



HTML + CSS

Here is where your website begins

講師: 劉國良

ENTER



HTML 歷史



- [Tim Berners-Lee](#) invented www (1989)
- HTML (1991)
- HTML+ (1993)
- HTML 2.0 (1995)
- HTML 3 (1997)
- HTML 4.01 (1999)
- XHTML 1.0 (2000)
- HTML 5 初稿 (2008)
- HTML 5 [WHATWG](#) 版本 (2012)
- HTML 5 [W3C](#) 版本 (2014)



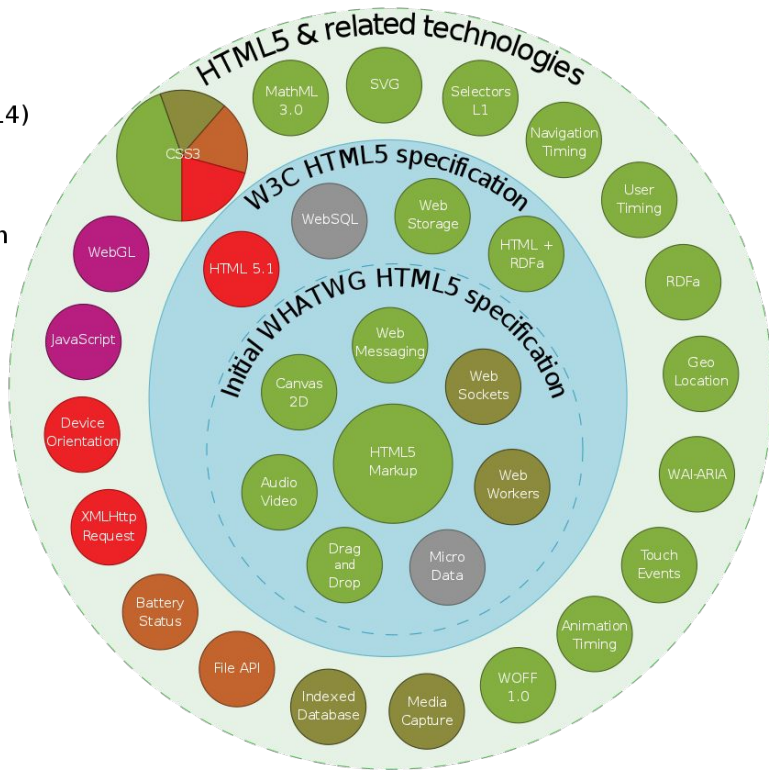
HTML5 包含內容



HTML5

Taxonomy & Status (October 2014)

-  Recommendation/Proposed
-  Candidate Recommendation
-  Last Call
-  Working Draft
-  Non-W3C Specifications
-  Deprecated or inactive





HTML 是什麼？



- 全名是 Hyper Text Markup Language
- 是建立網頁的標準語言
- 用來描述一個網頁的架構與內容
- HTML 包含了一系列的標籤
- 讓瀏覽器知道該如何呈現網頁內容
- HTML 的標籤用來標示每一段內容物是什麼，例如“這是標題”、“這是一段文字”、“這是一個連結”...等。



HTML 標籤總結



- HTML 是一種標記語言 (*markup language*)
- HTML: 包含了一系列的**元素**(elements)
- 元素: 包含了**標籤**(tag) 與**內容**(content)





HTML5 標準寫法



<html>標籤

```
<!DOCTYPE html>
```

第一行代表這是使用 HTML5 格式的網頁

```
<html lang="en">
```

<head>標籤

```
<head>
```

<head> 預設裡面會放, 表示資訊用的 <meta> 以及網頁顯示抬頭 <title>

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Document</title>
```

```
</head>
```

<body>標籤

```
<body>
```

```
  <h1>Hello World</h1>
```

```
</body>
```

<body> 預設裡面不放任何東西 是給使用者自行輸入想要出現在網頁的資訊。

例如這邊使用 HTML 的 <h1> 標籤, 並在標籤內寫 Hello World , 網頁就能幫我們生成一個文字標題的 內容了。

```
</html>
```



HTML：標籤說明



元素
Element

<p> 這是內容 </p>

起始標籤
Opening tag

內容
content

結束標籤
Closing tag

元素	由起始標籤、結束標籤、內容所組成
起始標籤	大於、小於的符號「< >」放入元素名稱，如上面的例子 <p>
結束標籤	在結尾加上結束標籤，如果寫錯位置，HTML 結構會大亂，CSS 修改範圍也會整個錯亂。 新手最容易犯錯就是結束標籤位置，務必細心處理結束標籤。
內容	這個元素的內容，以上面的例子來說，內容就是四個中文字。



HTML: 空元素



