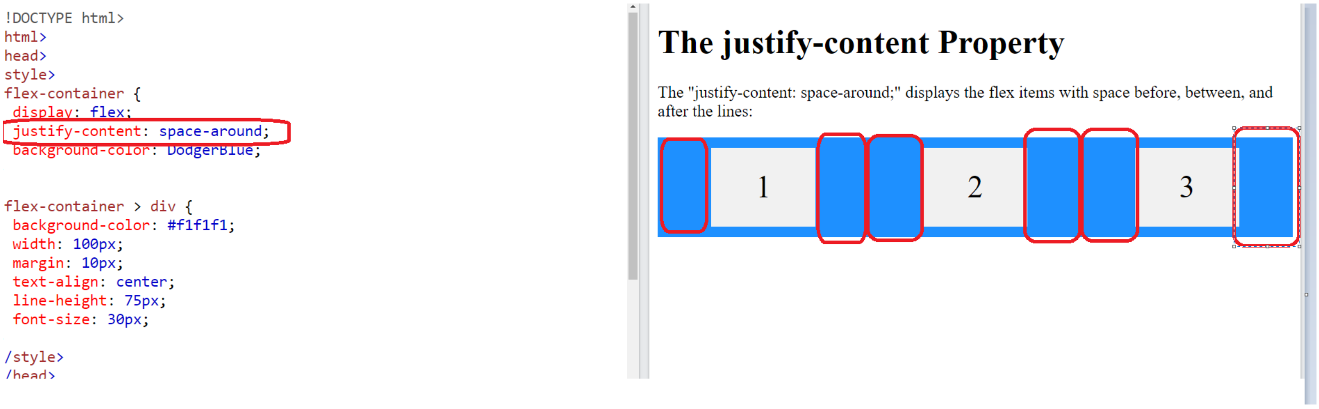
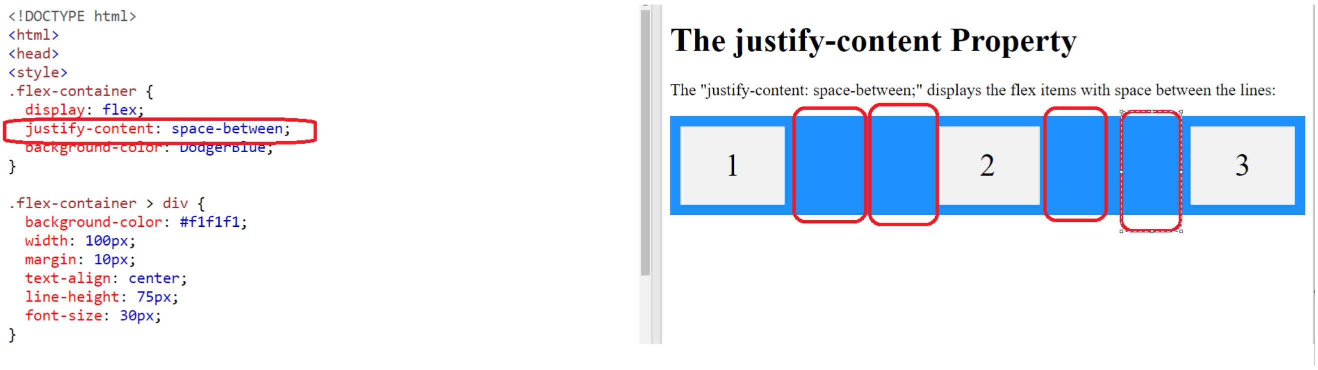


```
!DOCTYPE html>
html>
head>
style>
flex-container {
 display: flex;
 justify-content: space-around;
 background-color: DodgerBlue;


flex-container > div {
 background-color: #f1f1f1;
 width: 100px;
 margin: 10px;
 text-align: center;
 line-height: 75px;
 font-size: 30px;

/style>
/head>
```

The justify-content Property

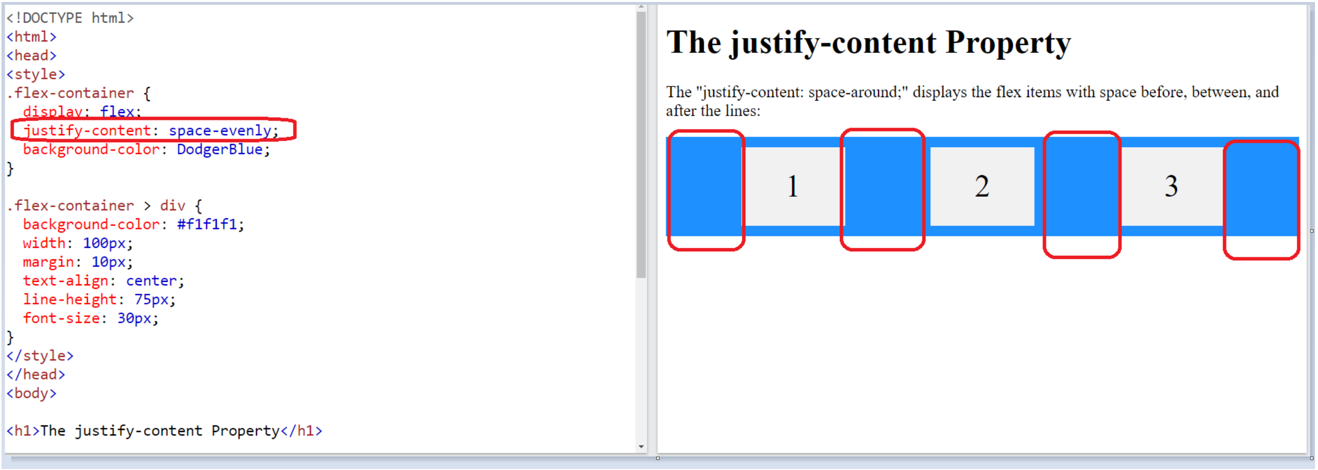
The "justify-content: space-around;" displays the flex items with space before, between, and after the lines:



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

.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
```

The justify-content Property

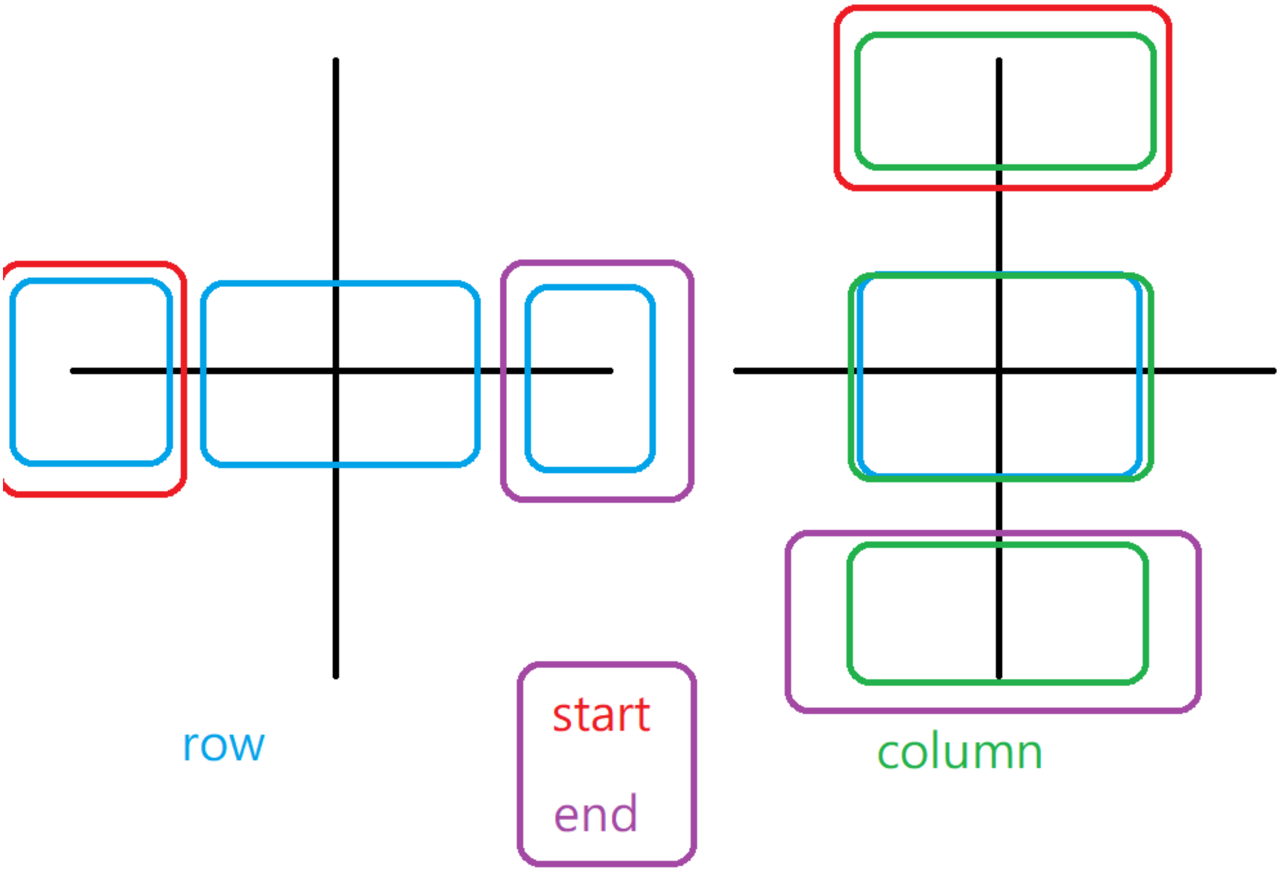
The "justify-content: space-between;" displays the flex items with space between the lines:

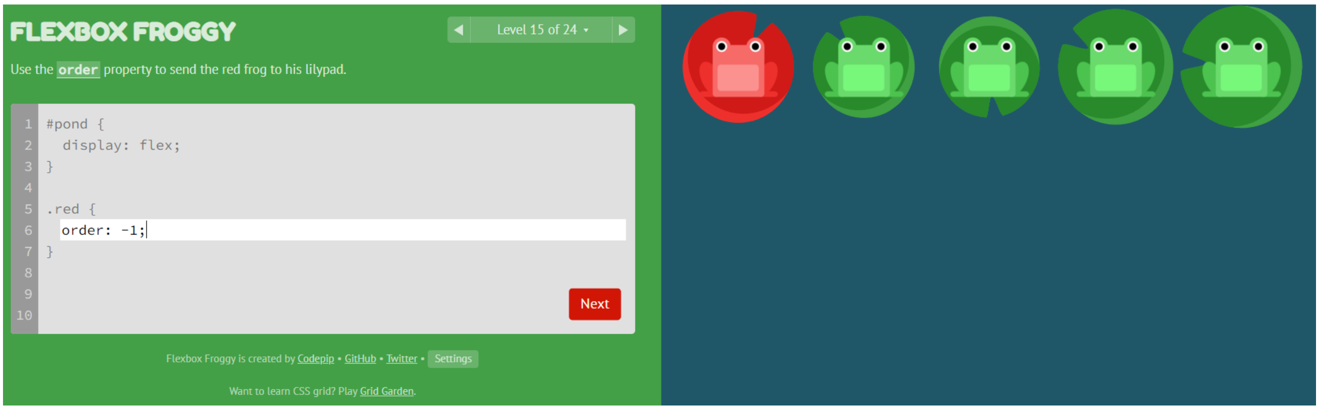


