

Cascading Style Sheets (II)

CSS 樣式表

清單的 CSS 樣式

清單的屬性

- list-style-type 設定清單的編號類型
- list-style-position 設定清單的編號位置
- list-style-image 設定清單的影像編號
- list-style 設定清單的所有屬性
- counter-reset 設定清單編號的起始號碼
- counter-increment 設定清單編號的增加量
- display: list-item 設定清單的替代元素
- content 在元素的前面或後面插入新的內容
- Reference
 - <http://www.w3.org/TR/CSS2/generate.html#list-style>

清單的類型

- 語法

list-style-type: disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower-greek | lower-alpha | lower-latin | upper-alpha | upper-latin | hebrew | armenian | georgian | cjk-ideographic | hiragana | katakana | hiragana-iroha | katakana-iroha | **none** | inherit

- 項目符號：disc, circle, square, box, check, diamond, hyphen

- 項目編號：decimal, ..., katakana-iroha

- e.g. **list-style-type.html**

```
li {list-style-type: disc;}
```

設定清單的位置

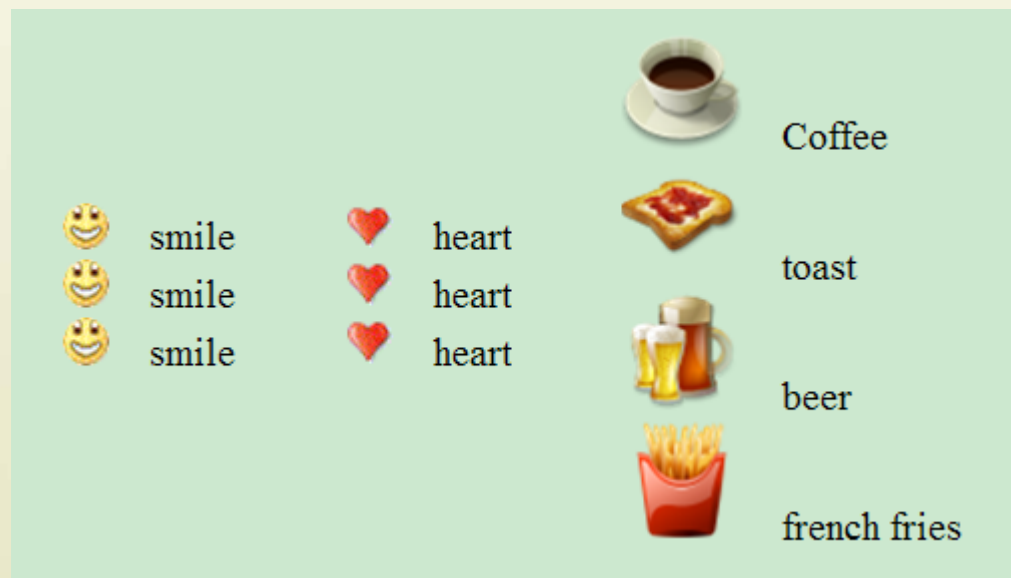
- 使用 `list-style-position` 設定清單的編號位置
- 語法：`list-style-position: inside | outside | inherit`
 - `outside`: The marker box is outside the principal block box.
 - `inside` : The marker box is the first inline box in the principal block box, after which the element's content flows.
- e.g. [list-style-position.html](#)
`li.outside {list-style-position: outside;}`
`li.inside {list-style-position: inside;}`

設定清單的影像

- 使用影像來替代清單編號
- 語法：list-style-image: <uri> | none | inherit
- e.g. [list-style-image.html](#)

```
ol {list-style-image: url(smile.gif);}
ul {list-style-image: url(heart.gif);}
```

- ps: list-style-image, list-style-type 同時設定則 image 優先



設定清單的所有屬性

- 使用 `list-style` 屬性來設定清單的所有屬性：

- e.g.

```
ol {list-style: url(smile.gif) disc outside;}
```

```
ul {list-style: lower-alpha inside;}
```

