



VIETNAM
AUSTRALIA
Vocational College

Bài 08 - Thiết kế Website theo bố cục Flexbox



Nội dung


1. Tổng quan về bố cục Flexbox
2. Các thuộc tính của Flex Container
3. Các thuộc tính của Flex Item
4. Canh chỉnh giao diện với Auto Margin
5. Xây dựng bố cục Music App với Flexbox
6. Thiết kế Responsive với Flexbox



Tổng quan về bố cục Flexbox

❑ Tại sao chúng ta nên sử dụng Flexbox?

- ♦ Tạo bố cục thông minh cho một trang web với CSS đã **tồn tại khá lâu với việc sử dụng một số thủ thuật** với **table**, thuộc tính **float**,...để bố cục cách hiển thị phù hợp.
- ♦ Nhưng **float**, **table** cũng **đi cùng khá nhiều lỗi phát sinh ngoài ý muốn** mặc dù chúng ta đã rất **cẩn thận quản lý**, **bố cục của chúng** -> Có cách nào tốt hơn không?
- ♦ Giải pháp của vấn đề trên sẽ **được giải quyết với** một **cú pháp hiện đại, gọn gàng hơn với** mô hình bố cục **CSS Flexbox**.



CSS FLEXBOX

Complex flexbox example

First article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Second article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Smile Laugh Wink Shrug

Blush

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat ennui. Organic small batch paleo 8-bit. Intelligent umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed squid chartreuse put a bird on it. Listic pickled man bun cornhole heirloom art party.



Tổng quan về bố cục Flexbox

❑ Flexbox là gì?

♦ **Bố cục Flexbox** (gọi ngắn gọn là **Flex**): Là **một phương pháp hiệu quả để bố cục, canh chỉnh và phân phối không gian giữa các phần tử trong trang web ngay cả khi viewport và size của các phần tử của bạn không xác định và/hoặc động.**

Complex flexbox example

First article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Second article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Smile

Laugh

Wink

Shrug

Blush

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat ennui. Organic small batch paleo 8-bit. Intelligentsia umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed squid chartreuse put a bird on it. Listicle pickled man bun cornhole heirloom art party.



Tổng quan về bố cục Flexbox

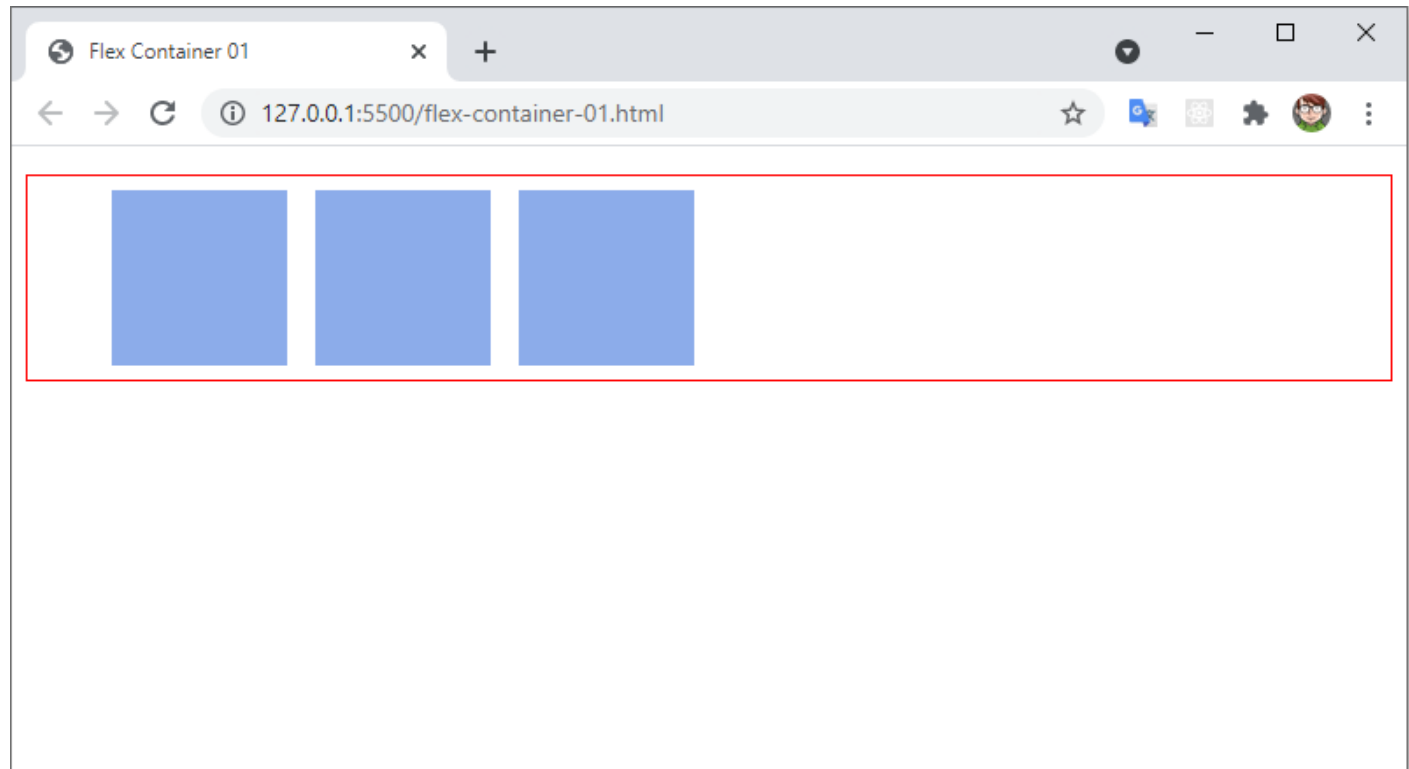
❑ Để sử dụng Flexbox:

♦ Để bắt đầu sử dụng bố cục Flexbox, bạn cần **định nghĩa** một **phần tử thùng chứa linh hoạt** (**flex-container**) và thiết lập thuộc tính **display:flex** hoặc **display:inline-flex** cho phần tử **flex-container** này.

```
<body>
  <!-- parent element -->
  <ul>
    <!-- first child element -->
    <li></li>
    <!-- second child element -->
    <li></li>
    <!-- third child element -->
    <li></li>
  </ul>
</body>
```

```
/* Make parent element a flex container 01 */
ul {
  /*display: flex or inline-flex*/
  display: flex;
  border: 1px solid red;
}

/* Style the list items in flex container 01 */
li {
  list-style-type: none;
  width: 100px;
  height: 100px;
  background-color: #8cacea;
  margin: 8px;
}
```





Tổng quan về bố cục Flexbox

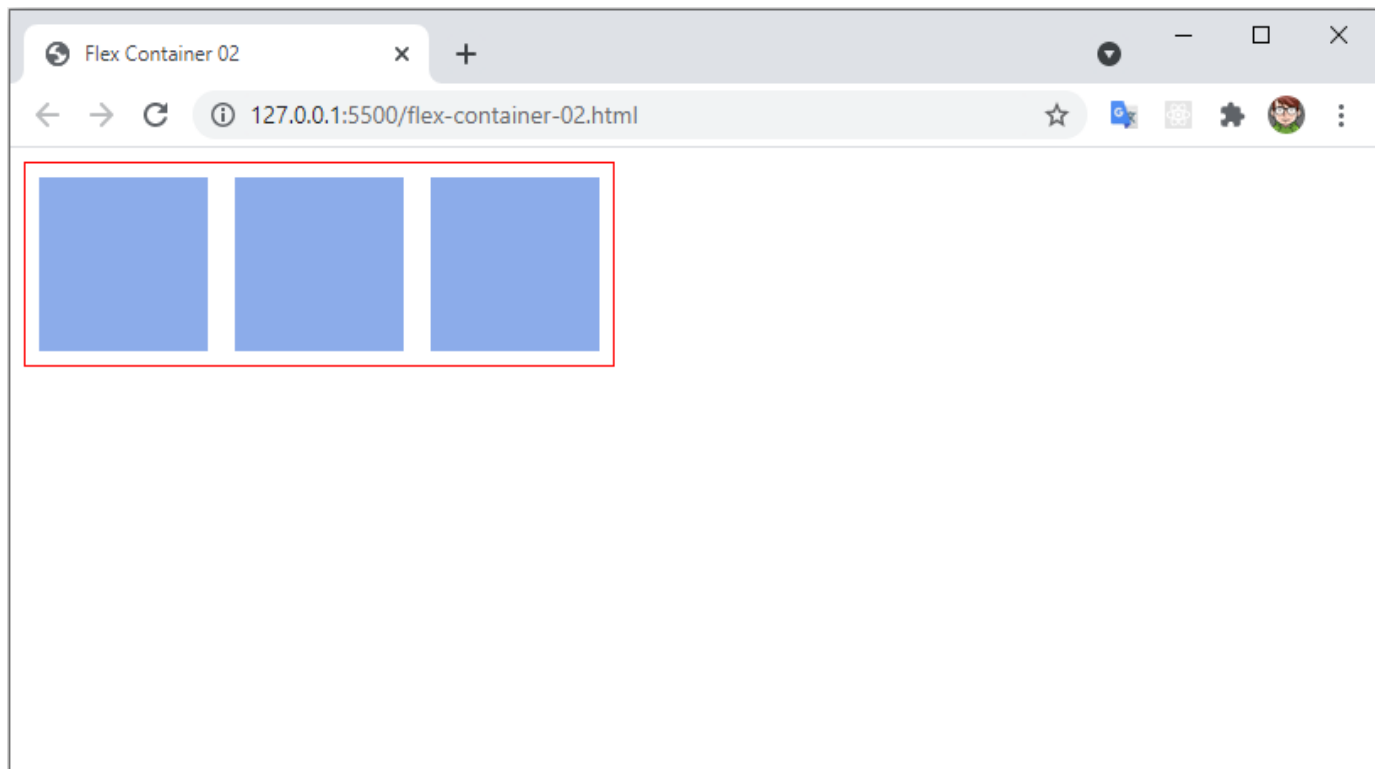
❑ Để sử dụng Flexbox:

♦ Để bắt đầu sử dụng bố cục Flexbox, bạn cần **định nghĩa** một **phần tử thùng chứa linh hoạt** (**flex-container**) và thiết lập thuộc tính **display:flex** hoặc **display:inline-flex** cho phần tử **flex-container** này.

```
<body>
  <!-- parent element -->
  <div id="container">
    <!-- first child element -->
    <div class="items"></div>
    <!-- second child element -->
    <div class="items"></div>
    <!-- third child element -->
    <div class="items"></div>
  </div>
</body>
```

```
/* Make parent element a flex container 02 */
#container {
  /*display: flex or inline-flex*/
  display: inline-flex;
  border: 1px solid red;
}

/* Style the list items in flex container 02 */
.items {
  height: 100px;
  width: 100px;
  background-color: #8cacea;
  margin: 8px;
}
```

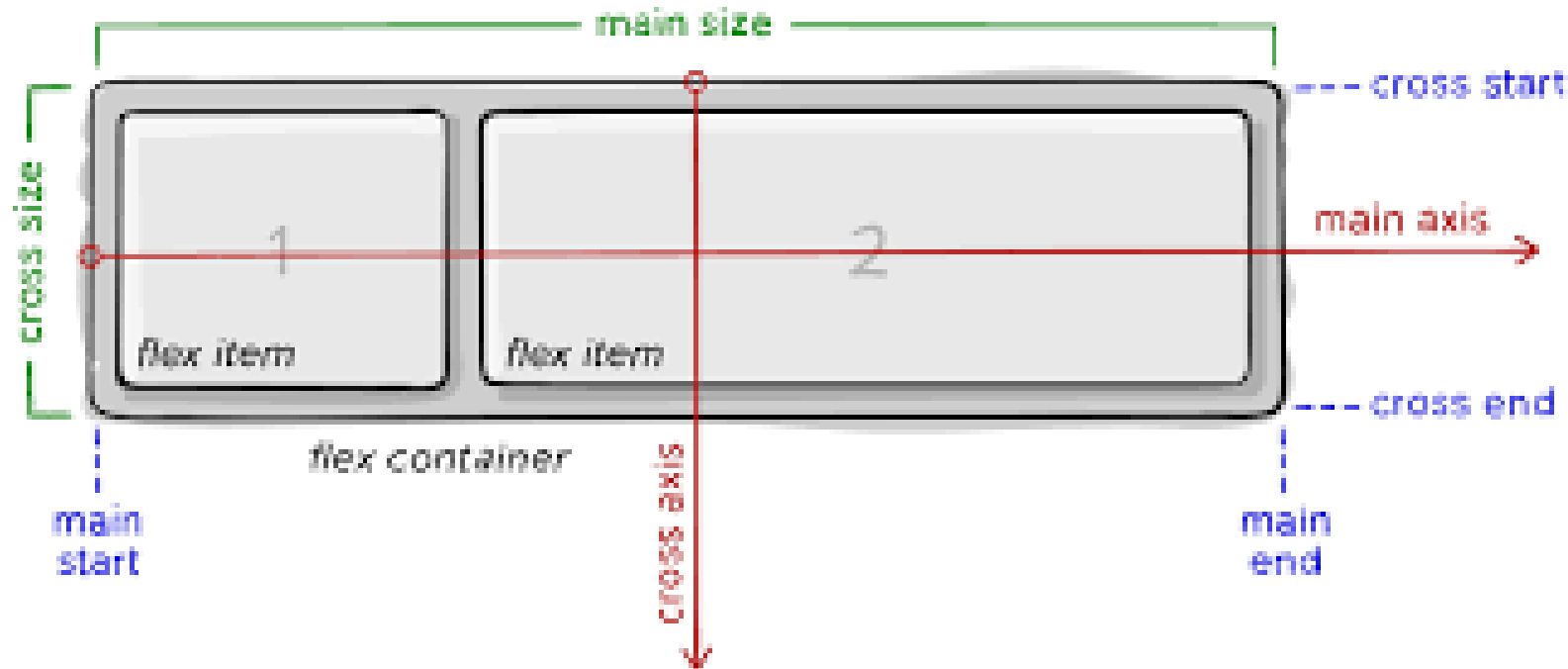




Các thuộc tính của Flex Container

❑ Thuộc tính flex-direction:

- ◆ Flex-direction **giúp kiểm soát hướng** mà **các flex items đặt dọc** theo **trục chính**.
- ◆ Flex-direction **có 4 giá trị phổ biến**: row, column, row-reverse, column-reverse.
- ◆ Về mặt kỹ thuật, **flex-direction chia** làm **2 hướng chính**: **main-axis** và **cross-axis**.





