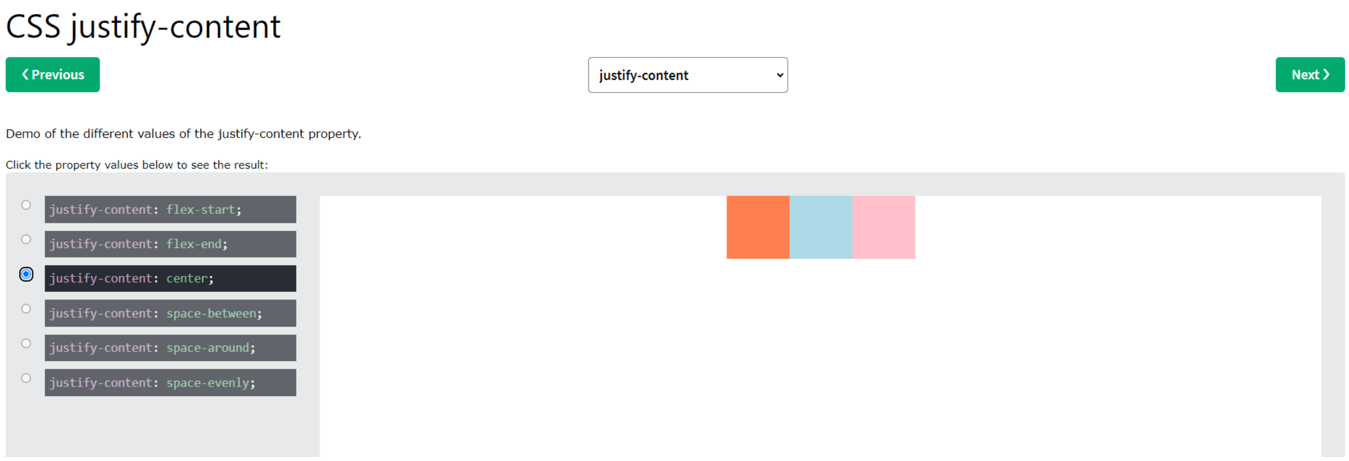
container-row
橫 rwd

container-column
直 沒有rwd
要先設定height 1000px
讓item height 300px切換

https://www.w3schools.com/cssref/playdemo.php?filename=playcss_justify-content

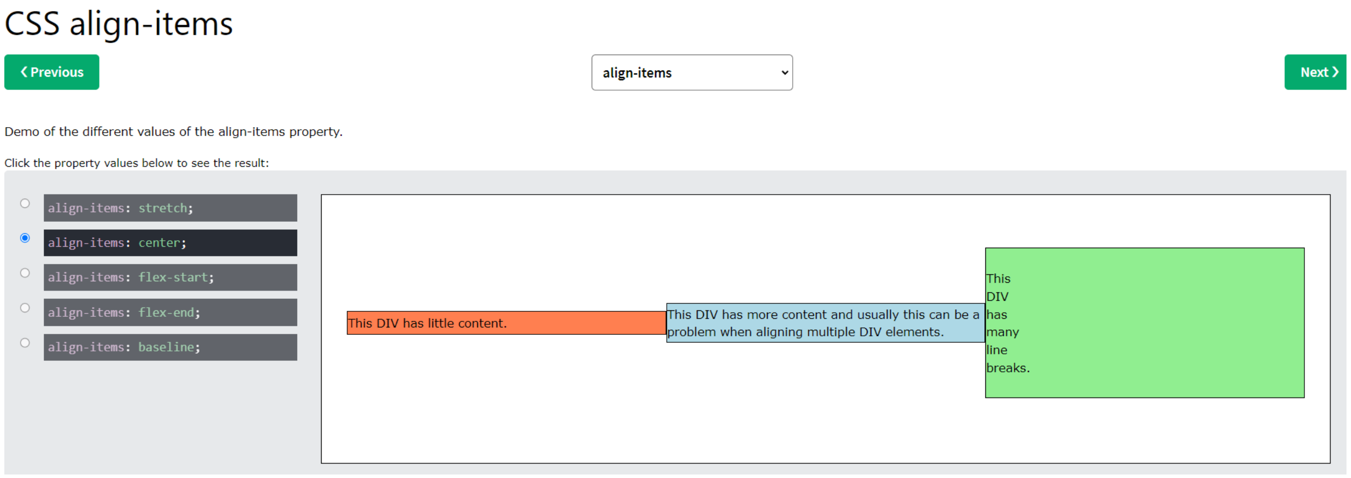


Demo of the different values of the justify-content property.

Click the property values below to see the result:

○ justify-content: flex-start;
○ justify-content: flex-end;
● justify-content: center;
○ justify-content: space-between;
○ justify-content: space-around;
○ justify-content: space-evenly;

https://www.w3schools.com/cssref/playdemo.php?filename=playcss_align-items



CSS align-items

Demo of the different values of the align-items property.

Click the property values below to see the result:

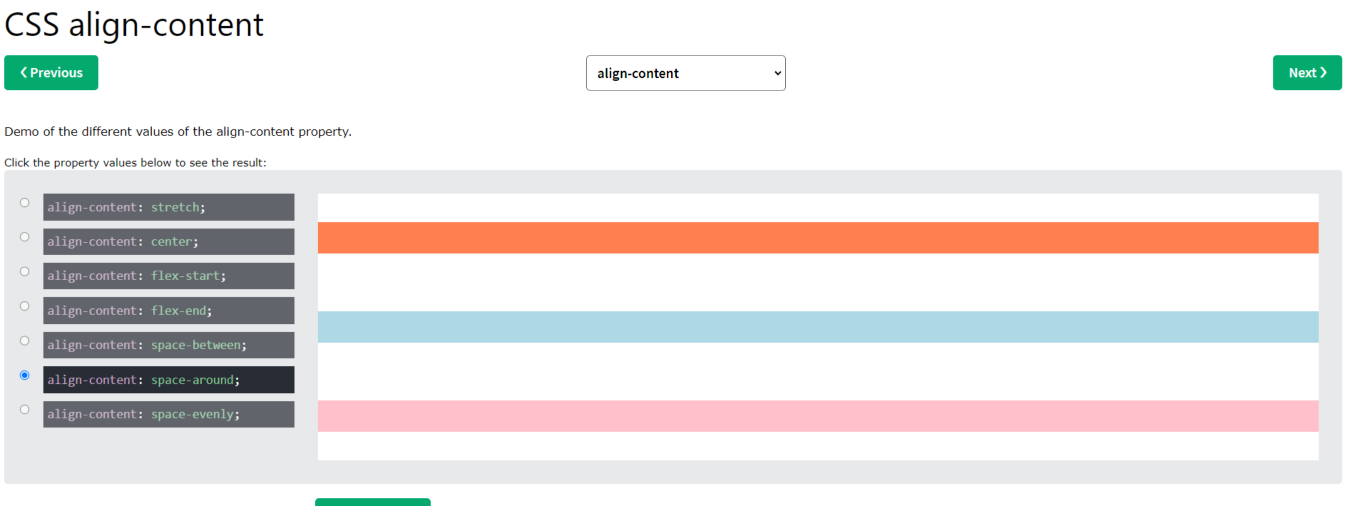
○ `align-items: stretch;`
● `align-items: center;`
○ `align-items: flex-start;`
○ `align-items: flex-end;`
○ `align-items: baseline;`

This DIV has little content.

This DIV has more content and usually this can be a problem when aligning multiple DIV elements.