設定清單的替代元素

- 使用 `display` 屬性值 `list-item`，來指定 `p` 元素擔任清單元素的角色：

- e.g. [list-display.html](#)

```
p {  
  display: list-item;  
  list-style-type: disc;  
  list-style-position: inside;  
}
```


Automatic Counter

- counter-reset
 - creates new counters on an element
 - e.g. `h1 {counter-reset : chapter;}`
 - 如果沒有在 `counter-reset` 屬性內指定號碼，預設值是 0，所以編號會從 1 開始。可以指定從幾號開始編起，e.g. `h1 {counter-reset : chapter 5;}`
這表示編號要從 6 開始。
- counter-increment
 - increase counter
 - e.g. `h1 {counter-increment : chapter 2;}`
- e.g. [counter.html](#)

counter(), counters()

- counter(), counters() : get counter value
 - counter(<ident> [, [<counter-style> | none]]?)
 - counters(<ident>, <string> [, [<counter-style> | none]]?)
 - join the nesting depth counter into a single string
- 註: 在巢狀結構中，counter 會依結構形成一個陣列，例如 counter 名稱為 cnt，第一層用 cnt[0], 第二層用 cnt[1], ...各自獨立，不會互相干擾

Custom Counter Style

- 自訂 counter 的樣式符號, e.g. [counter-style.html](#)

```
@counter-style triangle {  
  system: cyclic;  
  symbols: '\25BA';  
  suffix: " ";  
}  
li { list-style-type: triangle; }
```

- CSS 3 新增, Chrome 還沒支援, 請用 FireFox!
- <http://www.w3.org/TR/css-counter-styles-3/>

CSS 的排版模型和定位樣式

CSS 的排版模型

- 在正常版型 (normal flow) 中，區塊元素是由上而下，垂直地一個接一個排列，左右空間則是整個被區塊元素佔用；行內元素則由左而右排列，只佔用所需要的空間。
 - **note:** 文字元素是由左而右，由上而下排列。但如果地區語言是由右而左閱讀，例如阿拉伯文與希伯來文，文字元素就是由右而左排列。
- 當瀏覽器視窗重新調整大小的時候，區塊元素也會跟著調整寬度，而行內元素則是重新調整位置。
- 可利用 position, float, display, ...等改變 flow 方式

position

- 設定元素的定位類型
- Syntax: `position: static | relative | absolute | fixed | inherit;`
 - `static`: 瀏覽器的預設方式，元素依照出現的順序排列
 - `relative`: **offset relative to normal flow position**
 - `absolute`:
 - Absolutely positioned boxes are taken out of the normal flow.
 - **relative to the first parent element that has a position other than static**
 - `fixed`: the box is fixed with respect to the initial containing block [註] and does not move when scrolled

position - relative

- shift the positioned box according to the *top*, *bottom*, *left*, *right* property value. e.g. layout/position-relative.html
 - top|left|bottom|right: <length>|<percentage>|auto|inherit
 - the offset is with respect to the top|left|bottom|right edges of the box itself (i.e., the box is given a position in the normal flow)
 - for {top|left|bottom|right} offset, the box is shifted {down|right|up|left}, so top:2cm; is equivalent to bottom:-2cm; it's not meaningful to specify both top and bottom (or left and right) for the same element, because one of the values will be ignored.
- the positioned box still remain in the static flow
- the positioned box may overlap the following boxes

position - absolute

- Absolutely positioned boxes are taken out of the normal flow.
- The box's position (and possibly size) is specified with the `top`, `bottom`, `right`, `left` properties.
 - unlike `relative`, `top`, `bottom`, `left`, `right` can be used at the same time
- relative to the first parent element that has a position other than `static`
 - 相對於定位類型為非 `static` 的最近父元素，也就是父元素必須是 `relative`，`absolute`，或 `fixed` 的元素，如果沒有父元素，就使用 `initial containing block`。
- e.g. <layout/position-absolute.html>

position - fixed

- the box is fixed with respect to the initial containing block (browser window)
- a fixed box does not move when scrolled
- The fixed box position (and possibly size) is specified with the top, bottom, right, left properties.
- For media="print" a fixed box will be repeated on each printed page.
- e.g. <layout/position-fixed.html>

註: Containing Block

- One block-level element sits inside another block-level element then the outer block is known as **containing block**, which is generated by the parent element.
- The position and size of an element's box is sometimes calculated relative to the containing block of the element.
 - <http://www.w3.org/TR/CSS21/visudet.html#containing-block-details>
- The containing block for the root element is called the **initial containing block**, which cover whole screen/page area.

