CSS

網頁設計X視覺特效專題班

Lecturer: LinJer (林哲)

evin92@gmail.com

網頁前端-進階
CSS

第一堂：初探CSS基礎入門

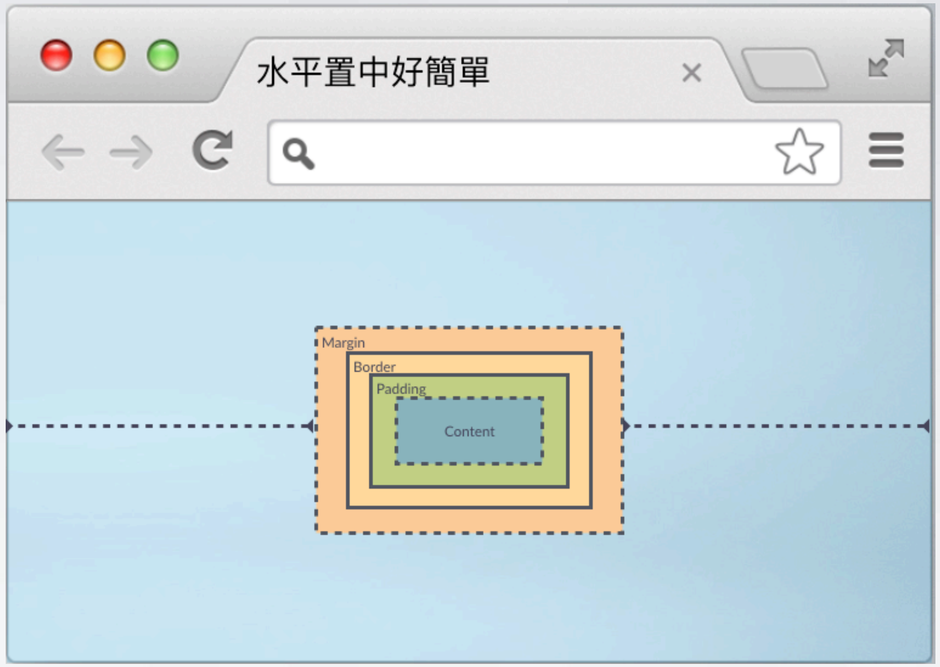
第二堂：進階CSS編寫技巧

第三堂：解構CSS網頁排版

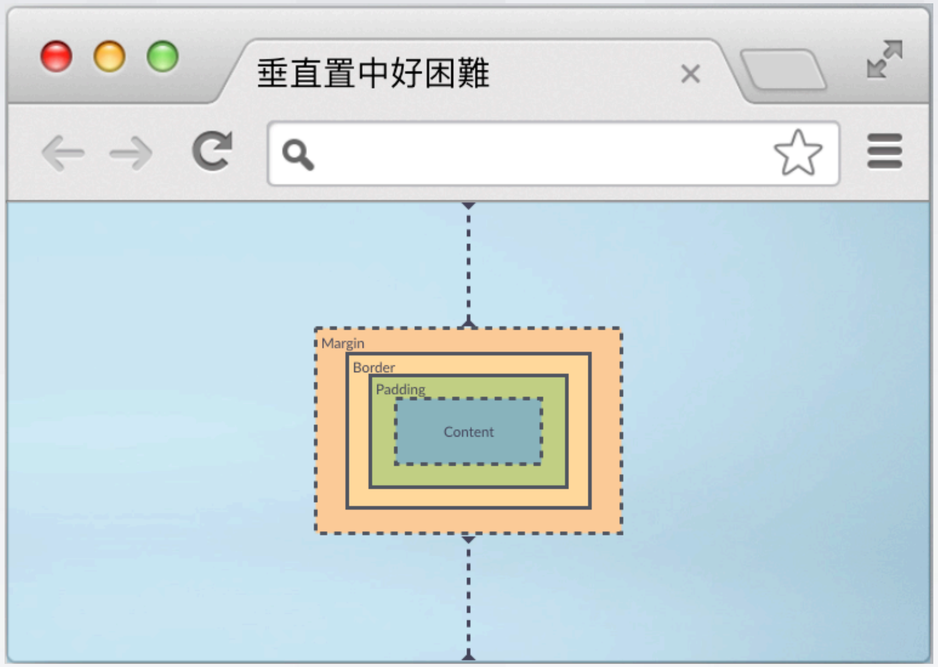
第四堂：玩轉CSS視覺特效

排版中的大哉問-置中

水平置中

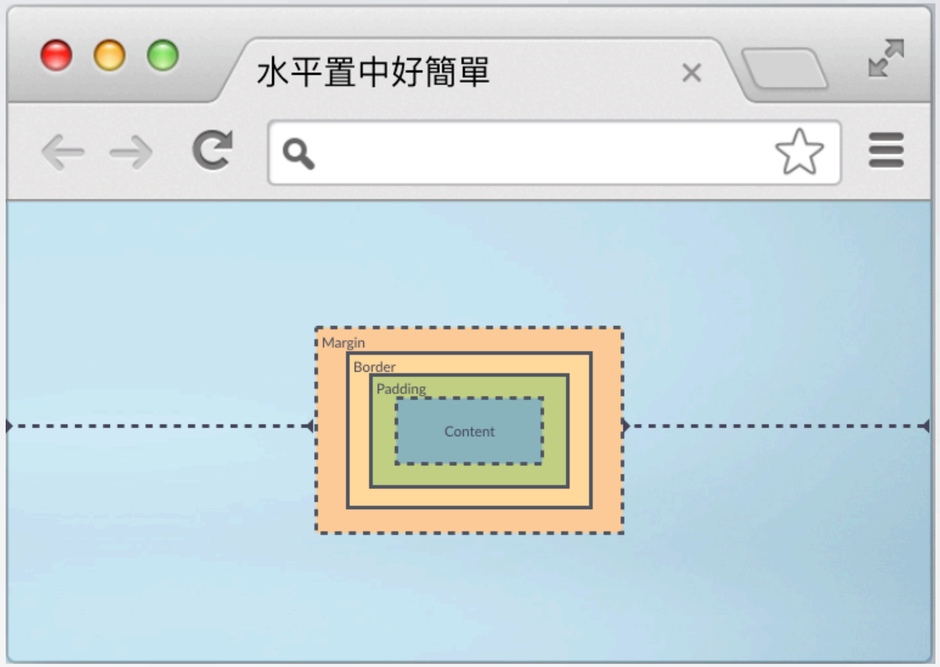


垂直置中



水平置中

查看範例



行內元素:

```
text-align: center;
```

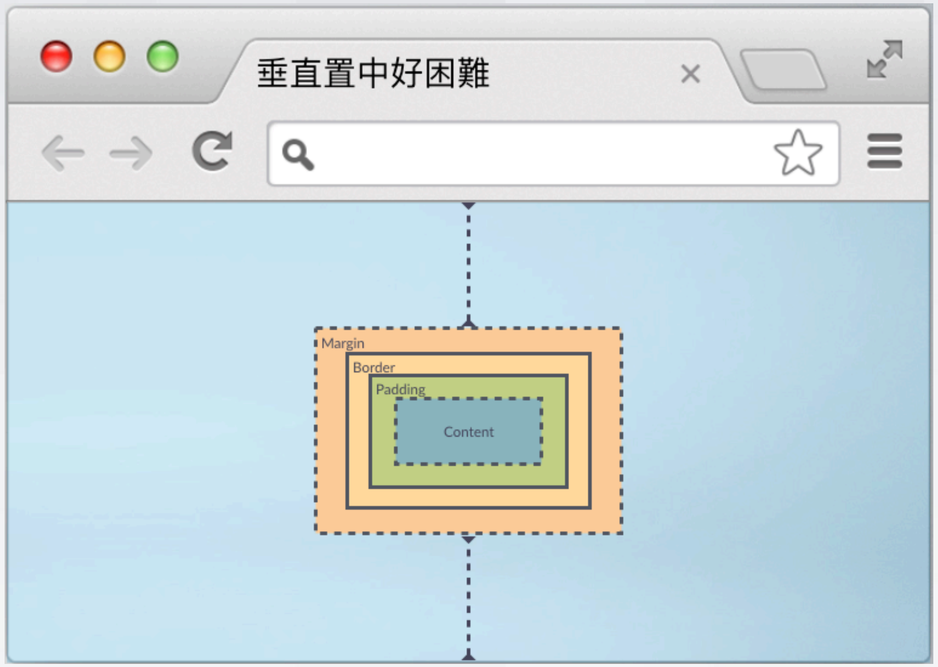
區塊元素:

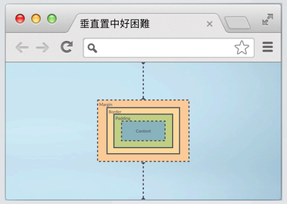
```
margin-left: auto;
margin-right: auto;
```

垂直置中

垂直置中是從過去以來
CSS最難達成目標之一

探討一些目前較好的解法





垂直置中 解法一:
絕對定位法(position: absolute)

前往練習

1

width: 18em;
height: 7em;

2

position: absolute;
top: 50%;
left: 50%;