This DIV has many line breaks.

https://www.w3schools.com/cssref/css3_pr_align-content.php

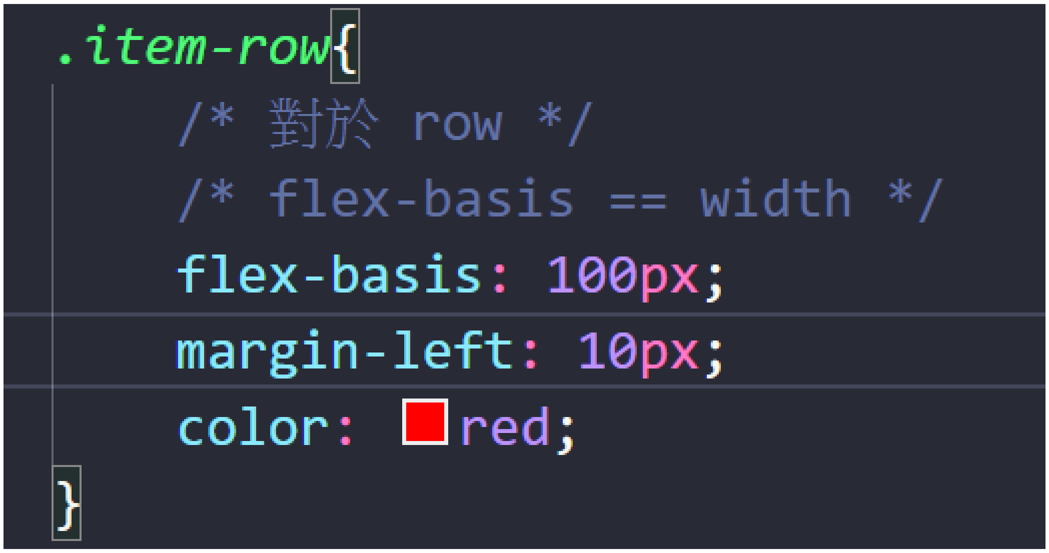


CSS align-content

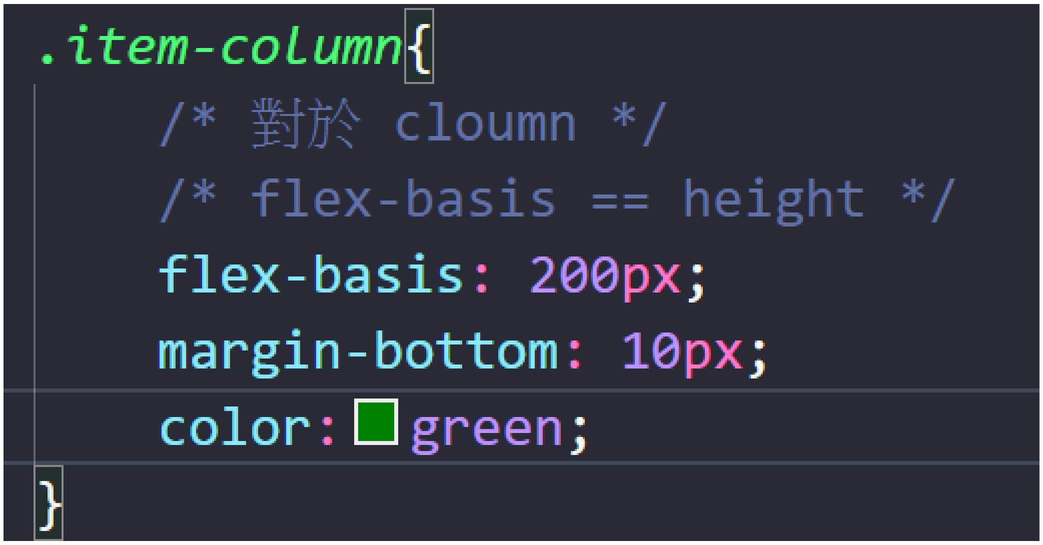
Demo of the different values of the align-content property.

Click the property values below to see the result:

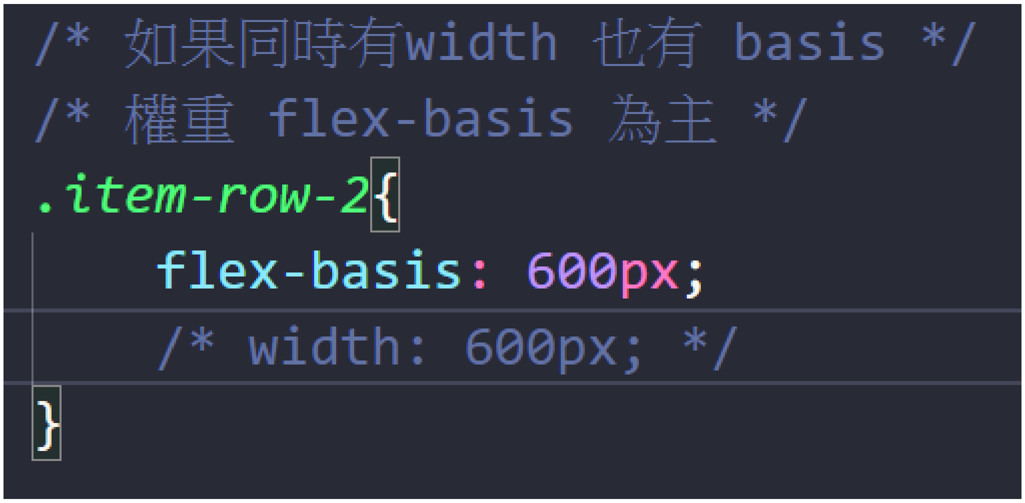
○ `align-content: stretch;`
○ `align-content: center;`
○ `align-content: flex-start;`
○ `align-content: flex-end;`
○ `align-content: space-between;`
● `align-content: space-around;`
○ `align-content: space-evenly;`