Các thuộc tính của Flex Container

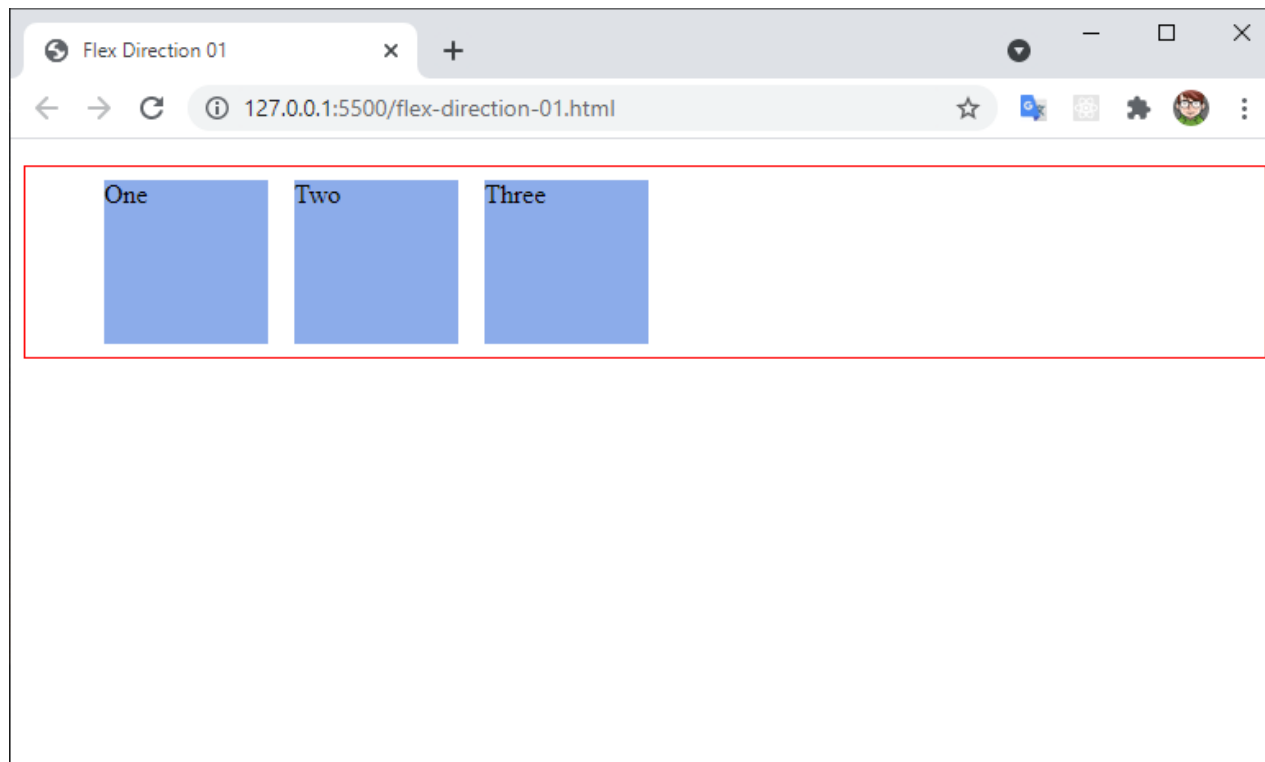
❑ Thuộc tính flex-direction:

- ♦ flex-direction giúp kiểm soát hướng mà các flex items đặt dọc theo trục chính.
- ♦ flex-direction có 4 giá trị phổ biến: **row**, column, row-reverse, column-reverse.
- ♦ Về mặt kỹ thuật, flex-direction chia làm 2 hướng chính: main-axis và cross-axis.

```
<!-- where ul represents a flex container -->
<ul>
  <!-- first child element -->
  <li></li>
  <!-- second child element -->
  <li></li>
  <!-- third child element -->
  <li></li>
</ul>
```

```
ul {
  display: flex;
  /* flex-direction: row or column, row-reverse,
  column-reverse; */
  flex-direction: row;
  border: 1px solid red;
}

li {
  list-style-type: none;
  width: 100px;
  height: 100px;
  background-color: #8cacea;
  margin: 8px;
}
```





Các thuộc tính của Flex Container

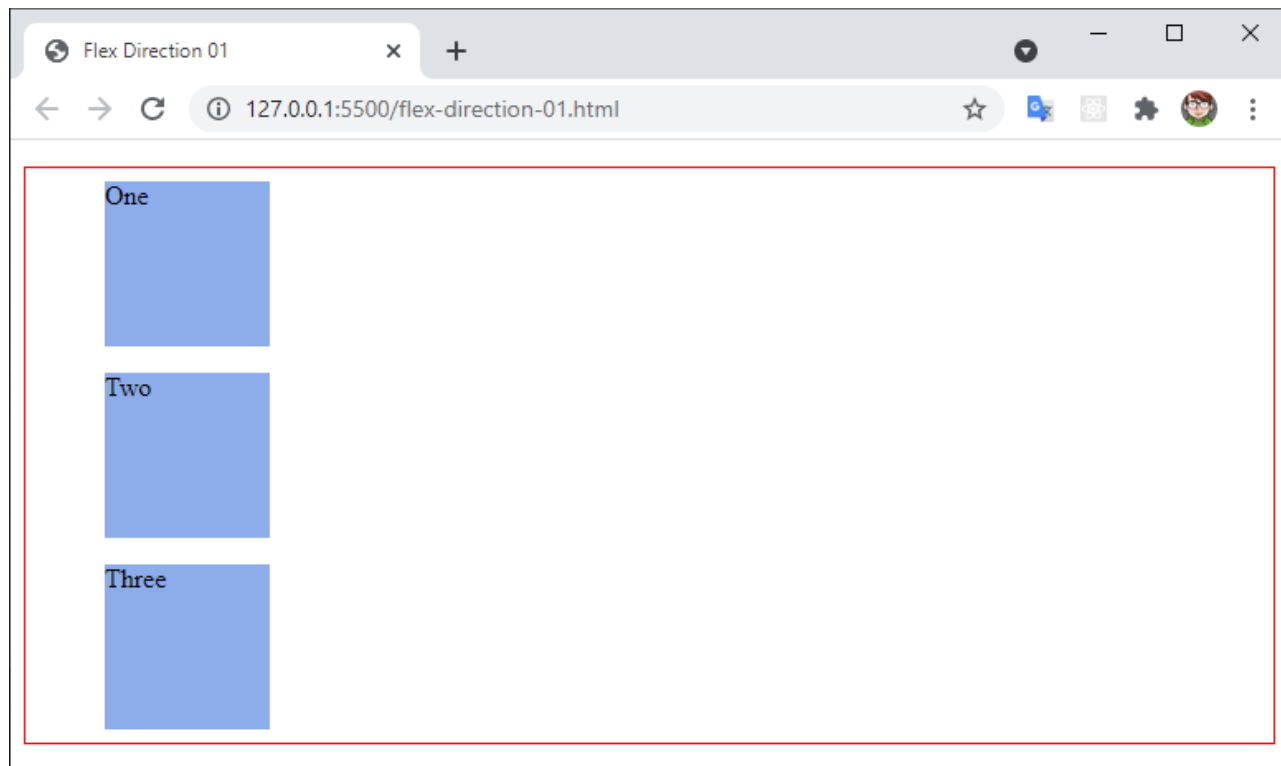
❑ Thuộc tính flex-direction:

- ♦ flex-direction giúp kiểm soát hướng mà các flex items đặt dọc theo trục chính.
- ♦ flex-direction có 4 giá trị phổ biến: row, column, row-reverse, column-reverse.
- ♦ Về mặt kỹ thuật, flex-direction chia làm 2 hướng chính: main-axis và cross-axis.

```
<!-- where ul represents a flex container -->
<ul>
  <!-- first child element -->
  <li></li>
  <!-- second child element -->
  <li></li>
  <!-- third child element -->
  <li></li>
</ul>
```

```
ul {
  display: flex;
  /* flex-direction: row or column, row-reverse,
  column-reverse; */
  flex-direction: column;
  border: 1px solid red;
}

li {
  list-style-type: none;
  width: 100px;
  height: 100px;
  background-color: #8cacea;
  margin: 8px;
}
```





Các thuộc tính của Flex Container

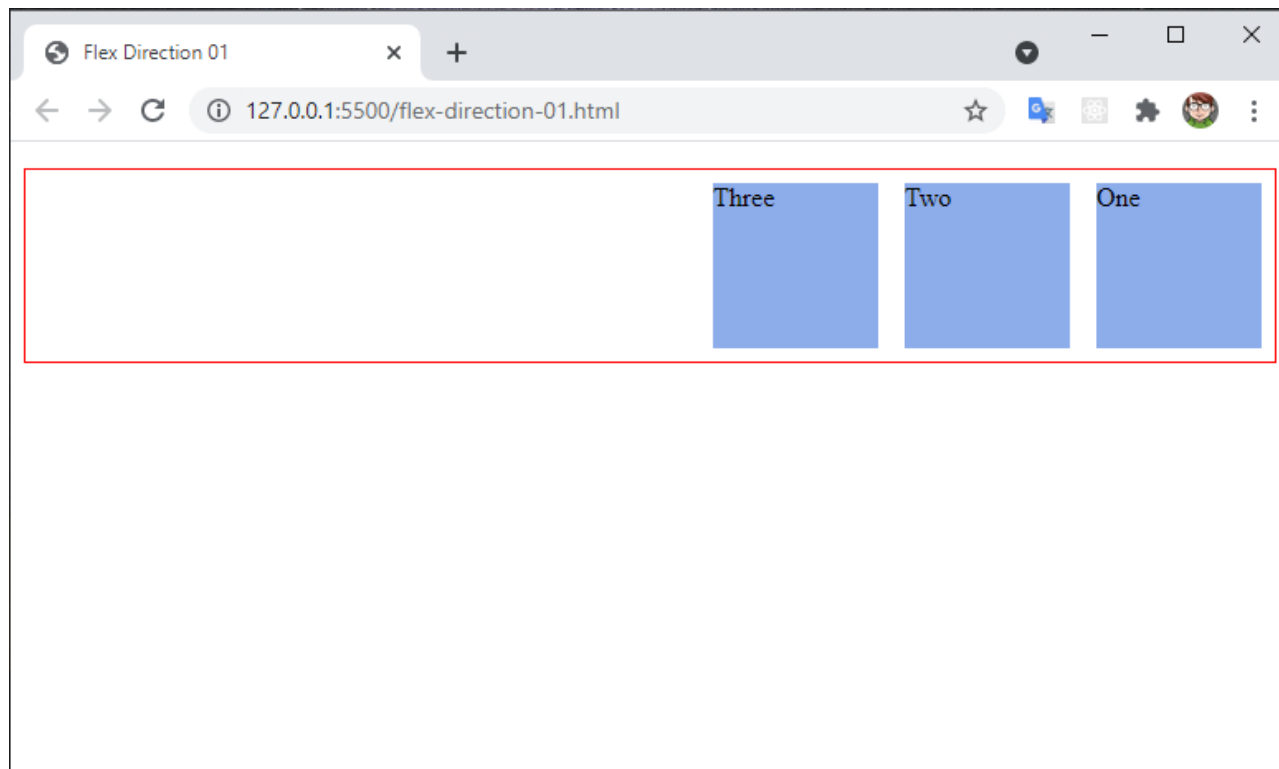
❑ Thuộc tính flex-direction:

- ♦ flex-direction **giúp kiểm soát hướng** mà các flex items đặt dọc theo **trục chính**.
- ♦ flex-direction **có 4 giá trị phổ biến**: row, column, **row-reverse**, column-reverse.
- ♦ Về mặt kỹ thuật, flex-direction **chia** làm **2 hướng chính**: main-axis và cross-axis.

```
<!-- where ul represents a flex container -->
<ul>
  <!-- first child element -->
  <li></li>
  <!-- second child element -->
  <li></li>
  <!-- third child element -->
  <li></li>
</ul>
```

```
ul {
  display: flex;
  /* flex-direction: row or column, row-reverse,
  column-reverse; */
  flex-direction: row-reverse;
  border: 1px solid red;
}

li {
  list-style-type: none;
  width: 100px;
  height: 100px;
  background-color: #8cacea;
  margin: 8px;
}
```





Các thuộc tính của Flex Container

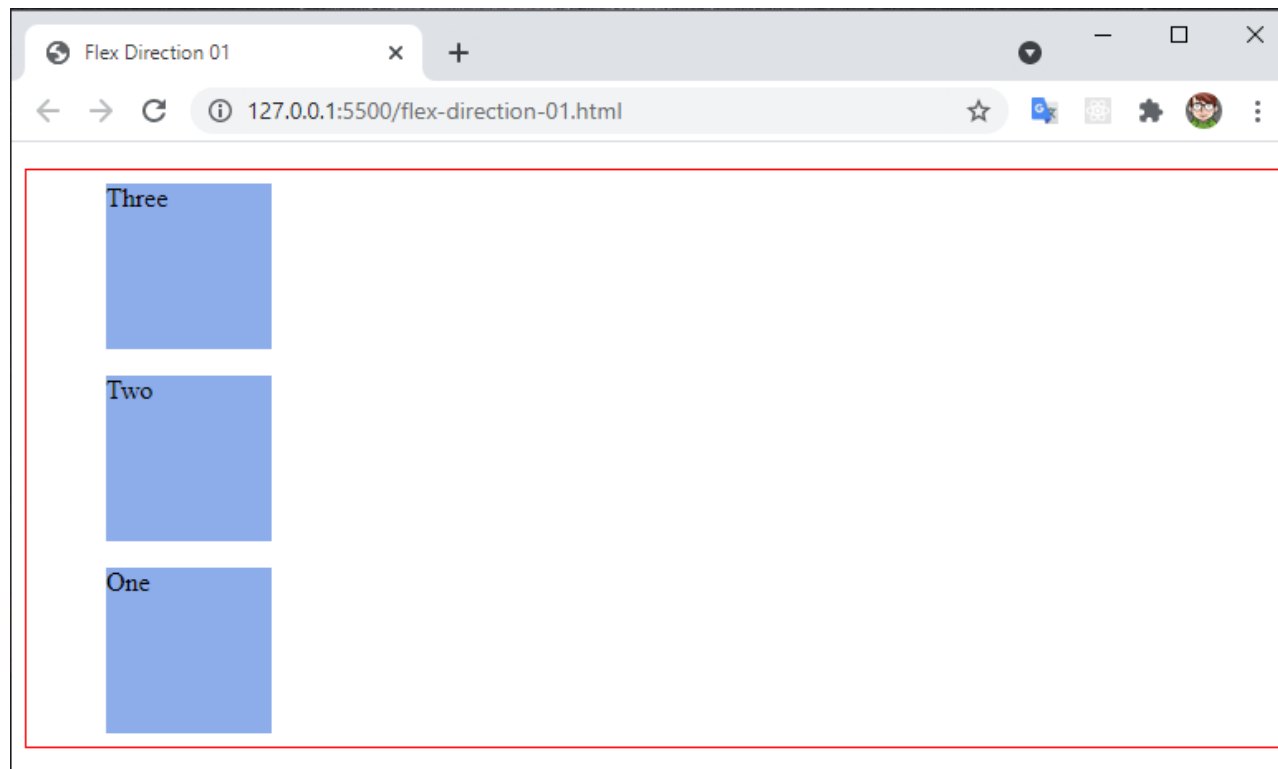
❑ Thuộc tính flex-direction:

- ◆ flex-direction giúp kiểm soát hướng mà các flex items đặt dọc theo trục chính.
- ◆ flex-direction có 4 giá trị phổ biến: row, column, row-reverse, **column-reverse**.
- ◆ Về mặt kỹ thuật, flex-direction chia làm 2 hướng chính: main-axis và cross-axis.

```
<!-- where ul represents a flex container -->
<ul>
  <!-- first child element -->
  <li></li>
  <!-- second child element -->
  <li></li>
  <!-- third child element -->
  <li></li>
</ul>
```

```
ul {
  display: flex;
  /* flex-direction: row or column, row-reverse,
  column-reverse; */
  flex-direction: column-reverse;
  border: 1px solid red;
}

li {
  list-style-type: none;
  width: 100px;
  height: 100px;
  background-color: #8cacea;
  margin: 8px;
}
```



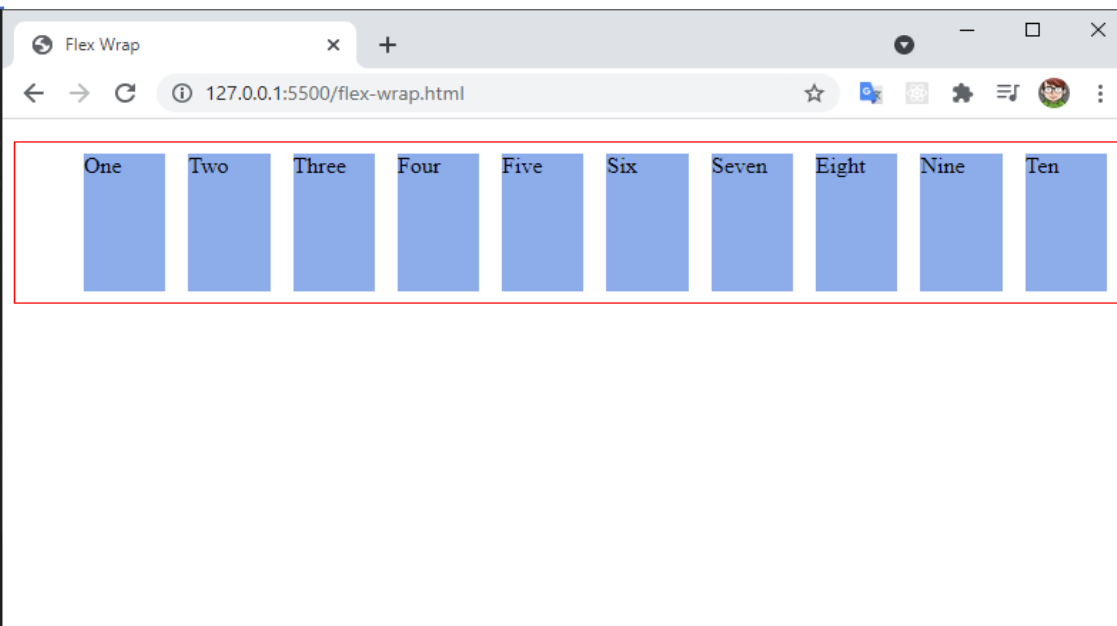


Các thuộc tính của Flex Container

❑ Thuộc tính flex-wrap:

- ♦ flex-wrap giúp bao bọc các flex items và quyết định có xuống dòng hay không.
- ♦ flex-wrap có 3 giá trị phổ biến:
 - nowrap, wrap, wrap-reverse.

```
<!-- where ul repre ul {  
<ul>  
  <li>One</li>  
  <li>Two</li>  
  <li>Three</li>  
  <li>Four</li>  
  <li>Five</li>  
  <li>Six</li>  
  <li>Seven</li>  
  <li>Eight</li>  
  <li>Nine</li>  
  <li>Ten</li>  
</ul>  
  display: flex;  
  flex-direction: row;  
  flex-wrap: nowrap;  
  border: 1px solid red;  
  li {  
    list-style-type: none;  
    width: 100px;  
    height: 100px;  
    background-color: #8cacea;  
    margin: 8px;  
  }  
}
```



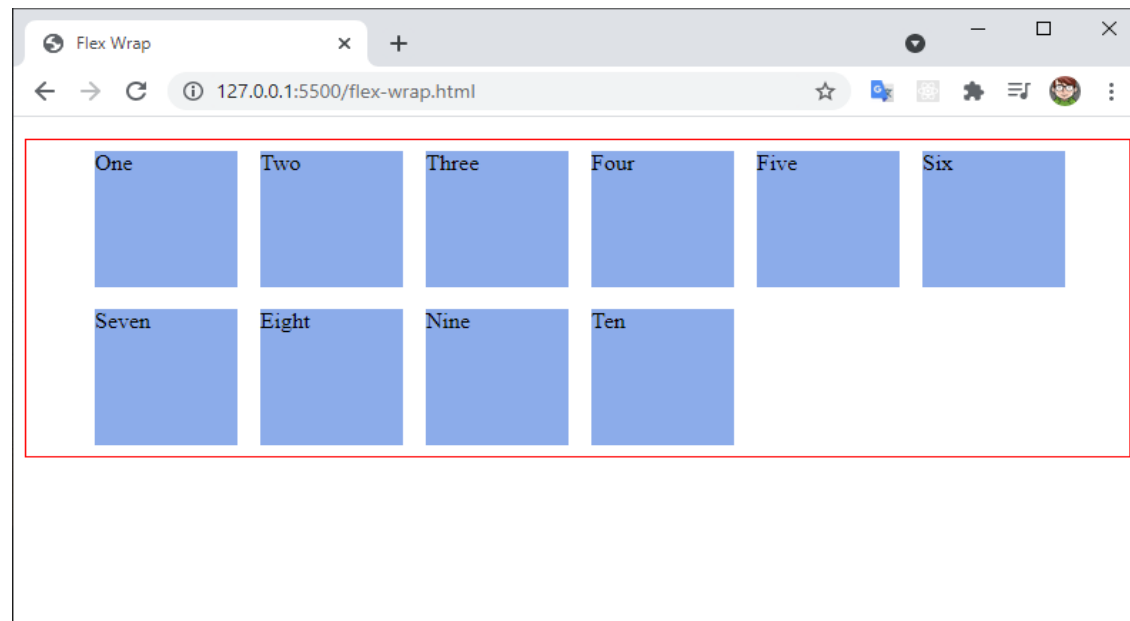


Các thuộc tính của Flex Container

❑ Thuộc tính flex-wrap:

- ♦ flex-wrap giúp bao bọc các flex items và quyết định có xuống dòng hay không.
- ♦ flex-wrap có 3 giá trị phổ biến:
 - nowrap, **wrap**, wrap-reverse.

```
<!-- where ul repres ul {  
<ul>  
  <li>One</li>  
  <li>Two</li>  
  <li>Three</li>  
  <li>Four</li>  
  <li>Five</li>  
  <li>Six</li>  
  <li>Seven</li>  
  <li>Eight</li>  
  <li>Nine</li>  
  <li>Ten</li>  
</ul>  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  border: 1px solid red;  
  li {  
    list-style-type: none;  
    width: 100px;  
    height: 100px;  
    background-color: #8cacea;  
    margin: 8px;  
  }  
}
```



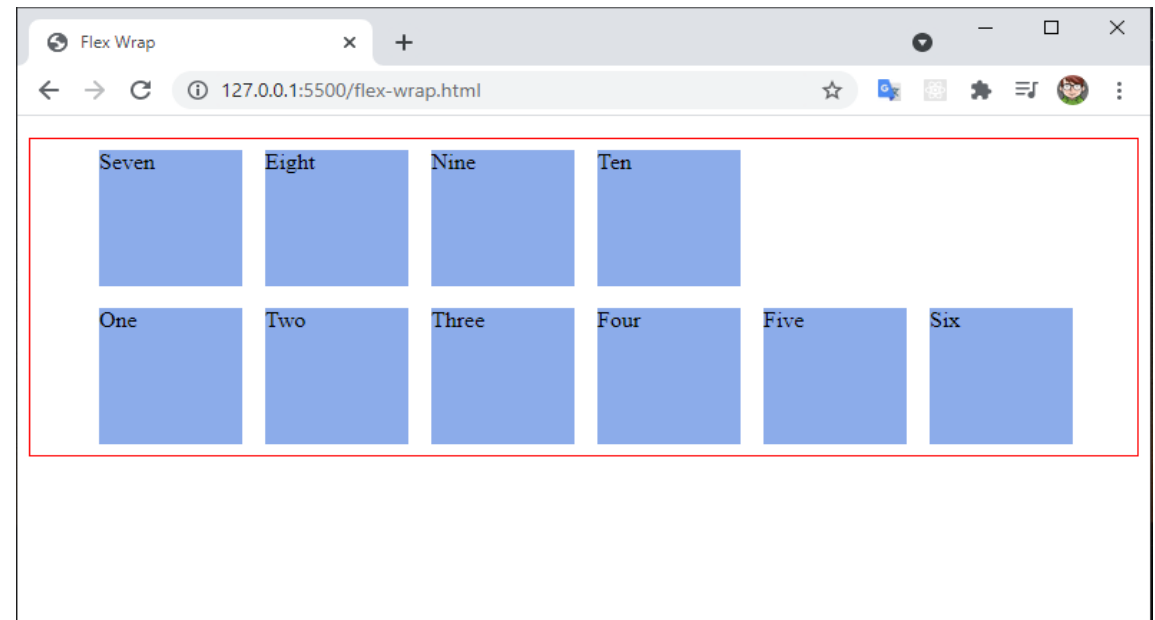


Các thuộc tính của Flex Container

❑ Thuộc tính flex-wrap:

- ♦ flex-wrap giúp bao bọc các flex items và quyết định có xuống dòng hay không.
- ♦ flex-wrap có 3 giá trị phổ biến:
 - nowrap, wrap, **wrap-reverse**.

```
<!-- where ul represents ul {  
<ul>  
  <li>One</li>  
  <li>Two</li>  
  <li>Three</li>  
  <li>Four</li>  
  <li>Five</li>  
  <li>Six</li>  
  <li>Seven</li>  
  <li>Eight</li>  
  <li>Nine</li>  
  <li>Ten</li>  
</ul>  
  }  
  li {  
    list-style-type: none;  
    width: 100px;  
    height: 100px;  
    background-color: #8cace6;  
    margin: 8px;  
  }  
}
```





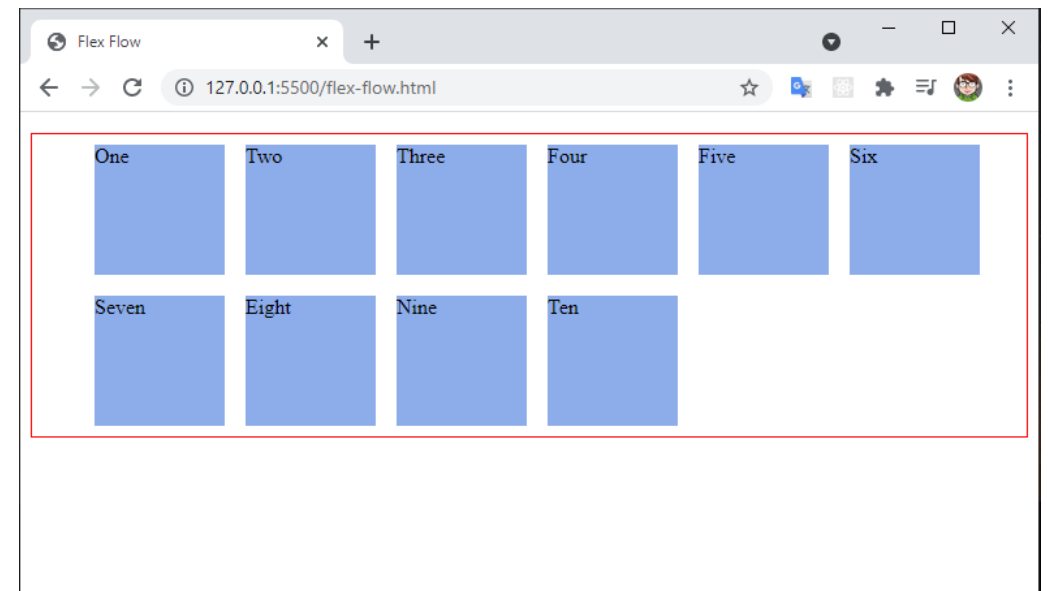
Các thuộc tính của Flex Container

❑ Thuộc tính flex-flow:

- ♦ flex-flow giúp kết hợp 2 thuộc tính flex-direction và flex-wrap lên các flex items.
- ♦ flex-flow có một giá trị phổ biến: **row wrap**, row nowrap, column wrap, column nowrap, row wrap-reverse, column wrap-reverse.

```
<!-- where ul represents 

 {  
<ul>  
  <li>One</li>  
  <li>Two</li>  
  <li>Three</li>  
  <li>Four</li>  
  <li>Five</li>  
  <li>Six</li>  
  <li>Seven</li>  
  <li>Eight</li>  
  <li>Nine</li>  
  <li>Ten</li>  
</ul>  
  }  
  li {  
    list-style-type: none;  
    width: 100px;  
    height: 100px;  
    background-color: #8cacea;  
    margin: 8px;  
  }  
}
```



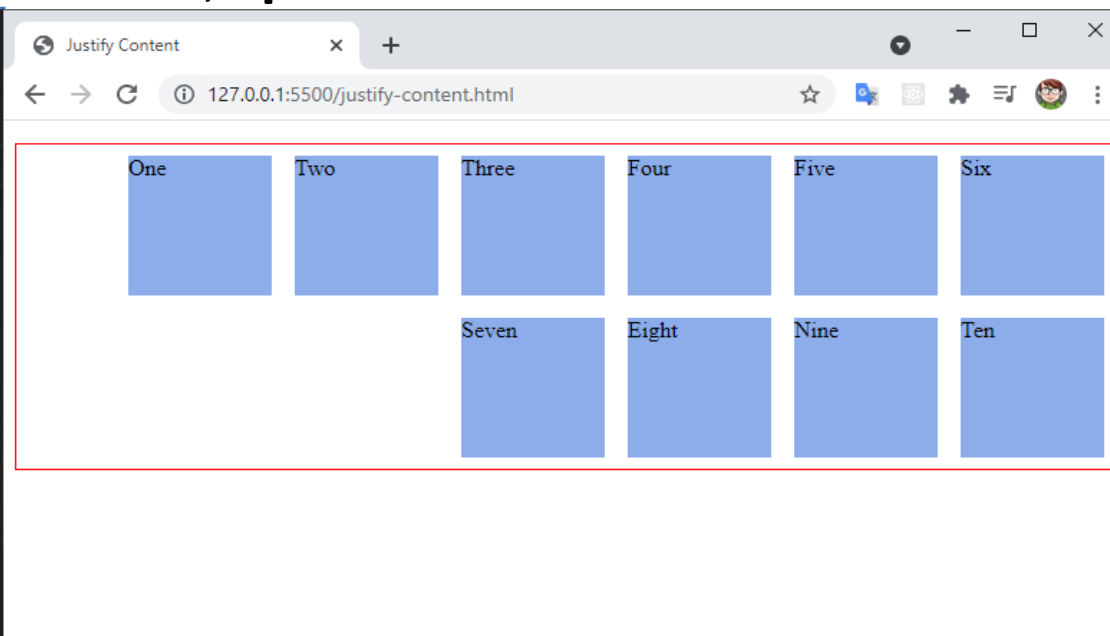


Các thuộc tính của Flex Container

❑ Thuộc tính justify-content:

- ♦ justify-content giúp canh chỉnh nội dung của các flex items trên trục chính.
- ♦ Có 5 giá trị phổ biến:
 - flex-start, **flex-end**, center, space-between, space-around.

```
<!-- where ul repre ul {  
<ul>  
  <li>One</li>  
  <li>Two</li>  
  <li>Three</li>  
  <li>Four</li>  
  <li>Five</li>  
  <li>Six</li>  
  <li>Seven</li>  
  <li>Eight</li>  
  <li>Nine</li>  
  <li>Ten</li>  
</ul>  
  display: flex;  
  flex-flow: row wrap;  
  justify-content: flex-end;  
  border: 1px solid red;  
  li {  
    list-style-type: none;  
    width: 100px;  
    height: 100px;  
    background-color: #8cacea;  
    margin: 8px;  
  }  
}
```





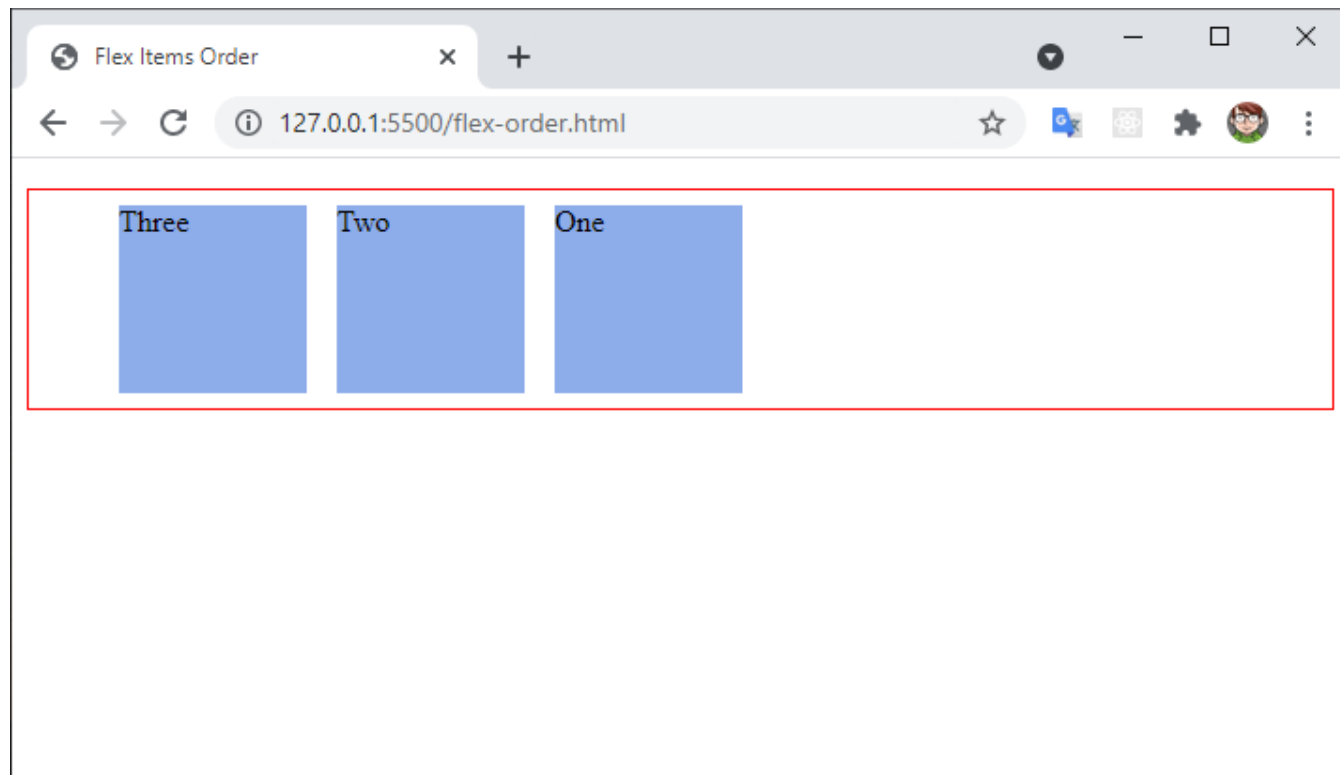
Các thuộc tính của Flex Item

❑ Thuộc tính order:

- ♦ order cho phép sắp xếp lại **thứ tự của các flex items** bên trong flex-container.
- ♦ Giá trị mặc định của thuộc tính order là **0**.
- ♦ order **hỗ trợ làm việc** trong **các bài toán liên quan tới thứ tự** của **các phần tử**.

```
<!-- where ul represents flex items order  
<ul>  
  <li>One</li>  
  <li>Two</li>  
  <li>Three</li>  
</ul>
```

```
/* select first li element within the ul */  
li:nth-child(1) {  
  /*give it a value equal 3*/  
  order: 3;  
}  
  
/* select second li element within the ul */  
li:nth-child(2) {  
  /*give it a value equal 2*/  
  order: 2;  
}  
  
/* select third li element within the ul */  
li:nth-child(3) {  
  /*give it a value equal 1*/  
  order: 1;  
}
```



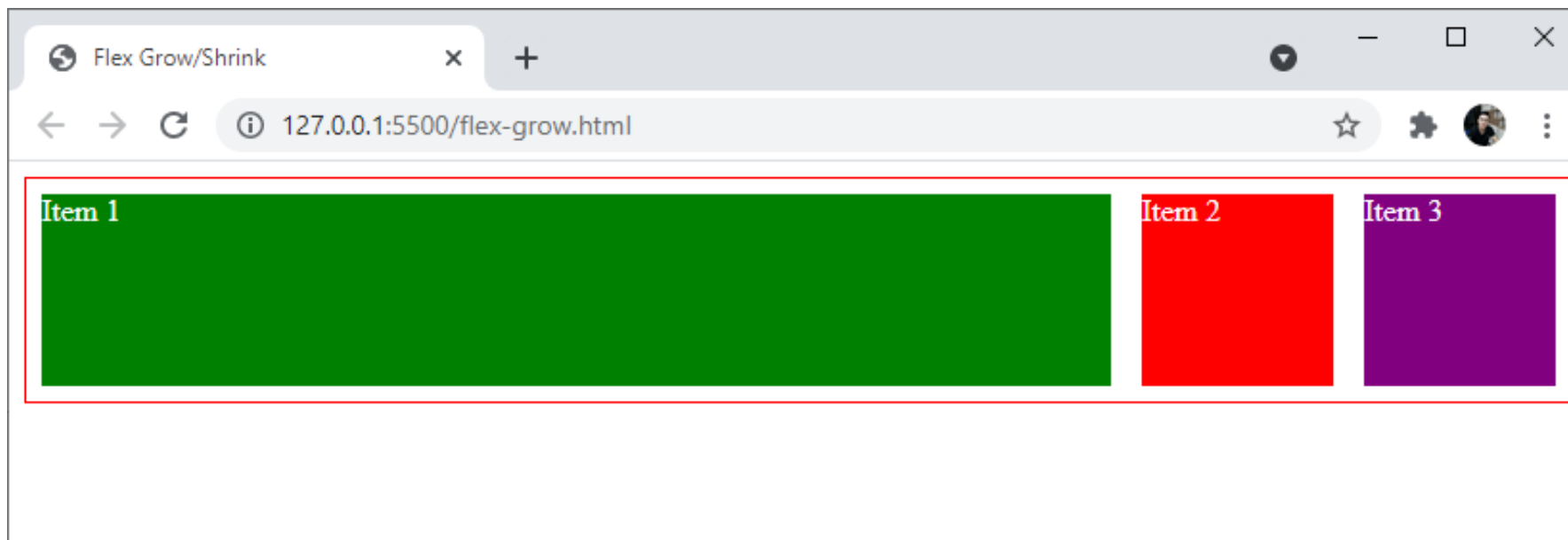


Các thuộc tính của Flex Item

❑ Thuộc tính flex-grow:

♦ Đặc tính **phát triển linh hoạt (flex-grow)** cho phép bạn **sử dụng khoảng trống còn lại trong một flex-container**, hoặc **tăng (grow) độ rộng phần tử có flex-grow lên bao nhiêu lần so với các phần tử còn lại**. **Giá trị** của **flex-grow** phải **là các số nguyên dương**.

♦ **Ví dụ 1:** Thêm **flex-grow: 1** vào Item 1 ta thấy item-1 chiếm hết khoảng trống còn lại. Item 2, 3 sẽ bị đẩy ra nhưng vẫn giữ nguyên độ rộng vì mặc định Item 2, 3 có **flex-grow: 0**



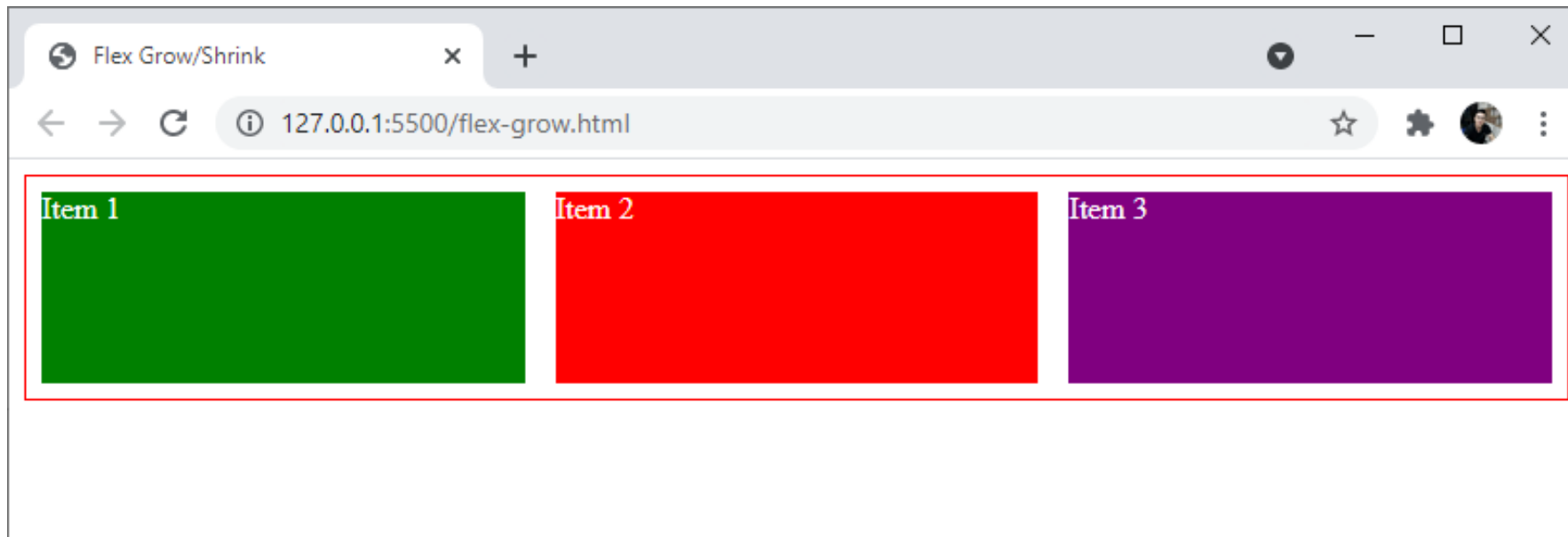


Các thuộc tính của Flex Item

❑ Thuộc tính flex-grow:

♦ Đặc tính **phát triển linh hoạt (flex-grow)** cho phép bạn **sử dụng khoảng trống còn lại trong một flex-container**, hoặc **tăng (grow) độ rộng phần tử có flex-grow lên bao nhiêu lần so với các phần tử còn lại**. **Giá trị của flex-grow phải là các số nguyên dương**.

♦ **Ví dụ 2:** Thêm **flex-grow: 1** vào Item 1, 2, 3 ta thấy tất cả các Item có độ rộng như nhau. Vì chúng ta đã thiết lập cho chúng tỉ lệ như nhau.



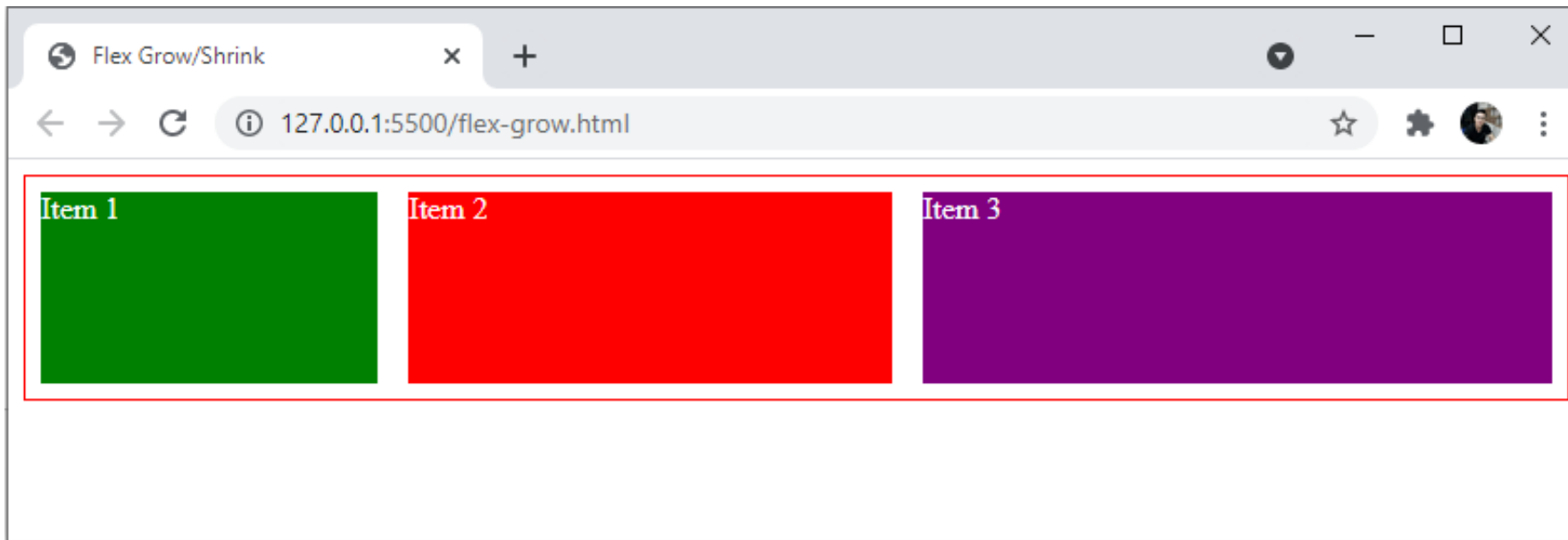


Các thuộc tính của Flex Item

❑ Thuộc tính flex-grow:

♦ Đặc tính **phát triển linh hoạt** (flex-grow) cho phép bạn **sử dụng khoảng trống còn lại trong một flex-container**, hoặc **tăng** (grow) **độ rộng phần tử có flex-grow lên bao nhiêu lần so với các phần tử còn lại**. **Giá trị** của flex-grow phải **là các số nguyên dương**.

♦ **Ví dụ 3:** Thêm **flex-grow: 1; flex-grow: 2; flex-grow: 3;** vào lần lượt các Item 1, 2, 3 ta thấy các Item sẽ có tỉ lệ tương ứng **1:2:3** với nhau. Tổng: **1+2+3=6** -> **1/6 : 2/6 : 3/6**



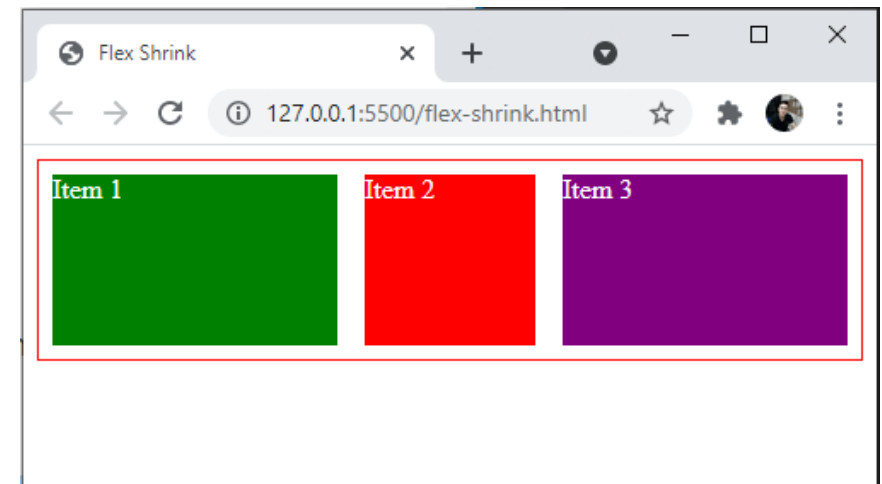
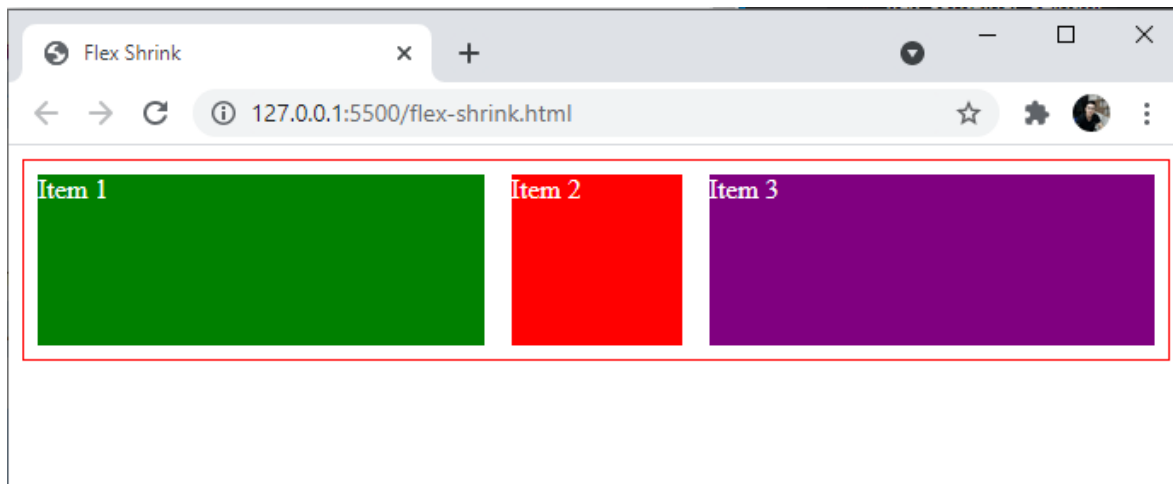


Các thuộc tính của Flex Item

❑ Thuộc tính flex-shrink:

♦ Đặc tính **co rút linh hoạt (flex-shrink)** cho phép bạn **co khoảng trống của từng phần tử lại trong một flex-container** khi **thay đổi width của parent xuống**. **Giá trị của flex-shrink phải là các số nguyên dương**.

♦ **Ví dụ**: Thêm **flex-shrink: 0; flex-shrink: 1; flex-shrink: 1;** vào lần lượt các Item 2, 1, 3 ta thấy Item 2 không co lại khi ta co màn hình lại. Còn 2 item còn lại bị co lại tỉ lệ 1:1.