```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  justify-content: space-evenly;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>The justify-content Property</h1>
```

The justify-content Property

The "justify-content: space-around;" displays the flex items with space before, between, and after the lines:



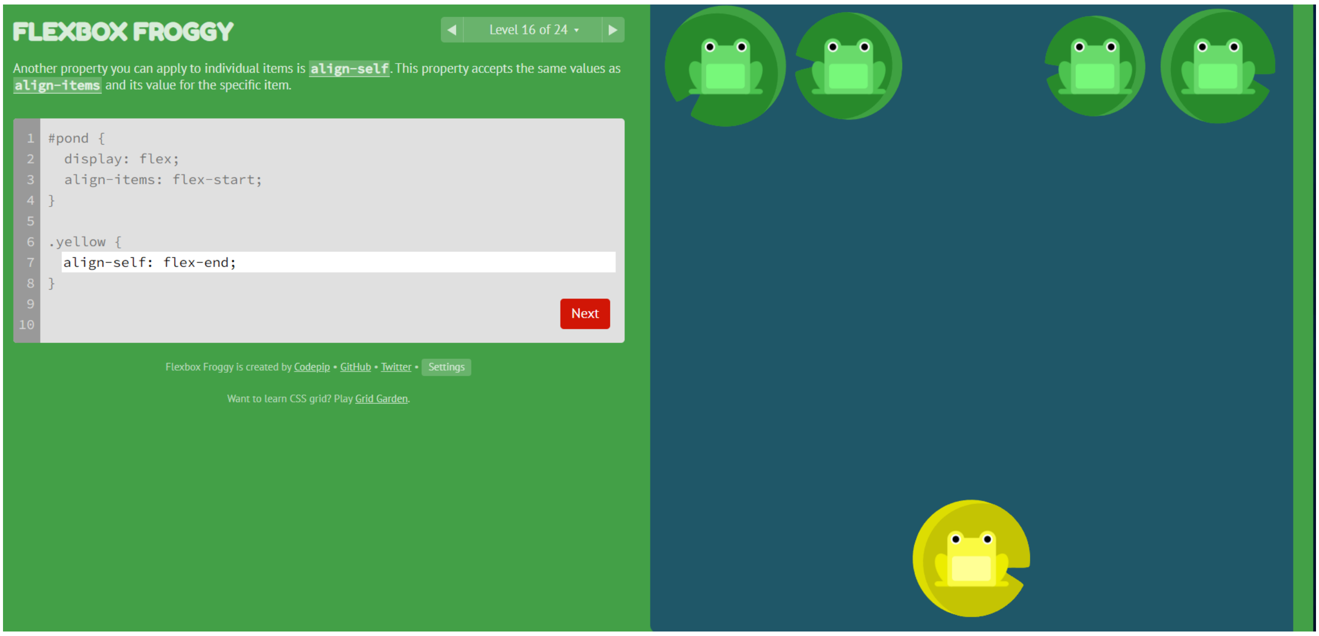


Use the `order` property to send the red frog to his lilypad.

```
#pond {
  display: flex;
}

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.

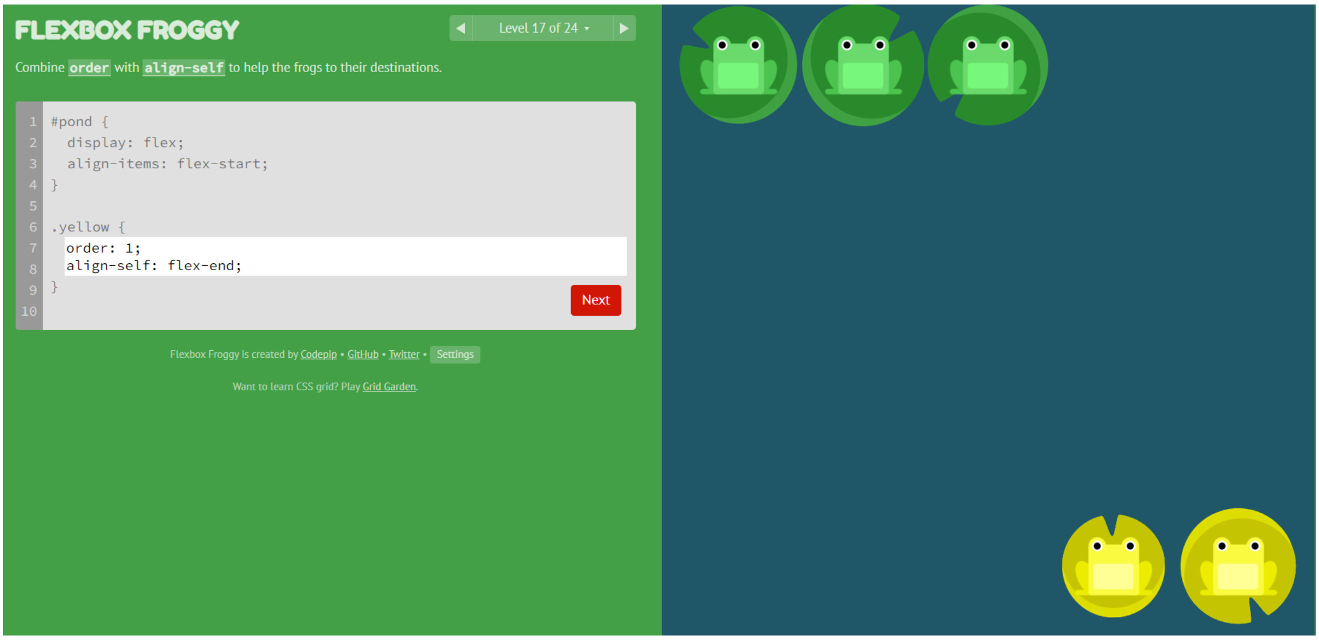


Another property you can apply to individual items is `align-self`. This property accepts the same values as `align-items` and its value for the specific item.

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

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.