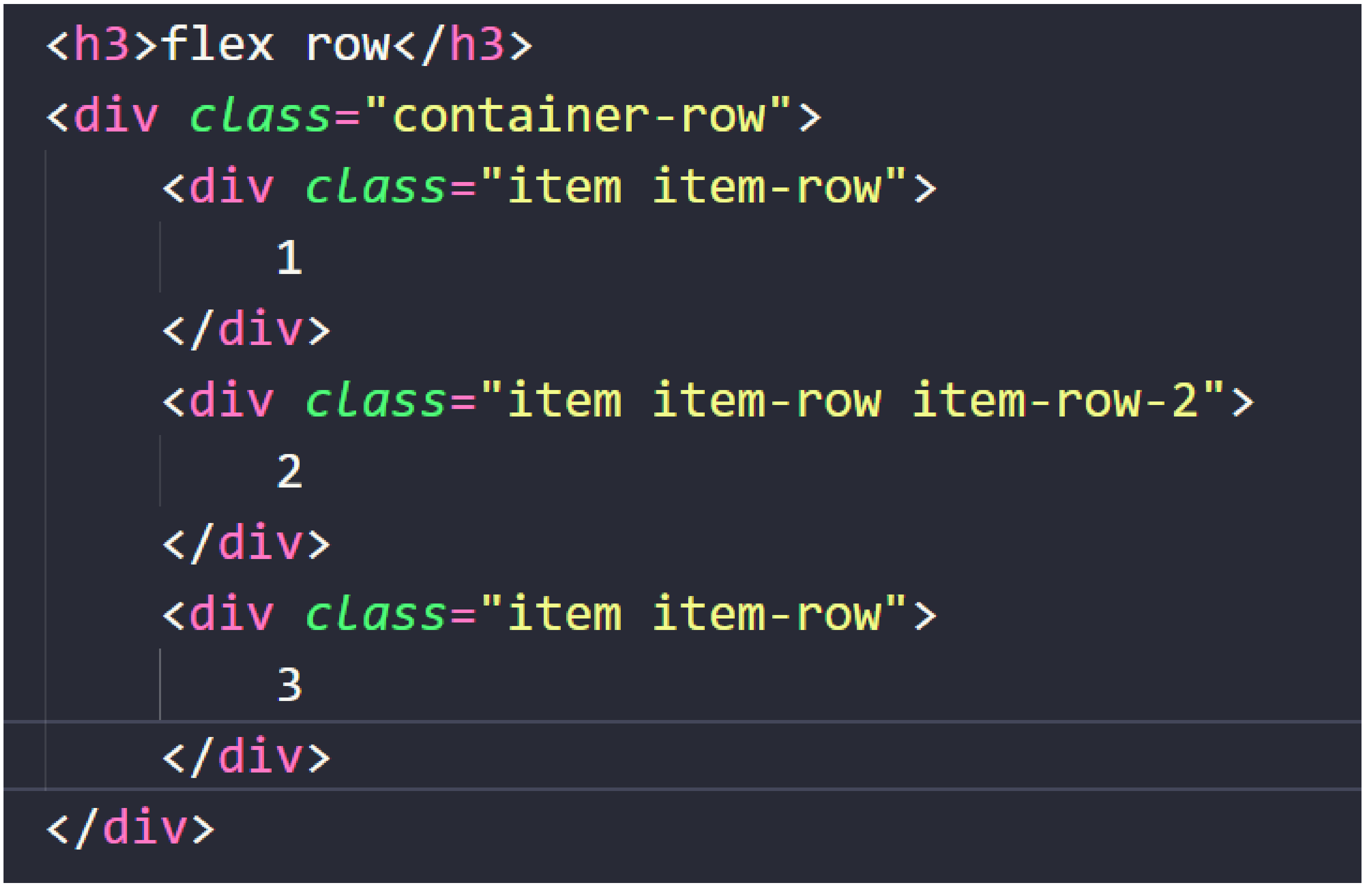
```css
.item-row{
    /* 對於 row */
    /* flex-basis == width */
    flex-basis: 100px;
    margin-left: 10px;
    color: 🟥red;
}
```



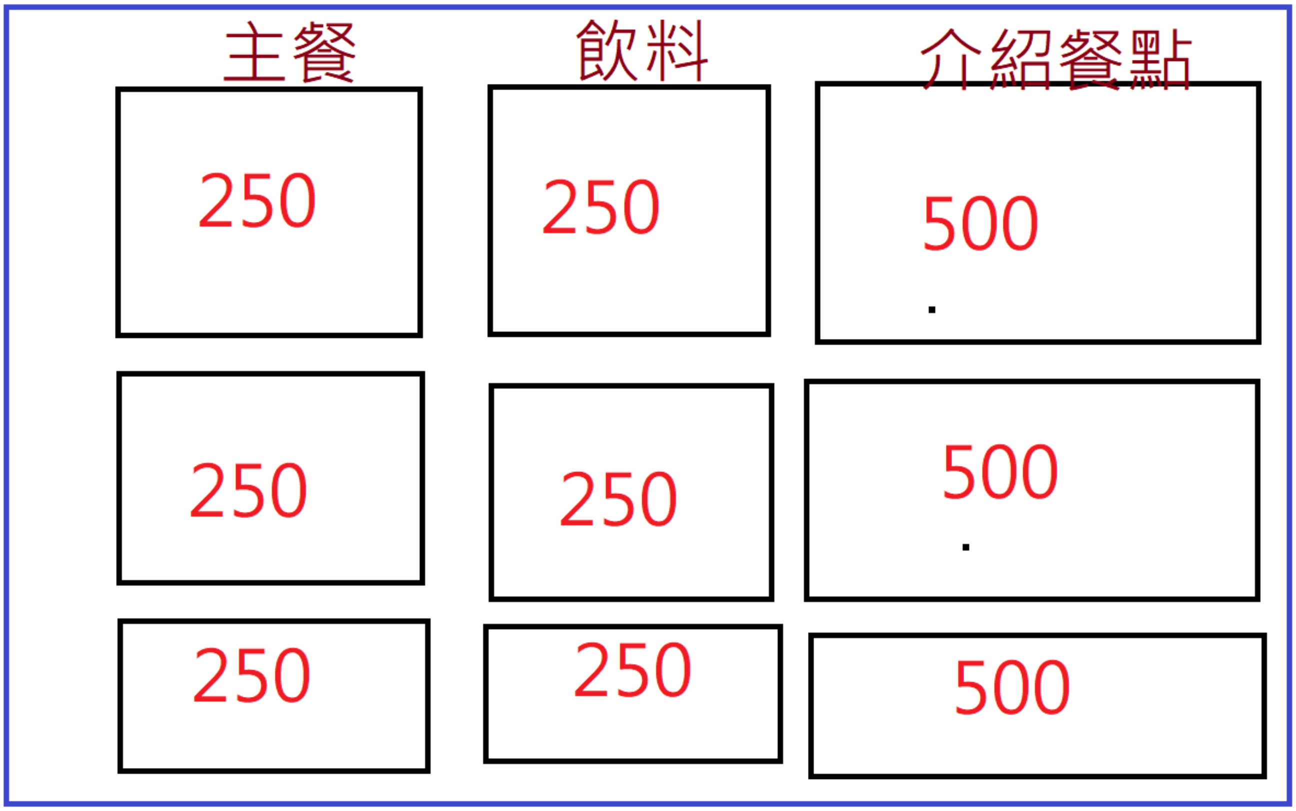
```css
.item-column{
    /* 對於 cloumn */
    /* flex-basis == height */
    flex-basis: 200px;
    margin-bottom: 10px;
    color:🟩green;
}
```



```css
/* 如果同時有width 也有 basis */
/* 權重 flex-basis 為主 */
.item-row-2{
    flex-basis: 600px;
    /* width: 600px; */
}
```



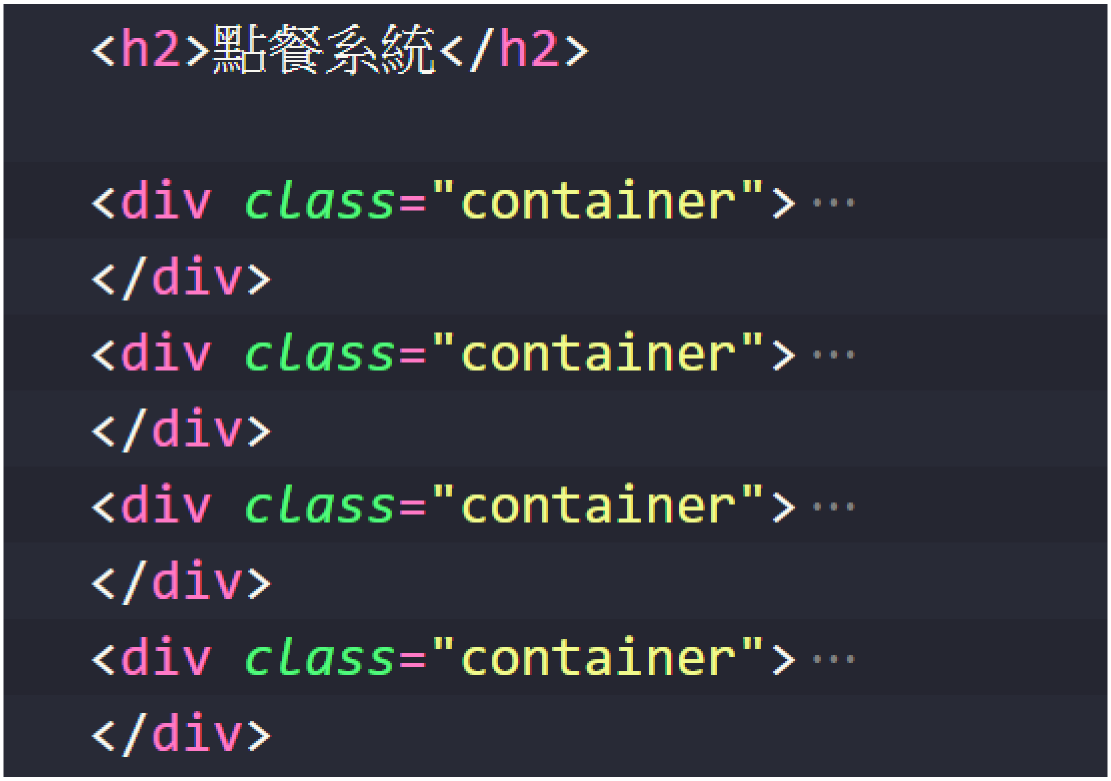
```html
<h3>flex row</h3>
<div class="container-row">
    <div class="item item-row">
        1
    </div>
    <div class="item item-row item-row-2">
        2
    </div>
    <div class="item item-row">
        3
    </div>
</div>
```