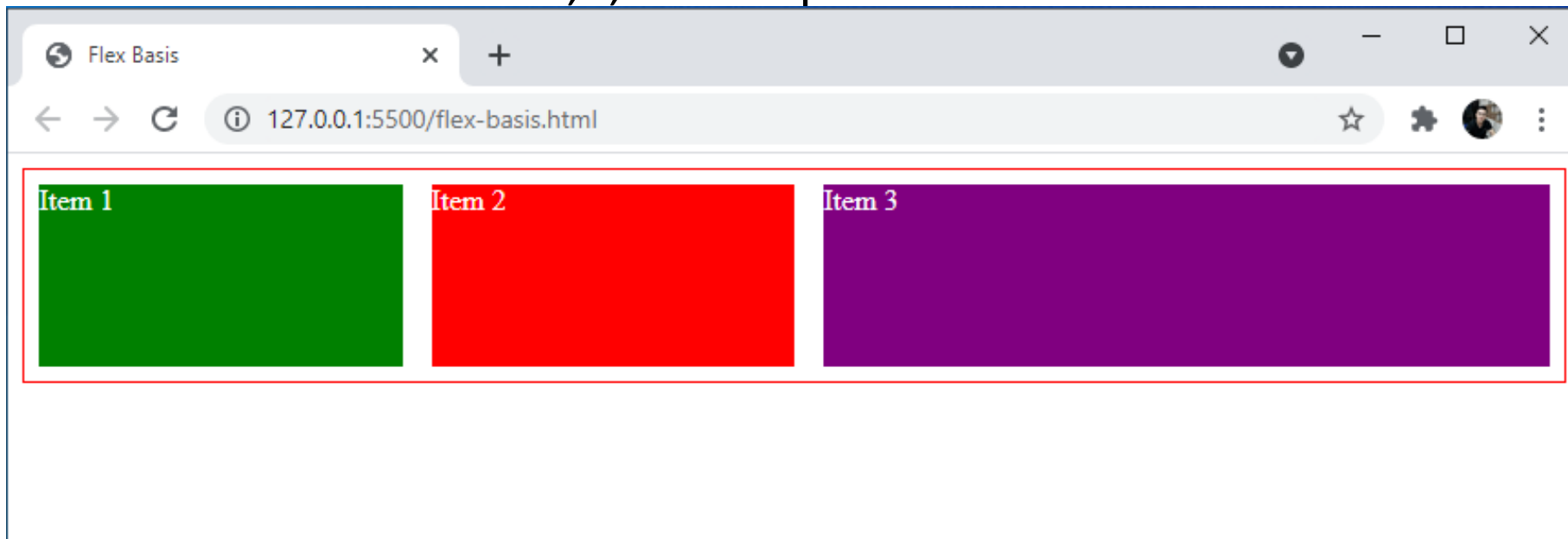




Các thuộc tính của Flex Item

❑ Thuộc tính flex-basis:

- ◆ Thuộc tính flex-basis **chỉ định kích thước ban đầu của một flex-item.**
- ◆ **Giá trị mặc định** của flex-basis là **auto**.
- ◆ **Giá trị** của flex-basis **sử dụng phổ biến** là: %, ems, rems, px,...
- ◆ **Ví dụ:** Thêm **flex-basis: 400px;** vào lần lượt các Item 3 ta thấy Item 3 to lên gấp đôi width ban đầu của cả 3 item 1,2,3 là 200px.





Các thuộc tính của Flex Item

❑ Rút gọn 3 thuộc tính flex-grow, flex-shrink, flex-basis:

- ♦ Thuộc tính **flex** cho phép rút gọn cả 3 thuộc tính **flex-grow**, **flex-shrink**, **flex-basis** cùng 1 lúc.
- ♦ Khuyến khích nên dùng thuộc tính **flex** để rút gọn 3 thuộc tính trên của flex-item.

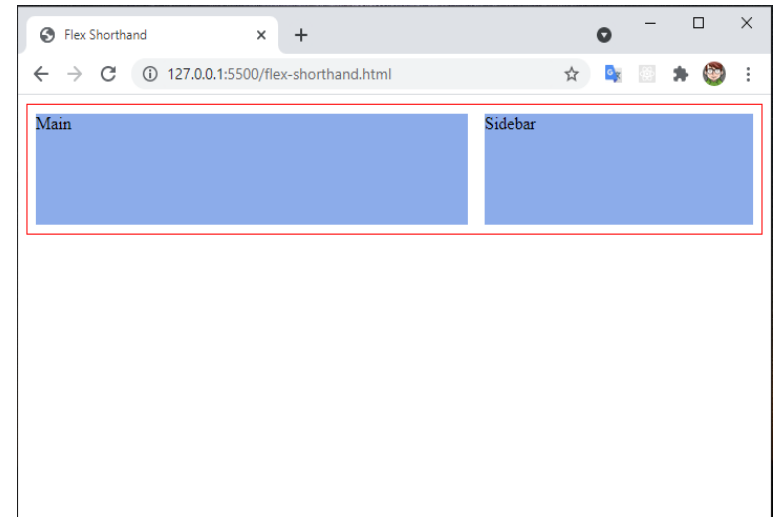
```
<!-- where ul represents flex shorthand -->
<div id="container">
  <p class="items main">Main</p>
  <p class="items sidebar">Sidebar</p>
</div>
```

```
#container {
  display: flex;
  flex-flow: row wrap;
  justify-content: flex-start;
  border: 1px solid red;
}
```

```
.items {
  list-style-type: none;
  width: 100px;
  height: 100px;
  background-color: #8cacea;
  margin: 8px;
}

.main {
  /* Shorthand of 3 properties: flex-grow: 2;
  flex-shrink: 1; flex-basis: auto;*/
  flex: 2 1 auto;
}

.sidebar {
  /* Shorthand of 3 properties: flex-grow: 1;
  flex-shrink: 1; flex-basis: auto;*/
  flex: 1 1 auto;
}
```



- ♦ **flex: 2 1 auto;** -> flex-grow: 2; -> nở ra lấp khoảng trống còn lại của parent div.
-> flex-shrink: 1; -> co lại khi điều chỉnh giảm độ rộng của parent div.
-> flex-basis: auto; -> thiết lập giá trị độ rộng cho phần tử là auto.

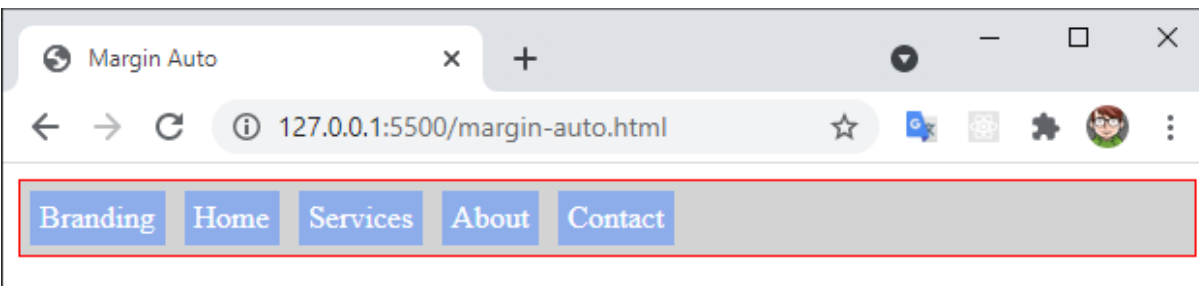


Canh chỉnh giao diện với Auto Margin

❑ Thuộc tính auto-margin:

- ◆ Khi bạn sử dụng **margin: auto** trên một **flex item**, **hướng** (trái, phải hoặc cả hai) có giá trị **auto** sẽ chiếm bất kỳ khoảng trống nào có sẵn.
- ◆ Bạn thường sử dụng thuộc tính này để canh chỉnh khoảng cách giữa các flex item.

```
<!-- where ul represents margin auto -->
<ul>
  <li>Branding</li>
  <li>Home</li>
  <li>Services</li>
  <li>About</li>
  <li>Contact</li>
</ul>
```



```
ul {
  display: flex;
  border: 1px solid red;
  background-color: lightgrey;
  margin: 0;
  padding: 0;
}

li {
  list-style-type: none;
  background-color: #8cacea;
  color: white;
  margin: 5px;
  padding: 5px;
  flex: 0 0 auto;
}
```



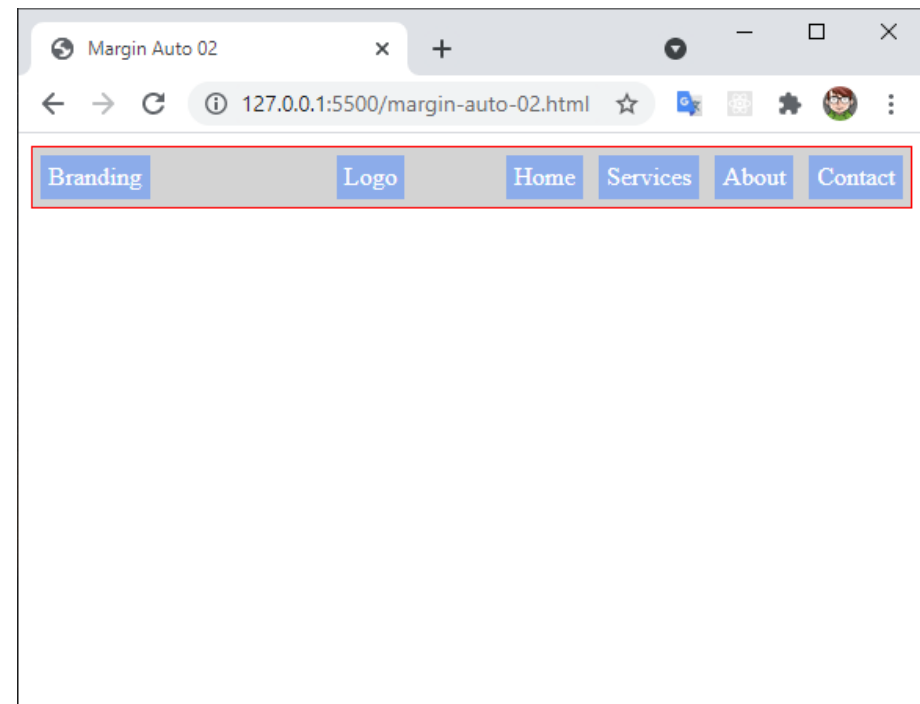

Canh chỉnh giao diện với Auto Margin

❑ Thuộc tính auto-margin:

- ◆ Sử dụng **margin-left: auto, margin-right: auto** trên một **flex item** để có thể canh trái, phải, giữa tùy mục đích sử dụng.
- ◆ Bạn thường sử dụng thuộc tính này để canh chỉnh từng thành phần cho menu.

```
<!-- where ul represer  
<ul>  
  <li>Branding</li>  
  <li>Logo</li>  
  <li>Home</li>  
  <li>Services</li>  
  <li>About</li>  
  <li>Contact</li>  
</ul>
```

```
ul { ...  
}  
  
li { ...  
}  
  
li:nth-child(1) {  
  margin-right: auto;  
}  
  
li:nth-child(2) {  
  margin-left: auto;  
  margin-right: auto;  
}
```

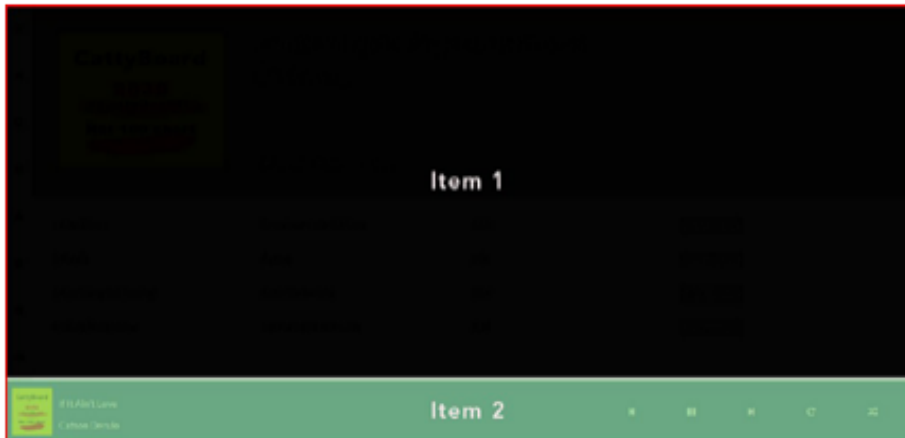




Xây dựng bố cục Music App với Flexbox

❑ Thực hành xây dựng bố cục giao diện Music App:

- ♦ Phân tích **trang HTML ban đầu** có **cấu trúc** bao gồm **2 phần** là **main** và **header**.
- ♦ Sau đó, chuẩn bị **trang HTML đơn giản** và **khởi tạo** một **flex-container cho body**.



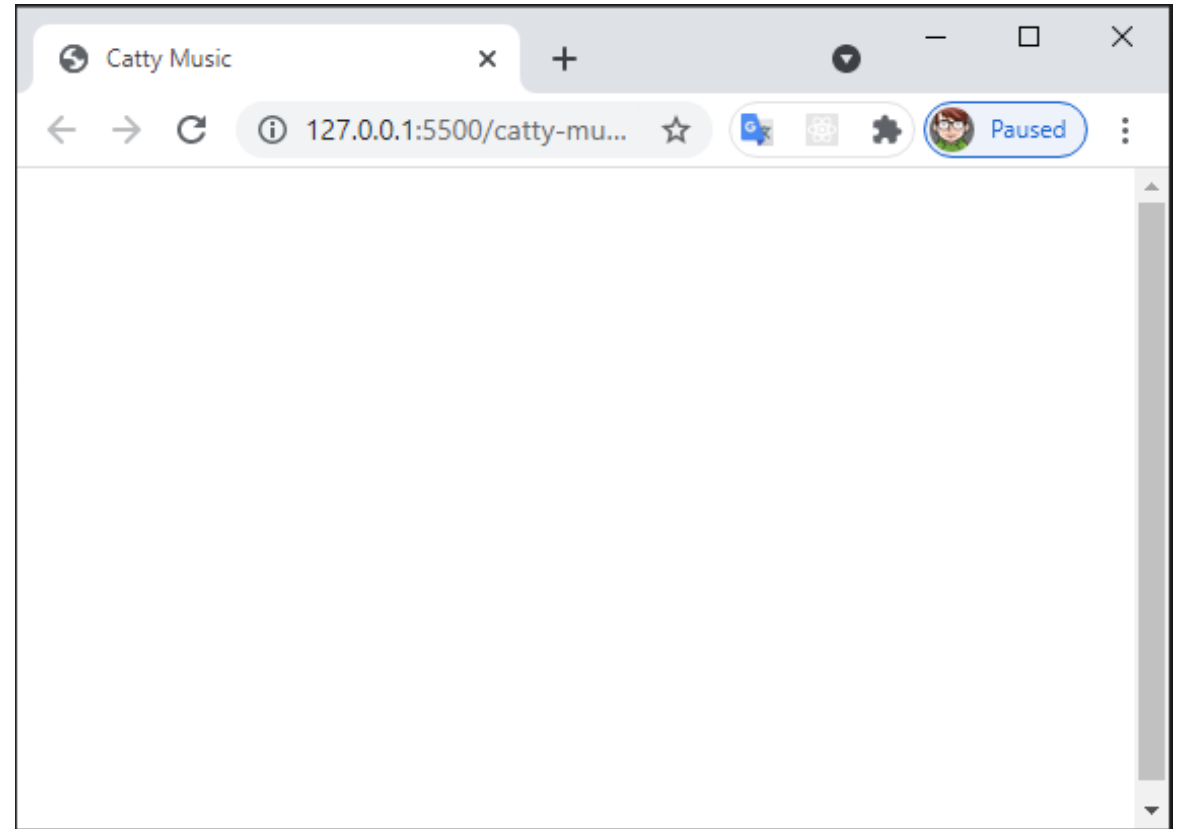
```
<head>
  <title>Catty Music</title>
  <link href="./css/catty-music.css"
        rel="stylesheet" type="text/css">
</head>

<body>
  <!--to contain the main section
  | of the app-->
  <main></main>

  <!--to contain the music control
  | buttons and song details-->
  <footer></footer>
</body>
```

```
html,
body {
  height: 100%;
}

body {
  display: flex;
  flex-direction: column;
}
```