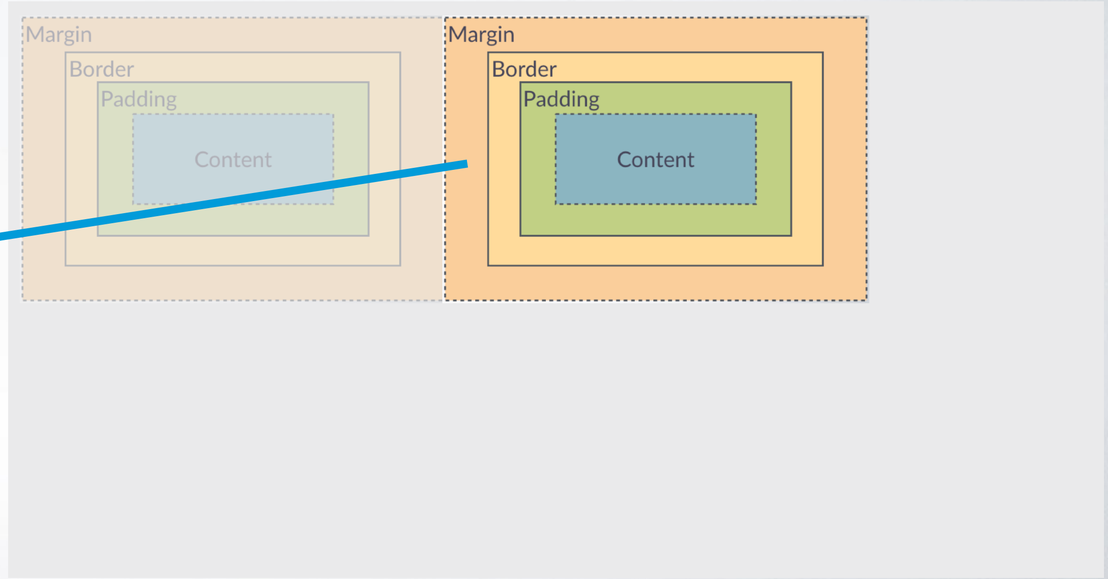
3

margin-top: -3.5em;
margin-left: -9em;

CSS重要屬性: position

CSS 版面配置中用來：
改變區塊的特定位置

- position: static (靜態的: 預設)
- position: relative (相對的)
- position: absolute (絕對的)
- position: fixed (固定的)



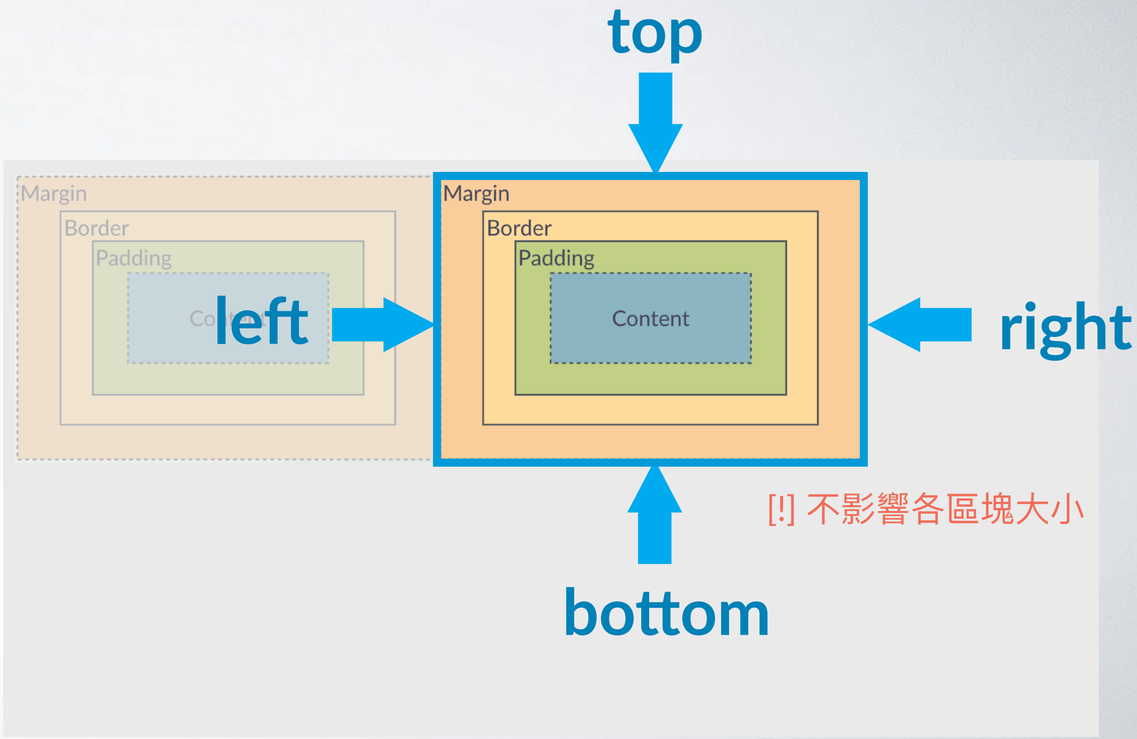
照著HTML順序配置，自動排列在頁面上，
不會被特別定位

CSS重要屬性: position

CSS 版面配置中用來：
改變區塊的特定位置

- position: static (靜態的: 預設)
- **position: relative (相對的)**
- position: absolute (絕對的)
- position: fixed (固定的)

前往範例



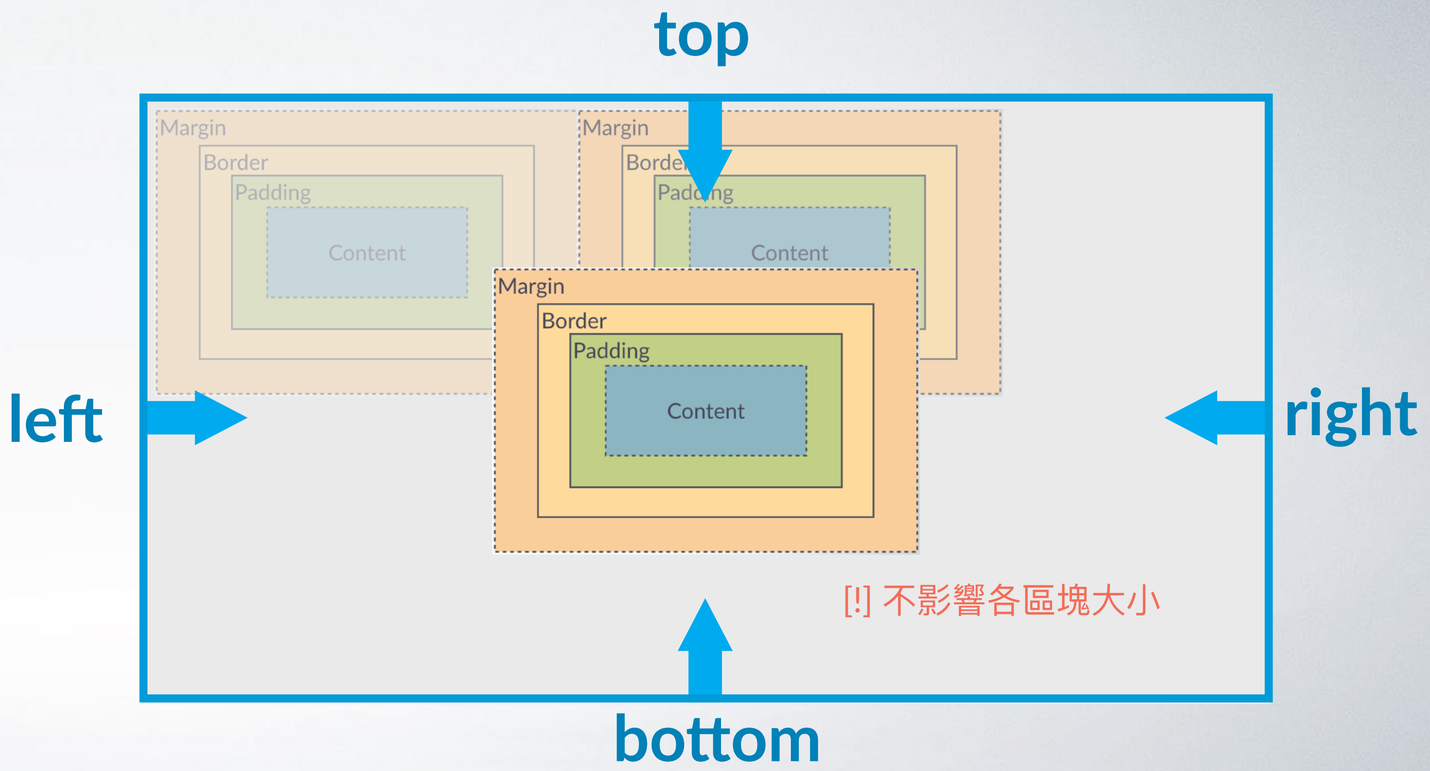
原本在static時的位置，
一旦設定任一(或二)四方向屬性，
會以自己邊界做「相對」移動

CSS重要屬性: position

CSS 版面配置中用來：
改變區塊的特定位置

- position: static (靜態的: 預設)
- position: relative (相對的)
- **position: absolute (絕對的)**
- position: fixed (固定的)