1.html
container
item
2.css
container bg-color
item bg-color
3.set flexbox container / item
4.完成一組 正確 container + 3 item 照片 文字.....
5.把上面整組container+3item copy 總共三組
6.把上面整組container+3item copy 貼上一次
更改為title

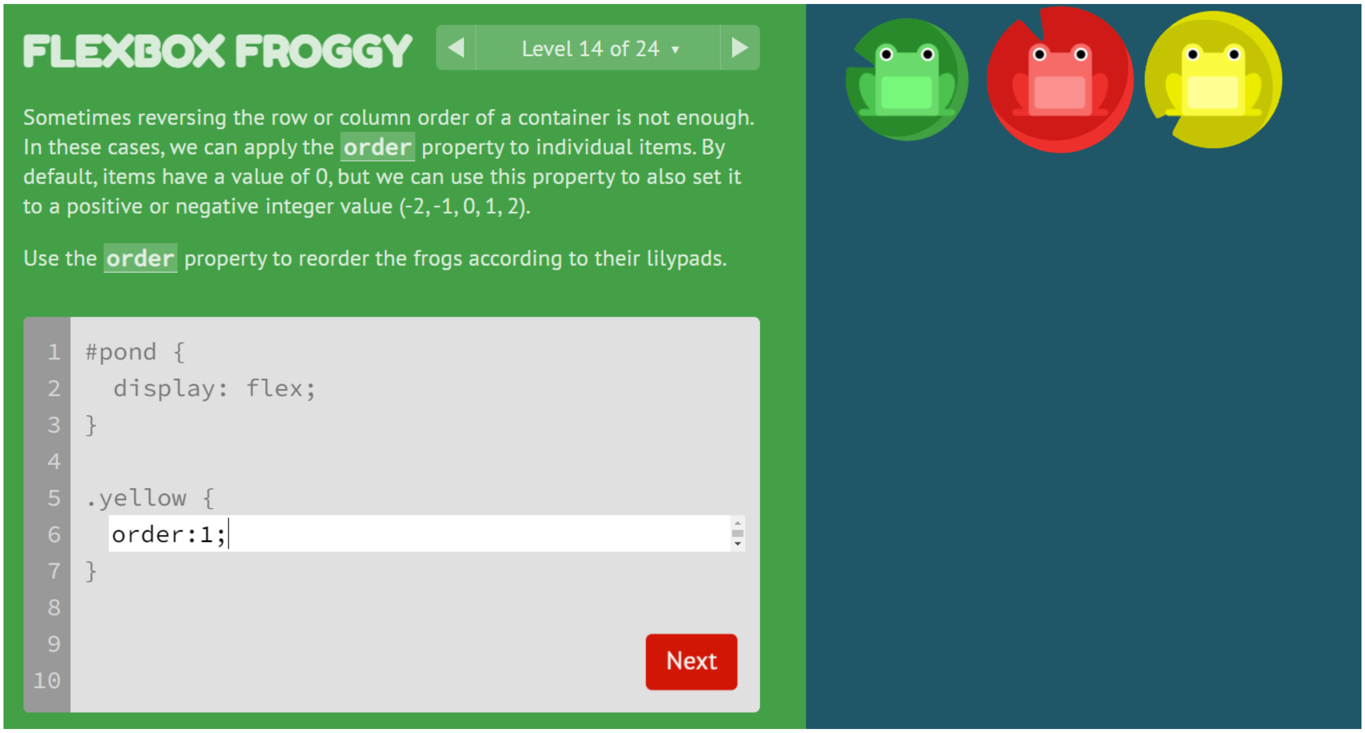
container-title
container
container

container



```
<h2>點餐系統</h2>

<div class="container">···
</div>
<div class="container">···
</div>
<div class="container">···
</div>
<div class="container">···
</div>
```

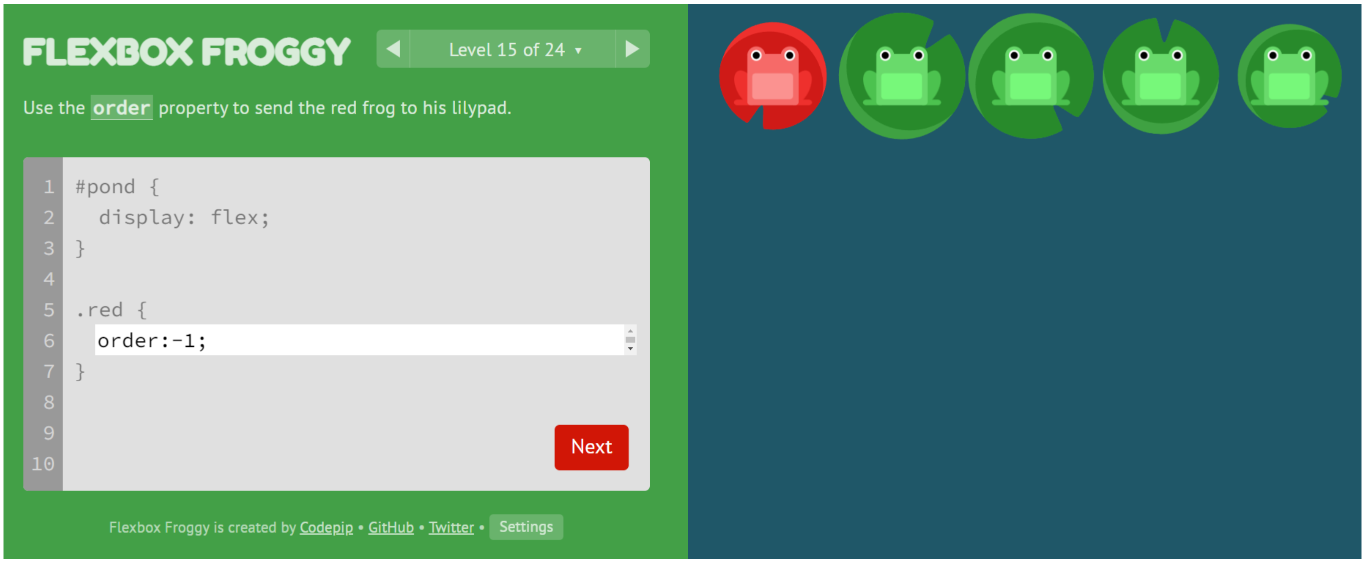


Sometimes reversing the row or column order of a container is not enough. In these cases, we can apply the `order` property to individual items. By default, items have a value of 0, but we can use this property to also set it to a positive or negative integer value (-2, -1, 0, 1, 2).

