



網頁程式設計

CSS樣式設計

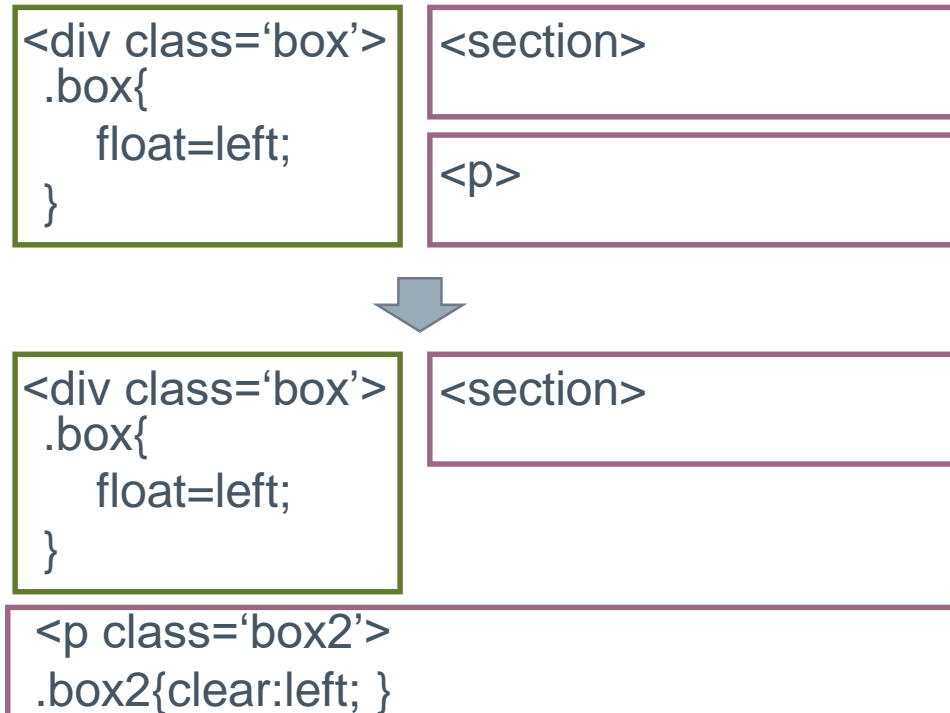
Instructor: 馬豪尚

Float

- › float: 這個屬性可以將一個正常順序中的元素放在容器的左側或右側
- › float: none → 預設
- › float: left → 將元素放在容器的左側
- › float: right → 將元素放在容器的右側
- › 用這個屬性可以做到類似文繞圖(block)的效果，緊鄰著設定float:left或right的inline box會預設排在其旁邊的位置

Clear

- › clear可以設定inline box的哪一邊不要緊鄰著文繞圖box，使其變成正常順序的下方顯示
- › clear: none | left | right | both



Float搭配clear版面設計

頁首

```
header { width: 100%; background: #eaeaea; }
```

導覽列

```
nav { width: 100%; color: white; background: black; }
```

主要內容

```
main {  
  width: 70%;  
  height: 300px;  
  background: skyblue;  
  float: left;  
}
```

側邊欄

```
aside {  
  width: 30%;  
  height: 300px;  
  background: pink;  
  float: right;  
}
```

頁尾

```
footer { width: 100%; background: #eaeaea; clear: both; }
```

頁首 header { width: 100%; background:#eaeaea; }

導覽列 nav {width: 100%; color: white; background: black; }

主要內容

```
main {  
  width: 70%;  
  height: 300px;  
  background: skyblue;  
  float: left  
}
```

側邊欄

```
aside {  
  width: 30%;  
  height: 300px;  
  background: pink;  
  float: right;  
}
```

頁尾 footer { width: 100%; background: #eaeaea; clear: both; }

頁首

```
header { width: 100%; background:#eaeaea; }
```

導覽列

```
nav {width: 100%; color: white; background: black; }
```

主要內容

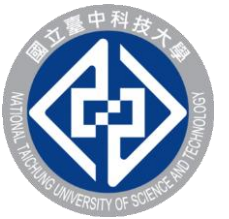
```
main {  
  width: 70%;  
  height: 300px;  
  background: skyblue;  
  float: left;  
}
```

側邊欄

```
aside {  
  width: 30%;  
  height: 300px;  
  background: pink;  
  float: right;  
}
```