前往範例



原本在static時的位置，
一旦設定任一(或二)四方向屬性，
會以其所處的父層元素邊界做「相對」移動

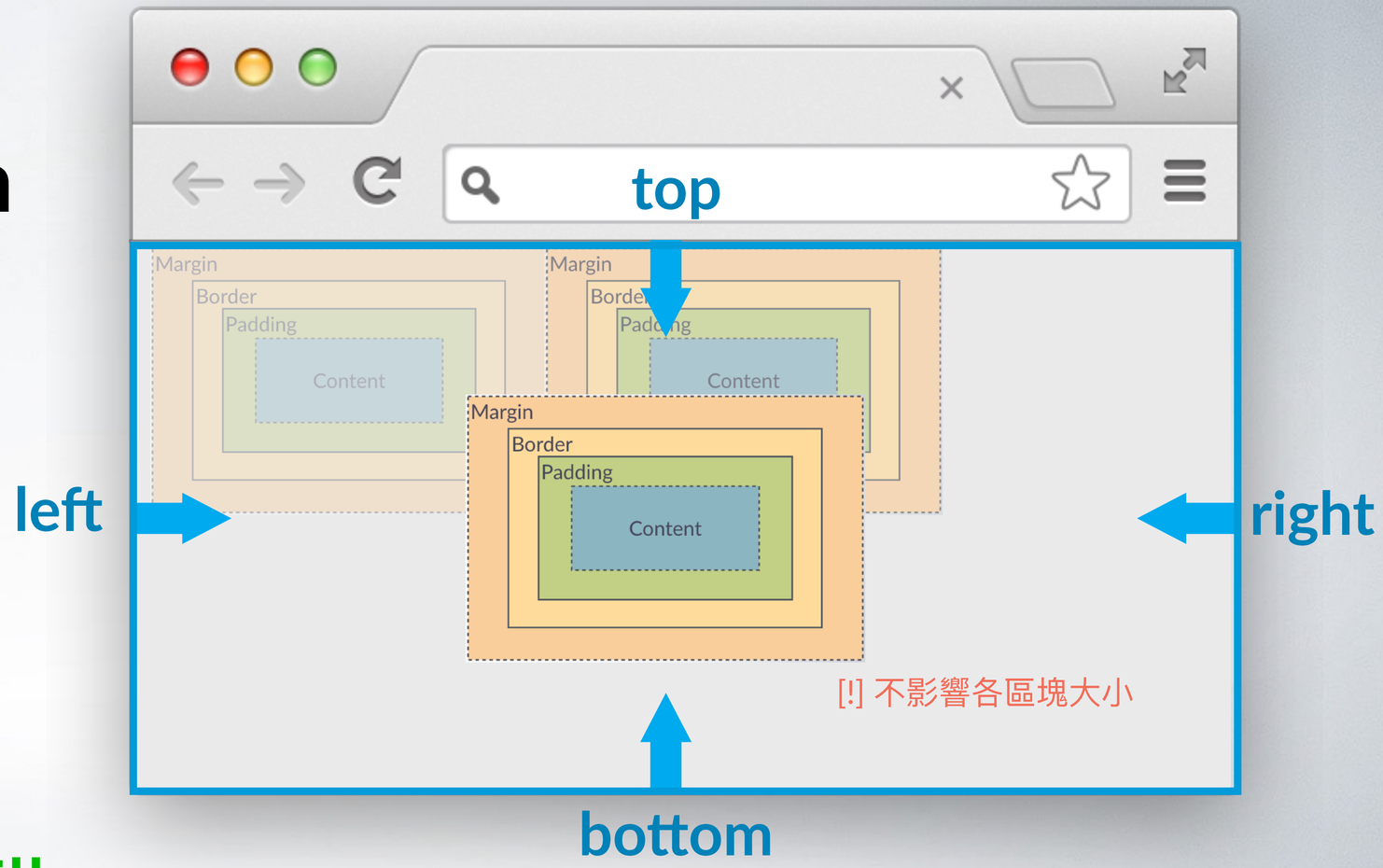
CSS重要屬性: position

CSS 版面配置中用來：
改變區塊的特定位置

- position: static (靜態的: 預設)
- position: relative (相對的)
- position: absolute (絕對的)
- **position: fixed (固定的)**

陰魂不散!!

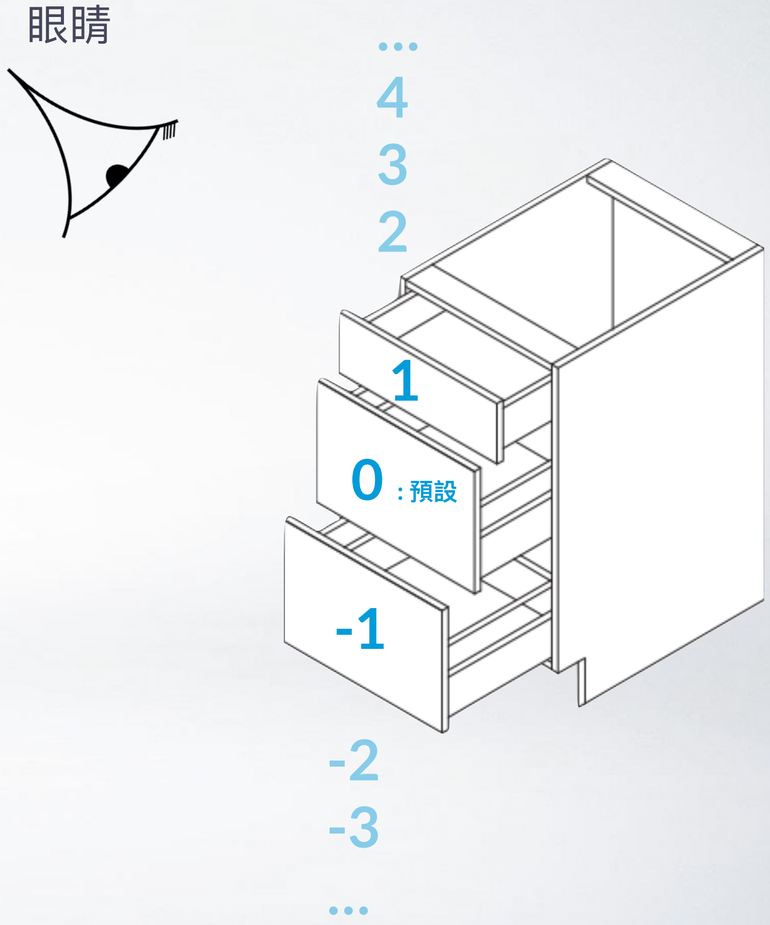
前往範例



原本在static時的位置，
一旦設定任一(或二)四方向屬性，
會以瀏覽器視窗邊界做「相對」移動

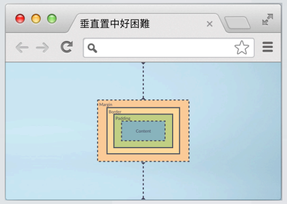
CSS屬性: z-index

設置元素的 Z 方向位置，
z-index 數字越大的在越上面，
反之則在越下面

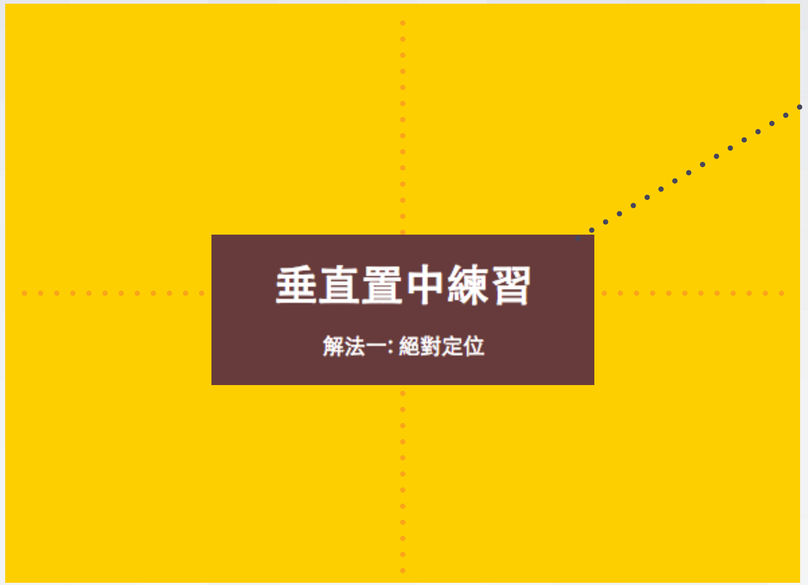


z–index : 抽屜層數（整數）

前往範例練習



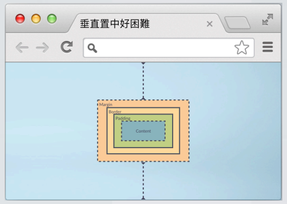
垂直置中 解法一:
絕對定位法(position: absolute)



簡化它!!

```
width: 18em;
height:  7em;

position: absolute;
top: 50%;
left: 50%;

margin-top: -3.5em;
margin-left: -9em;
```

```
width: 18em;
height:  7em;

position: absolute;
top: calc( 50% - 3.5em );
left: calc( 50% - 9em );

~~margin-top: -3.5em;~~
~~margin-left: -9em;~~
```



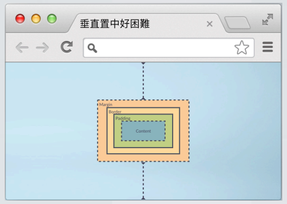
垂直置中 解法一:
絕對定位法(position: absolute)

問題:此解需固定寬高尺寸
想讓寬高由內容決定
該如何解決？

~~width: 18em;~~
~~height: 7em;~~

position: absolute;
top: calc(50% - 3.5em);
left: calc(50% - 9em);

?????