Use the `order` property to reorder the frogs according to their lilypads.

```
#pond {
  display: flex;
}

.yellow {
  order:1;
}
```

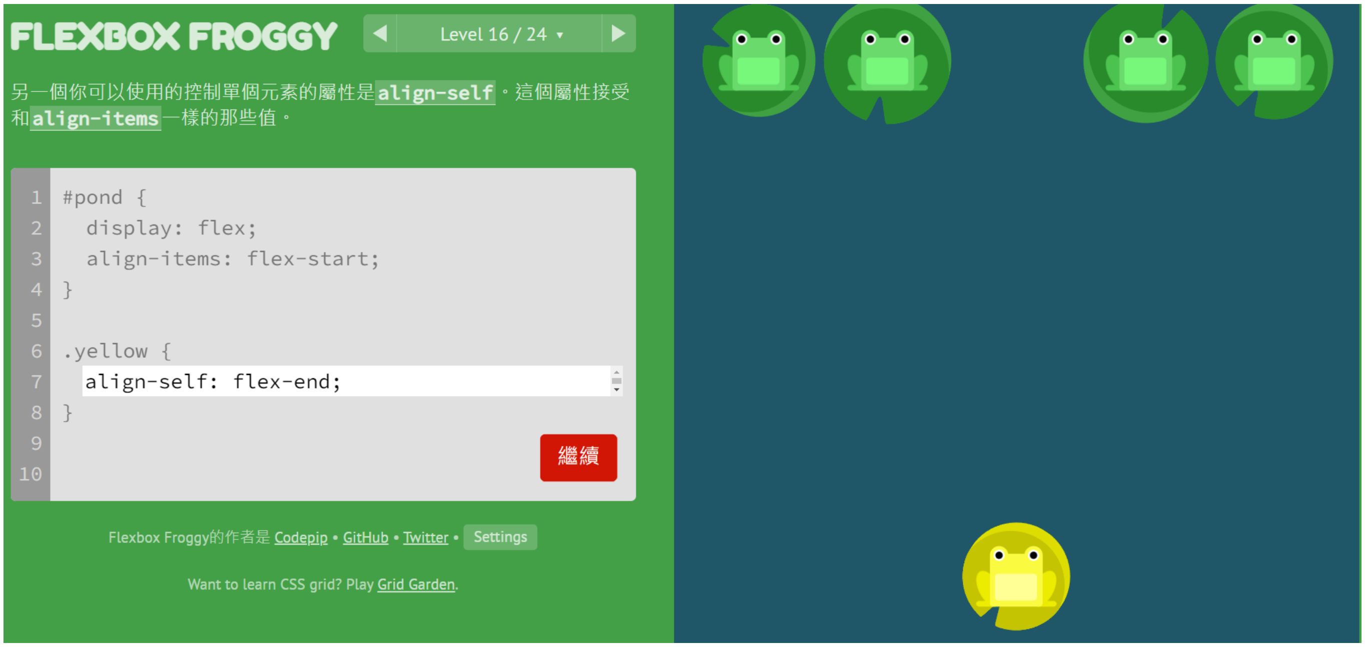


Use the order property to send the red frog to his lilypad.

```
#pond {
  display: flex;
}

.red {
  order:-1;
}
```

Flexbox Froggy is created by Codepip • GitHub • Twitter •



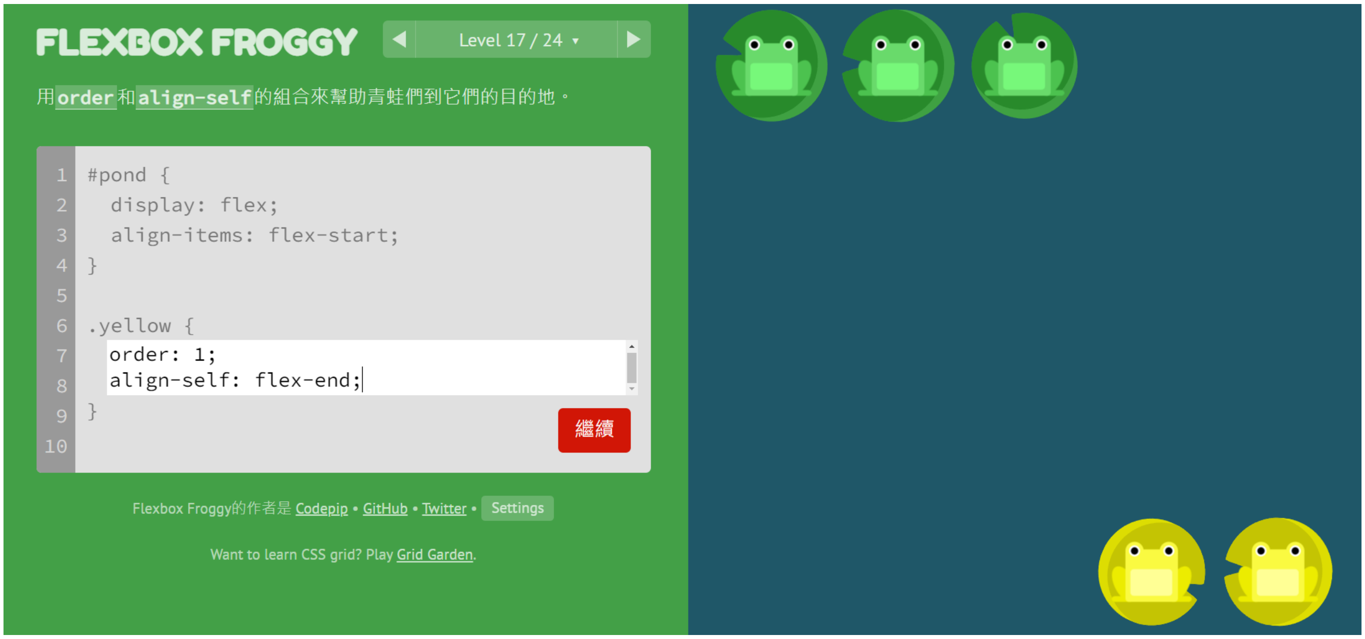
另一個你可以使用的控制單個元素的屬性是align-self。這個屬性接受和align-items一樣的那些值。

```
#pond {
  display: flex;
  align-items: flex-start;
}

.yellow {
  align-self: flex-end;
}
```

Flexbox Froggy的作者是 Codepip • GitHub • Twitter •

Want to learn CSS grid? Play Grid Garden.