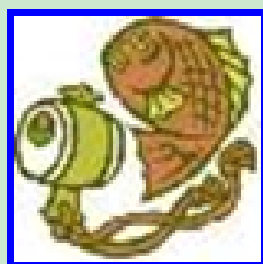
浮動定位 – float 屬性 ^{1/2}

- float 設定元素浮動的位置
 - Box can be floated to the left/right. Content flows on the right/left side of the box
 - 語法：float: left | right | none;
 - 設定 float 會產生 block box，e.g. <a> 本來是 inline-level element，若設定浮動，則會產生 block box (如同設定了 display: block; 一樣)
 - e.g. <layout/float.html>
- clear 清除浮動元素

float 屬性 2/2

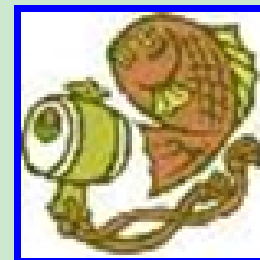
- 浮動的元素會維持在父元素的内容區塊範圍裡面。
- 浮動元素的四周會維持所設間距。周圍的文字會圍繞在浮動元素的四周。
- 如果浮動元素的高度比父元素高，父元素後面的元素不會因此改變位置，但會調整內容來配合浮動元素。
- 浮動 element 的 margin 設為負值時，浮動 element 會與週圍元件重疊
- e.g. <layout/float-overlap.html>

浮動的效果 1/2



開始元素先 **float: left**，後面才接文字
根據 CSS 的排版模型，在正常版型 (normal flow) 中，文字元素是由上而下，由左而右排列。float 屬性值 left 讓元素浮動在左邊，right 讓元素浮動在右邊，none 則是讓元素照原來位置排列不浮動。

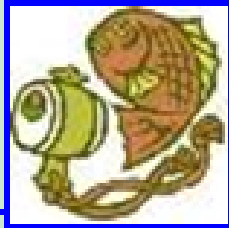
開始元素先 **float: right**，後面才接文字
根據 CSS 的排版模型，在正常版型 (normal flow) 中，文字元素是由上而下，由左而右排列。float 屬性值 left 讓元素浮動在左邊，right 讓元素浮動在右邊，none 則是讓元素照原來位置排列不浮動。



浮動的效果 2/2

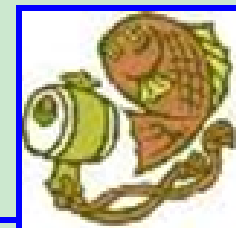
先顯示文字，結尾元件才 `float: left`

根據 CSS 的排版模型，在正常版型(normal flow)中，文字元素是由上而下，由左而右排列。 `float` 屬性值 `left` 讓元素浮動在左邊，`right` 讓元素浮動在右邊，`none` 則是讓元素照原來位置排列不浮動。



先顯示文字，結尾元件才 `float: right`

根據 CSS 的排版模型，在正常版型(normal flow)中，文字元素是由上而下，由左而右排列。 `float` 屬性值 `left` 讓元素浮動在左邊，`right` 讓元素浮動在右邊，`none` 則是讓元素照原來位置排列不浮動。

