Some confusing CSS styles

```
body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
```

display: flex;

CSS Flexbox (Flexible Box)

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects like HTML, CSS, JavaScript, Python, SQL, Java, and many, many more.
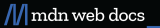
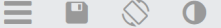W³ https://www.w3schools.com/css/css3_flexbox.asp

Basic concepts of flexbox - CSS: Cascading Style Sheets | MDN

The flexible box layout module, usually referred to as flexbox, was designed as a one-dimensional layout model, and as a method that could offer space distribution between items in an interface and powerful

M https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flexible_box_layout/Basic_concepts_of_flexbox

Result Size: 625 x 391   Get your own website

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>Create a Flex Container</h1>
```
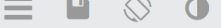
Create a Flex Container

| 1 | 2 | 3 |

A Flexible Layout must have a parent element with the *display* property set to *flex*.

Direct child elements(s) of the flexible container automatically becomes flexible items.

Result Size: 625 x 391   Get your own website

```
.flex-container {
  /*display: flex;*/
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>Create a Flex Container</h1>

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

Create a Flex Container

1

2

3

A Flexible Layout must have a parent element with the *display* property set to *flex*.
Direct child elements(s) of the flexible container automatically becomes flexible items.
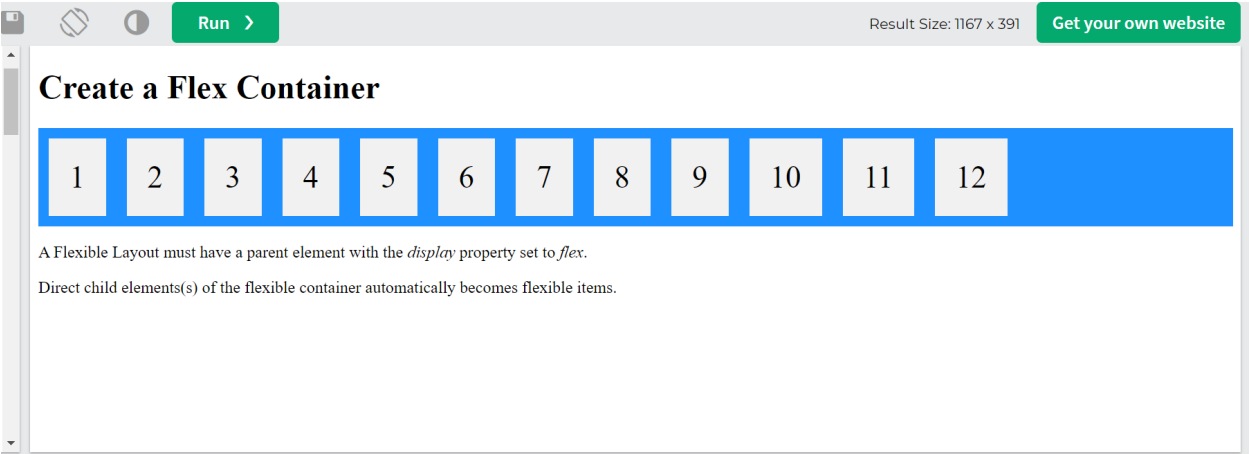
💡 The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