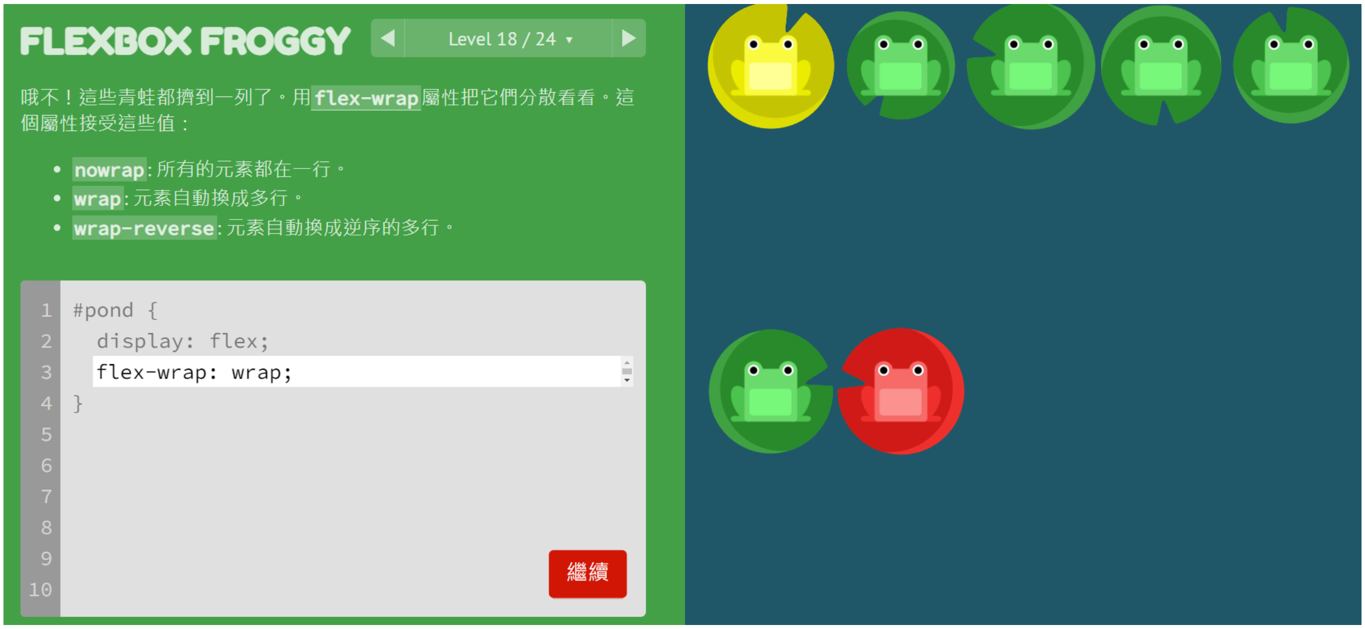
用`order`和`align-self`的組合來幫助青蛙們到它們的目的地。

```
#pond {
  display: flex;
  align-items: flex-start;
}

.yellow {
  order: 1;
  align-self: flex-end;
}
```

Flexbox Froggy的作者是 Codepip • GitHub • Twitter •

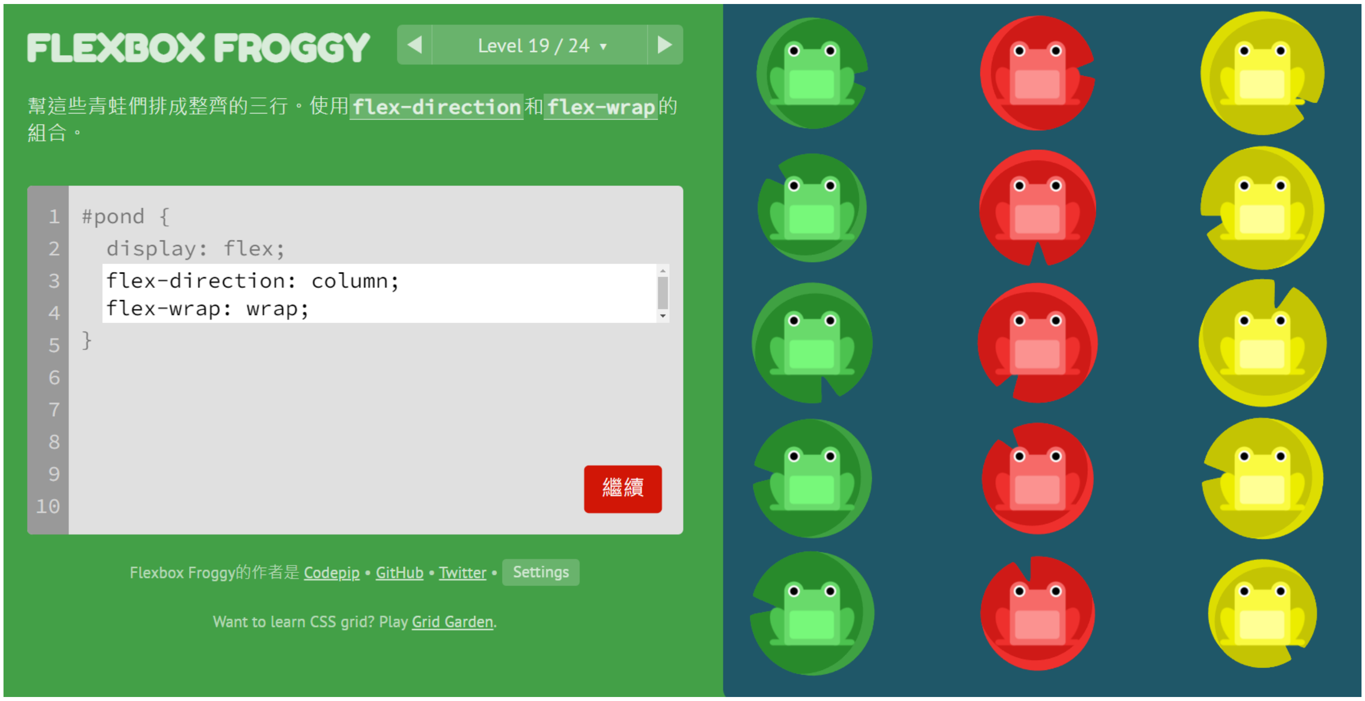
Want to learn CSS grid? Play Grid Garden.



哦不！這些青蛙都擠到一列了。用`flex-wrap`屬性把它們分散看看。這個屬性接受這些值：

- `nowrap`: 所有的元素都在一行。
- `wrap`: 元素自動換成多行。
- `wrap-reverse`: 元素自動換成逆序的多行。

```
#pond {
  display: flex;
  flex-wrap: wrap;
}
```