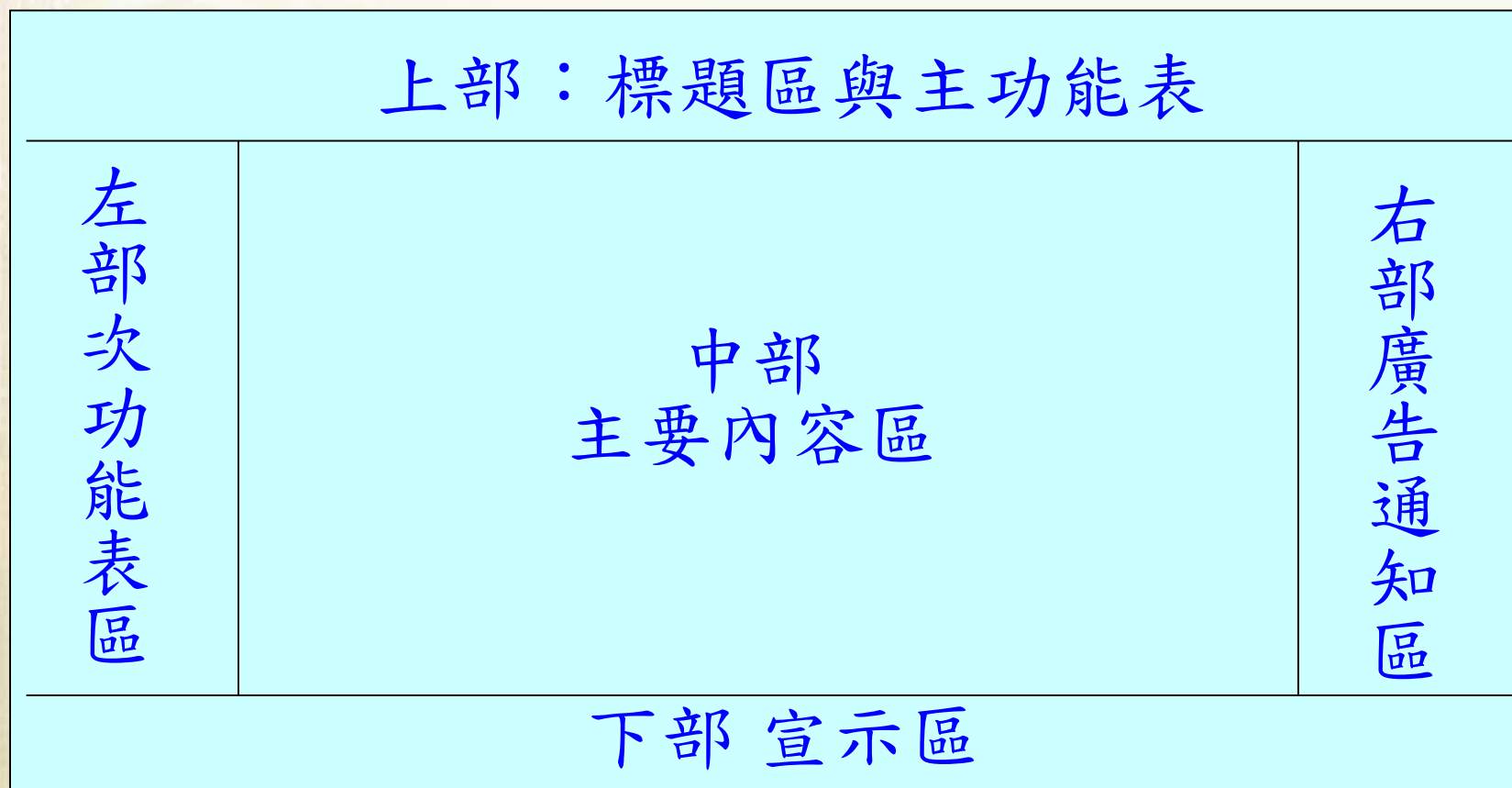


clear 屬性

- 用來清除 float 屬性的作用，設定clear屬性的區塊元素會依照正常的排版形式，移到浮動元素下面一行顯示
- Syntax:
clear: none | left | right | both | inherit
- e.g. [layout/float-clear.html](#)
``
Some Text
`<h2 style="clear: left;">恢復正常顯示</h2>`