Xây dựng bố cục Music App với Flexbox

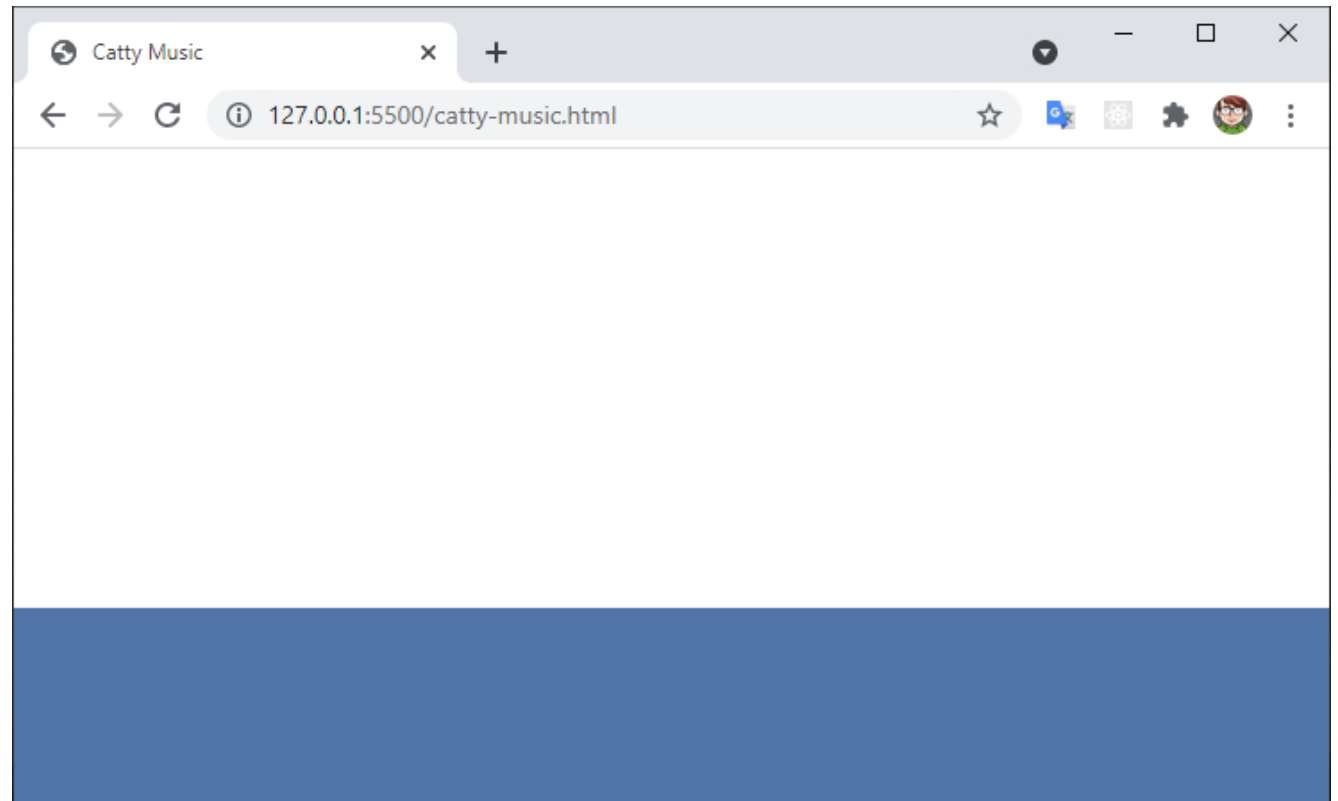
❑ Thực hành xây dựng bố cục giao diện Music App:

◆ Phần **chân trang** (footer) **chứa các nút điều khiển âm nhạc** nằm ở cuối trang trong khi **phần chính** (main) **lấp đầy không gian còn lại**.

```
body {
  display: flex;
  flex-direction: column;
  background-color: #fff;
  margin: 0;
  font-family: Lato, sans-serif;
  color: #222;
  font-size: 0.9em;
}

main {
  flex: 1 0 auto;
}

footer {
  /*don't grow or shrink,
  just stay at a height of 90px */
  flex: 0 0 90px;
  padding: 10px;
  color: #fff;
  background-color: rgba(61, 100, 158, 0.9);
}
```





Xây dựng bố cục Music App với Flexbox

❑ Thực hành xây dựng bố cục giao diện Music App:

- ◆ Thêm các liên kết về font chữ vào thẻ head để sử dụng các **phông chữ** và **hình icon nhỏ gọn vào** thanh **sidebar** lúc sau.

```
<head>
  <title>Catty Music</title>
  <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">
  <link rel="stylesheet" type="text/css"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
  <link href="./css/catty-music.css" rel="stylesheet" type="text/css">
</head>
```

- ◆ Thêm **flex container** vào **main** để chia vị trí với **sidebar** lúc sau, main flex-grow là **1**.

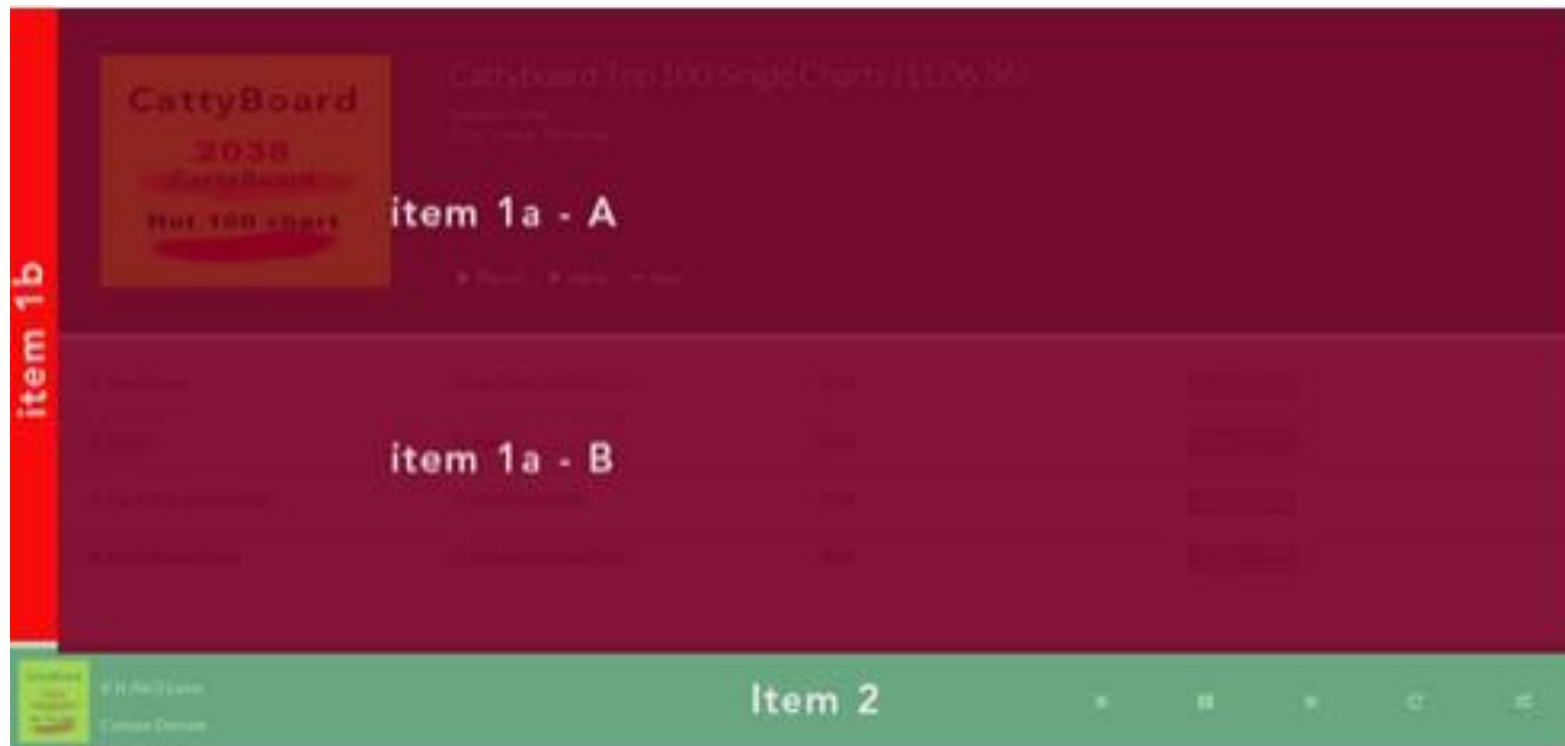
```
main {
  /*Is a flex item*/
  flex: 1 0 auto;
  /*I just included this! - now a flex container
  with flex items: sidebar & main content section*/
  display: flex;
}
```



Xây dựng bố cục Music App với Flexbox

☐ Thực hành xây dựng bố cục giao diện Music App:

- ♦ Phân tích giao diện hiện tại phần 1a ra **thành 2 phần**: item 1a – A và item 1a – B.
- ♦ Thêm thẻ aside chứa các icon cho thanh sidebar **giúp hiển thị các tính năng** của **giao diện app nghe nhạc**: home, search, user,...Và chỉnh sửa CSS cho sidebar.





Xây dựng bố cục Music App với Flexbox

❑ Thực hành xây dựng bố cục giao diện Music App:

- ♦ Phân tích giao diện hiện tại phần 1a ra **thành 2 phần**: item 1a – A và item 1a – B.
- ♦ Thêm thẻ aside chứa các icon cho thanh sidebar **giúp hiển thị các tính năng** của **giao diện app nghe nhạc**: home, search, user,... Và chỉnh sửa CSS cho sidebar.

```
<!--to contain the main section
of the app-->
<main>
  <!--This represents the sidebar and contained
  in it are icon sets from font-awesome-->
  <aside>
    <i class="fa fa-bars"></i>
    <i class="fa fa-home"></i>
    <i class="fa fa-search"></i>
    <i class="fa fa-volume-up"></i>
    <i class="fa fa-user"></i>
    <i class="fa fa-spotify"></i>
    <i class="fa fa-cog"></i>
    <i class="fa fa-soundcloud"></i>
  </aside>
  <section class="content">
    <!--This section will house everything other
    than the sidebar-->
  </section>
</main>
```

```
aside {
  /*do not grow or shrink. Stay fixed at 40px*/
  flex: 0 0 40px;
  /*Now a flex-container too*/
  display: flex;
  /*stack icons vertically*/
  flex-direction: column;
  /*since direction is changed, this works on
  the vertical direction*/
  justify-content: space-around;
  /*direction is changed! Align-items affects the
  horizontal direction. Place Icons in the center*/
  align-items: center;
  /*make me pretty*/
  background-color: #f2f2f2;
}

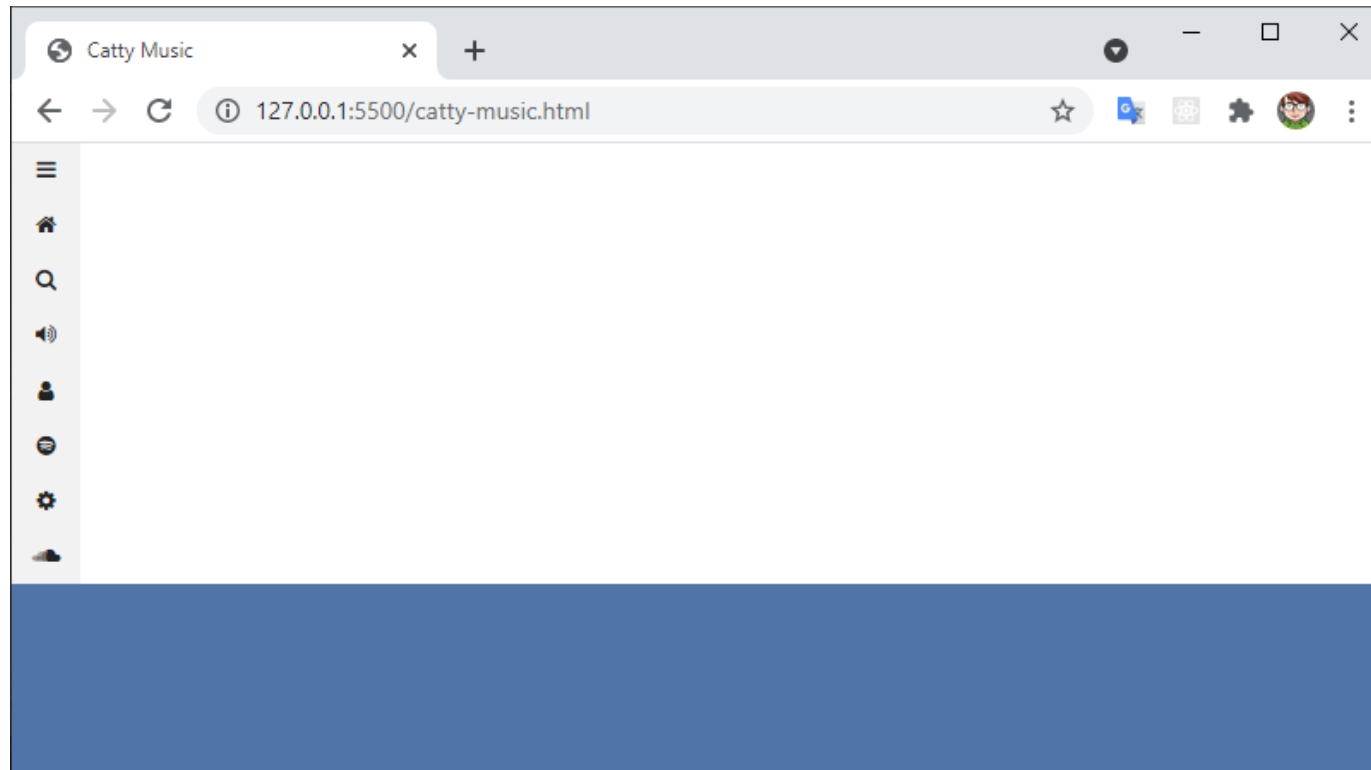
/*font size for the icons*/
aside i.fa {
  font-size: 0.9em;
}
```



Xây dựng bố cục Music App với Flexbox

☐ Thực hành xây dựng bố cục giao diện Music App:

- ♦ Phân tích giao diện hiện tại phần 1a ra **thành 2 phần**: item 1a – A và item 1a – B.
- ♦ Thêm thẻ aside chứa các icon cho thanh sidebar **giúp hiển thị các tính năng** của **giao diện app nghe nhạc**: home, search, user,... Và chỉnh sửa CSS cho sidebar.