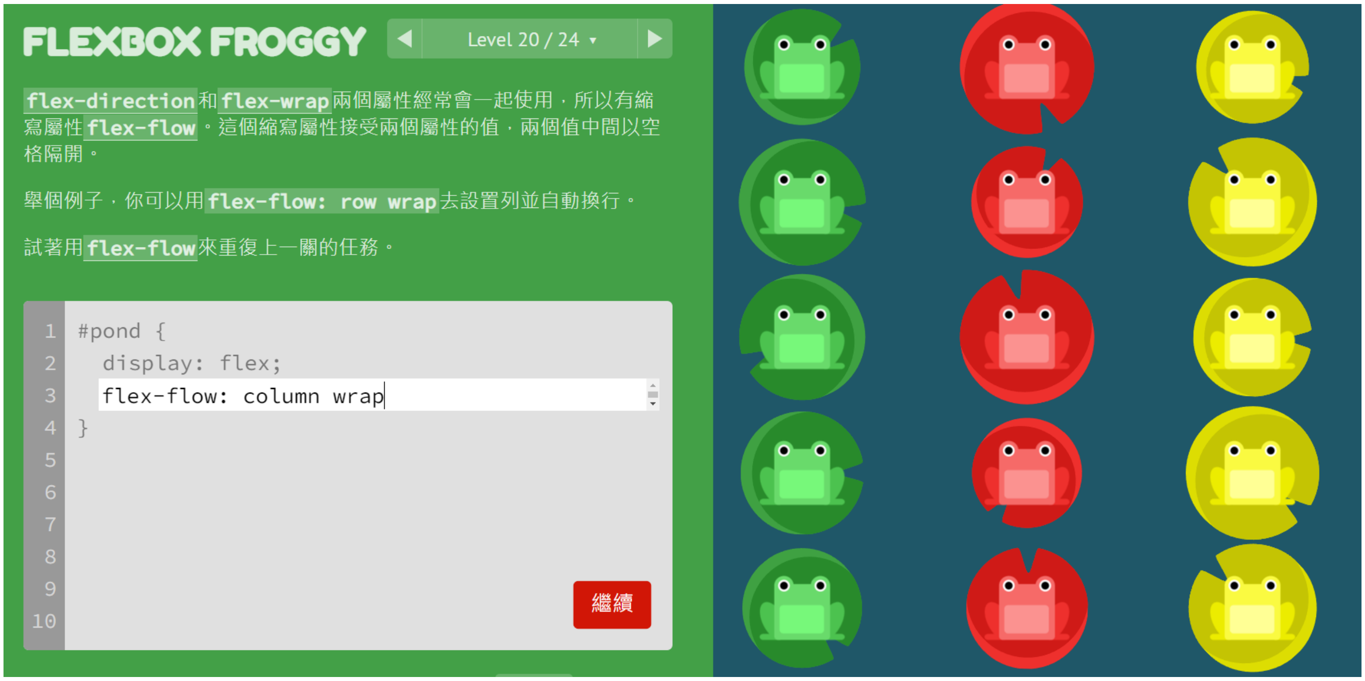


幫這些青蛙們排成整齊的三行。使用`flex-direction`和`flex-wrap`的組合。

```
#pond {
  display: flex;
  flex-direction: column;
  flex-wrap: wrap;
}
```

Flexbox Froggy的作者是 Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.

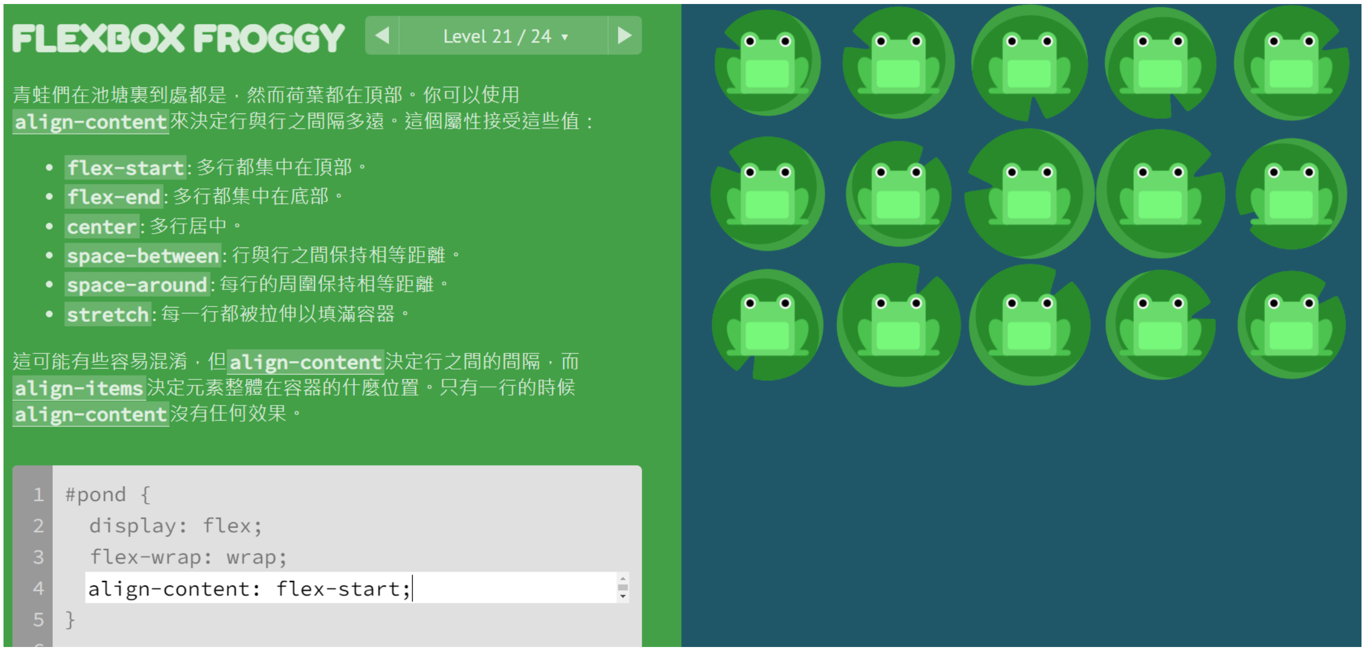


`flex-direction`和`flex-wrap`兩個屬性經常會一起使用，所以有縮寫屬性`flex-flow`。這個縮寫屬性接受兩個屬性的值，兩個值中間以空格隔開。

舉個例子，你可以用`flex-flow: row wrap`去設置列並自動換行。

試著用`flex-flow`來重復上一關的任務。

```
#pond {
  display: flex;
  flex-flow: column wrap
}
```

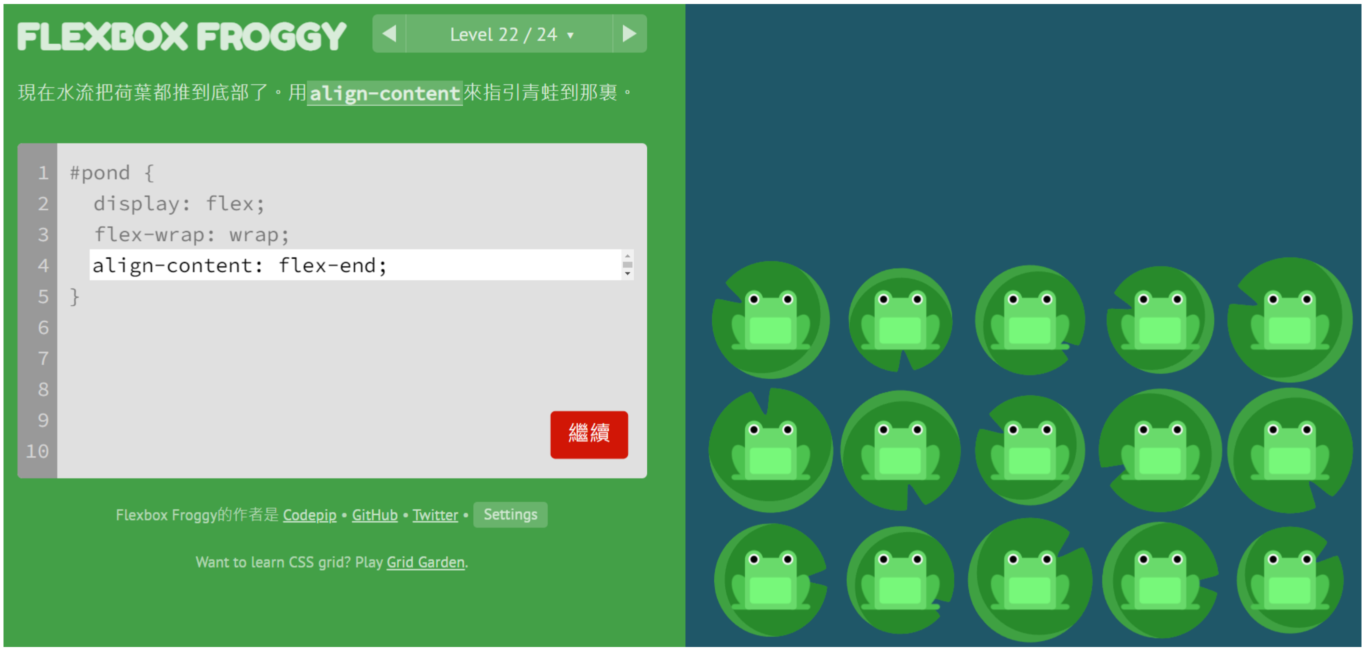


青蛙們在池塘裏到處都是，然而荷葉都在頂部。你可以使用`align-content`來決定行與行之間隔多遠。這個屬性接受這些值：

- `flex-start`: 多行都集中在頂部。
- `flex-end`: 多行都集中在底部。
- `center`: 多行居中。
- `space-between`: 行與行之間保持相等距離。
- `space-around`: 每行的周圍保持相等距離。
- `stretch`: 每一行都被拉伸以填滿容器。