```

```



標籤

tag



屬性1

attributes



值

value



屬性2

attributes



值

value



屬性3

attributes



值

value

由於空元素沒有結尾標籤，所以要顯示的內容都會放在標籤的各屬性裡面。

常見空元素: <input>
 <hr>





CSS 的三種引用方式



使用 CSS 的時候，有三種方法：

1.在 head 的地方用 link 的方式引用

2.在 head 的地方用 `<style> </style>` 的標籤 123

3.在 HTML 標籤裡面直接寫 `style="....."` (inline - style)

- 業界通常用 1 的方法，因為設計有一致性，方便多個網頁都能引用同一個 CSS style
- 上課為了方便講解，使用第 2 種作法，也能避免引用錯誤 CSS 檔案造成學習困難
- 第 3 種不建議使用，主要是未來程式碼維護不容易，修改時必須一行一行去找來修改，而且還有 CSS 權重過重，不易維護的問題。





HTML: 巢狀元素



元素裡面可以再放元素，我們稱之為「**巢狀元素(nesting element)**」

`<p>`戶外``下午茶最值得推薦的點心``，充滿奶香的鬆餅加上酸甜莓果，大人小孩看了都心動，若再配上咖啡跟美景就更完美。`</p>`

例如：你想強調「下午茶最值得推薦的點心」，我們就可以在`<p>` 標籤裡面，多加一個行內元素 `` 來標記文字，就能對這個標籤做CSS 修改顏色、字體等。

切記，一般處理巢狀結構的時候，我們不會在行內元素(inline) 裡面包區塊元素(block)





HTML: 標籤種類



1. 行內元素 (display: inline;)
2. 區塊級元素 (display: block;)
3. 結合兩者的特性 (display: inline-block;)



HTML: 行內元素



行內元素 (display: inline;)

- 元素可在同一行內呈現, 圖片或文字均不換行
- 不可設定長寬, 元素的寬高由它的內容撐開
- padding, margin 上、下的設定不會有效果





HTML: 區塊元素



區塊元素 (display:block;)

- 元素寬度預設會撐到最大，使其寬度占滿整個容器
- 可以設定長寬 / margin / padding，但仍會占滿一整行
- 設定 margin, padding 上下左右都會影響排版



HTML: inline-block 元素



結合兩者特性的 inline-block (行內級別的區塊元素)

- 以 inline 的方式呈現, 元素不會佔滿整行或換行
- 同時擁有 block 的屬性, 可以設定長寬 / margin / padding
- 設定 margin, padding 會影響排版



HTML: 標題元素



文件標題 (heading)

讓你呈現這些內容的主題，就像一本書有書名、章節名稱和副標題，一份HTML文件也有類似的概念。HTML 最多可以有 6 層的 heading，從 `<h1>` 到 `<h6>`，雖然通常我們只使用 3 ~ 4 層。

```
<h1>My main title</h1>  
<h2>My top level heading</h2>  
<h3>My subheading</h3>  
<h4>My sub-subheading</h4>
```





HTML: 文字元素



段落 (paragraph)

<p> 元素內容會放一個文字段落，在呈現一般文字時，這是我們最常用到的，而且他是區塊元素，適合拿來做文章段落使用。

```
<p>This is a single paragraph </p>
```

如果要在其中強調文字，可以再加 行內元素的標籤來做為 highlight 使用。只要有標籤，我們未來就能方便地使用 css 去修改它的樣式。





HTML: 常見的連結元素



連結 (link)

連結對於網頁來說是非常重要的。若要讓使用者可以方便換頁，我們需要用到這個元素。其中 *a* 代表了「anchor」錨點的英文簡稱。

```
<a href="https://www.google.com/">Google</a> (外部連結)
```

```
<a href="./2.inline.html">Inline Page</a> (伺服器內部連結)
```

```
<a href="#myID">goto myID's position</a> (網頁內部連結)
```

href 這個屬性名稱比較不直觀，不太好記，但它代表的是：*hypertext reference*的縮寫。





HTML: 常見的清單元素



清單 (list)

清單至少會包含兩個元素，以下是最常見的清單類，無順序性與有順序性的：

1. **無順序性清單 (Unordered lists)** 代表這些項目的順序改變，不影響任何是，例如購物清單。項目會包含在 `` 裡面。
2. **有順序性清單 (Ordered lists)** 代表這些項目的順序是有意義的，例如食譜裡的製作步驟。項目會包含在 `` 裡面。

每個項目則分別放在 `` (list item) element 裡面。





CSS 樣式：文字水平對齊



水平對齊 text-align

text-align CSS屬性定義行內內容(例如文字)如何相對它的區塊級父元素對齊。text-align 並不控制區塊元素的對齊，只控制它的行內內容的對齊。

常用的幾種對齊屬性參數：

```
text-align: left;  
text-align: right;  
text-align: center;  
text-align: justify;  
text-align: justify-all;  
text-align: start;  
text-align: end;  
text-align: match-parent;
```

left

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nam
pellentesque vehicula con valis.

start (ltr)

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nam
pellentesque vehicula con valis.

start (rtl)

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nam
pellentesque vehicula con valis

right

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nam
pellentesque vehicula con valis.

end (ltr)

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nam
pellentesque vehicula con valis.

end (rtl)

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nam
pellentesque vehicula con valis

center

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nam
pellentesque vehicula con valis.

justify

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nam
pellentesque vehicula con valis.

justify-all

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nam
pellentesque vehicula con valis.





CSS 樣式：文字垂直對齊



垂直對齊 vertical-align

CSS 的屬性 `vertical-align` 用來指定行內元素 (inline) 或表格單元格 (table-cell) 元素的垂直對齊方式。

`vertical-align` 屬性可被用於兩種環境：

1. 使行內元素 box-model 與其容器垂直對齊。
2. 垂直對齊表格單元內容。

注意：`vertical-align` 只對行內元素、表格單元格元素生效。不能垂直對齊區塊元素。

可使用預設的對齊值

```
vertical-align: baseline;  
vertical-align: sub;  
vertical-align: super;  
vertical-align: text-top;  
vertical-align: text-bottom;  
vertical-align: middle;  
vertical-align: top;  
vertical-align: bottom;
```

或使用自訂的對齊值

```
/* <length> values */  
vertical-align: 10em;  
vertical-align: 4px;  
  
/* <percentage> values */  
vertical-align: 20%;
```





HTML 常見的基本架構 - 清單列表



Order List 有序清單

```
<ol>
  <li><a href="/1.html5.html">Link1</a></li>
  <li><a href="/2.inline.html">Link2</a></li>
  <li><a href="/3.block.html">Link3</a></li>
  <li><a href="/4.inline-block.html">Link4</a></li>
</ol>
```

Unorder List 無序清單

```
<ul>
  <li><a href="/1.html5.html">Link1</a></li>
  <li><a href="/2.inline.html">Link2</a></li>
  <li><a href="/3.block.html">Link3</a></li>
  <li><a href="/4.inline-block.html">Link4</a></li>
</ul>
```





CSS 樣式: 清單列表符號



`list-style-type:` 可以設置列表元素前的標示符號

例如:

```
list-style-type: disc;    實心圓點 (預設)
list-style-type: circle;  空心圓點
list-style-type: square;  實心方塊
list-style-type: decimal; ;
list-style-type: georgian;
list-style-type: trad-chinese-informal;
list-style-type: kannada;
List-style-type: '-';
list-style-type: "\1F44D";
```

`list-style-type` 屬性可以應用在任何 `display` 的值為 `list-item` 的元素上。此外，由於這個屬性是可繼承的，它可以設置在父元素上 (通常是 `` 或 ``) 以便應用於所有子元素 (``)。





HTML 常見的基本架構 - 導覽列的連結



Link1 Link2 Link3 Link4

```
ul {  
  list-style-type: none;  
  background-color: lightblue;  
  margin: 0;  
  padding: 0 20px;  
}  
  
ul > li {  
  display: inline-block;  
  margin: 10px;  
  line-height: 2em;  
}  
  
ul > li > a {  
  text-decoration: none;  
  color: #333;  
}
```





HTML: img 圖片標籤



```

```

- src - 圖片的路徑
- alt - 圖片讀取失敗時的文字顯示
- title - 滑鼠移動到圖片停留跳出文字





HTML 路徑規則(相對路徑)



```

```

相對路徑

/	根目錄
../	上一層目錄
./	同一層目錄





HTML 路徑規則 (絕對路徑)



```

```

絕對路徑

網站主機位置+資料夾名稱+檔案名稱





HTML 常見的基本架構 - 圖片



```
<div class="img-wrap">  
    
</div>
```



CSS: 圖片處理技巧



1. 會在父層包一個 `<div>` (或其他例如 `<picture>` 、`<figure>` 標籤)
2. 利用父層是 block 元素, 去處理寬高以及排版較方便
3. 最後讓 `` 跟父層做一樣寬高, 即可完成圖片處理

```
.img-wrap{  
  width: 400px;  
  height: 400px;  
  overflow: hidden;  
}
```

```
.img-wrap>img{  
  width: 100%;  
  height: 100%;  
  object-fit: cover;  
}
```



CSS: 圖片處理技巧



object-fit:

幫助我們自動處理圖片的工具，可以設定許多屬性



原圖

- 1.cover (最常用)
- 2.contain (次常用)
- 3.fill (預設)
- 4.none (特殊情況用)



cover



contain



fill



none





HTML 常見的基本架構 - 卡片範例



莓果鬆餅

戶外下午茶最值得推薦的點心，充滿奶香的鬆餅加上酸甜莓果，大人小孩看了都心動，若再配上咖啡跟美景就更完美。



莓果鬆餅

戶外下午茶最值得推薦的點心，充滿奶香的鬆餅加上酸甜莓果，大人小孩看了都心動，若再配上咖啡跟美景就更完美。



莓果鬆餅

戶外下午茶最值得推薦的點心，充滿奶香的鬆餅加上酸甜莓果，大人小孩看了都心動，若再配上咖啡跟美景就更完美。





HTML 常見的基本架構 - 卡片程式碼



```
<div class="card">
```

```
  
```

```
  <h3>莓果鬆餅</h3>
```

```
  <p>戶外<span>下午茶最值得推薦的點心</span>，充滿奶香的鬆餅加上酸甜莓  
果，大人小孩看了都心動，若再配上咖啡跟美景就更完美。</p>
```

```
</div>
```





CSS: 權重介紹



我們會將權重分成五個等級，你可以想像成個、十、百、千、萬數字單位，就能清楚計算權重大小了。

萬

!important

千位數

inline-style

百位數

ID 名稱

十位數

CLASS 名稱

個位數

HTML 標籤





CSS: 排版技巧 - 盒子理論

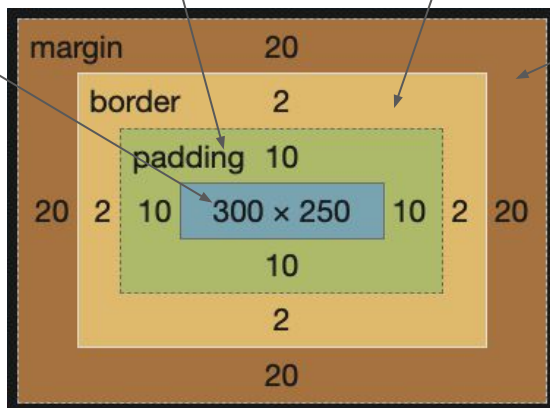


盒子理論 - Box Model

瀏覽器的渲染引擎會根據標準之一的 CSS box model 將所有元素表示為一個矩型的盒子，並且決定這些盒子的大小、位置、及屬性（顏色、背景、邊框...等）

每一個盒子皆由四個區域組成：

內容區 (content)、內邊距 (padding)、邊框 (border)、外邊距 (margin)





CSS: 排版技巧 - 盒子理論



Box-sizing:

如何計算我的盒子到底有多寬？

下方皆為寬度300px，但由於 `box-sizing` 屬性的差異，會得到不同的結果。

如果設定為 `content-box` (預設)，則
CSS 的寬度屬性計算僅包含到藍色區域。



如果設定為 `border-box` (常用)，則 CSS
的寬度屬性計算是包含到淺橘色的邊框區域。





inline-block 詭異的 4px 問題



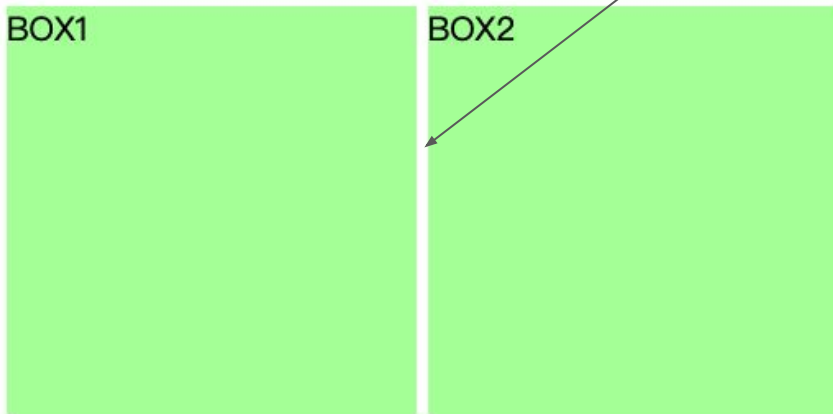
```
<style>
  .box {
    display: inline-block;
    width: 200px;
    height: 200px;
    background-color: #afa;
  }
</style>
</head>