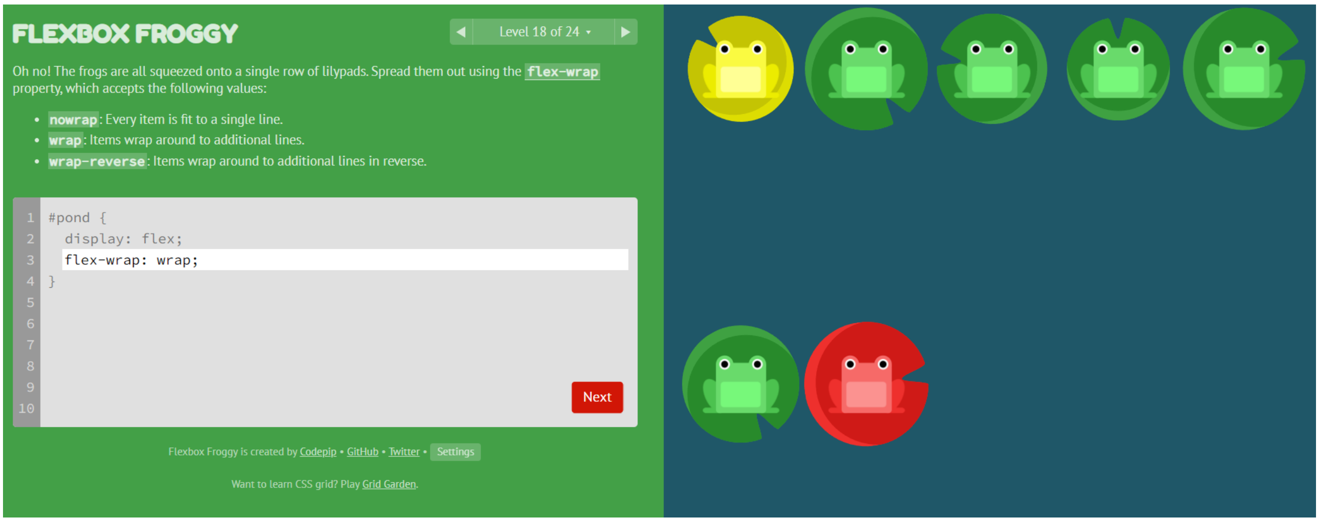


Combine `order` with `align-self` to help the frogs to their destinations.

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

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.



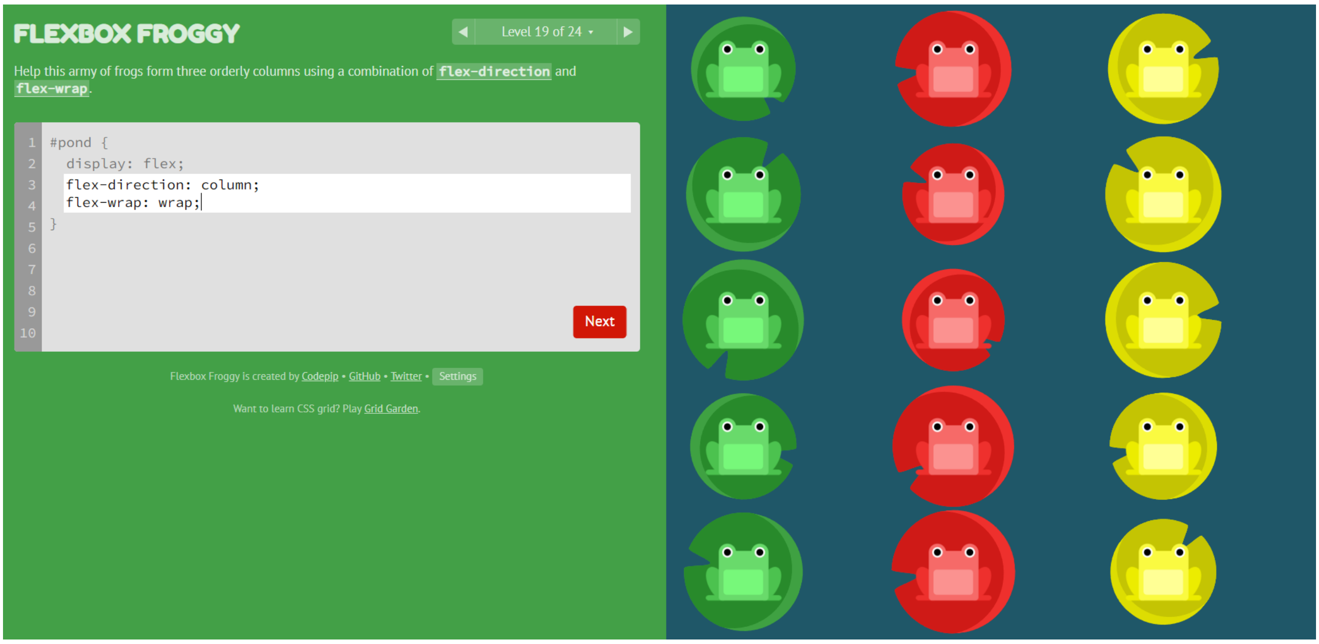
Oh no! The frogs are all squeezed onto a single row of lilypads. Spread them out using the `flex-wrap` property, which accepts the following values:

- `nowrap`: Every item is fit to a single line.
- `wrap`: Items wrap around to additional lines.
- `wrap-reverse`: Items wrap around to additional lines in reverse.

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.