典型版面 1/2

- 網頁設計首先要決定版面的配置
- 常見版面配置是將畫面分為五部分

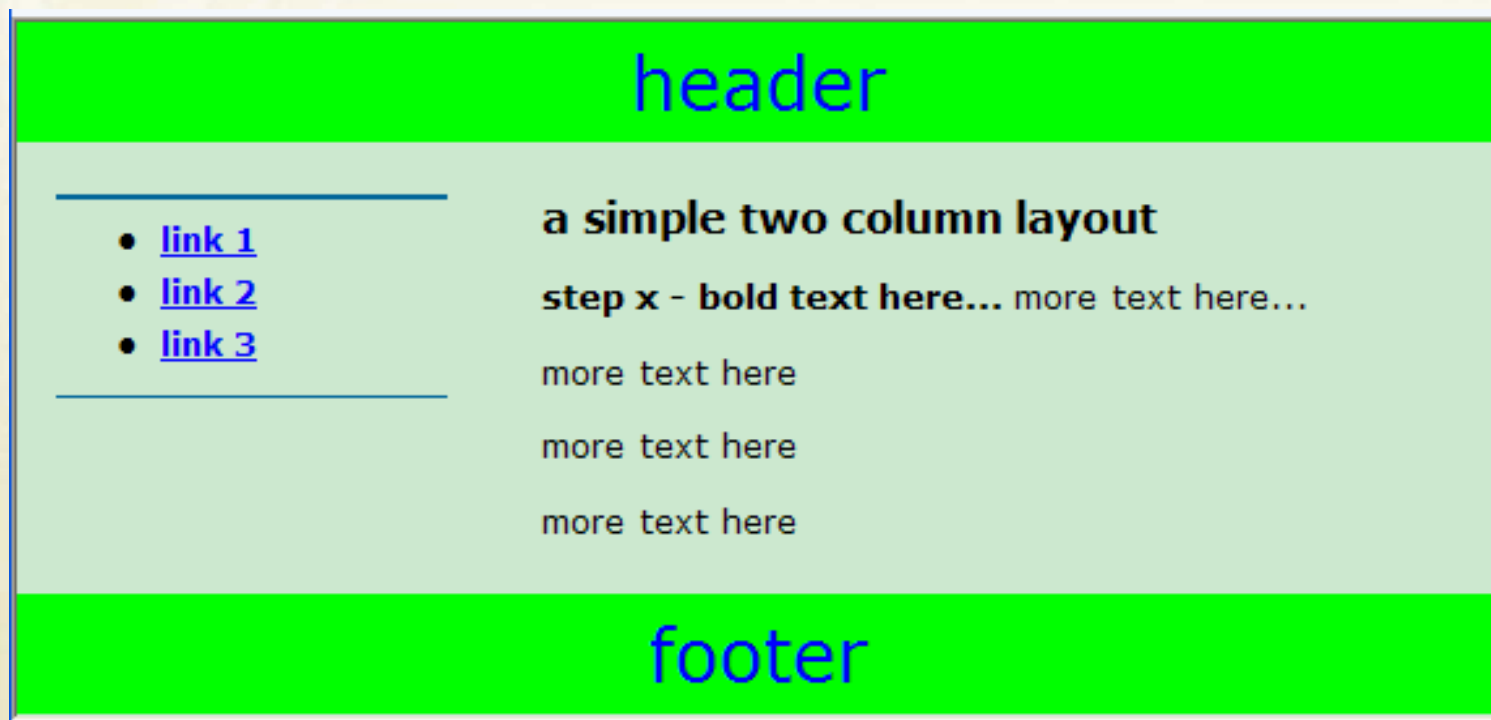


典型版面 2/2

- 上部 header：放 Banner、單位名稱、代表性圖案，底部通常擺放文字主功能表或快捷列
- 下部 footer：聯絡資訊(電話、住址、EMail)、版權宣示、計數器...等等
- 中間 content:
 - 左部：次功能表或重要網頁連結
 - 中部：網頁主要內容
 - 右部：訊息、通知、廣告