<body>
  <div class="box">BOX1</div>
  <div class="box">BOX2</div>
</body>
```

在 inline-block 模式下，元素 HTML 標籤間的空白、換行及 Tab 字元將佔用一格空間(依 HTML 規範，連續多個空白只會保留一個)這就是 box 之間會間隔 4px 空隙的由來。

BOX1

BOX2





區塊元素詭異的 margin 上下邊界問題



Margin Collapsing 邊界重疊

```
<style>
  p:nth-child(1) {
    margin-bottom: 20px;
  }
  p:nth-child(2) {
    margin-top: 50px;
  }
</style>
```

<p>下邊界範圍會...</p>

<p>...跟這個元素的上邊界範圍重疊。</p>

p 938.4 x 22.4

兩個 p 之間距離只有
50px, 不是 70px

...跟這個元素的上邊界範圍重疊。





區塊元素詭異的 margin 問題 1

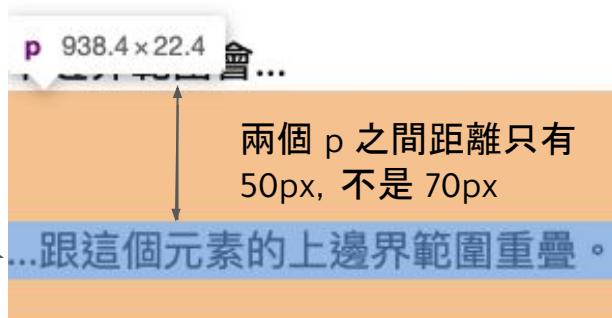


Margin Collapsing 邊界重疊

```
<style>
  p:nth-child(1) {
    margin-bottom: 20px;
  }
  p:nth-child(2) {
    margin-top: 50px;
  }
</style>
```

`<p>`下邊界範圍會...`</p>`
`<p>`...跟這個元素的上邊界範圍重疊。`</p>`

當一個 Block 的下邊界(margin-bottom)
和另一個 Block 的上邊界(margin-top) 都有設定時,
只會留下最大那個 值, 這種情況我們稱為 **邊界重疊**。





區塊元素詭異的 margin 問題 2



Margin Collapsing 邊界重疊

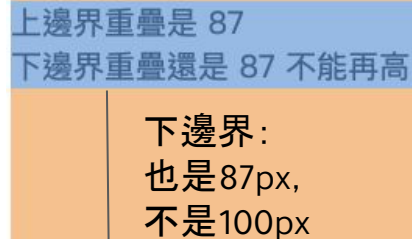
```
<style>
  section {
    margin-top: 13px;
    margin-bottom: 87px;
  }
  header {
    margin-top: 87px;
  }
  footer {
    margin-bottom: 13px;
  }
</style>
```

```
<section>
  <header>上邊界重疊是 87</header>
  <footer>下邊界重疊還是 87 不能再高</footer>
</section>
```

如果第一個子元素跟父元素的上邊界範圍 (margin-top) 貼在一起, 也會發生邊界重疊的情況。除非父元素有設定邊框 (border)、內距 (padding)、內容設定為 inline 或是有加上 clear-fix, 這些都會隔開子元素和父元素的屬性。



section 938.4 x 44.8





區塊元素詭異的 margin 問題 3



Margin Collapsing 邊界重疊

```
<style>
  p {
    margin: 0px;
  }
  div {
    margin-top: 13px;
    margin-bottom: 50px;
  }
</style>
```