垂直置中 解法一:
絕對定位法(position: absolute)

解法一缺點: 需使用absolute

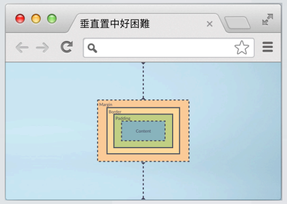
解法: 加入 transform 讓元素位移

position: absolute;
top: 50% ;
left: 50% ;

transform: translate(-50%, -50%);

此百分比是根據其元素本身寬/高

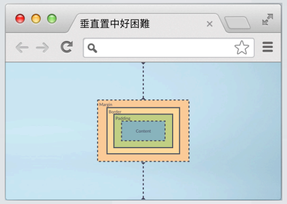
前往範例



垂直置中 解法二:
半可視高度法(margin)

?

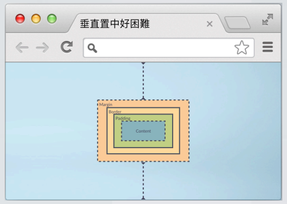
```
width: 18em;
margin: 50% auto 0;
transform: translateY(-50%);
```



垂直置中 解法二:
半可視高度法(margin)

?

```
width: 18em;
margin: 50% auto 0;
transform: translateY(-50%);
```



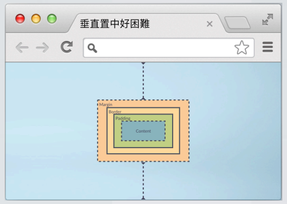
垂直置中 解法二:
半可視高度法(margin: 50vh auto 0)

解法二缺點: 受限於視窗可視寬高，小區塊怎麼辦？

vw: 視窗可視範圍寬度的百分比
vh: 視窗可視範圍高度的百分比

Ex.
1vw 表示視窗寬度的1%長，
100vw 表示視窗寬度100%全長

```
width: 18em;
margin: 50vh auto 0;
transform: translateY(-50%);
```



垂直置中 解法三:
Flex法(display: flex;)

解法三缺點: 瀏覽器老舊不支援CSS3

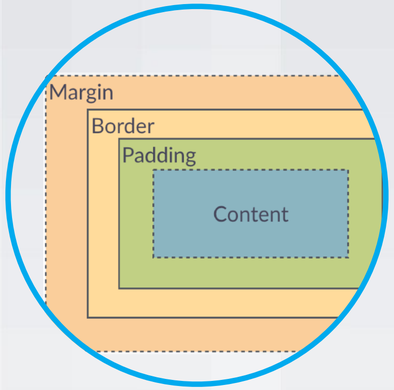

```
body{
    display: flex;
}
body>.main{
   margin: auto;
   width: 300px;
}
```

不只水平置中，垂直也置中

前往練習

排版流程

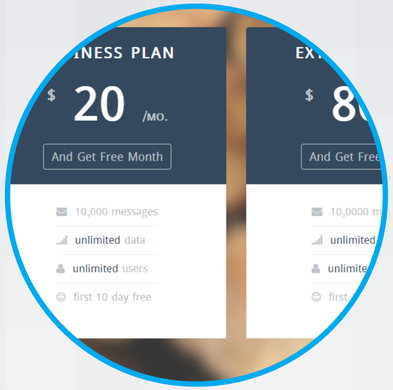
1



Box大小

調整頁面中
各元素的box大小

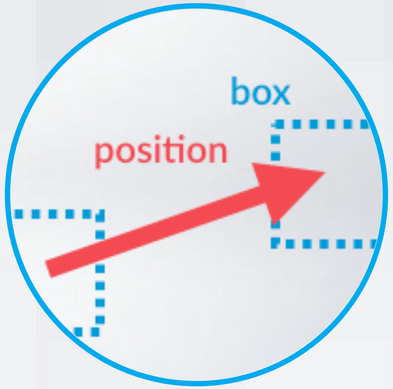
2



Box佈局

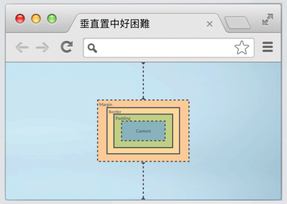
讓box依需求
在不同區塊中排列

3



Box定位

設定box在版面中
最終位置



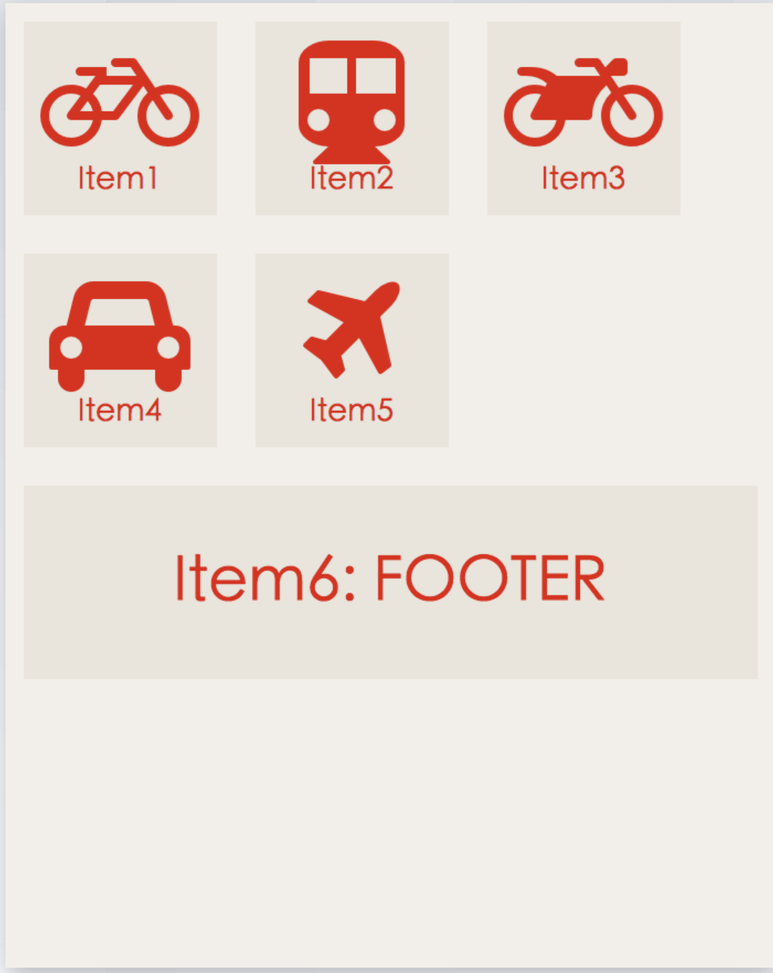
四種佈局模式:

Trick!

1. 表格標籤(table)佈局 前往範例
2. 浮動屬性(float)佈局: float, clear 前往範例
3. 外行內區(inline-block)佈局 前往範例

CSS3

4. **彈性方塊(flexbox)佈局** 前往範例



浮動屬性(float)佈局

CSS 版面配置中用來：
定義區塊的浮動(區塊原地強行靠左或右)

- float: none (預設)
- float: left (靠左浮動)
- float: right (靠右浮動)

前往範例練習#1

[提示]

1.任何元素都是可以變成浮動的

2.任一文檔產生以：由左往右，從上至下來繪製畫面

文字浮動(對inline元素)

float:none	Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Hello! I am span !!! Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
float:left	Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex Hello! I am span !!!ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

前後文不一致，因強行靠左

浮動屬性(float)佈局

CSS 版面配置中用來：
定義區塊的浮動(區塊原地強行靠左或右)

- float: none (預設)
- float: left (靠左浮動)
- float: right (靠右浮動)

前往範例練習#2

圖片浮動(文繞圖 對block元素)

float:none	 Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's
float:left	 Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was

浮動技巧

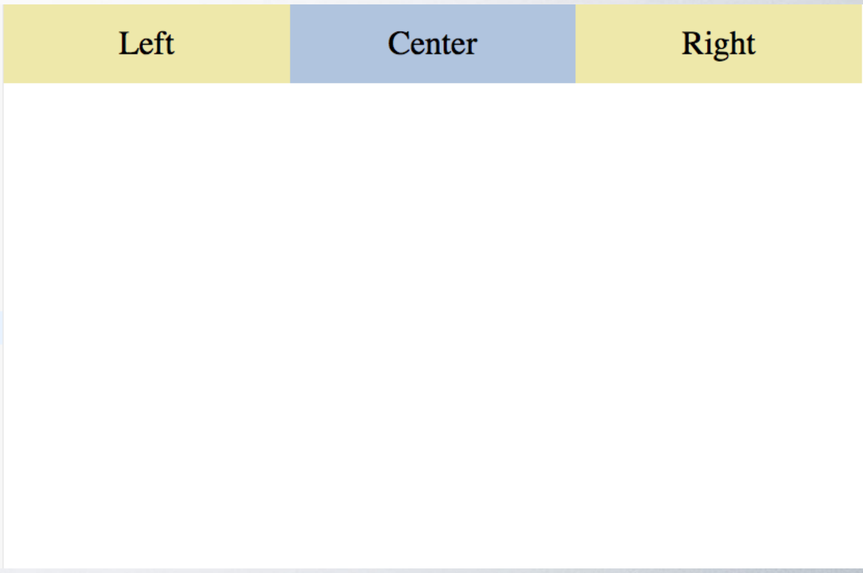
float 讓 div 不會排擠別人

(強行靠左，產生堆疊效果)

前往範例練習#3

```html
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css"
href="http://yui.yahooapis.com/3.4.1/build/cssr
eset/cssreset-min.css">
</head>

<body>

    <div>Left</div><div>Center</div>
<div>Right</div>

</body>
</html>
```

```css
div:nth-of-type(even){
  background: LightSteelBlue  ;
}

div{
  float: left;
  width: 33.3%;
  background:PaleGoldenRod ;
  text-align: center;
  padding: 10px 0px;
}
```