頁尾

```
footer { width: 100%; background: #eaeaea; clear: both; }
```



頁首

```
header { width: 100%; background:#eaeaea; }
```

導覽列

```
nav {width: 100%; color: white; background: black; }
```

主要內容

```
main {  
  width: 70%;  
  height: 300px;  
  background: skyblue;  
  float: left  
}
```

頁尾

```
footer { width: 100%; background: #eaeaea; clear: both; }
```

側邊欄

```
aside {  
  width: 30%;  
  height: 300px;  
  background: pink;  
  float: right;  
}
```

Float搭配clear版面設計

頁首 header { width: 100%; background: #eaeaea; }

區塊1

```
#one {  
  width: 32%;  
  height: 300px;  
  background: hotpink;  
  margin-right: 2%;  
  float: left;  
}
```

區塊2

```
#two {  
  width: 32%;  
  height: 300px;  
  background: deepskyblue;  
  margin-right: 2%;  
  float: left;  
}
```

區塊3

```
#three {  
  width: 32%;  
  height: 300px;  
  background: orange;  
  float: right;  
}
```

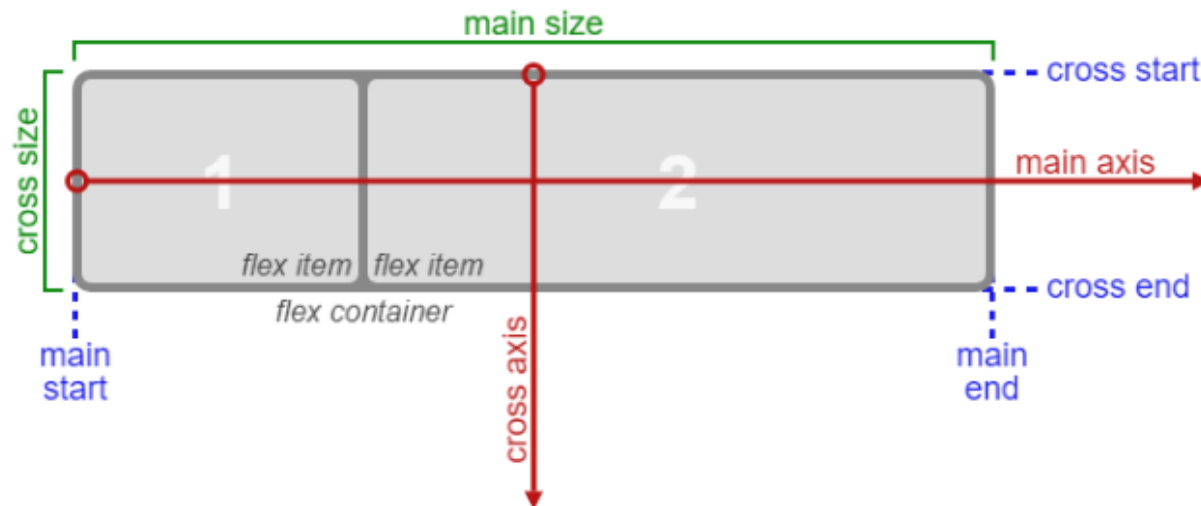
頁尾 footer { width: 100%; background: #eaeaea; clear: both; }



Flex Layout彈性版面設計

Flex Box Layout 彈性版面

- › Flexbox 顧名思義就是彈性的盒子
 - Flex Container(彈性容器)作為彈性版面的父元素容器
 - Flex Item(彈性項目)放在container裡面的子元素，一個父元素可以放多個子元素
- › Flexbox 具有主軸起點、終點、尺寸與交錯軸起點、終點、尺寸的特性可以進行布局規畫



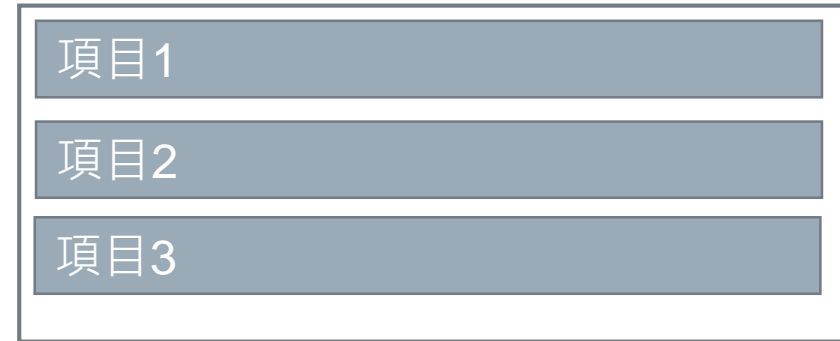
彈性版面 Flex Box Layout

› Flex Box的設定方式和排序

```
<div class="container">  
  <div class="item1">項目1</div>  
  <div class="item2">項目2</div>  
  <div class="item3">項目3</div>  
</div>
```