```
<p>下邊界範圍是 50</p>
<div></div>
<p>上邊界範圍也是 50</p>
```

空的元素：

當同一個元素上邊界範圍可以直接貼到下邊界範圍時，也會發生邊界重疊。

下邊界範圍是 50

Div 的 margin 實際上只有 50px。

上邊界範圍也是 50

空內容 div，因為中間沒有高度，上下邊界為重疊在一起的狀態，沒辦法做出上、下兩種 margin。





CSS: 排版技巧 - float 浮動排版



浮動 - float

可以輕鬆做到雜誌文繞圖的效果



Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquam, culpa
perferendis? Beatae dolore est corrupti itaque, dignissimos expedita quam
maxime. Obcaecati possimus ipsa dolor sit qui non tempore culpa hic neque
est, maiores ipsam, laudantium, sequi ipsum incidunt quasi? Nobis accusamus
maxime veritatis tempore qui magnam facere nisi repudiandae deleniti ipsum
dolorum id quo, quidem labore ab quaerat iure alias exercitationem vel ea
minus dignissimos? Non officia unde ut ducimus dolorum, asperiores libero
cumque doloribus cum corporis cupiditate aliquam numquam distinctio rerum
nam excepturi sunt nostrum dicta tenetur fugiat odio sit beatae adipisci! Et

tempora doloremque rem fugiat maxime non laudantium aliquam animi facilis nobis veniam commodi aut, dicta voluptas
dolorem fugit sint ut laboriosam officiis minima veritatis voluptatum suscipit nam? Ratione officia vel quis quo ipsum velit
numquam itaque exercitationem. Ipsum, aperiam veritatis similique nulla minima eveniet rerum dolor amet? Iste rerum
culpa modi provident. Architecto iure, enim, quaerat accusantium sit nam praesentium nulla nostrum pariatur sapiente
labore officia, nemo omnis fuga ullam doloribus quia ab rerum? Cumque similique provident, molestiae voluptates id
autem dolore nostrum quibusdam optio nesciunt ea tempore cupiditate, praesentium ad ducimus officiis qui preferendis
consequatur rem quasi? Iure a, porro accusamus voluptatibus molestias doloribus ipsam!



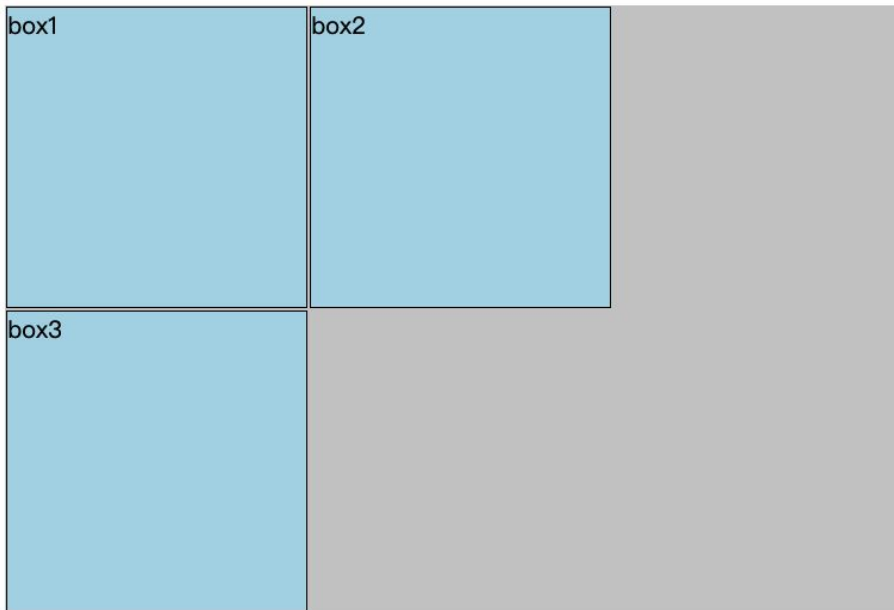


CSS: 排版技巧 - float 浮動排版



浮動 - float

像氣泡一樣的往上浮動，但也有其缺點，排版時需要非常注意寬度是否足夠。



Box3 應該跟著一起往上浮動橫排，但由於 邊框 跟 外邊距各 1px 寬度造成空間不足，只能排第二排，畫面排版就跟你想要的不一樣。

只要超過容器總寬度，即使只差 1px 也會往下掉，造成排版上的困擾。

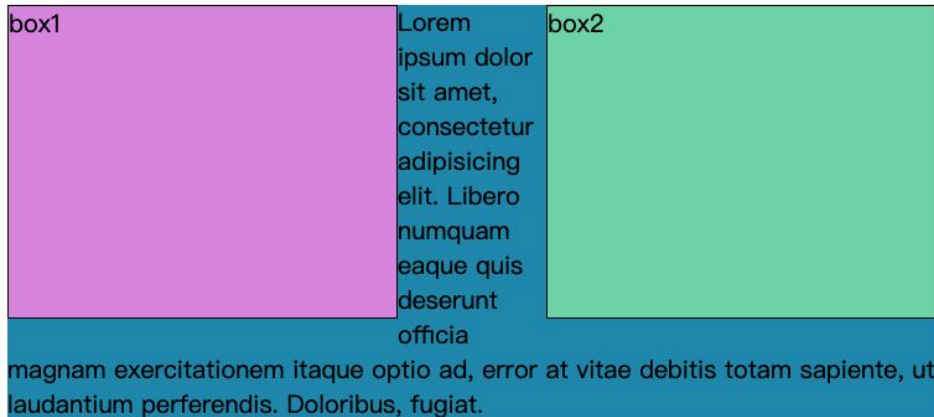


CSS: 排版技巧 - float 浮動排版



浮動 - float

還有一個普遍的問題，就是浮動起來之後的元素，其他 box-model 會辨識不到。



Box 1 往左浮動，Box2 往右浮動

中間的空隙，造成 Box3 判斷可以放文字，因此變成一個詭異的文繞圖效果，但這通常不是我們想要的排版效果。

只能靠 clear-fix 解決。





CSS: 排版技巧 - table 表格排版



```
<table>
  <tr>
    <th>國家</th>
    <th>首都</th>
    <th>人口</th>
    <th>語言</th>
  </tr>
  <tr>
    <td>USA</td>
    <td>Washington D.C.</td>
    <td>309 million</td>
    <td>English</td>
  </tr>
  <tr>
    <td>Sweden</td>
    <td>Stockholm</td>
    <td>9 million</td>
    <td>Swedish</td>
  </tr>
</table>
```

國家	首都	人口	語言
USA	Washington D.C.	309 million	English
Sweden	Stockholm	9 million	Swedish



CSS: 排版技巧 - table 表格排版



```
<table>
  <tr>
    <th>項目</th>
    <th>金額</th>
    <th>總金額</th>
  </tr>
  <tr>
    <td>iPhone 11</td>
    <td>$24,900</td>
    <!-- rowspan - 垂直合併儲存格 -->
    <td rowspan="2">$31,390</td>
  </tr>
  <tr>
    <td>AirPods</td>
    <td>$6,490</td>
  </tr>
</table>
```

項目	金額	總金額
iPhone 11	\$24,900	\$31,390
AirPods	\$6,490	



CSS: 排版技巧 - table 表格排版