CSS3
彈性方塊(flexbox)佈局

父容器:

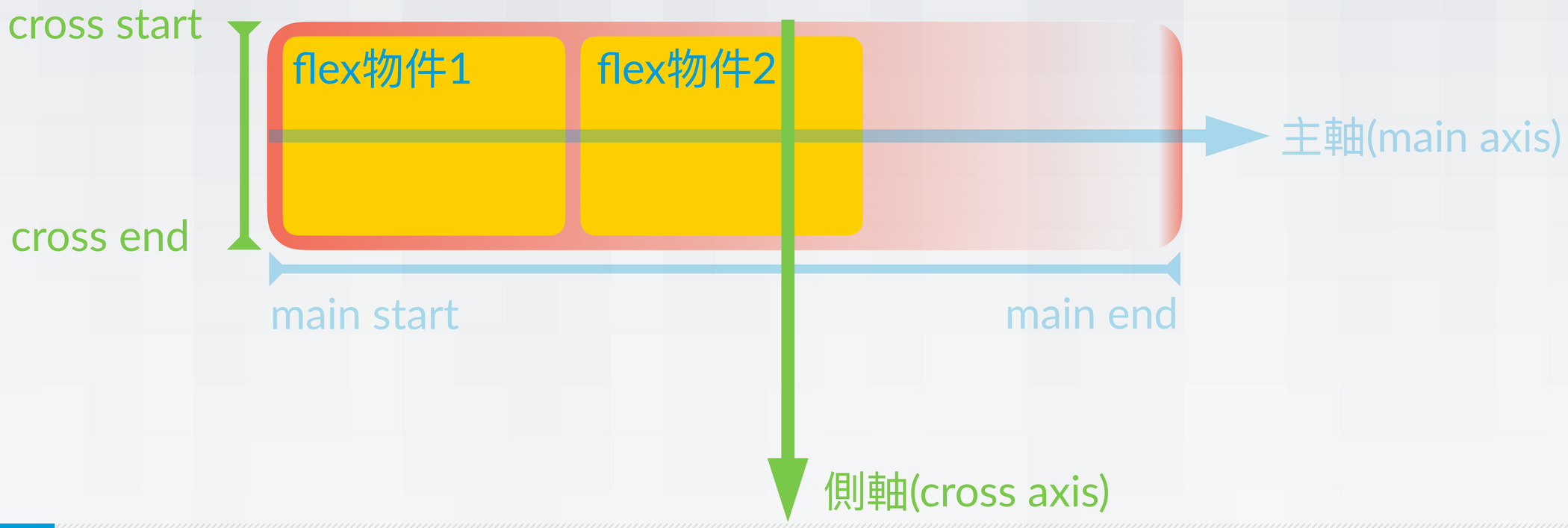
```
.container {
  display: flex;
}
```

OR

```
.container {
  display: inline-flex;
}
```

```
父容器: {
    子物件1,
    子物件2,
    ....
}
```

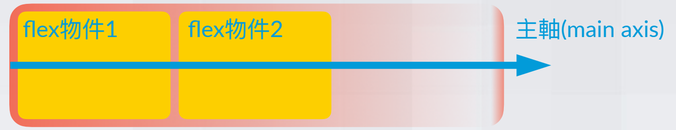
CSS3 flexbox 彈性方塊



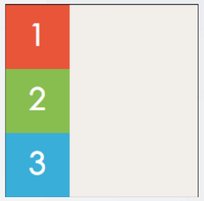
元素排列方向 flex-direction

flex-direction:

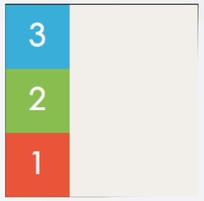
- row(預設值)：由左到右，上往下



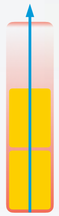
- row-reverse：由右到左，上往下



- column：由上到下，左往右



- column-reverse：由下到上，左往右

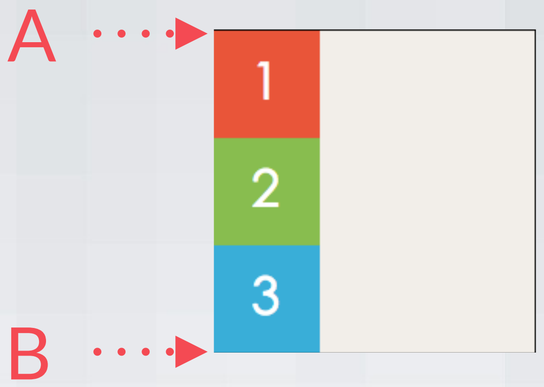


[前往範例]

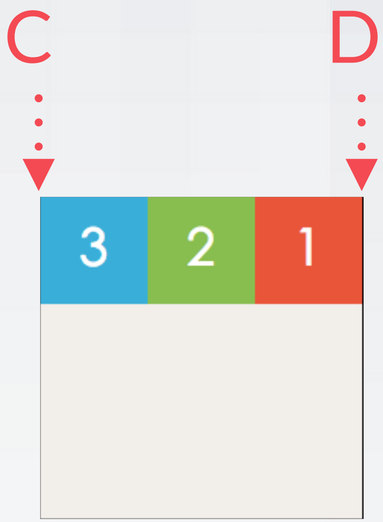
小問答時間

如右分別
設定不同的 **flex-direction**
各**main-end**在哪裡？

答案: ___ ___



flex-direction: column

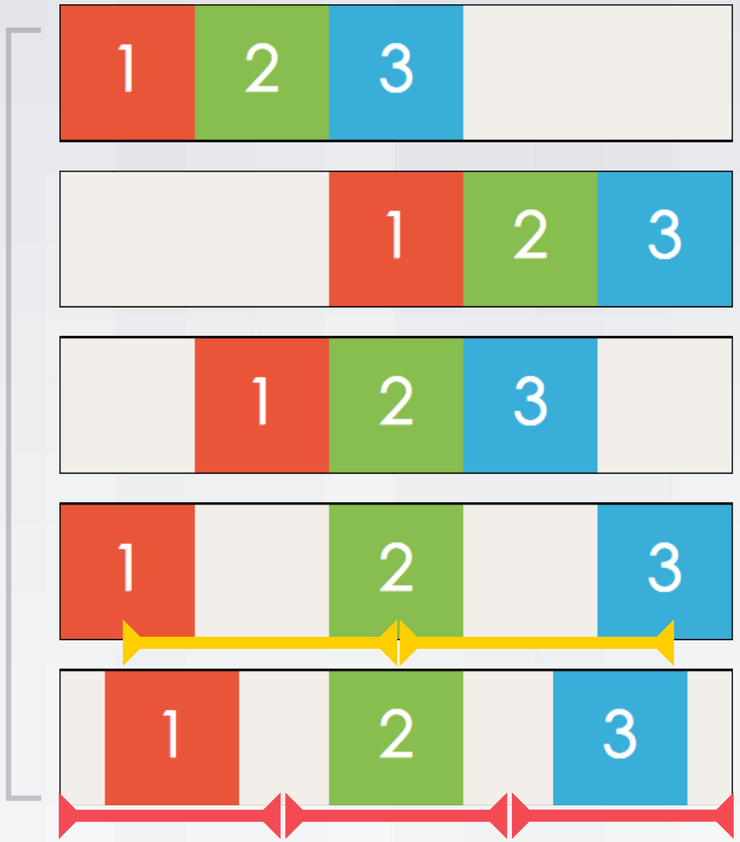


flex-direction: row-reverse

(左右貼齊-內容)

主軸(水平)對齊方式 justify-content

justify-content:

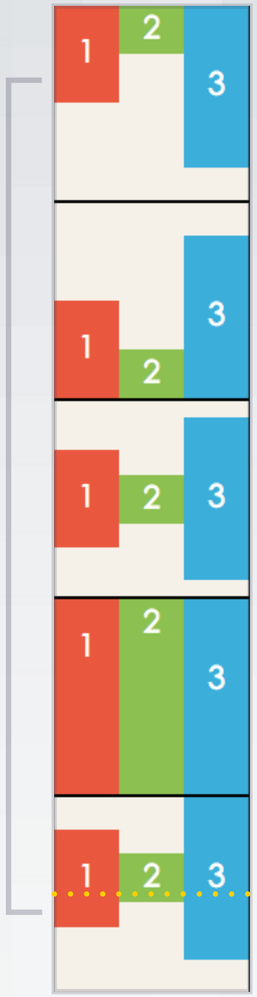


- flex-start(預設值)：從最左邊的 main start 對齊
- flex-end：從最右邊的 main end 對齊
- center：水平置中
- space-between：左右對齊 main start/ end，後平均分配
- space-around：平均分配內容元素

前往範例

側軸(垂直)對齊方式 (對準-物件) align-items

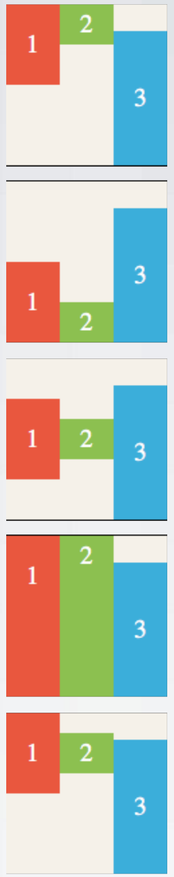
align-items:



- flex-start(預設值)：從最上方的 cross start 對齊
- flex-end：從最下面的 cross end 對齊
- center：垂直置中
- stretch：將元素全部撐開至父層(container)的高度
- baseline：以所有元素的基線作為對齊標準