Can be used as a replacement of float.
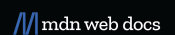
If add more divs:
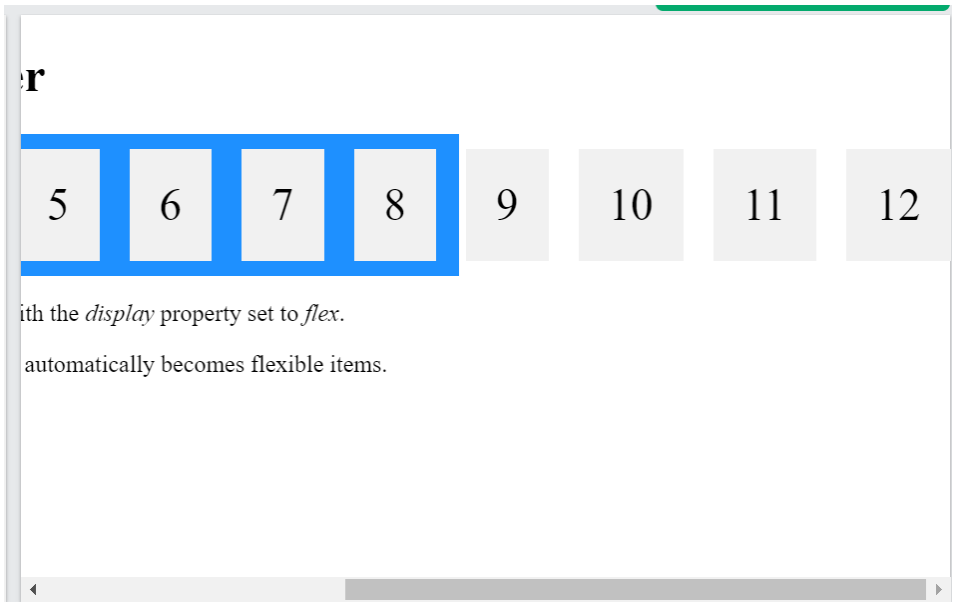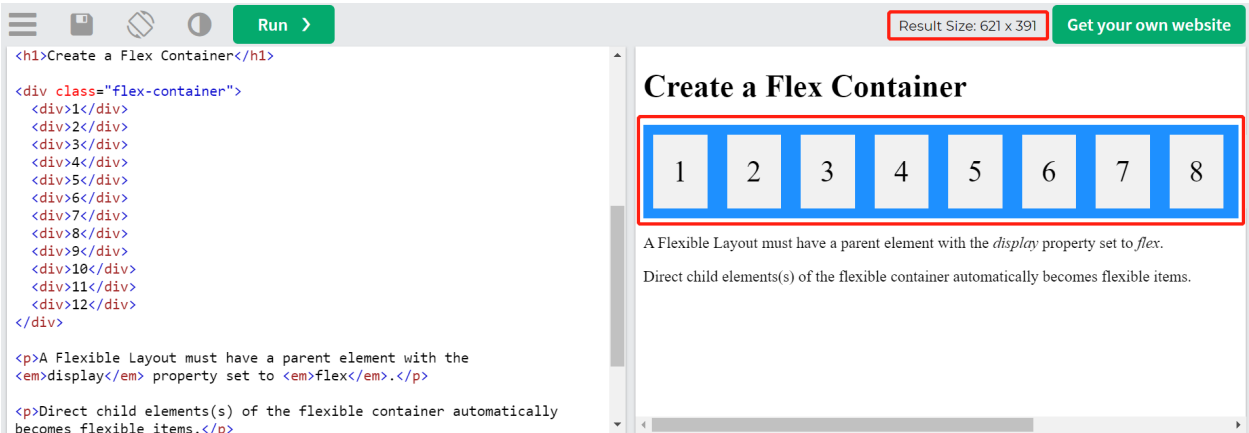
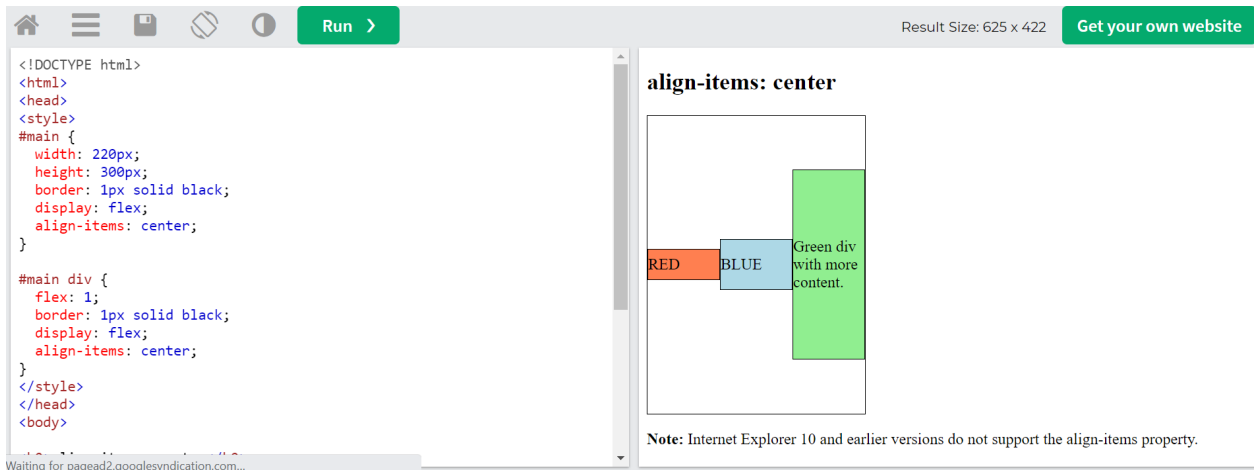any divs outside will not share the blue background.