```
<table>
  <tr>
    <th>項目</th>
    <th>金額</th>
  </tr>
  <tr>
    <td>iPhone 11</td>
    <td>$24,900</td>
  </tr>
  <tr>
    <td>AirPods</td>
    <td>$6,490</td>
  </tr>
  <tr>
    <td colspan="2">總金額:
$31,390</td>
  </tr>
</table>
```

項目	金額
iPhone 11	\$24,900
AirPods	\$6,490
總金額: \$31,390	



CSS: 排版技巧 - table



項目	金額
iPhone 11	\$24,900
AirPods	\$6,490
iPad Pro	\$25,900
總金額	\$57,290

```
<table>  
  <thead>  
    <tr>  
      <th>項目</th>  
      <th>金額</th>  
    </tr>  
  </thead>  
  <tbody>  
    <tr>  
      <td>iPhone 11</td>  
      <td>$24,900</td>  
    </tr>  
    <tr>  
      <td>AirPods</td>  
      <td>$6,490</td>  
    </tr>  
    <tr>  
      <td>iPad Pro</td>  
      <td>$25,900</td>  
    </tr>  
  </tbody>  
  <tfoot>  
    <tr>  
      <th>總金額</th>  
      <td>$57,290</td>  
    </tr>  
  </tfoot>  
</table>
```



CSS 排版技巧 - grid 網格排版



box1

box2

box3

box4

box5

box6

box7

box8

box9

box10

box11

box12



CSS 排版技巧 - grid 網格排版



display : grid

將下面這三個屬性都設定在父層：

```
display:grid;
```

```
grid-template-columns: 240px auto;
```

定義容器有多少欄 (給寬度值)

```
grid-template-rows: 100px 200px 1fr;
```

定義容器有多少列 (給高度值)

注意！設定單位除了 px 之外，還有 1fr 與 auto 的關係。

如果想設定欄位之間的距離的話使用gap 這個參數，有下面三種寫法：

```
row-gap: 5px; // column-gap: 5px; // gap: 5px 5px;
```

其實 **display:grid** 就是類似表格的排版感覺，但是長寬設定更加彈性了。





CSS 排版技巧 - grid 網格排版



```
<style>
```

```
.container {
```

```
width: 960px;
```

```
height: 500px;
```

```
background-color: #ccc;
```

```
padding: 10px;
```

```
display: grid;
```

```
grid-template-columns: 240px auto 100px auto;
```

```
grid-template-rows: 100px 200px 1fr;
```

```
}
```

```
</style>
```



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

```
<div class="autobg">box1</div>
```

```
<div class="autobg">box2</div>
```

```
<div class="autobg">box3</div>
```

```
<div class="autobg">box4</div>
```

```
<div class="autobg">box5</div>
```

```
<div class="autobg">box6</div>
```

```
<div class="autobg">box7</div>
```

```
<div class="autobg">box8</div>
```

```
<div class="autobg">box9</div>
```

```
<div class="autobg">box10</div>
```

```
<div class="autobg">box11</div>
```

```
<div class="autobg">box12</div>
```

```
</div>
```



CSS 排版技巧 - grid 網格排版



```
<div class="container">  
  <div class="header"></div>  
  <div class="menu"></div>  
  <div class="content"></div>  
  <div class="footer"></div>  
</div>
```



Header

menu

content

footer





CSS 排版技巧 - grid 網格排版



<style>

```
.container {  
background-color: rgb(240, 213, 222);  
width: 800px;  
height: 430px;  
padding: 10px;  
display: grid;  
grid-template-columns: 240px auto;  
grid-template-rows: 100px 200px 1fr;  
gap: 5px 5px;  
grid-template-areas:  
    "header header "  
    "menu content"  
    "footer footer ";  
}
```