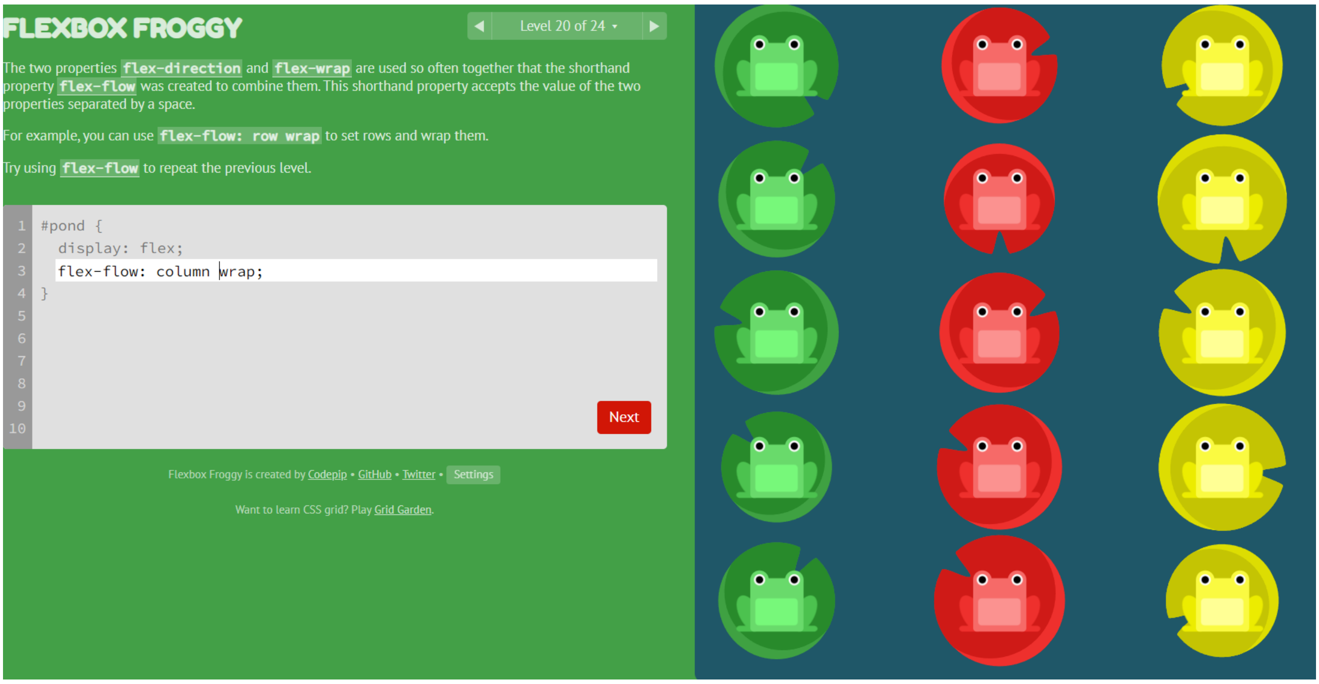


Help this army of frogs form three orderly columns using a combination of `flex-direction` and `flex-wrap`.

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.



The two properties `flex-direction` and `flex-wrap` are used so often together that the shorthand property `flex-flow` was created to combine them. This shorthand property accepts the value of the two properties separated by a space.

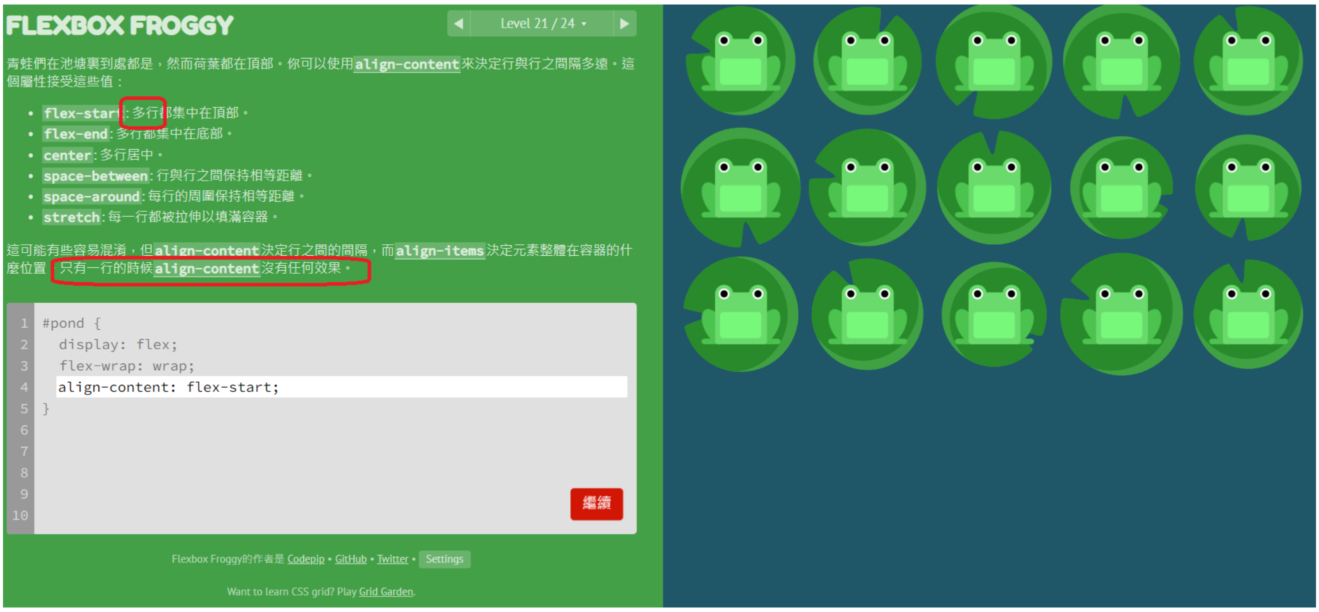
For example, you can use `flex-flow: row wrap` to set rows and wrap them.

Try using `flex-flow` to repeat the previous level.

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

Flexbox Froggy is created by Codepip • GitHub • Twitter •

Want to learn CSS grid? Play Grid Garden.