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

正常順序



彈性容器



Float和FlexBox的差別

- › float 的是針對單一元素去調整它浮動的位置
- › flexbox 則是調整該元素在父元素中主軸和交錯軸的分配。
- › 當我們使用 float 排版時，常常還會需要加入 clear 效果清除浮動，才不會跑版。

Flex box特性

- › 大多數的 flex 屬性，都是設定在 container 身上，而設定後的效果是套用在全部的 items 身上
- › Flex item會忽略 float 設定，還可以設定顯示順序，對佈局來說更方便。
- › float 和 clear 無法使用於彈性容器內
- › vertical-align 對彈性項目沒有效果（但對彈性項目裡面的內容還是有用的）
- › 虛擬元素 ::first-line 和 ::first-letter 無法使用於彈性容器本身

Flex Container(父元素)屬性

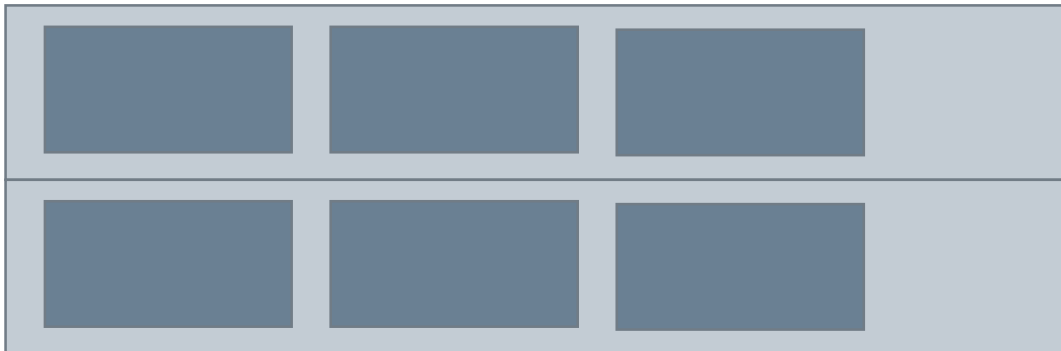
Flex Box Container屬性 display

› display: flex

- 當在 display 屬性宣告 flex，如果沒有設定寬高，彈性容器會像 block 一樣佔據了整行。



- 第二個彈性容器出現，就會像block把第二個彈性容器放到下一行



Flex Box Container屬性 display

› display: inline-flex

- 當在 display 屬性宣告 inline-flex，如果沒有設定寬高，彈性容器會像 inline-block 一樣被其中的子元素彈性項目的尺寸撐開



- 第二個彈性容器出現，它就會像 inline-block 和第二個彈性容器併排(如果畫面寬度可以放得下)



Flex Item彈性項目的排序方向

- › flex-direction: row | row-reverse | column | column-reverse
 - row→由左往右
 - row-reverse→由右往左
 - column→由上往下
 - column-reverse→由下往上

Flex Item的水平對齊方式

- › justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly | start | end | left | right;
 - flex-start → 對齊主軸線開頭
 - flex-end → 對齊主軸線尾端
 - center → 對齊主軸線中央
 - space-between → 將空白空間分配在東西之間
 - space-around → 將空白空間分配在東西兩側
 - space-evenly → 將空白剩下的空間均勻分配