```
.item {  
width: 100%;  
height: 100px;  
background-color: #ccc;  
}  
.header {  
grid-area: header;  
background-color: #aff;  
}  
.menu {  
grid-area: menu;  
background-color: #afa;  
min-height: 200px;  
}  
.content {  
grid-area: content;  
background-color: #faa;  
min-height: 200px;  
}  
.footer {  
grid-area: footer;  
background-color: #aaa;  
}  
</style>
```





CSS 排版技巧 - flex 彈性盒子



display : flex

- 現在的主流排版方式 全名為 flex-box
- display:flex 這個css屬性設定在 父層
- 由於很彈性，所以可調整的屬性有很多
- 主要常用是 水平置中、以及 垂直置中 兩種

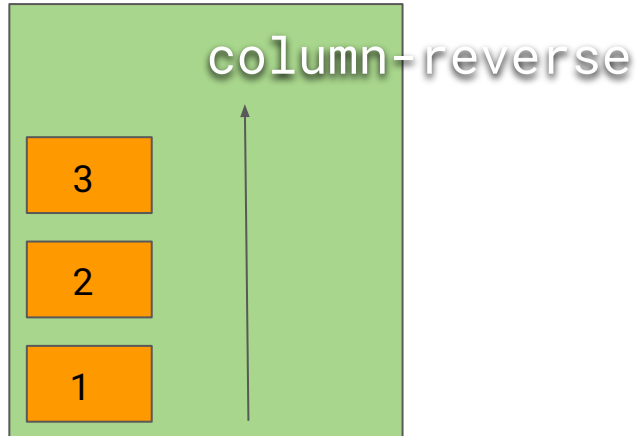
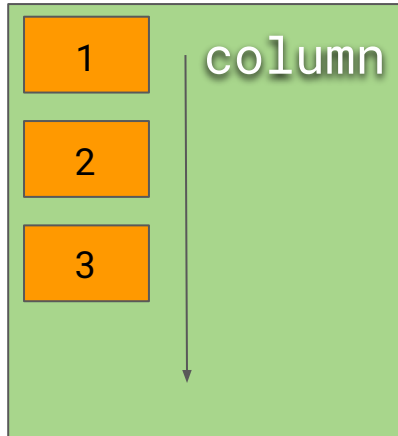




CSS 排版技巧 - flex-direction



主軸可以設定四種方向
預設是 row





CSS 排版技巧 - justify-content




主軸上的排列對齊方式, 預設是 flex-start

flex-start



flex-end



center



space-between



space-around



space-evenly

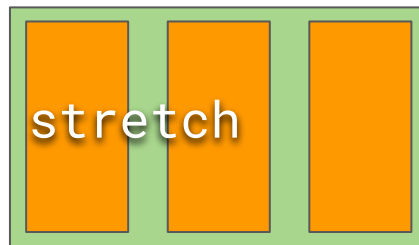
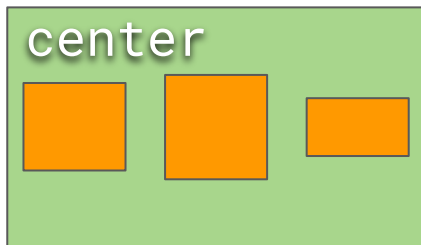
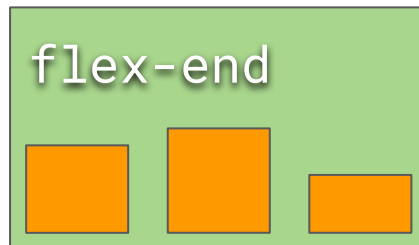
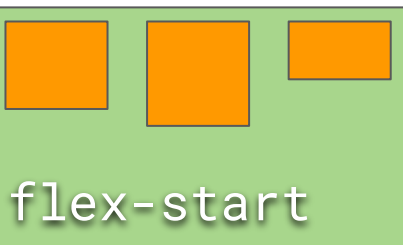




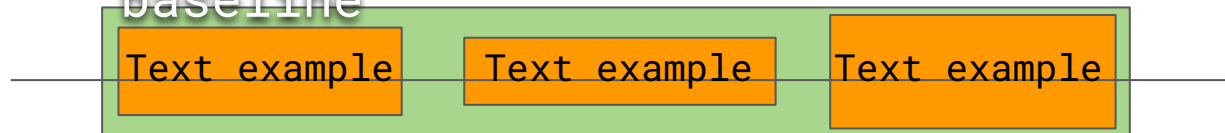
CSS 排版技巧 - align-items



在次軸上的排列對齊方式, 預設是 stretch



baseline





CSS 排版技巧 - flex-grow



當空間足夠的時候, 讓 box 彈性成長, 會自動計算。

預設為0, 需自行針對想彈性長大的 box 各別設定參數。

例如：