這可能有些容易混淆，但`align-content`決定行之間的間隔，而`align-items`決定元素整體在容器的什麼位置。只有一行的時候`align-content`沒有任何效果。

```
#pond {
  display: flex;
  flex-wrap: wrap;
  align-content: flex-start;
}
```

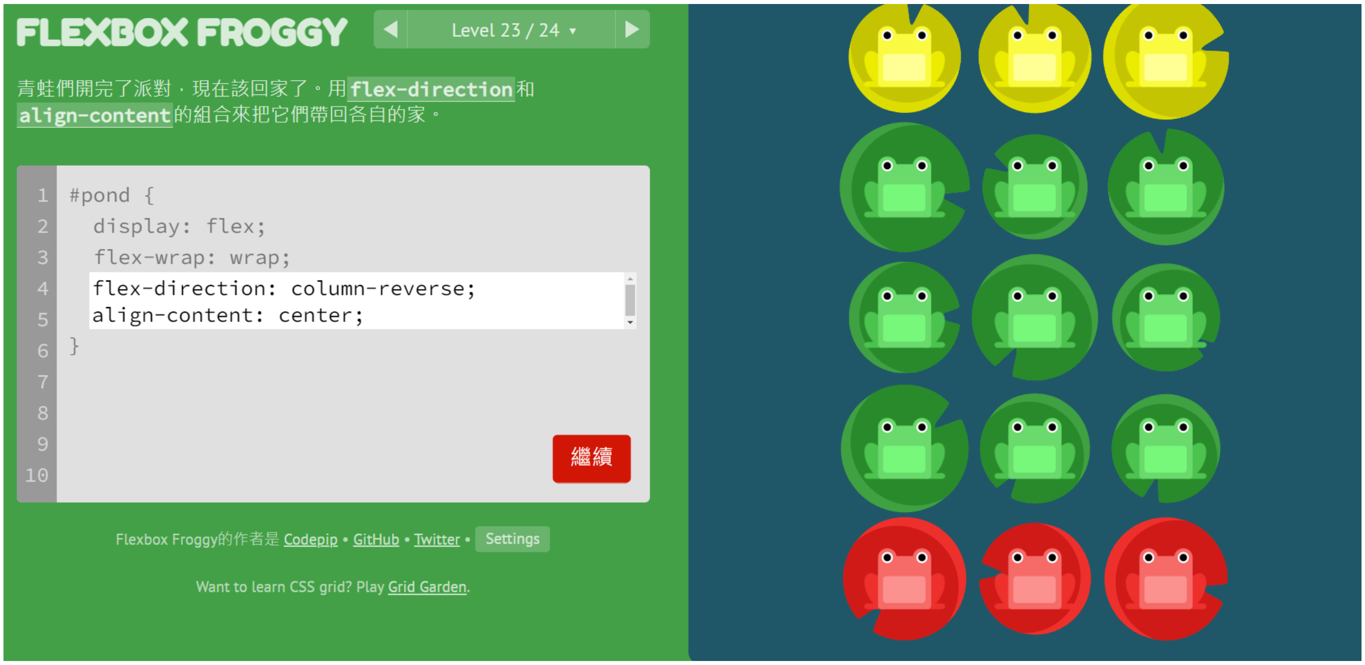


現在水流把荷葉都推到底部了。用`align-content`來指引青蛙到那裏。

```
#pond {
  display: flex;
  flex-wrap: wrap;
  align-content: flex-end;
}
```

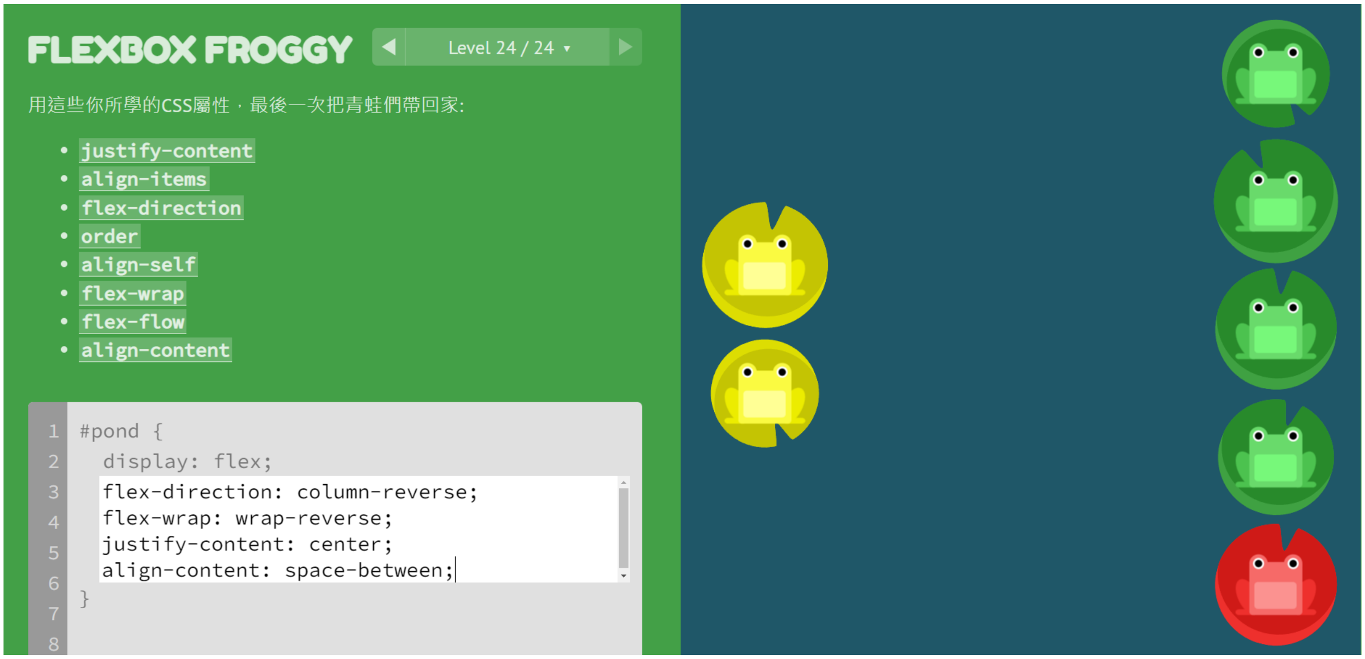
Flexbox Froggy的作者是 Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.


FLEXBOX FROGGY
◀ Level 23 / 24 ▾ ▶
繼續
Flexbox Froggy的作者是 Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden.

青蛙們開完了派對，現在該回家了。用flex-direction和align-content的組合來把它們帶回各自的家。

```
#pond {
  display: flex;
  flex-wrap: wrap;
  flex-direction: column-reverse;
  align-content: center;
}
```



用這些你所學的CSS屬性，最後一次把青蛙們帶回家:

- justify-content
- align-items
- flex-direction
- order
- align-self
- flex-wrap
- flex-flow
- align-content

```
#pond {
  display: flex;
  flex-direction: column-reverse;
  flex-wrap: wrap-reverse;
  justify-content: center;
  align-content: space-between;
}
```

hw0413

1.flexbox

2.flex-direction

3.justfy-content

4.align-items

5.flex-basis

6.align-self