Flex Item的水平對齊方式

› justify-content: flex-start 預設值，對齊主軸線最前端

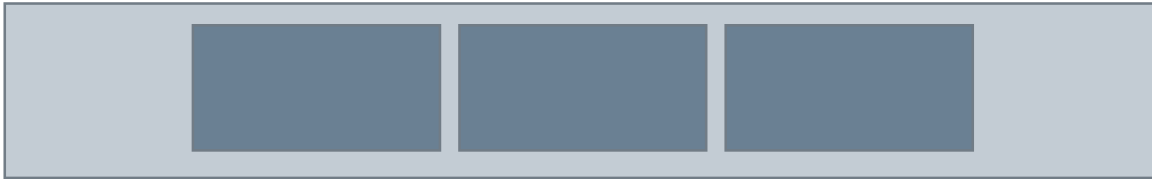


› justify-content: flex-end 對齊主軸線最終端



Flex Item的水平對齊方式

› justify-content: center 對齊主軸線中央

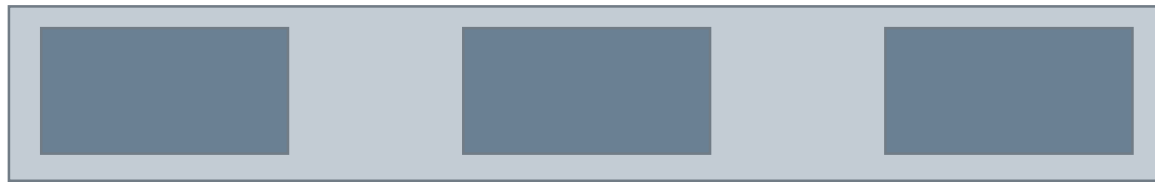


› justify-content: space-around 對齊中軸，平均分配空白寬度和間距

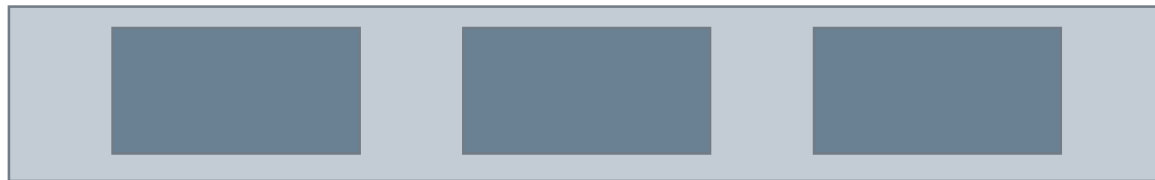


Flex Item的水平對齊方式

- › justify-content: space-between 平均分配空白，第一項和最後一項靠邊緣



- › justify-content: space-evenly → 將空白剩下的空間均勻分配



Flex Item的垂直對齊方式

- › align-items: stretch | flex-start | flex-end | center | baseline | first baseline | last baseline | start | end | self-start | self-end;
 - stretch → 延伸，將內容元素撐開至 flexbox 大小
 - flex-start → 對齊交錯軸線最前端
 - flex-end → 對齊交錯軸線最末端
 - center → 對齊交錯軸線中央
 - baseline → 對齊內容物的基線

Flex Item的垂直對齊方式

- › align-items: stretch 預設值，將內容元素撐開至 flexbox 大小



- › align-items: flex-start 對齊交錯軸線最前端

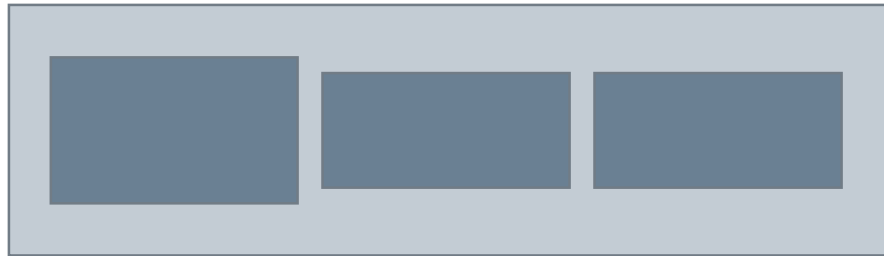


Flex Item的垂直對齊方式

› align-items: flex-end 對齊交錯軸線最末端



› align-items: center 對齊交錯軸線中央



Flex Item的垂直對齊方式

- › `align-items: baseline` 對齊內容物的基線，通常代表的是文字的底部



Flex Item的換行方式

- › Flex-wrap: nowrap | wrap | wrap-reverse
 - nowrap: 不換行
 - wrap: 自動換行，由上往下
 - wrap-reverse: 自動換行，由下往上

Flex Item的多行對齊方式(對齊整個內容)

- › align-content: flex-start | flex-end | center | space-between | space-around | stretch
 - flex-start → 對齊交錯軸線最前端
 - flex-end → 對齊交錯軸線最末端
 - center → 對齊交錯軸線中間(置中)
 - space-between → 第一行與最後一行分別對齊交錯軸線最前端與最末端，再以相等間隔排列剩下的
 - space-around → 每行平均分配每行間距
 - stretch → 延伸對齊到整個flexbox大小