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：

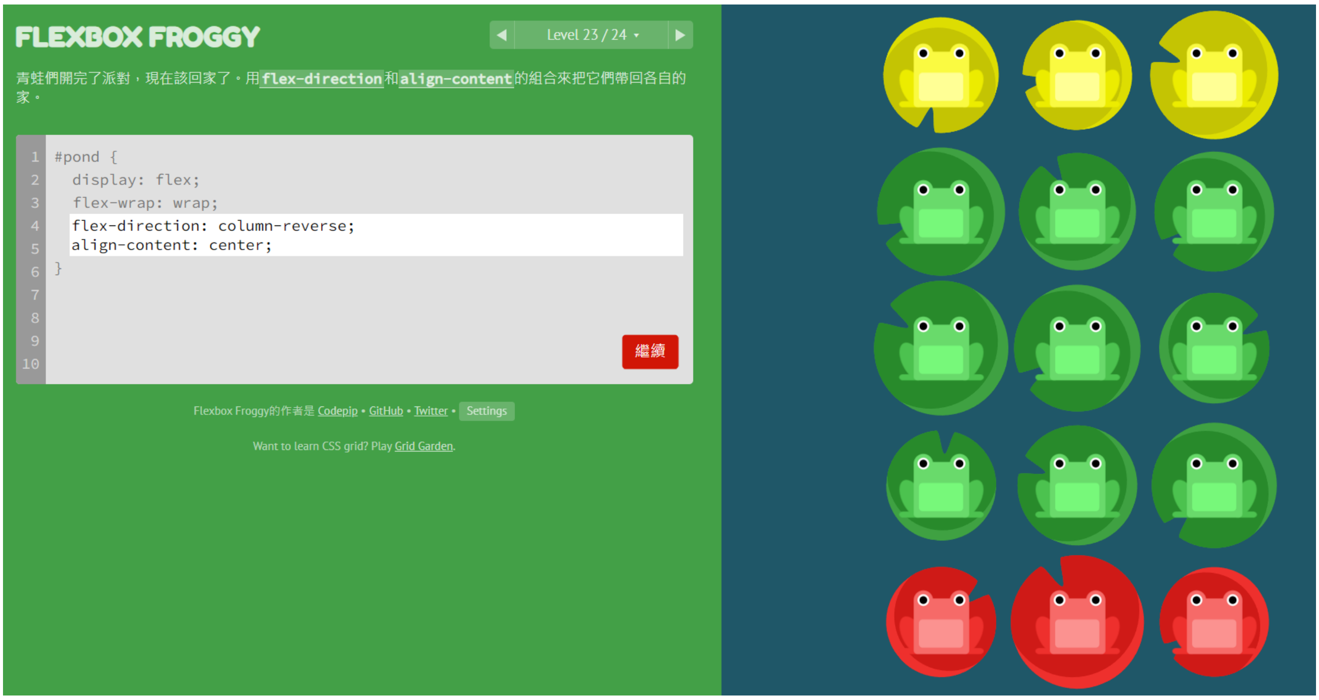
- `flex-start`：多行都集中在頂部。
- `flex-end`：多行都集中在底部。
- `center`：多行居中。
- `space-between`：行與行之間保持相等距離。
- `space-around`：每行的周圍保持相等距離。
- `stretch`：每一行都被拉伸以填滿容器。

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;
}
```

Flexbox Froggy的作者是 Codepip • GitHub • Twitter •

Want to learn CSS grid? Play Grid Garden.

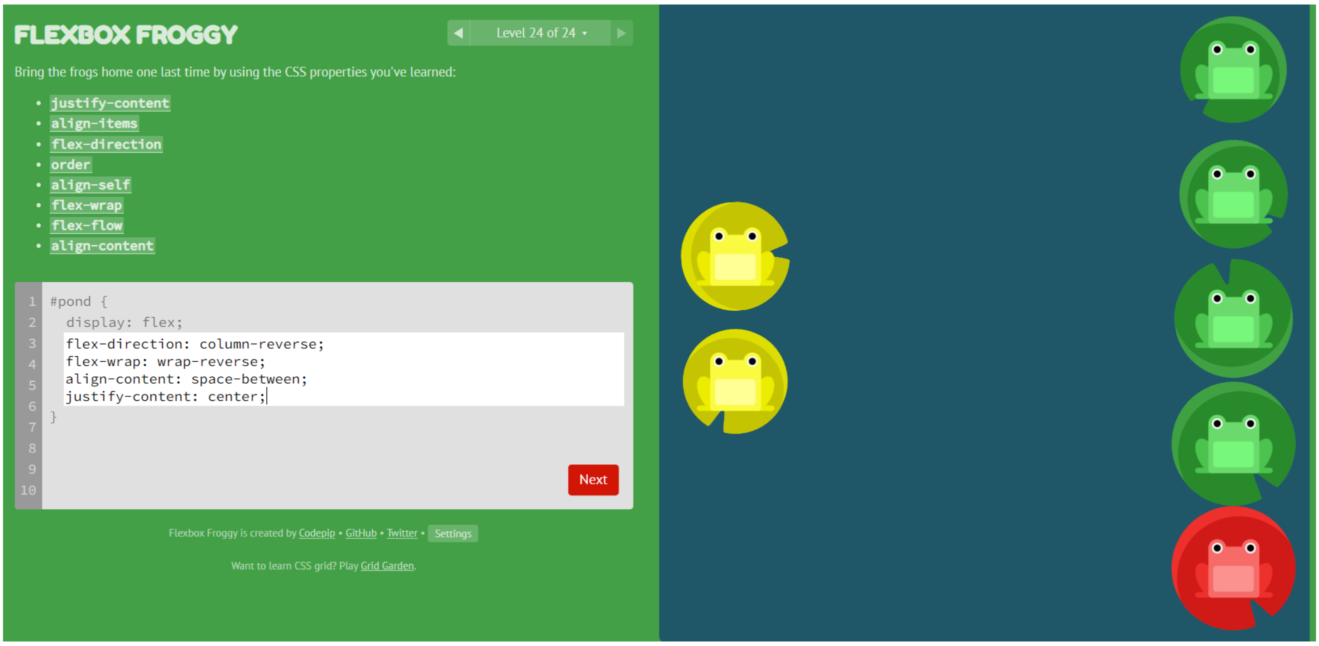


青蛙們開完了派對，現在該回家了。用`flex-direction`和`align-content`的組合來把它們帶回各自的家。

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

Flexbox Froggy的作者是 Codepip • GitHub • Twitter •

Want to learn CSS grid? Play Grid Garden.



Bring the frogs home one last time by using the CSS properties you've learned:

- `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;
  align-content: space-between;
  justify-content: center;
}
```

