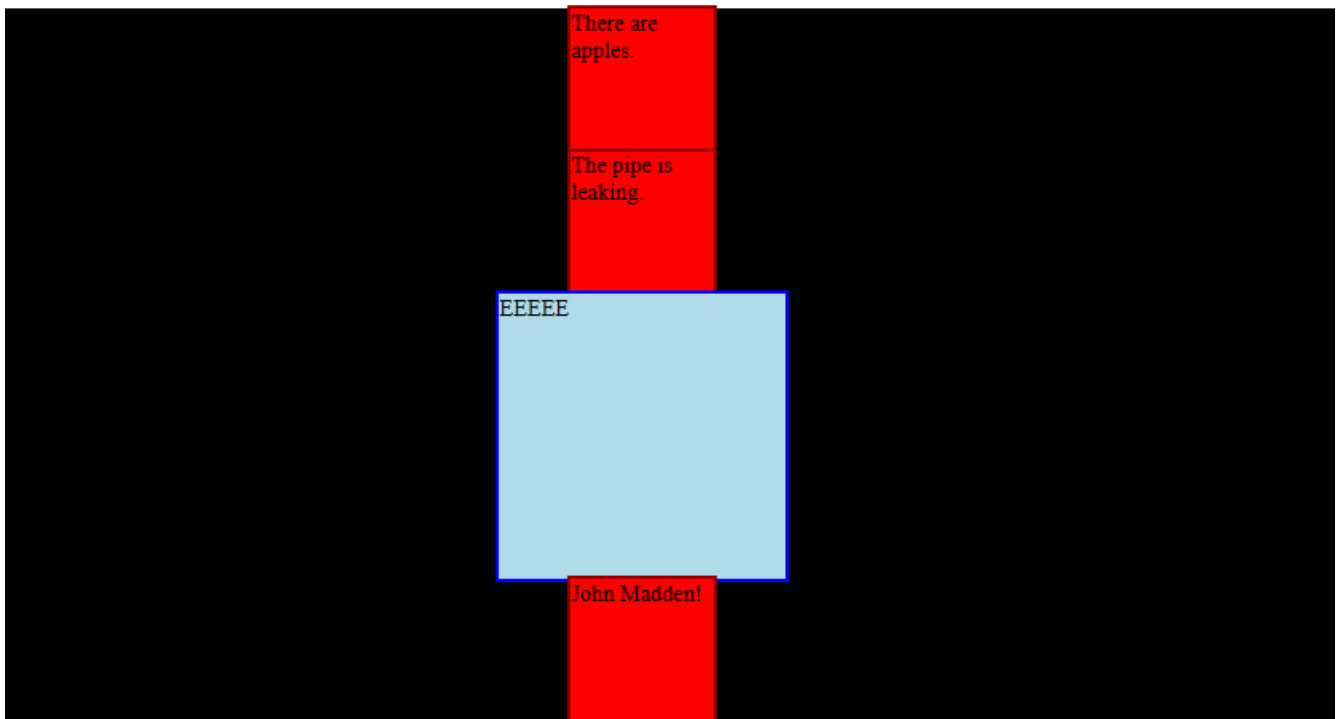
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>

    #squareHolder
    {
      background-color:black;
      display:flex;
      flex-direction: column;
      align-items: center;
    }

    .redsquare
    {
      width:100px;
      height:100px;
      background-color: red;
      outline:3px solid darkred;
    }