Flex Item(子元素屬性)

Flex Item的顯示順序

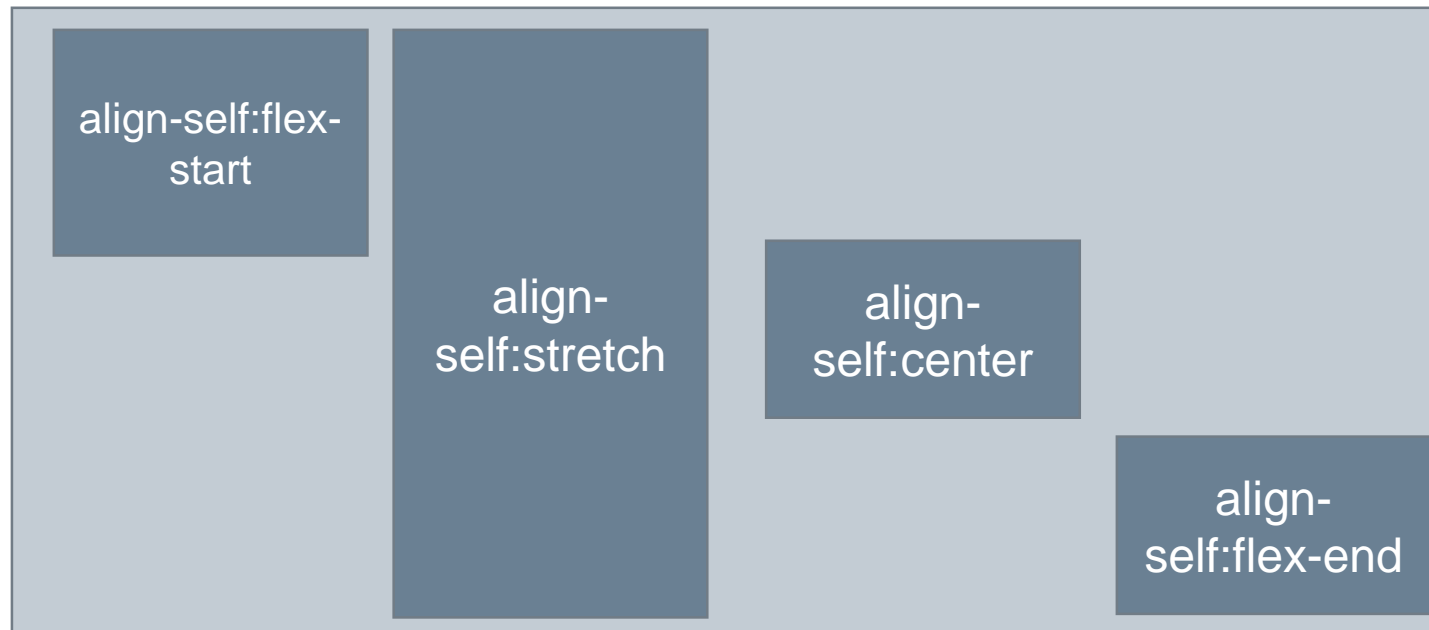
- › order 是針對flex item(子元素)去設定它的顯示順序
 - 正整數
 - 預設是0
 - 數字越大代表顯示順序越後面

個別Flex Item的屬性

- › 設定個別flex item的垂直對齊方式
 - align-self: stretch | flex-start | flex-end | center | baseline
 - stretch → 延伸到整個父元素高度
 - flex-start → 對齊父元素交錯軸線最前端
 - flex-end → 對齊父元素交錯軸線最末端
 - center → 對齊父元素交錯軸線中間(置中)
 - baseline → 對齊父元素的基準線
- › 設定個別flex item的大小
 - flex-basis: 長度 | 百分比 | auto
 - › 有點類似min-width，算是代替width用的，優先權比width高

個別Flex Item的align-self

› 分別設定每個Flex item的垂直位置



練習

- › 運用CSS Flex Box Layout重新定位你的論文瀏覽網頁架構，和CSS_flex.ppt長的一樣
 - 每一個框都是一個box，每個box的大小和排版的位置盡量要和ppt內的設定接近，每個box都要設定框線，框線粗度自行設定
 - 一張紙的大小代表你的最外層父元素Box的大小，這個不用設框線
 - Index Terms做絕對定位在最上面