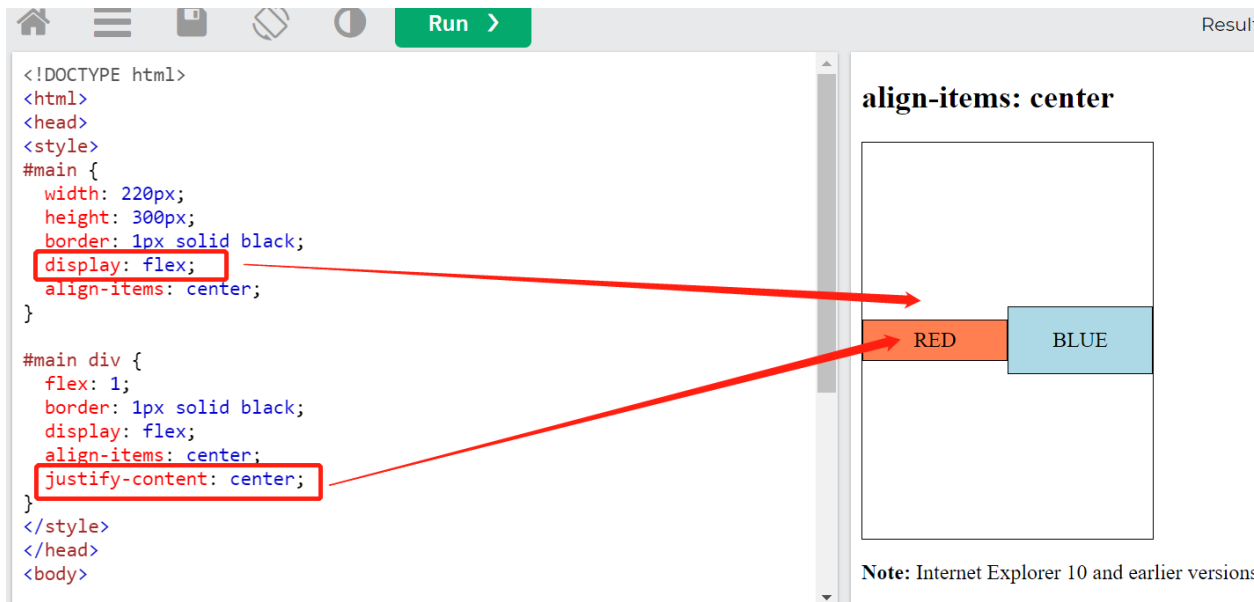If increase the size of the window, the blue background will also grow wider as well.