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

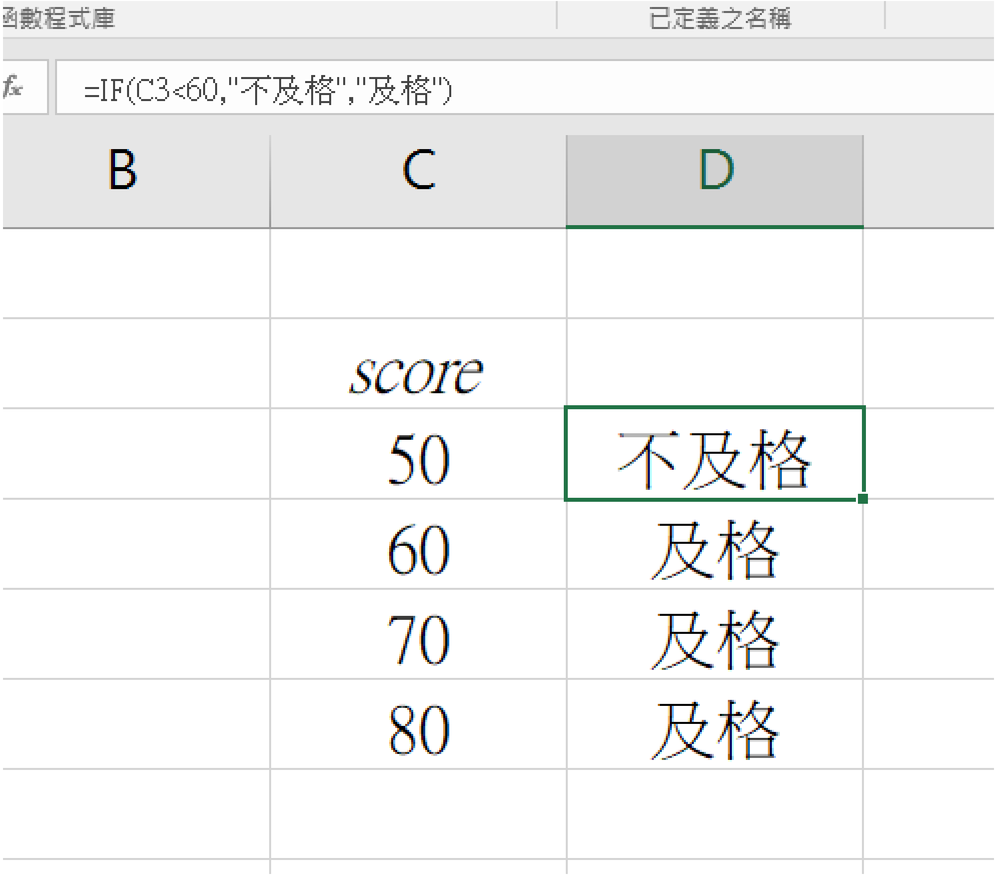
Want to learn CSS grid? Play Grid Garden.

flex-direction: column-reverse;

flex-wrap: wrap-reverse;

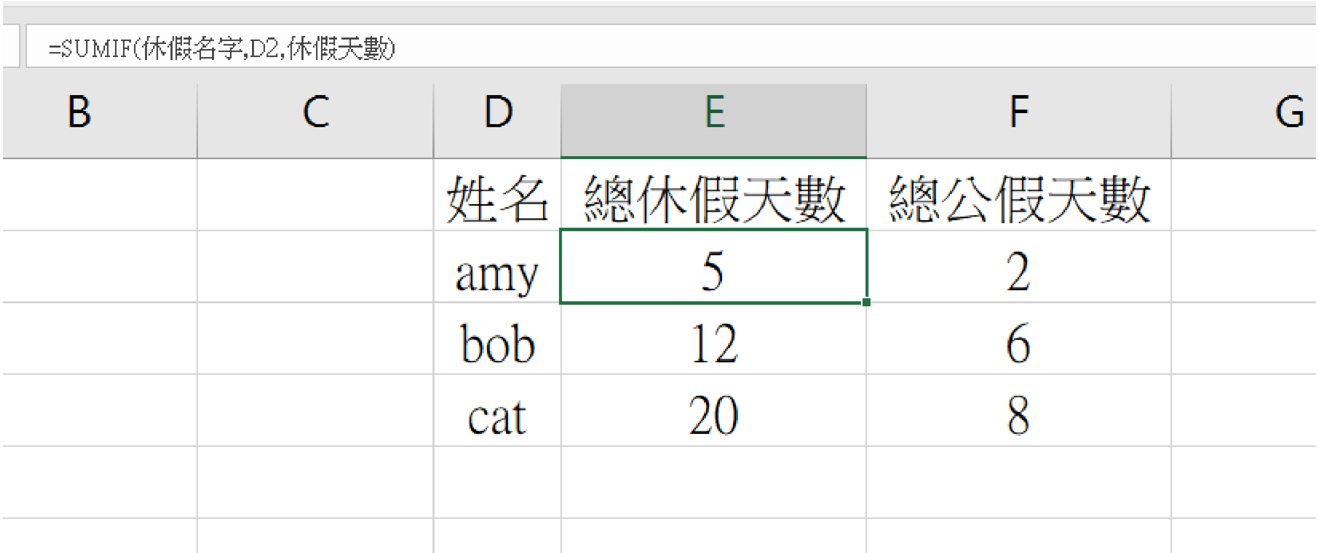
align-content: space-between;

justify-content: center;