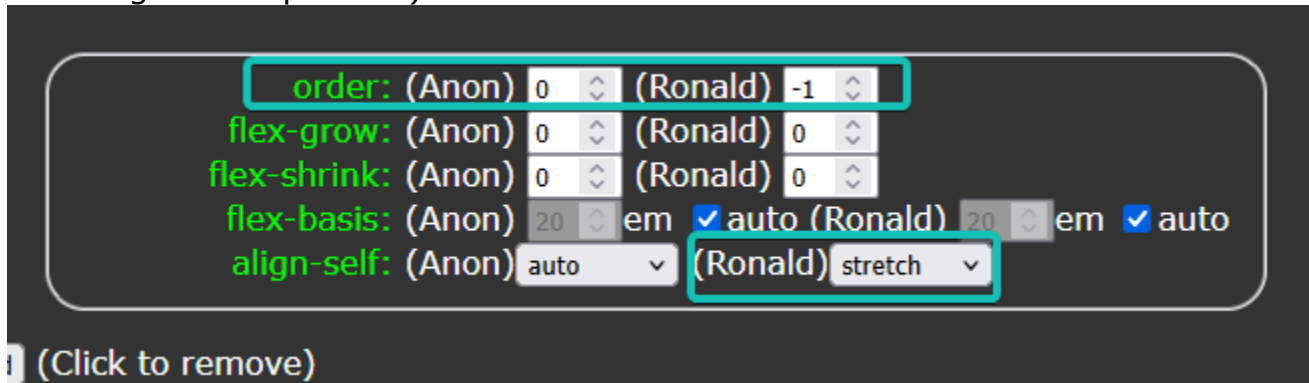
    .bluesquare
    {
      width:200px;
      height:200px;
      background-color: lightblue;
      outline: 3px solid blue;
    }
  </style>
</head>
<body>
  <div id="squareHolder">
    <div class="redsquare"></div>
    <div class="bluesquare"></div>
  </div>
</body>
</html>
```

```
</style>
</head>
<body>
  <div id = "squareHolder">
    <div class = 'redsquare'>There are apples.</div>
    <div class = 'redsquare'>The pipe is leaking.</div>
    <div class = 'bluesquare'>EEEEEE</div>
    <div class = 'redsquare'>John Madden!</div>
  </div>
</body>
</html>
```



*Not done yet*

Any changed field you made in the left box (order, flex-grow...) add directly to the CSS rule that targets types of elements (classes) or individual elements (id, according to best practice).



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>

    #squareHolder
    {
      background-color:black;
      display:flex;
      flex-direction: column;
      align-items: center;
    }

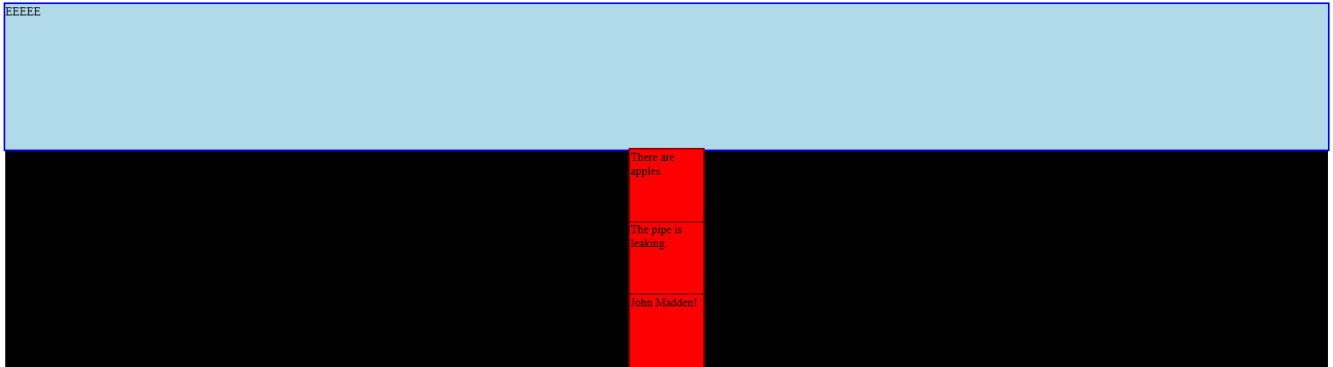
    .redsquare
    {
      width:100px;
      height:100px;
      background-color: red;
      outline:3px solid darkred;
    }

    .bluesquare
    {
      width:200px; This needs to be removed for stretch to work. Some troubleshooting may
be necessary.
      height:200px;
      background-color: lightblue;
      outline: 3px solid blue;
      order: -1;
      align-self: stretch;
    }

  </style>
```



```
</head>
<body>
  <div id = "squareHolder">
    <div class = 'redsquare'>There are apples.</div>
    <div class = 'redsquare'>The pipe is leaking.</div>
    <div class = 'bluesquare'>EEEEEE</div> This is the targeted element by it's type/class.
    <div class = 'redsquare'>John Madden!</div>
  </div>
</body>
</html>
```



Written by David Schriver

March 24, 2025

<https://www.linkedin.com/in/david-schrive-162083178/>