Let's simply named it as a quite flexible display method (based on the width of the window), take all the space.
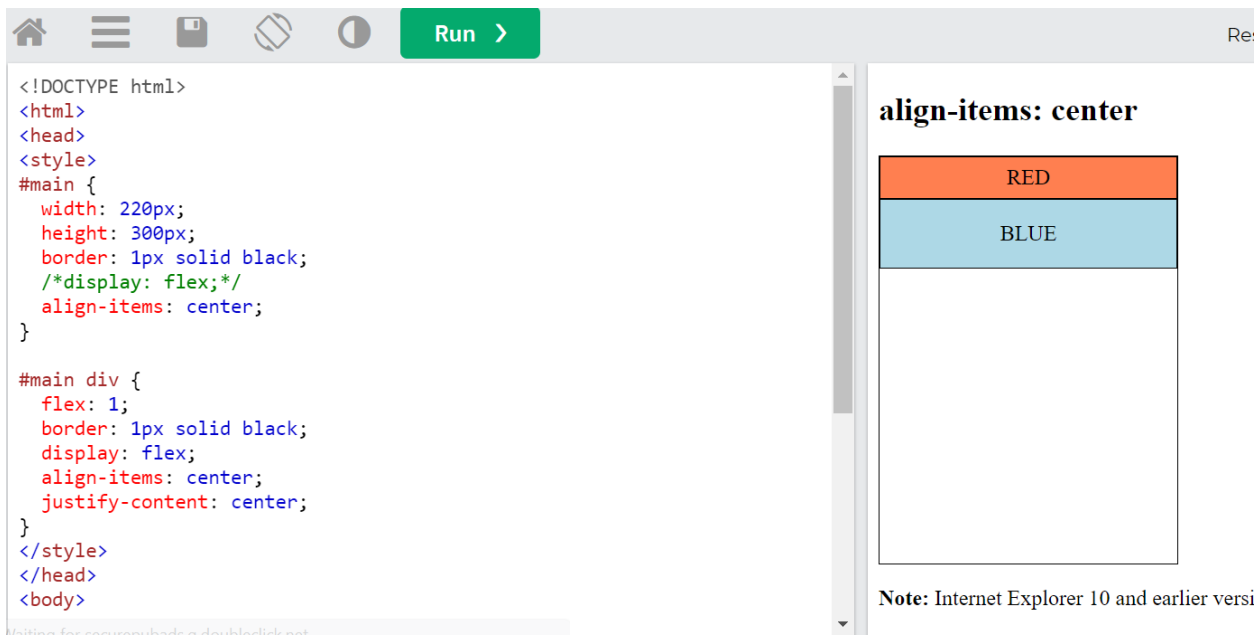
justify-content: center;

The `justify-content` property aligns the flexible container's items when the items do not use all available space on the main-axis (horizontally).

Tip: Use the `align-items` property to align the items vertically.

align-items: center;

```html
<!DOCTYPE html>
<html>
<head>
<style>
#main {
  width: 220px;
  height: 300px;
  border: 1px solid black;
  display: flex;
  align-items: center;
}

#main div {
  flex: 1;
  border: 1px solid black;
  display: flex;
  align-items: center;
}
</style>
</head>
<body>
```

Here we can also test display:flex. and justify-content: center;


```html
<!DOCTYPE html>
<html>
<head>
<style>
#main {
  width: 220px;
  height: 300px;
  border: 1px solid black;
  display: flex;
  align-items: center;
}

#main div {
  flex: 1;
  border: 1px solid black;
  display: flex;
  align-items: center;
  justify-content: center;
}
</style>
</head>
<body>
```

Even delete the green div, other divs will also take all the space.

justify-content: center make words align horizontally.

Without flex:


```html
<!DOCTYPE html>
<html>
<head>
<style>
#main {
  width: 220px;
  height: 300px;
  border: 1px solid black;
  /*display: flex;*/
  align-items: center;
}

#main div {
  flex: 1;
  border: 1px solid black;
  display: flex;
  align-items: center;
  justify-content: center;
}
</style>
</head>
<body>
```

height: 100vh;

100vh

`height: 100vh` = 100% of the viewport height

`height: 100%` = 100% of the parent's element height

That is why you need to add `height: 100%` on `html` and `body` , as they don't have a size by default

Something you have to know : if you use % for vertical margin or padding, % will be calculated on the **width** of the parent element, not the height.

100em

1em is **equal** to the **current font size**.

2em means 2 times the size of the current font.