- ◆ Tiếp tục viết mã cho item 1a – A và item 1a – B để **thêm nội dung** cho **phần content**
-> chỉnh sửa CSS: **content** hiển thị **container**; **music-head**, **music-list** canh **flex** hàng dọc.

```
.content {
  display: flex;
  flex: 1 1 auto;
  /*this makes sure the section grows to fill the entire available
  space and shrinks too*/
  flex-direction: column;
}

.music-head {
  flex: 0 0 280px;
  /*Same memo, don't grow or shrink -
  stay at 280px*/
  display: flex;
  padding: 40px;
  background-color: #4e4e4e;
}

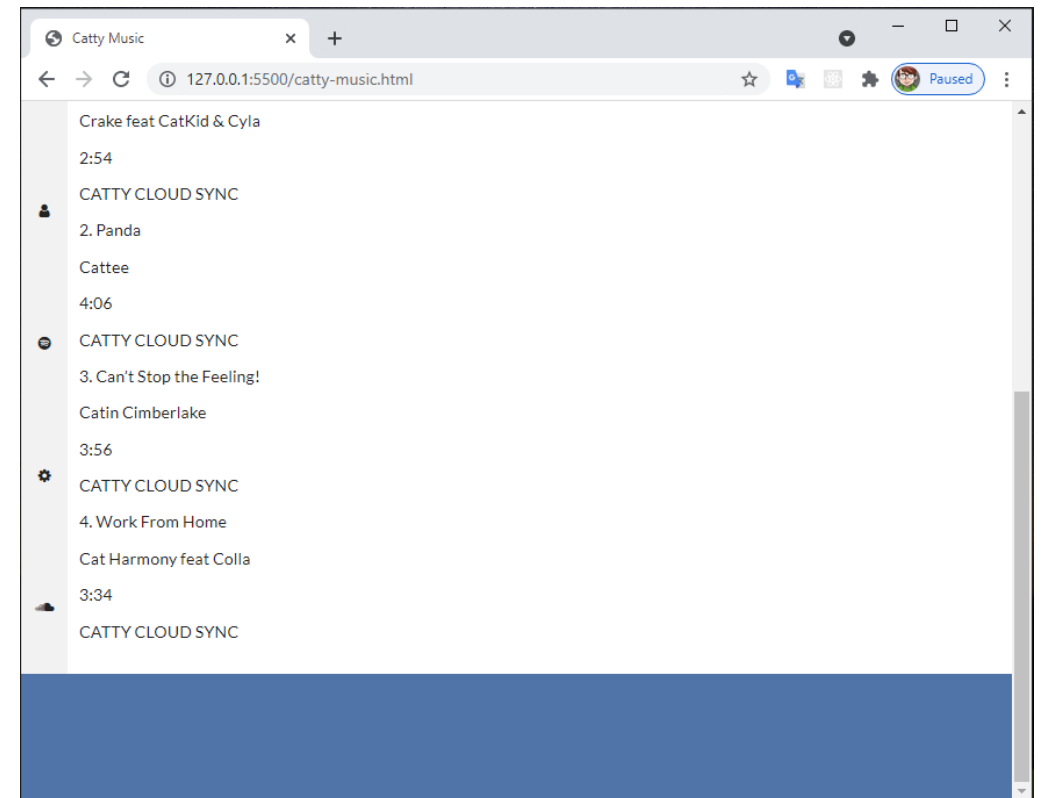
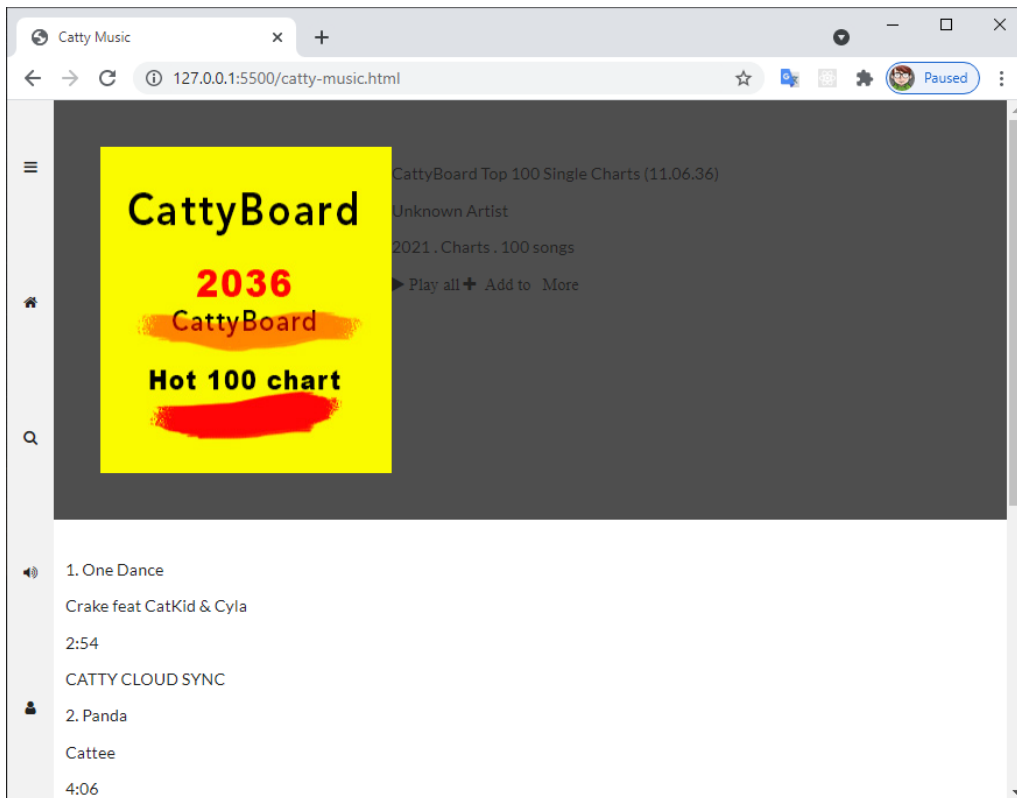
.music-list {
  flex: 1 0 auto;
  list-style-type: none;
  padding: 5px 10px 0px;
}
```




Xây dựng bố cục Music App với Flexbox

❑ Thực hành xây dựng bố cục giao diện Music App:

- ◆ Tiếp tục viết mã cho item 1a – A và item 1a – B để **thêm nội dung** cho **phần content**
- > chỉnh sửa CSS: **content** hiển thị **container**; **music-head**, **music-list** canh **flex** hàng dọc.



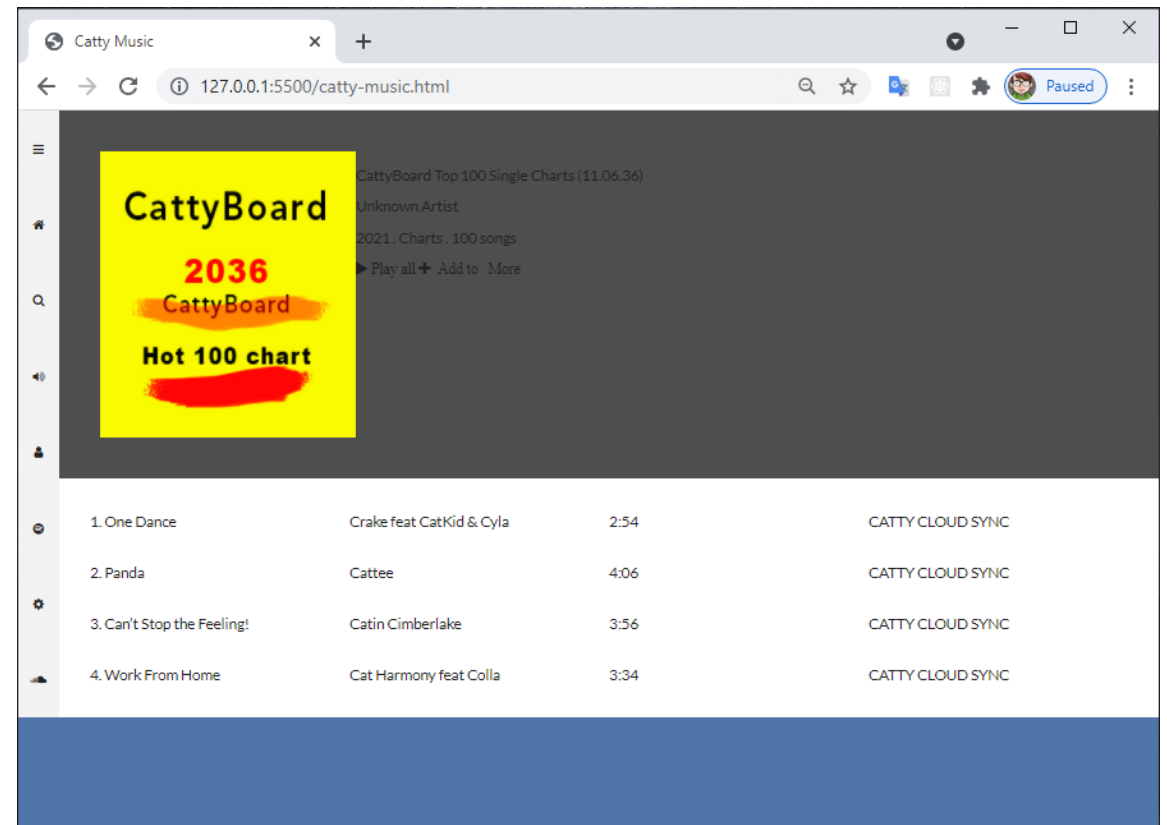


Xây dựng bố cục Music App với Flexbox

❑ Thực hành xây dựng bố cục giao diện Music App:

♦ Canh chỉnh sửa CSS cho danh sách các bài hát: li hiển thị **container**, canh khoảng **padding** cho từng dòng text; li p canh **flex** hàng dọc, độ lớn của **từng p** là 25%.

```
li {  
  /* Paragraphs are now displayed  
  |   on one line */  
  display: flex;  
  /* Some breathing space */  
  padding: 0 20px;  
  min-height: 50px;  
}  
  
li p {  
  /* This is the sweet sauce */  
  flex: 0 0 25%;  
}
```



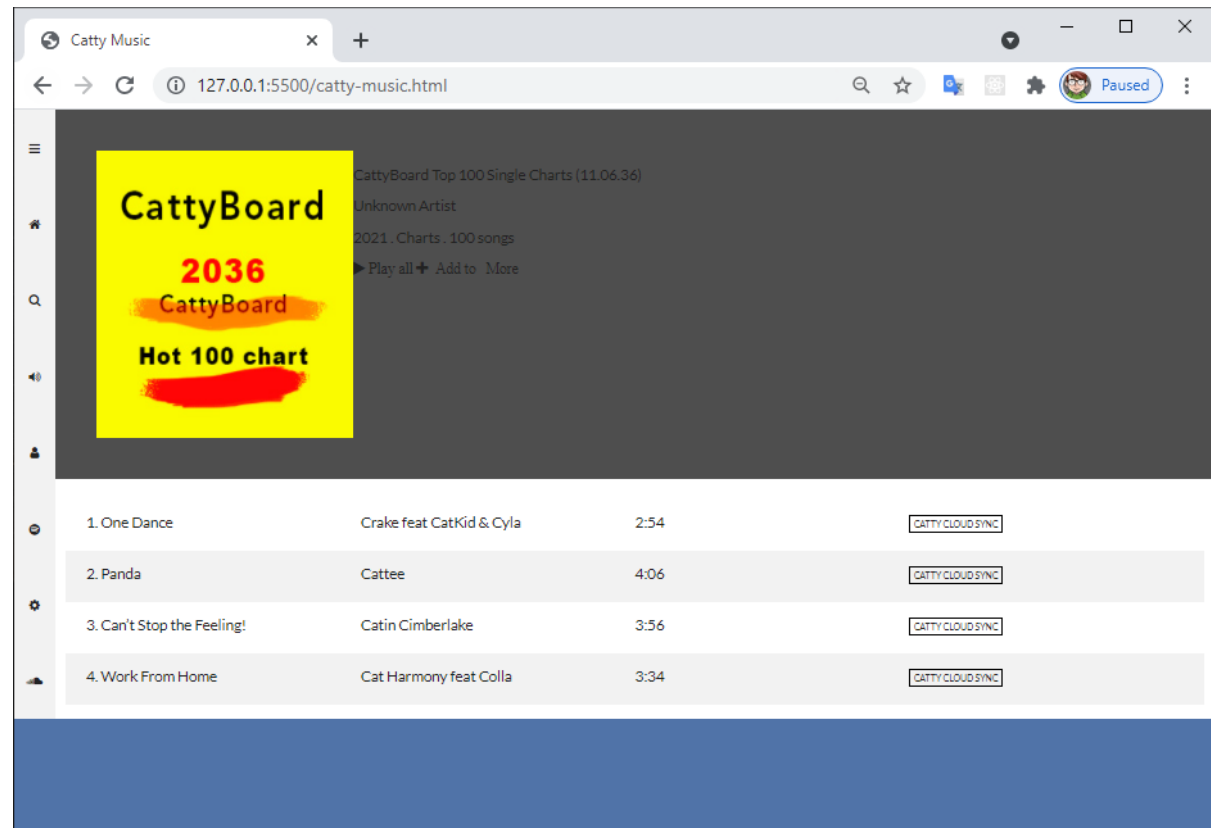


Xây dựng bố cục Music App với Flexbox

❑ Thực hành xây dựng bố cục giao diện Music App:

♦ Canh chỉnh sửa CSS cho danh sách các bài hát: li span với lớp catty-cloud hiển thị **border**, **padding**, **font-size**; các phần tử li:nth-child hiển thị màu nền xám nhạt (#f2f2f2).

```
li p {  
    /* This is the sweet sauce */  
    flex: 0 0 25%;  
}  
  
li span.catty-cloud {  
    border: 1px solid black;  
    font-size: 0.6em;  
    padding: 3px;  
}  
  
li:nth-child(2n) {  
    background-color: #f2f2f2;  
}
```





Xây dựng bố cục Music App với Flexbox

❑ Thực hành xây dựng bố cục giao diện Music App:

♦ Canh chỉnh sửa CSS cho danh sách các bài hát: music-head với lớp catty-music hiển thị **container**, hướng **column**; các phần tử **div:nth-child** của lớp **catty-music** đẹp hơn.

```
.catty-music {  
  flex: 1 1 auto;  
  display: flex;  
  flex-direction: column;  
  font-weight: 300;  
  color: ■ #fff;  
  padding-left: 50px;  
}  
  
.catty-music div:nth-child(1) {  
  margin-bottom: auto;  
}  
  
.catty-music div:nth-child(2) {  
  margin-top: 0;  
}
```

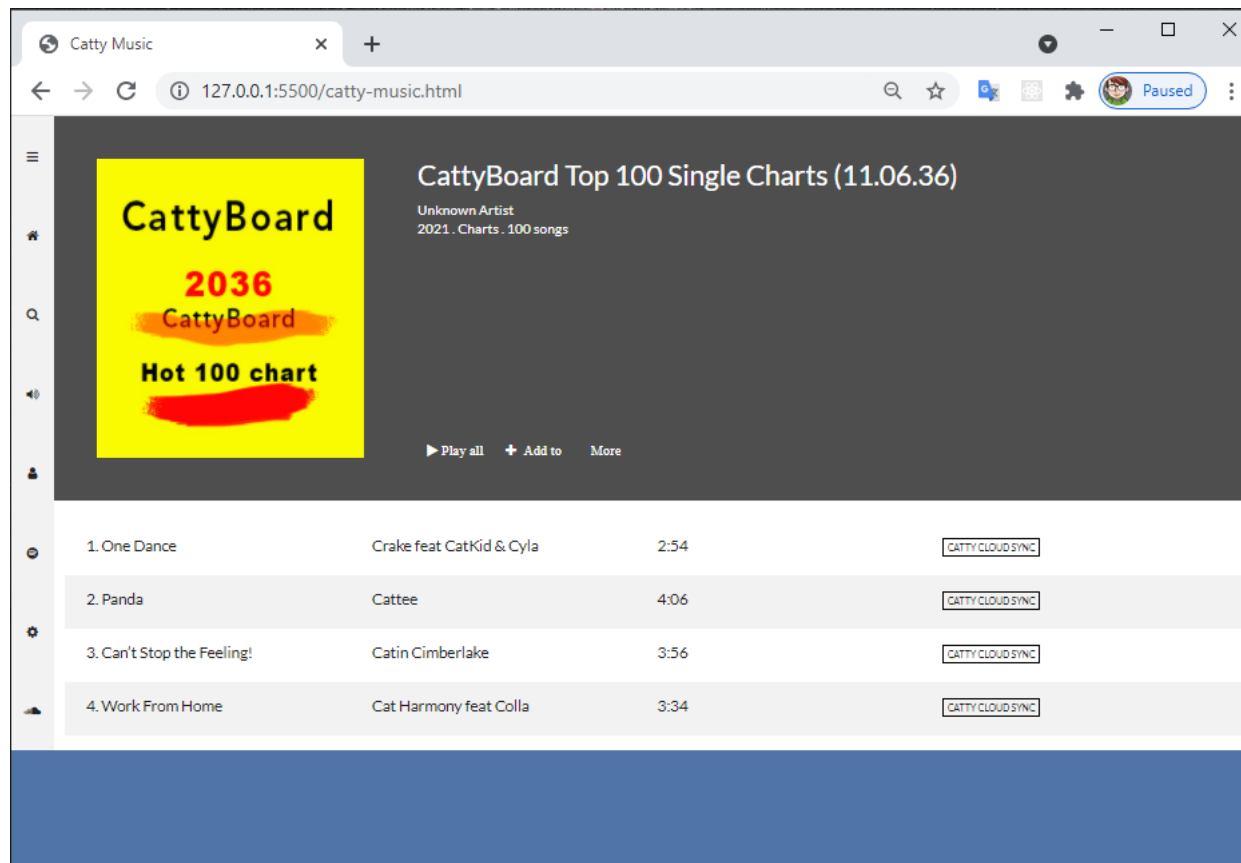
```
.catty-music div:nth-child(2) i.fa {  
  font-size: 0.9em;  
  padding: 0 0.7em;  
  font-weight: 300;  
}  
  
.catty-music div:nth-child(1) p:first-child {  
  font-size: 1.8em;  
  margin: 0 0 10px 0;  
}  
  
.catty-music div:nth-child(1) p:not(:first-child) {  
  font-size: 0.9em;  
  margin: 2px 0;  
}
```



Xây dựng bố cục Music App với Flexbox

☐ Thực hành xây dựng bố cục giao diện Music App:

♦ Canh chỉnh sửa **CSS** cho danh sách các bài hát: **music-head** với lớp **catty-music** hiển thị **container**, hướng **column**; các phần tử **div:nth-child** của lớp **catty-music** đẹp hơn.