前往範例

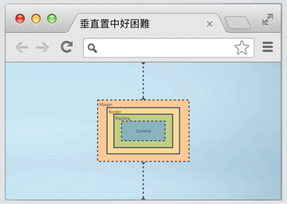
跳脫群體
自行定義
側軸(垂直)對齊方式
align-self



值與align-items一模一樣

但是需針對個別「子物件」

前往練習



垂直置中 解法三:
Flex法(display: flex;)

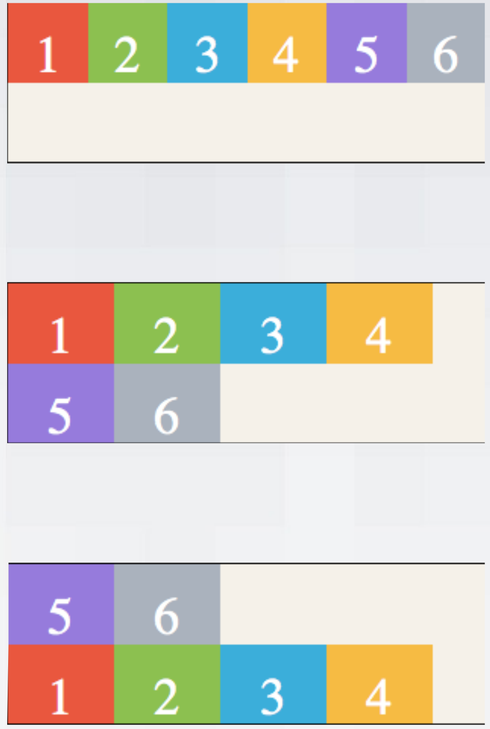
解法三缺點: 瀏覽器老舊不支援CSS3


```
body{
    display: flex;
    justify-content:  ?;
    align-items: ?;
}
body>.main{
  margin: auto;
  width: 300px;
}
```

前往驗證

不只一行:換行 flex-wrap

flex-wrap:



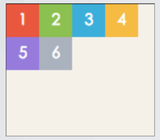
- nowrap (預設值)：單行(擠壓再擠壓)
- wrap：多行
- wrap-reverse：多行，但內容元素反轉

側軸(垂直)多行對齊 align-content (對準-內容)

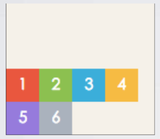
justify-content (左右貼齊-內容)

align-items (對準-物件)

align-content:

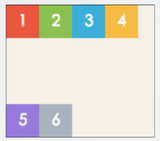


・flex-start(預設值)：從最上方的 cross start 對齊

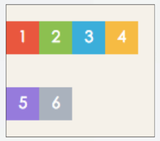


・flex-end：從最下面的 cross end 對齊

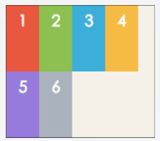
・center：垂直置中



・space-between：將第一行與最後一行分別對齊最上方與最下方



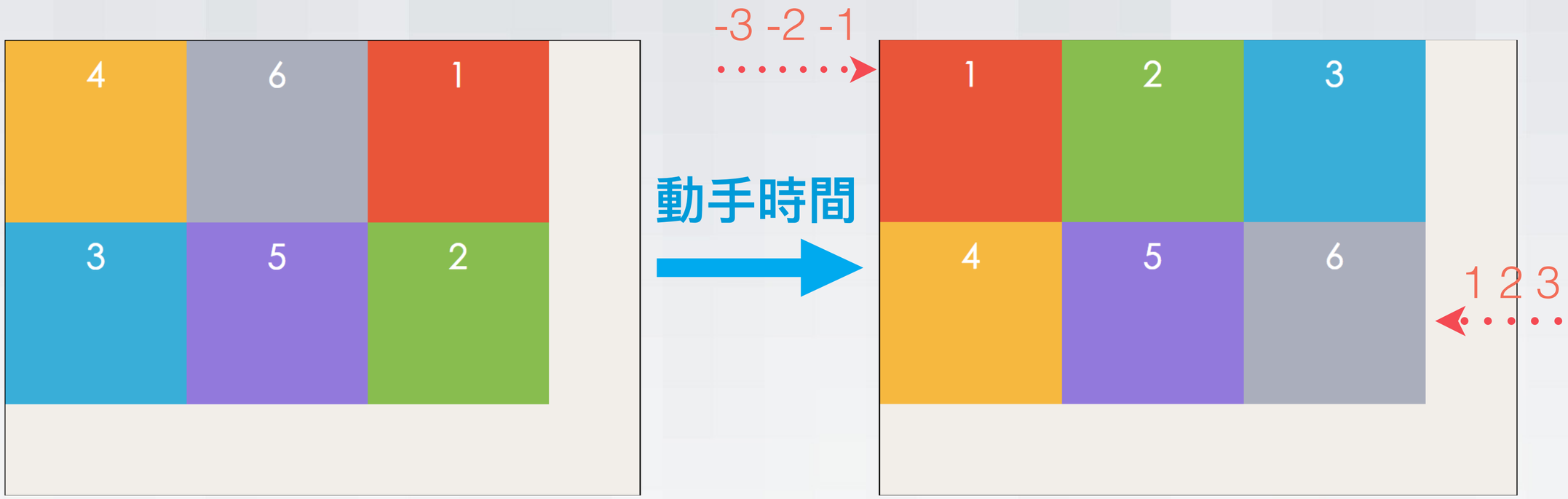
・space-around：每行平均分配間距



・stretch：每行平均分配間距，並將元素全部撐開

前往範例

現在來談談子物件
調整順序 order

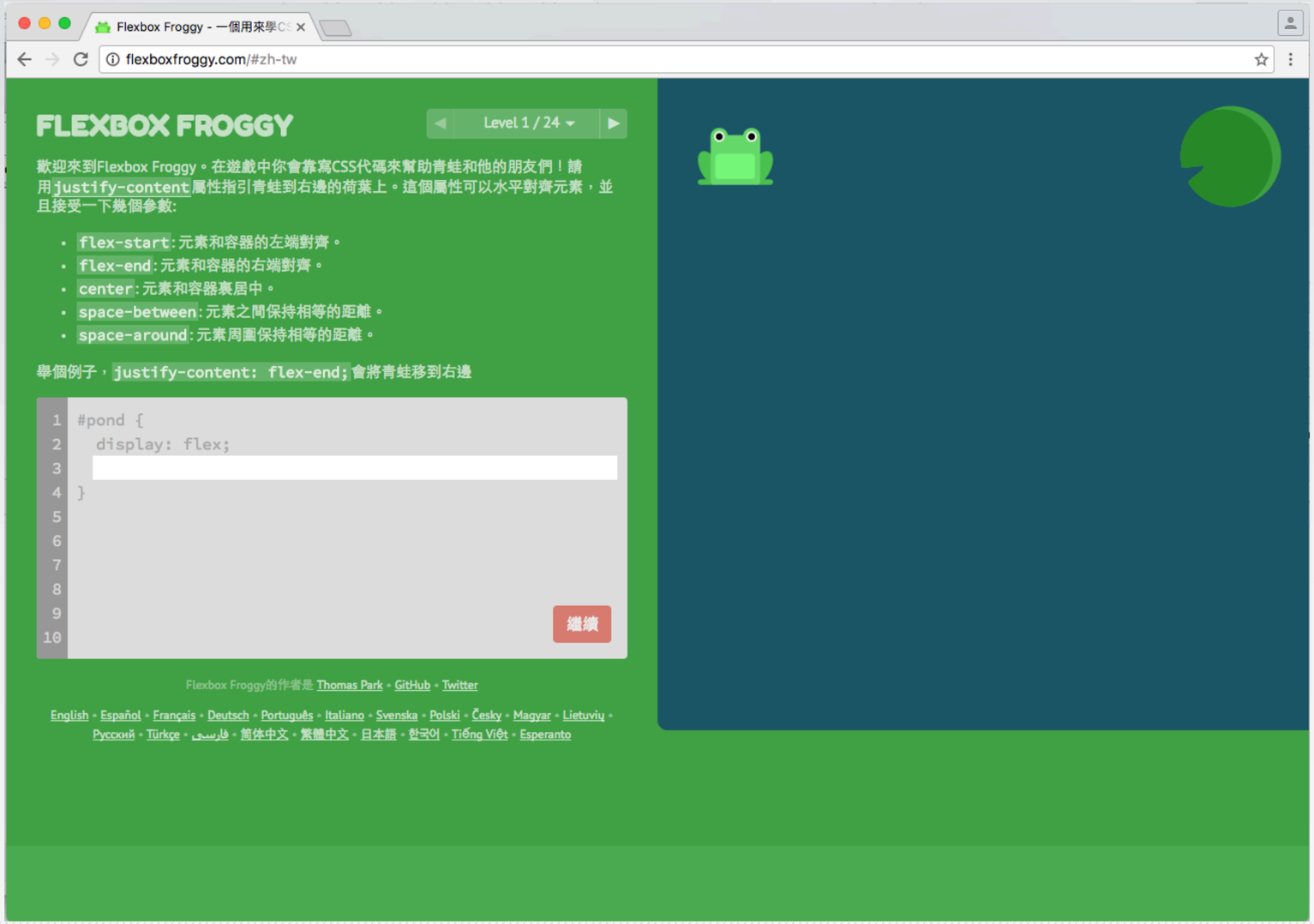


前往範例

遊戲時間

用你剛剛學的flexbox
來幫助青蛙和他的朋友們

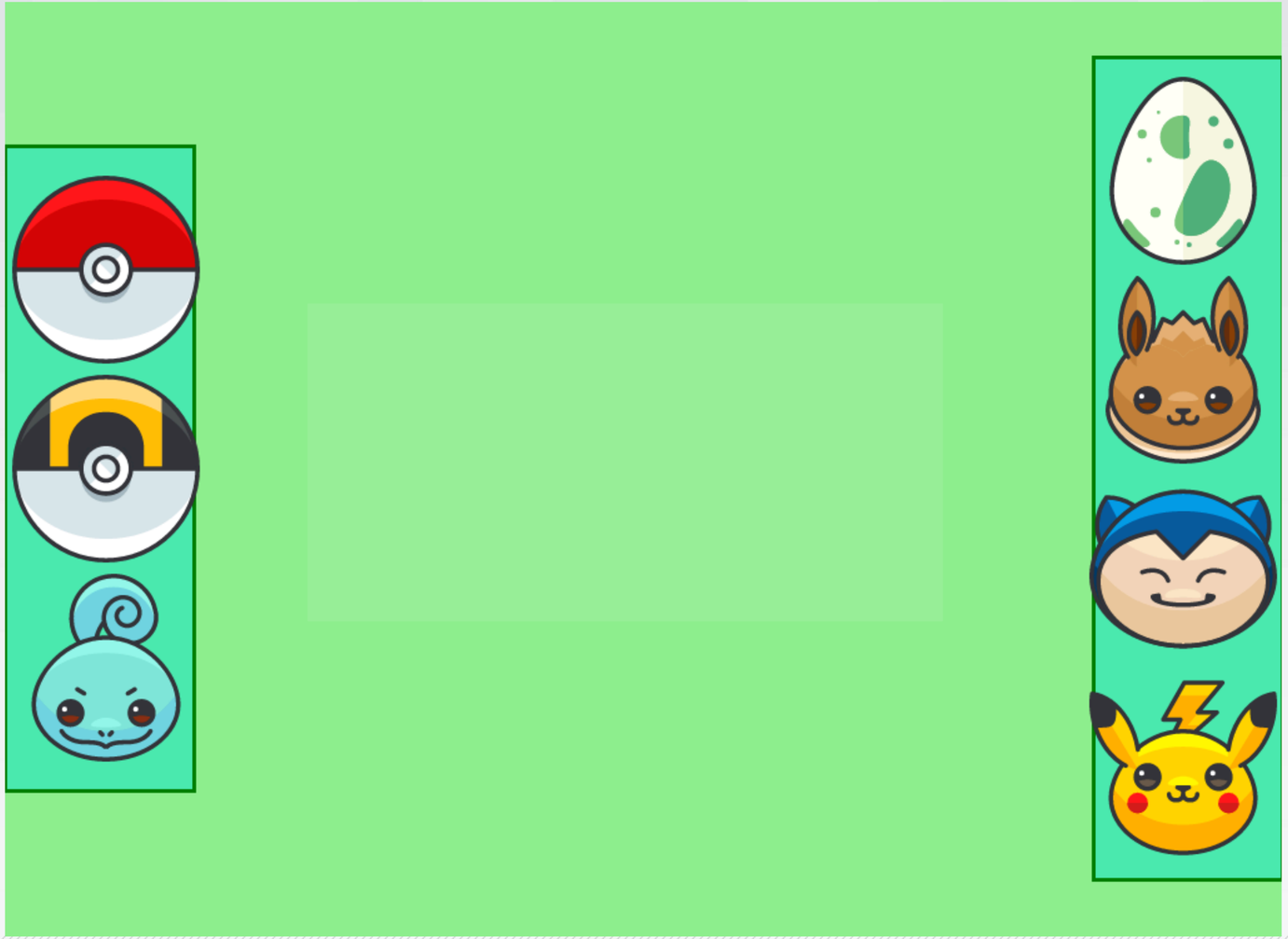
前往挑戰



成果驗收

用你剛剛幫助青蛙和他的朋友們
的最後那口氣(第24題答案)
幫助皮卡丘及他的朋友們

前往挑戰



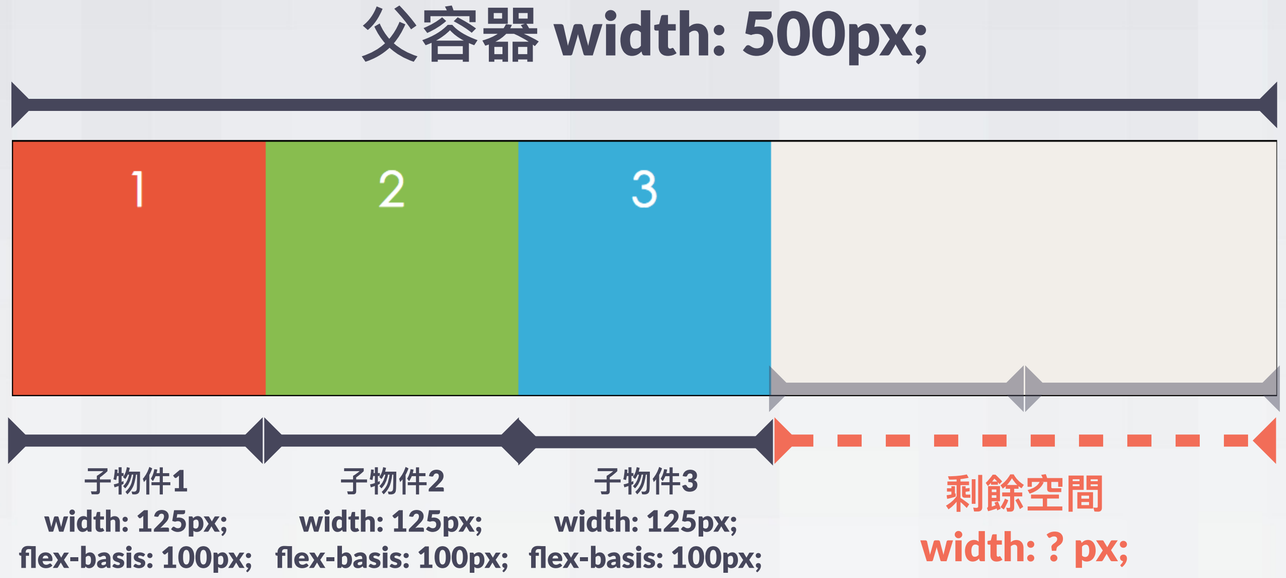
繼續來談談子物件

父容器內的剩餘空間有多大

子物件-寬度

flex-basis > width

在flexbox佈局中flex-basis說的算

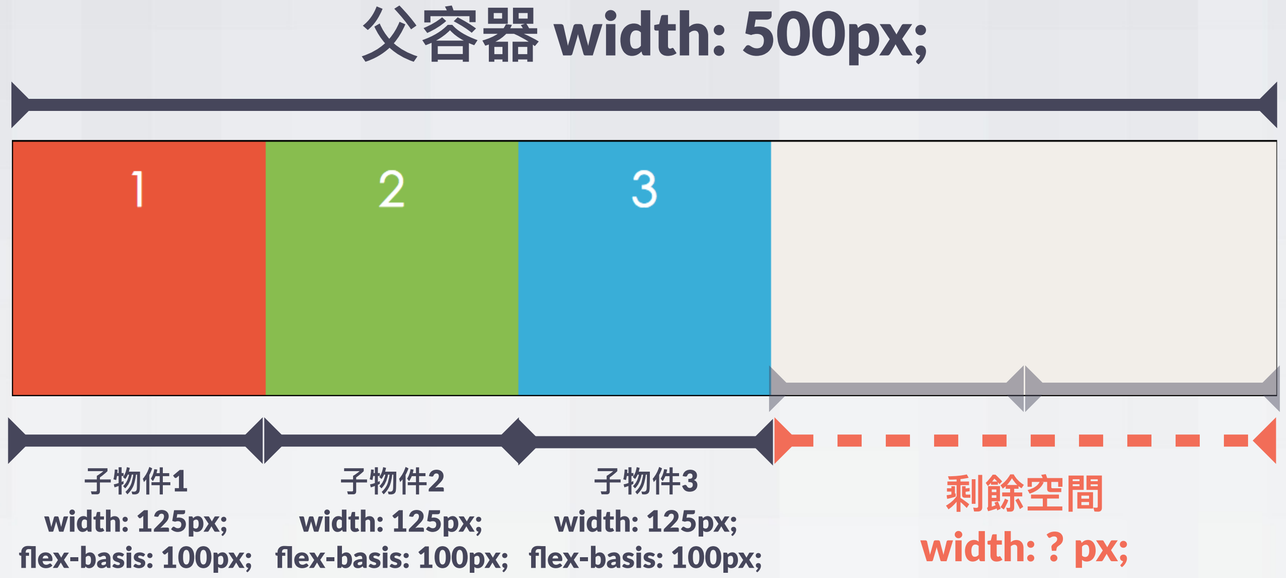


查看結果

子物件-寬度

flex-basis > width

在flexbox佈局中flex-basis說的算



查看結果

如何讓子物件
佔滿父容器寬度?