E.g., if an element is displayed with a font of 12 pt, then '2em' is 24 pt. The 'em' is a very useful unit in CSS, since it can adapt automatically to the font that the reader uses

100px

100%

fixed

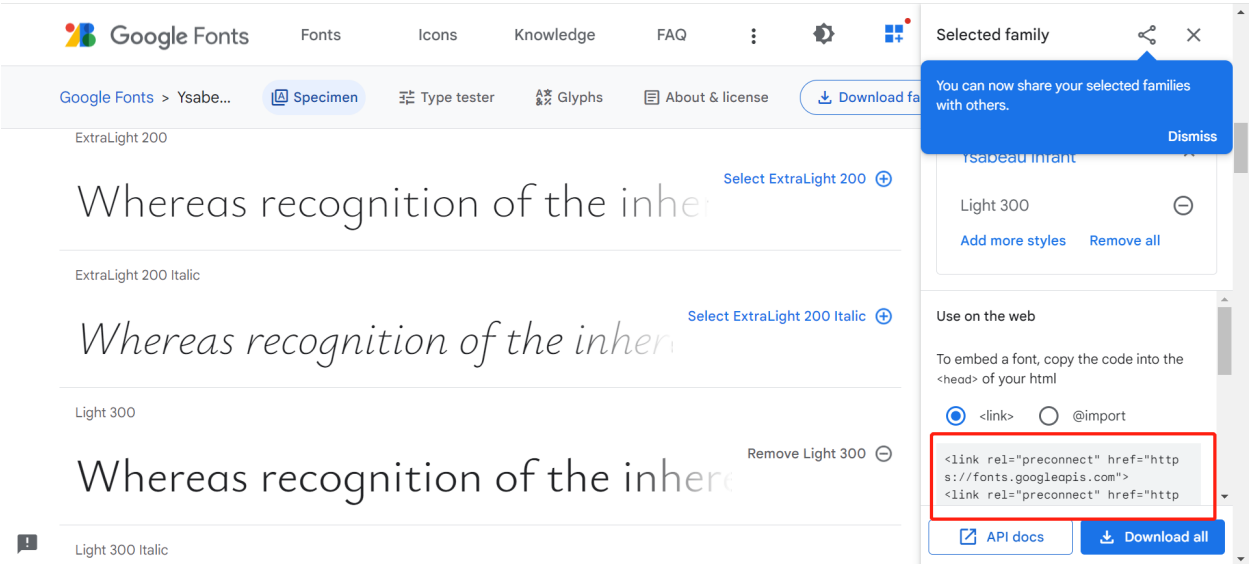# font style

Browse Fonts - Google Fonts

Making the web more beautiful, fast, and open through great typography

https://fonts.google.com/
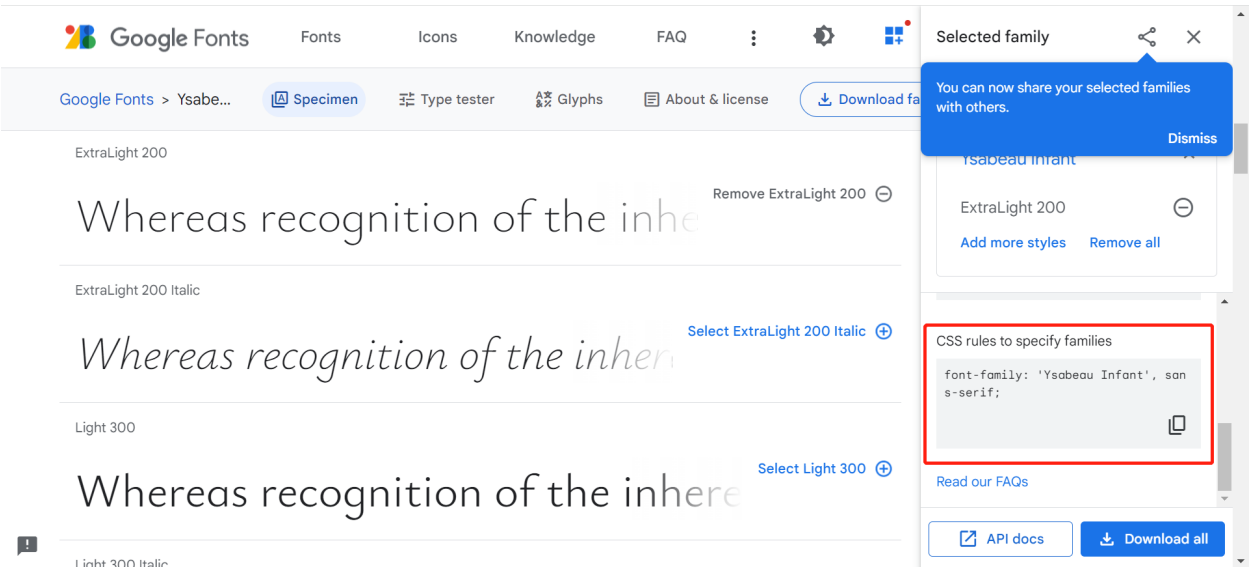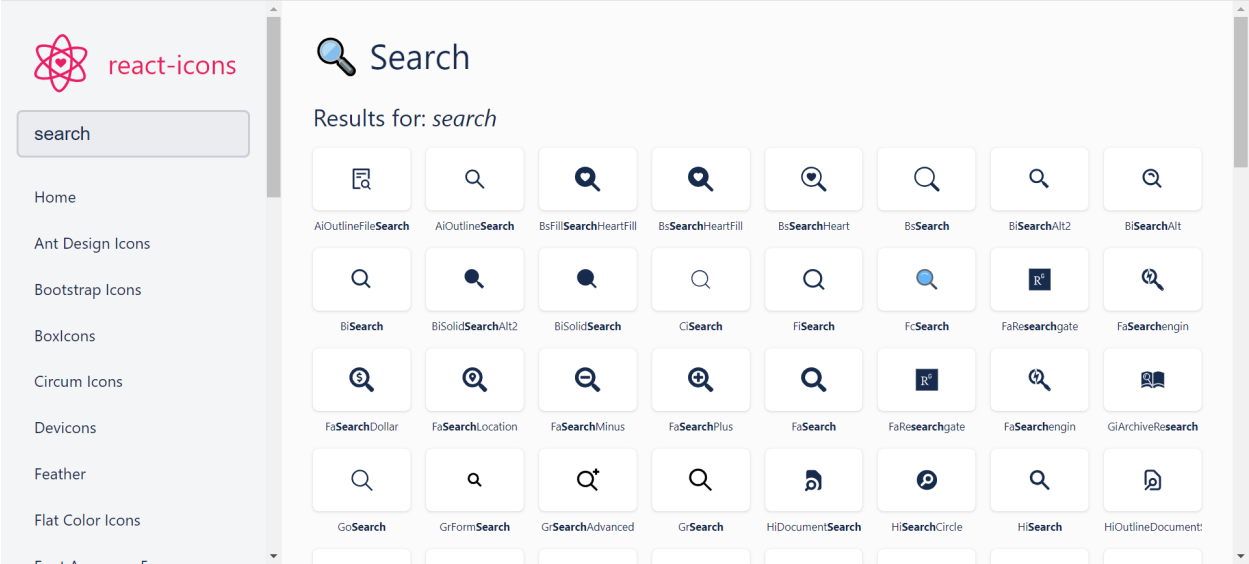

add↑link to html

add↓font to css

react Icons

React Icons

Include popular icons in your React projects easly with react-icons.

🔬 https://react-icons.github.io/react-icons/

search icons and pick what you like

backgroud image

```
background-image: url("https://source.unsplash.com/1600x900/?nature,water");
```

```
background-image: url("https://source.unsplash.com/1600x900/?landscape");
```

Also, we can control parameters in css:

```
document.body.style.backgroundImage =
"url('https://source.unsplash.com/1600x900/?" + name + "')"
```