Thiết kế Responsive với Flexbox

❑ Thực hiện thiết kế Responsive với Flexbox:

- ◆ Thêm **2 thẻ meta** để màn hình trình duyệt có thể nhận biết viewport của các thiết bị khi truy cập vào trình duyệt đó là **smartphone, tablet, laptop** hay **desktop**.

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Catty Music</title>
  <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">
  <link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/a
  <link href="./css/catty-music.css" rel="stylesheet" type="text/css">
</head>
```

- ◆ Thêm **nội dung** thẻ **html** vào **footer** để chuẩn bị thực hiện hiển thị responsive.

```
<div class="catty-music-small">
  
  <div>
    <p>If It Ain't Love</p>
    <p>Catson Derulo</p>
  </div>
</div>
```

```
<div class="catty-music-controls">
  <i class="fa fa-step-backward"></i>
  <i class="fa fa-pause"></i>
  <i class="fa fa-step-forward"></i>
  <i class="fa fa-rotate-right sm-hide"></i>
  <i class="fa fa-random sm-hide"></i>
</div>
```



Thiết kế Responsive với Flexbox

❑ Thực hiện thiết kế Responsive với Flexbox:

♦ Thêm các lớp **sm-hide**, **sm-text-center**, **sm-text-right** để chuẩn bị cho việc sử dụng **media queries** thực hiện Responsive Design.

- Thêm các lớp trên vào phần **aside**, **section class="catty-music"**

```
<aside class="sm-hide">
  <i class="fa fa-bars"></i>
  <i class="fa fa-home"></i>
  <i class="fa fa-search"></i>
  <i class="fa fa-volume-up"></i>
  <i class="fa fa-user"></i>
  <i class="fa fa-spotify"></i>
  <i class="fa fa-cog"></i>
  <i class="fa fa-soundcloud"></i>
</aside>
```

```
<section class="catty-music">
  <div>
    <p class="sm-text-center">CattyBoard Top 100 Single
    <p class="sm-hide">Unknown Artist</p>
    <p class="sm-hide">2021 . Charts . 100 songs</p>
  </div>
  <div class="sm-text-center">
    <!--Music controls-->
    <i class="fa fa-play">&nbsp;Play all</i>
    <i class="fa fa-plus"> &nbsp;Add to</i>
    <i class="fa fa-ellipsis-h">&nbsp;&nbsp;More</i>
  </div>
</section>
```



Thiết kế Responsive với Flexbox

❑ Thực hiện thiết kế Responsive với Flexbox:

♦ Thêm các lớp **sm-hide**, **sm-text-center**, **sm-text-right** để chuẩn bị cho việc sử dụng **media queries** thực hiện Responsive Design.

- Thêm các lớp trên vào phần **ul class="music-list"**, các **li**, và phần **footer** như sau:

```
<ul class="music-list">
  <li>
    <p>1. One Dance</p>
    <p class="sm-hide">Crake feat CatKid & Cyla</p>
    <p class="sm-text-right">2:54</p>
    <p class="sm-hide"><span class="catty-cloud">CATTY CLOUD
      SYNC</span></p>
  </li>
  <li>...
</li>
  <li>...
</li>
  <li>...
</li>
</ul>
```

```
<footer>
  <div class="catty-music-small">
    
    <div>
      <p>If It Ain't Love</p>
      <p>Catson Derulo</p>
    </div>
  </div>

  <div class="catty-music-controls">
    <i class="fa fa-step-backward"></i>
    <i class="fa fa-pause"></i>
    <i class="fa fa-step-forward"></i>
    <i class="fa fa-rotate-right sm-hide"></i>
    <i class="fa fa-random sm-hide"></i>
  </div>
</footer>
```

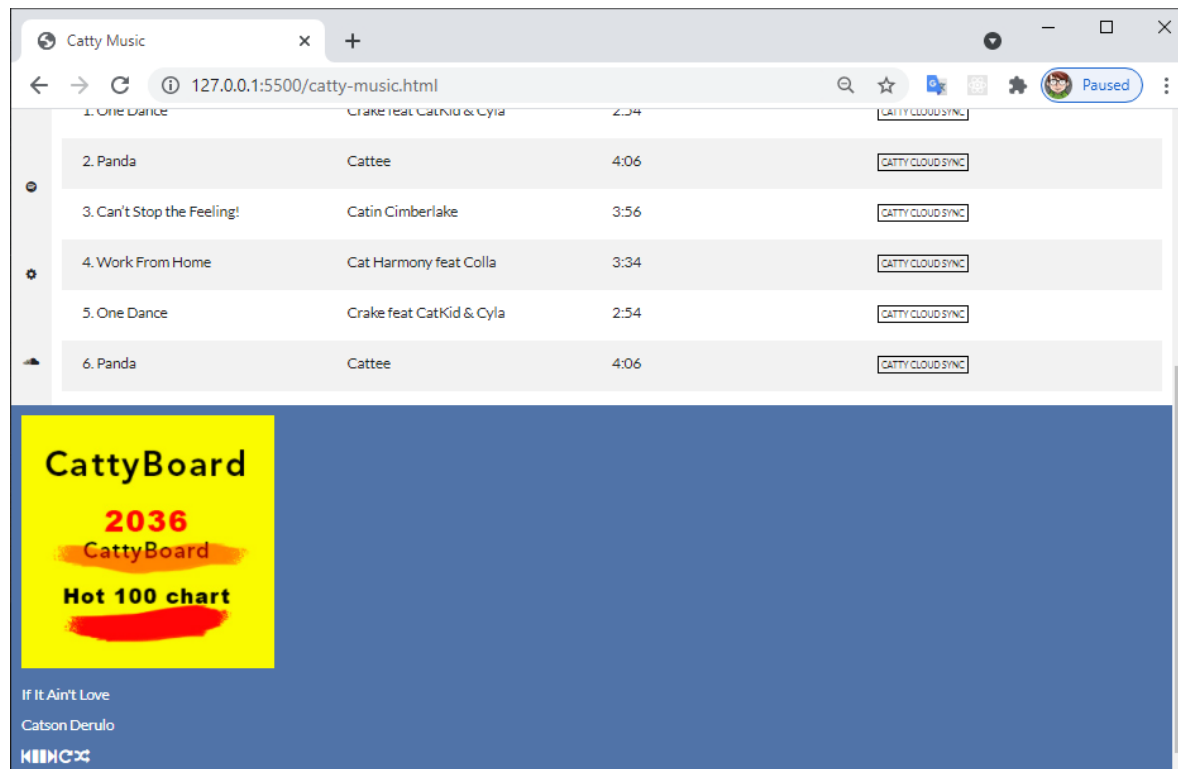



Thiết kế Responsive với Flexbox

❑ Thực hiện thiết kế Responsive với Flexbox:

♦ Thêm các lớp **sm-hide**, **sm-text-center**, **sm-text-right** để chuẩn bị cho việc sử dụng **media queries** thực hiện Responsive Design.

- Sau khi thêm các lớp trên, ta có giao diện hiện tại chỉ thay đổi nội dung ở footer.

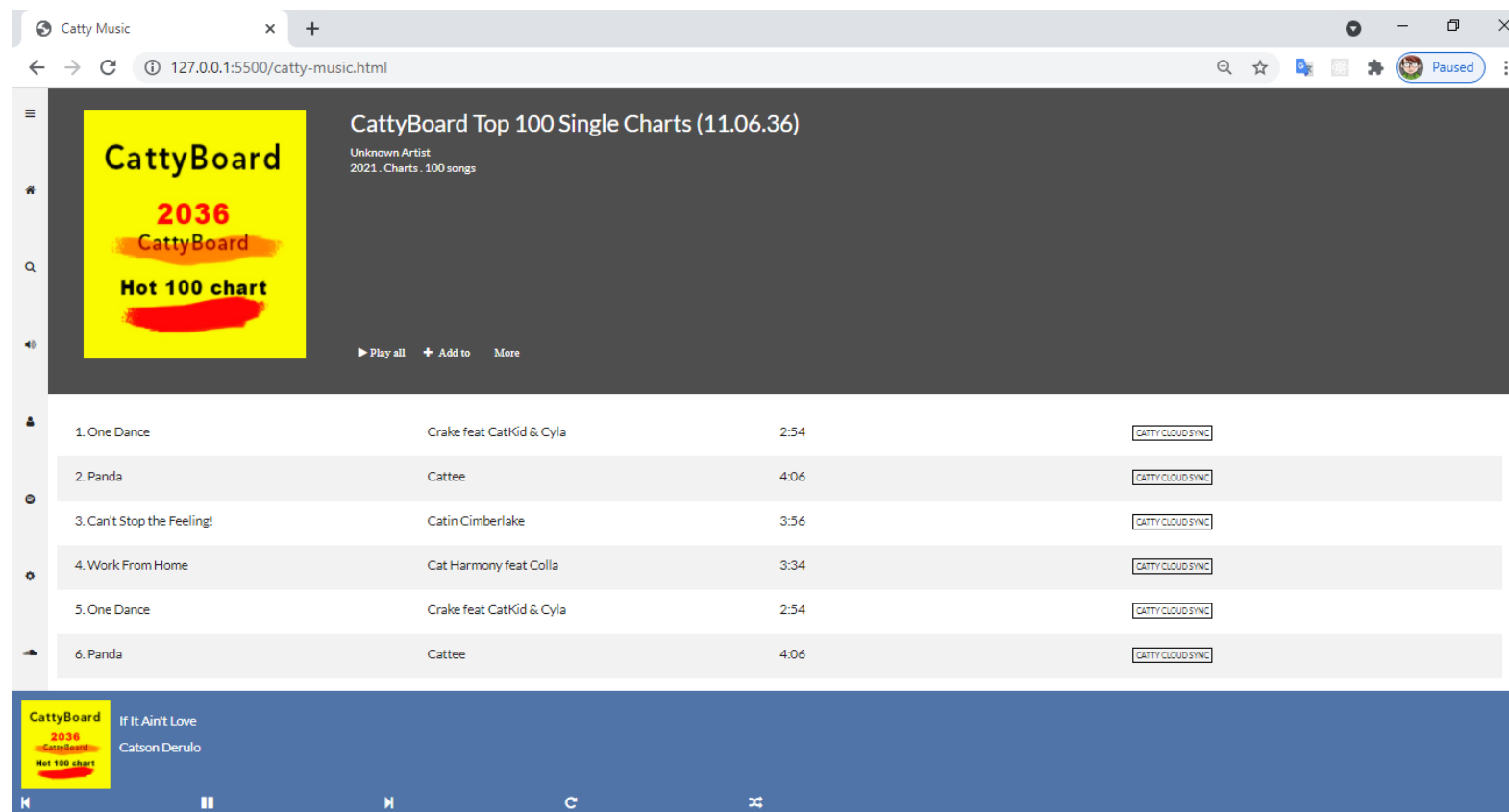




Thiết kế Responsive với Flexbox

❑ Thực hiện thiết kế Responsive với Flexbox:

♦ Chỉnh sửa CSS cho các lớp **sm-hide**, **sm-text-center**, **sm-text-right** và sử dụng **media queries** đã học để thực hiện Responsive. Trên desktop, laptop sẽ hiển thị như sau:

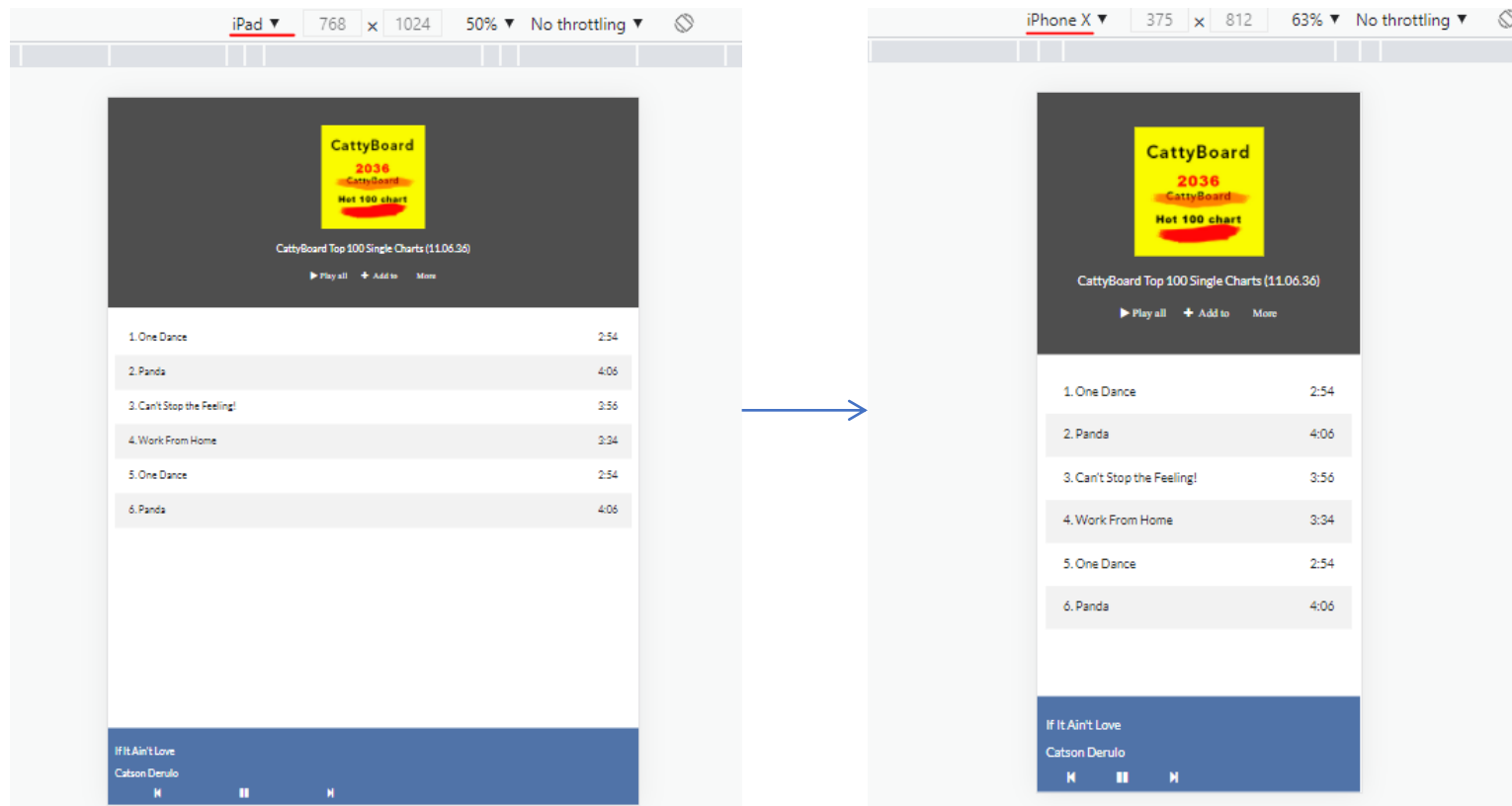




Thiết kế Responsive với Flexbox

❑ Thực hiện thiết kế Responsive với Flexbox:

♦ **Chỉn sửa CSS cho các lớp `sm-hide`, `sm-text-center`, `sm-text-right` và sử dụng `media queries` đã học để thực hiện Responsive. Trên tablet, mobile sẽ hiển thị như sau:**





Tổng kết nội dung bài học

- ☐ Tổng quan về bố cục Flexbox
- ☐ Các thuộc tính của Flex Container
- ☐ Các thuộc tính của Flex Item
- ☐ Canh chỉnh giao diện với Auto Margin
- ☐ Xây dựng bố cục Music App với Flexbox
- ☐ Thiết kế Responsive với Flexbox

Let's
Recap