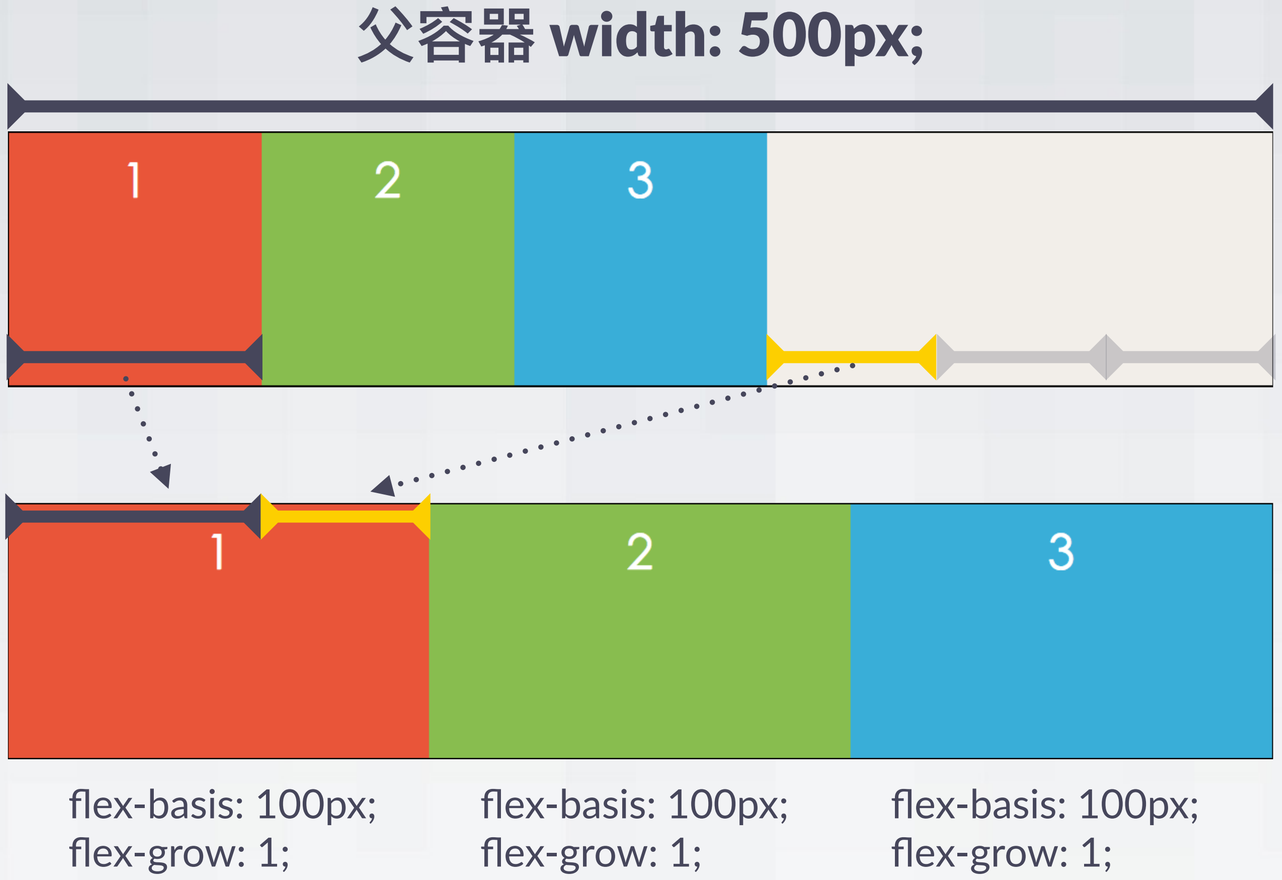
除了用flex-basis或width還有什麼方法？

查看結果

試試 flex-grow

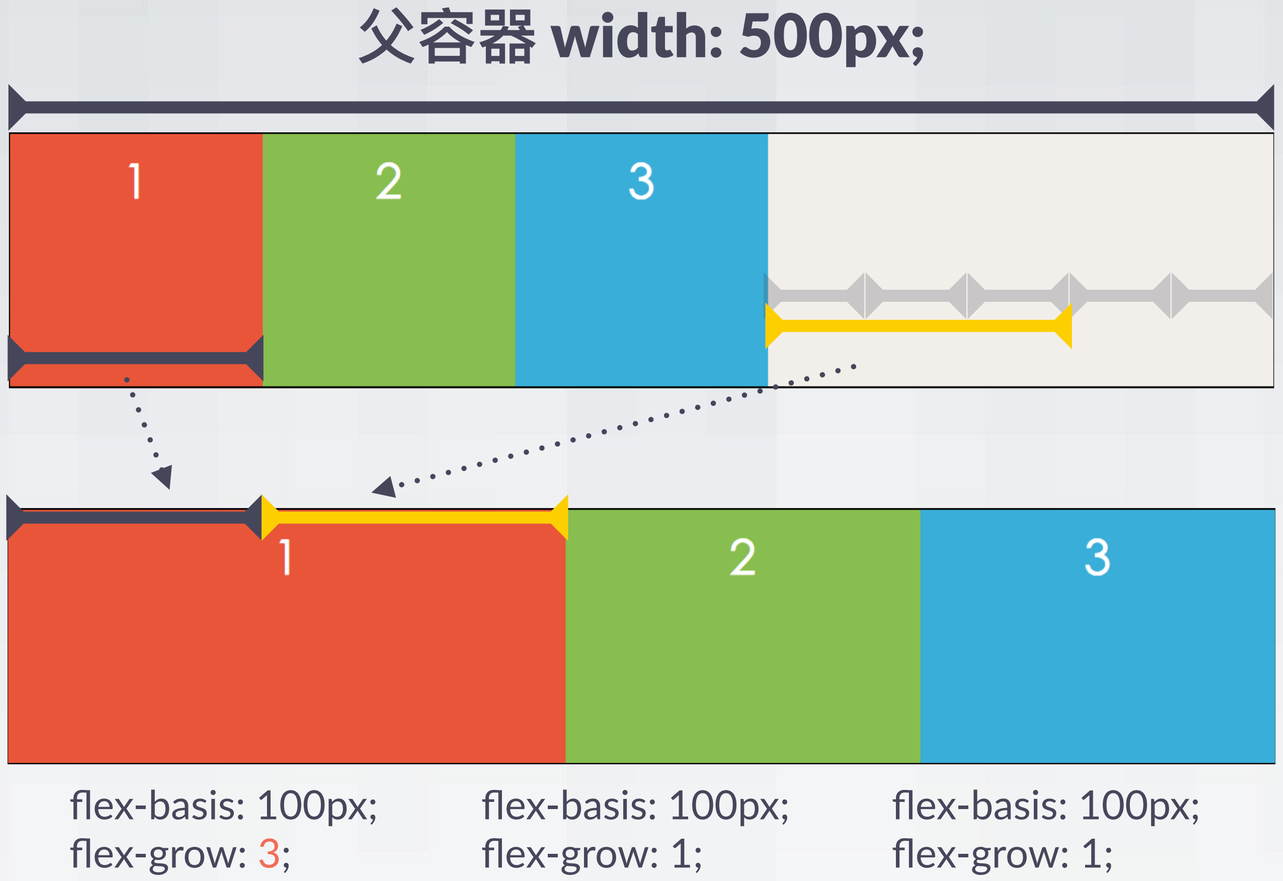
讓子物件「增長」點剩餘空間

每個物件「按比例」拿一點剩餘空間來用



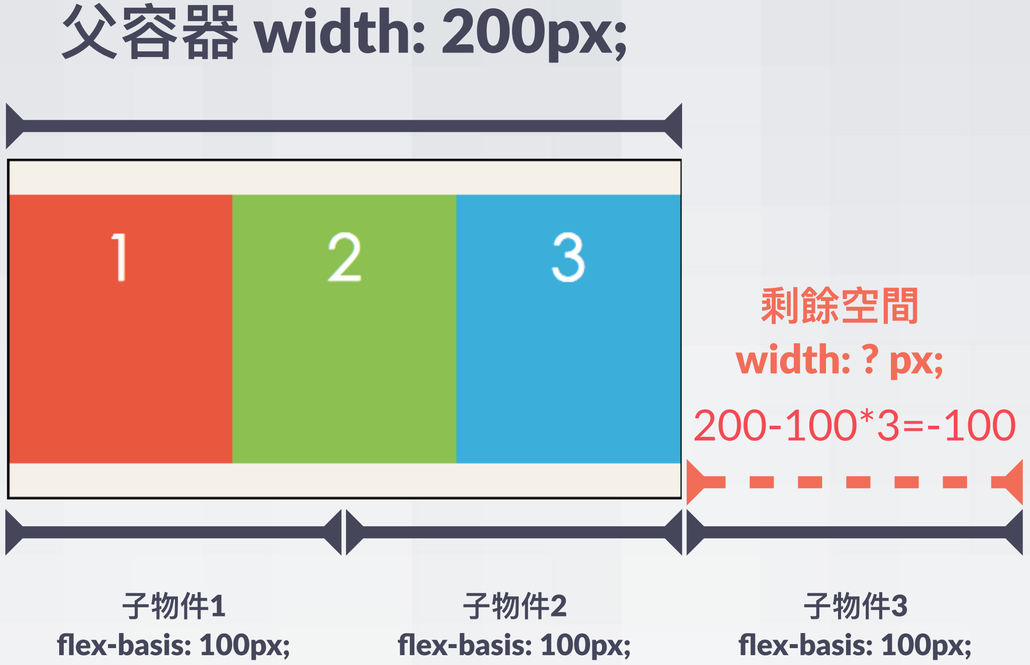
查看結果

改一下 flex-grow 比例



查看結果

如果父容器比子物件窄？
也就是剩餘空間是負的！



查看結果

flex-grow 增長比例
flex-shrink 壓縮比例