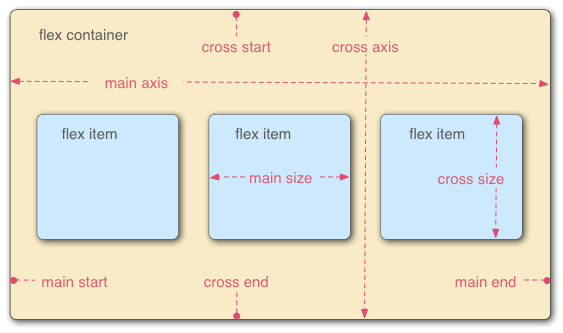
flex container(容器)可用的屬性

flexbox是CSS3裡較為特殊的排版方式，相較於一般常使用的float、position等屬性更為彈性，可以更有效率地做出一般方式所不易達成的效果。隨著響應式布局的興起，排版彈性大的flexbox使用率也頻繁了起來。  
flexbox盒子模型如下圖(圖片來源:[MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Using_CSS_flexible_boxes))，不同於其他一般的CSS屬性，它有專屬於自己的一些屬性和概念，flexbox具有水平的起點與終點(main start、main end)、垂直的起點與終點(cross start、cross end)、水平軸與垂直軸(main axis、cross axis)。



總體來說分為flex container(容器)與flex item(子項目)兩大類。

如何使用flexbox的功能呢？

只要在容器內加上display:flex或是display:inline-flex，容器和其內的項目就會直接套用flexbox的設定喔！

display:flex與display:inline-flex的差別

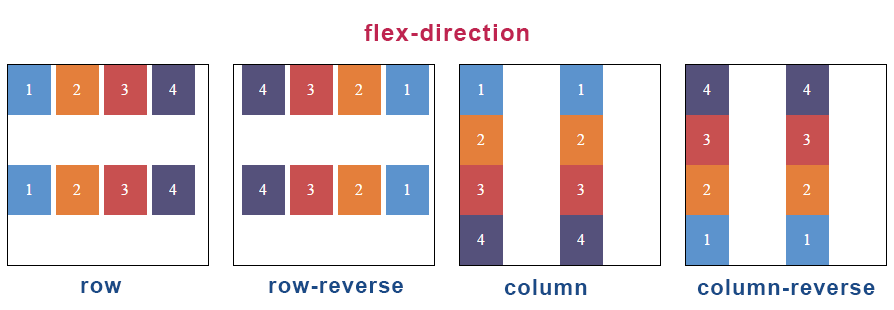
就跟display:block與display:inline-block差不多

此章先介紹flex container(容器)可用的屬性，有以下幾項：

**1. flex-direction**為容器內項目的排列方向

**row** (從左到右，再從上到下) / **row-reverse**(row的顛倒順序)

**column**(從上到下，再從左到右) / **column-reverse**(column的顛倒順序)

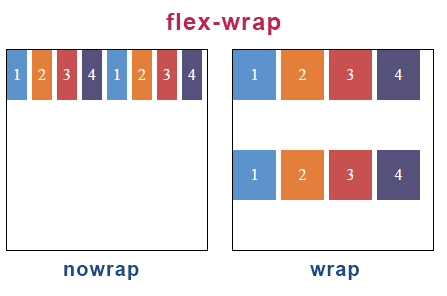


**2. flex-wrap**決定flex項目是否換行

**nowrap**：不換行(預設值)，不管有多少項目都會放在同一行內(寬度會自動調整)

**wrap**：換行，會依照內容或寬度自動換行。

**wrap-reverse**：換行，但順序相反



**3. flex-flow**是  flex-direction與  flex-wrap屬性的縮寫，就跟「border-width:1px;border-style:solid; border-color:#000」等於「 border:1px solid #000」是一樣的道理 flex-direction | flex-wrap 例：.box{flex-flow: row wrap}

**4. justify-content**決定flex項目與容器的水平對齊設定

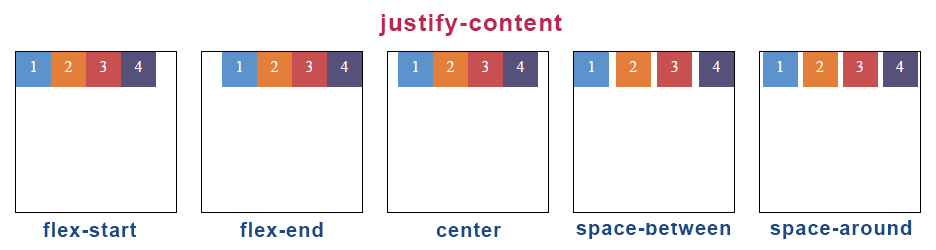
**flex-start**：預設值，flex項目對齊最左邊的main start(水平起點)

**flex-end**：flex項目對齊容器最左邊的main end(水平終點)

**center**：水平置中

**space-between**：平均分配  flex項目，最左與最右的flex項目會與main start及main end貼齊。

**space-around**：平均分配  flex項目，間距也是平均分配。



**5. align-items**與justify-content相反，決定的是flex項目與容器的垂直對齊設定

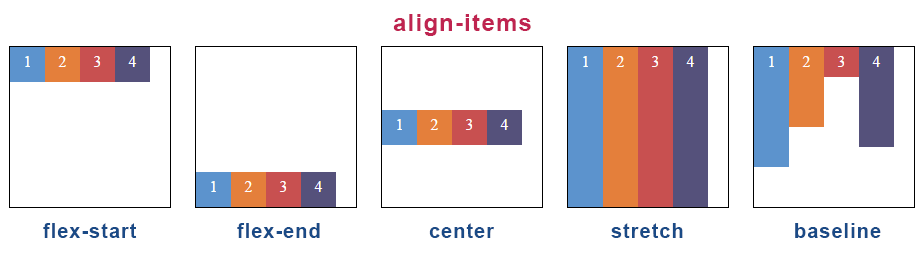
**flex-start**：預設值，對齊容器最上面的cross start(垂直起點)