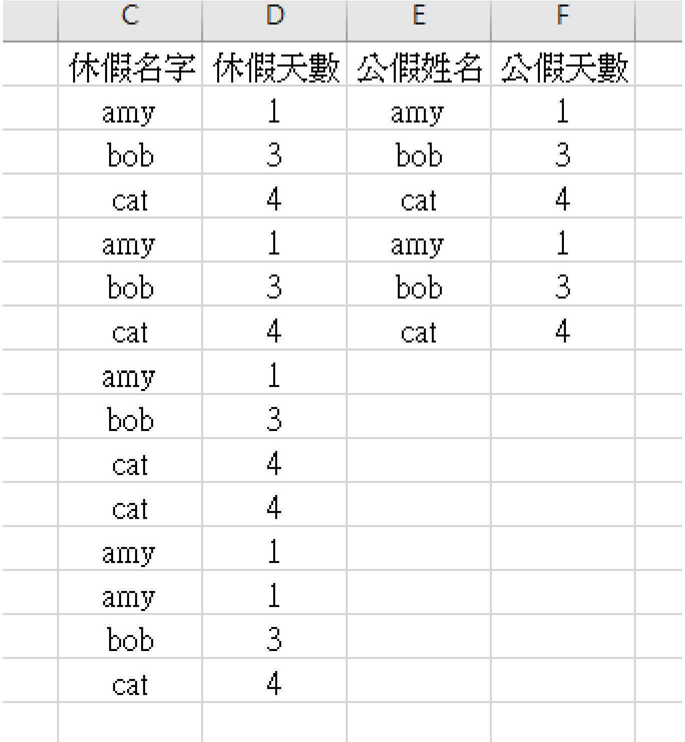
=IF(C3<60,"不及格","及格")

B	C	D
	score	
	50	不及格
	60	及格
	70	及格
	80	及格

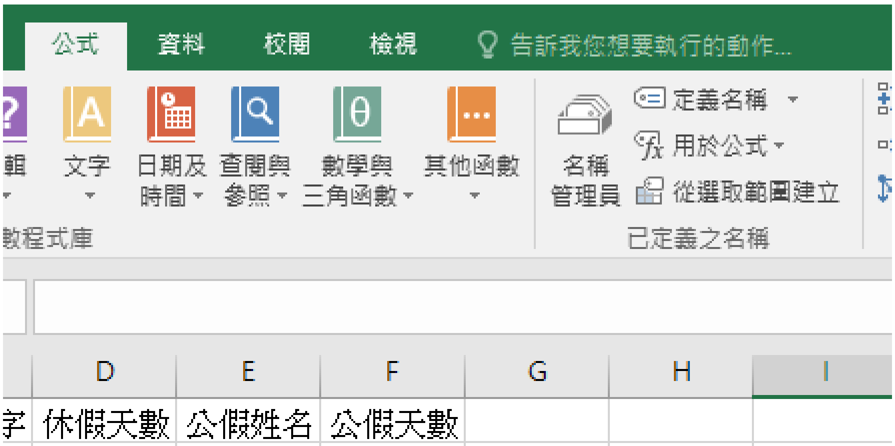


=SUMIF(休假名字,D2,休假天數)

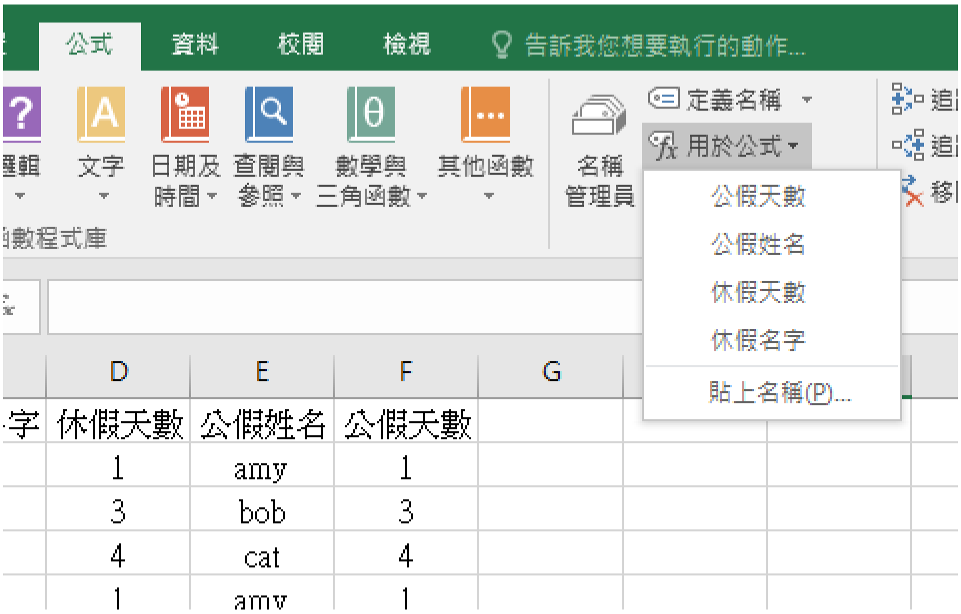
B	C	D	E	F	G
		姓名	總休假天數	總公假天數	
		amy	5	2	
		bob	12	6	
		cat	20	8	



C	D	E	F
休假名字	休假天數	公假姓名	公假天數
amy	1	amy	1
bob	3	bob	3
cat	4	cat	4
amy	1	amy	1
bob	3	bob	3
cat	4	cat	4
amy	1		
bob	3		
cat	4		
cat	4		
amy	1		
amy	1		
bob	3		
cat	4		



	D	E	F	G	H	I
字	休假天數	公假姓名	公假天數			



	D	E	F	G
字	休假天數	公假姓名	公假天數	
	1	amy	1	
	3	bob	3	
	4	cat	4	
	1	amy	1	