此參數有權重的概念, 會依照比例去分配, 計算出每個 box 長大多少





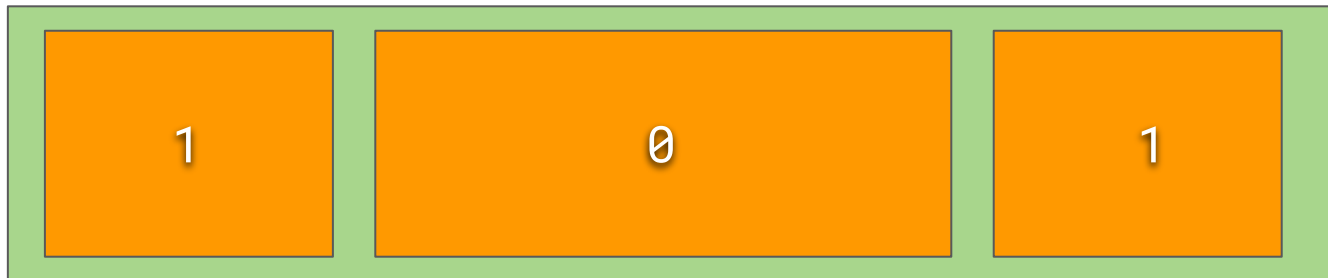
CSS 排版技巧 - flex-shrink



當空間不足夠的時候，讓 box 彈性收縮，會自動計算。

避免爆版超出版面，flex-shrink 預設為1，需自行針對不想彈性收縮的 box 各別設定參數為 0。

例如：





CSS 排版技巧 - flex-basis



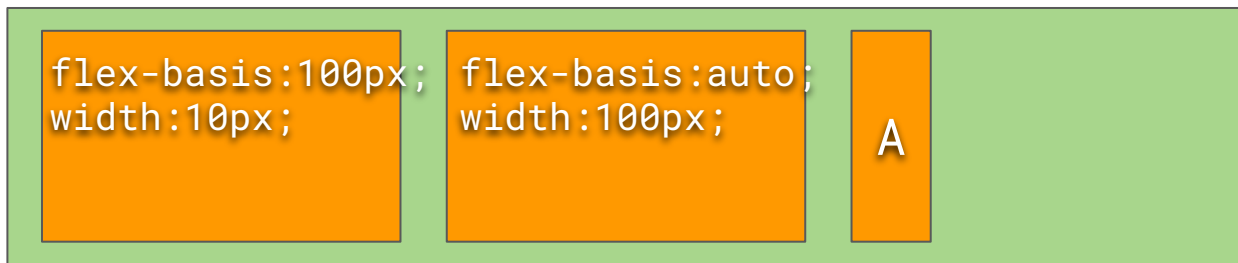
主軸為橫向的時候：

Flex-basis 其實就是決定 items 放進 flex 容器的寬度。

例1:沒給 flex-basis, 則預設值為 auto, 寬度取決於 width 的設定。

例2:如果沒給 width 設定, 則寬度由內容決定。

圖例：

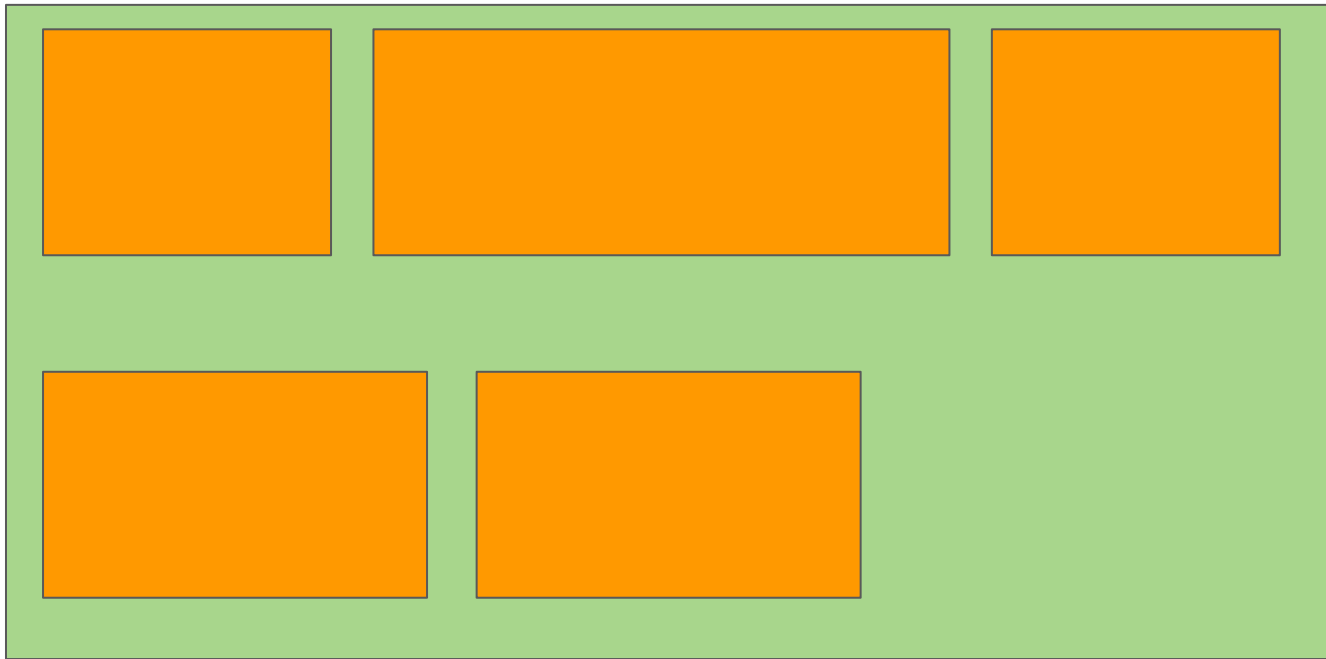


結論: max-width/min-width > flex-basis > width > content





CSS 排版技巧 - flex-wrap



決定超過容器寬度的時候是否要換行。



CSS 排版技巧 - align-content



多行 boxes 在次軸上的排列對齊方式, 預設是 stretch

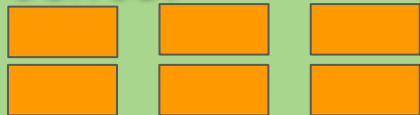
flex-start



flex-end



center



space-between



space-around



stretch



高度被限制不會伸展





CSS 排版技巧 - align-self



Align-self: start

Align-items: end

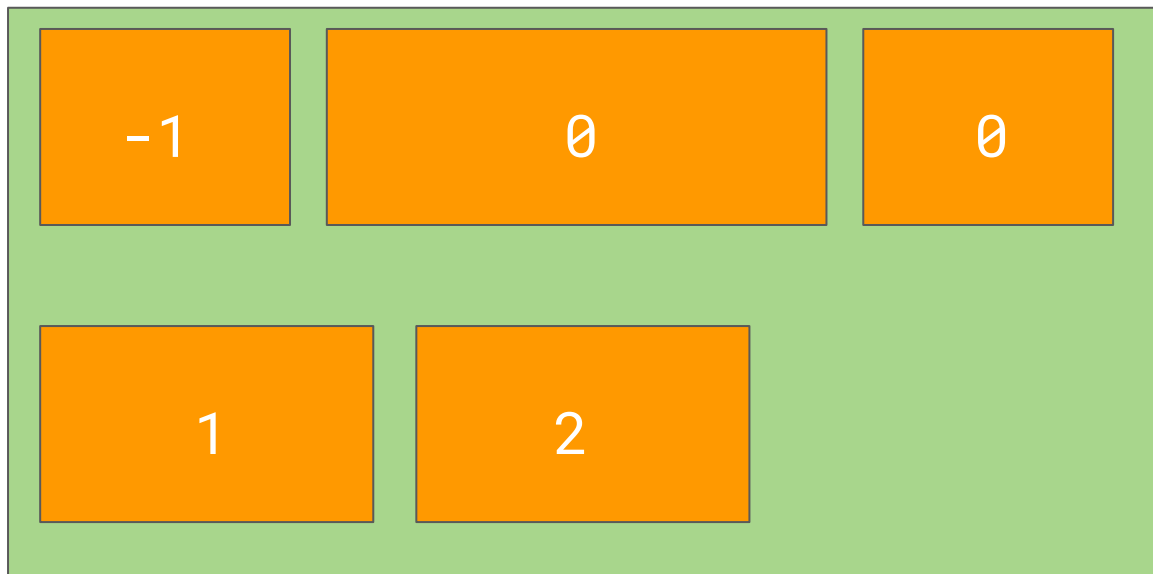
可針對單一個 **item** 自身改變其 **align** 的對齊屬性。



CSS 排版技巧 - order



不改變 HTML 架構的情況下隨時可以改變顯示順序



Order 會將 **items** 由小到大排列, 預設值為 **0**。



CSS 排版技巧 - 網頁實作



[Shop](#) [Lookbook](#) [About](#) [Contact](#)

Alameda



New In

Shop Now



Linen Top / Blue



Linen Top / Blue



Linen Top / Blue

Made with love in Los Angeles from all
natural eco-friendly materials.