**flex-end**：對齊容器最下面的cross end(垂直終點)

**center**：垂直置中

**stretch**：將flex項目撐開至容器的高度

**baseline**：以所有flex項目的基線作為對齊標準



**6. align-content**

align-items是針對內容為單行的項目進行處理，但若是多行的項目，就要使用align-content此屬性。

**flex-start**：預設值，對齊容器最上面的cross start(垂直起點)

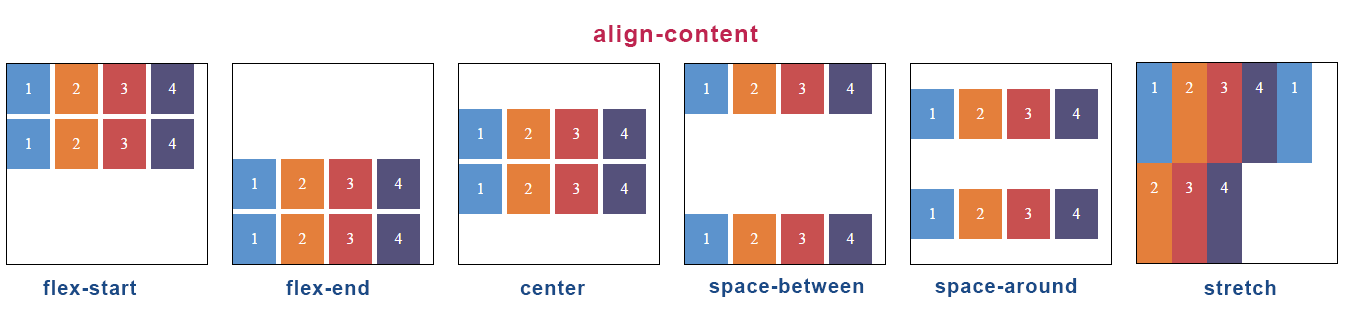
**flex-end**：對齊容器最下面的cross end(垂直終點)

**center**：垂直置中

**space-between**：第一行與最後一行分別貼齊最上方與最下方

**space-around**：行距平均分配

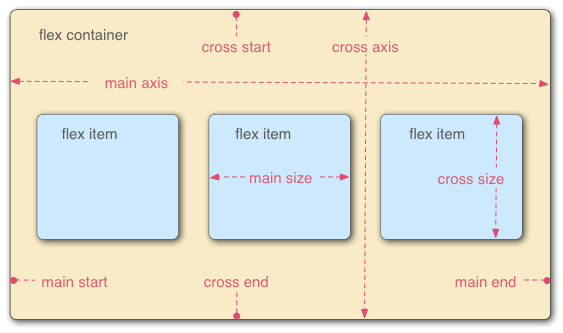
**stretch**：將flex項目撐開至容器的高度



flex items(子項目)可用的屬性

上回介紹了flex-container(容器)可使用的屬性，這次來介紹flex-items(子項目)可操作的屬性。

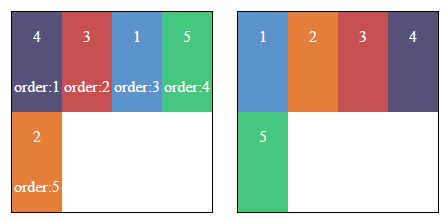
一樣先來張flexbox盒子模型圖複習一下(來源:MDN)



關於flex-items可操作的屬性，有以下幾項：

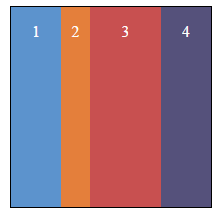
1. **order**可以決定項目的排列順序。

例：下面兩圖僅差在是否加入了order，若未加order指定順序的話會如右圖所示。左圖則是已加入order。



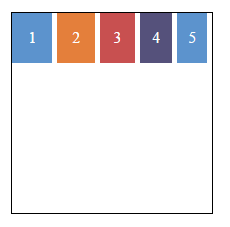
1. **flex-grow**可設定項目們在容器內的比例。

例：每個項目在容器內的比例是2:1:3:2



1. **flex-shrink**與flex-grow相反，當項目將容器擠爆時，flex-shrink會把項目們縮小設定的比例。會根據flex-shrink設定的值來決定縮小的倍數。

下圖的縮小倍數比：1:2:3:4:5



1. **flex-basis**項目的基本大小，有點類似min-width，算是代替width用的，優先權比width高，但若兩者其中之一為auto，會以沒設auto的為優先。

關於flex-grow、flex-shrink、flex-basis此三項屬性，[這邊](http://zhoon.github.io/css3/2014/08/23/flex.html)有更詳細的解釋

1. **flex**由flex-grow、flex-shrink、flex-basis三個屬性組成，例：.flex{flex:2 1 300px;}
2. **align-self**可以指定某個項目的行為，會覆蓋容器align-items的設定。

**auto**：預設值，繼承容器的align-items屬性，若沒有則為stretch

**flex-start：**對齊容器最上面的cross start

**flex-end：**對齊容器最下面的cross end

**center**：垂直置中

**stretch**：將flex項目撐開至容器的高度

**baseline**：以所有flex項目的基線作為對齊標準

例：下圖本來將容器設align-items: flex-start(靠齊最上方)。然而在第三項與第四項分別設了align-self: center(垂直置中)與align-self: flex-end(靠齊最下方)，覆蓋了容器align-items: flex-start的設定。