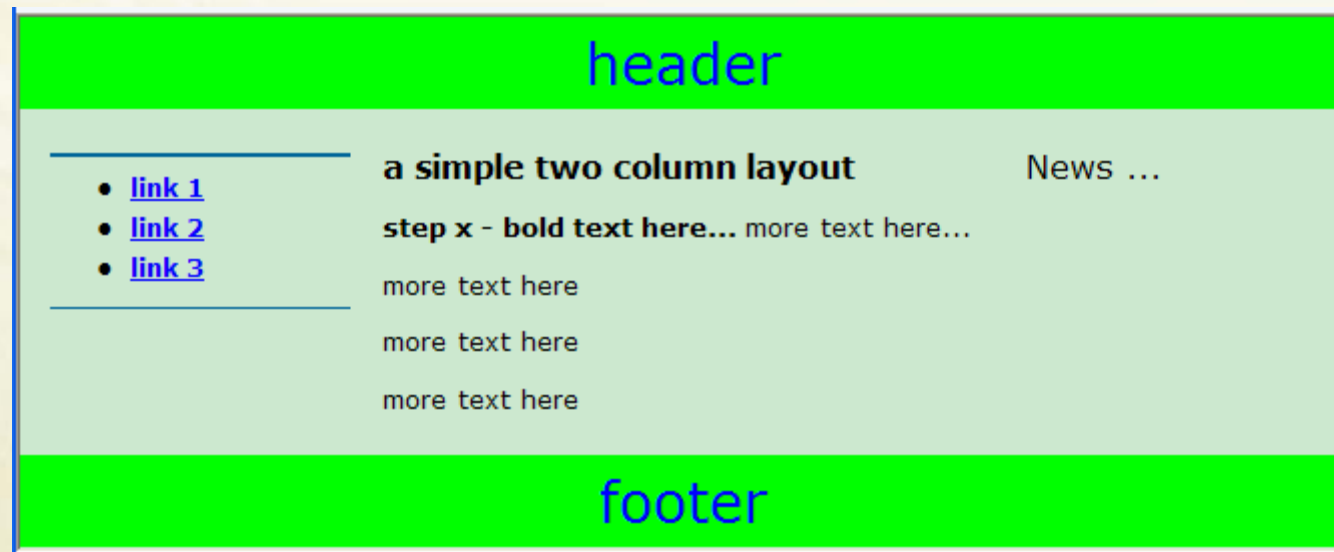
兩欄的版型 - Example

- 下圖，除了header, footer之外，中間content畫面分為兩部分，左邊的 navigation column 和 右邊的 content column



三欄的版型 - Example

- 除了原有兩欄外，右邊再加上 sidebar column，形成三欄版型



多欄版型

- 多欄的版區可利用幾種方式完成:
 - 使用元素浮動(float)
 - 使用絕對定位方式來建立 (position: absolute)
 - (CSS3) 使用 grid 樣式
 - (CSS3) 使用 box 樣式

多欄版型 - 利用元素浮動

- e.g. [layout/layoutFloat.html](#)

- 重要樣式碼

```
div#header { ... }
```

```
div#nav { float: left; width: 150px; margin: 20px 0 0 15px; }
```

```
div#sidebar { float: right; ... }
```

```
div#content { margin: 20px 185px 0 185px; }
```

```
div#footer { clear: left; }
```

```
<div id="nav"> ... </div>
```

```
<div id="sidebar"> ... </div>
```

```
<div id="content"> ... </div>
```

sidebar寫在content之前

多欄版型 - 使用絕對定位

- e.g. [layout/ layoutAbsolute.html](#)

- 重要樣式碼

```
div#header { ... }
```

```
div#nav {
```

```
    position: absolute; left:0px; top:46px; width:150px; }
```

```
div#sidebar {
```

```
    position: absolute; right:10px; top:46px; width:150px; }
```

```
div#content {margin:20px 10px 20px 185px; }
```

```
div#footer { ... }
```

CSS 3 Layout

- Flexible Box Layout
 - multiple columns layout without float
 - `display: box;`
 - `box-orient: horizontal`
 - `box-flex: 1 /* for child element */`
- Grid Layout
 - `display: grid;`
 - <https://www.w3.org/TR/css3-layout/>
- Template Layout

Multiple Columns Text

- Breaks flowing text into multiple columns, based on the width of the container.
 - column-width: the width of each column
 - column-gap: the gap width between columns
 - column-count can be specified in lieu of column-width
 - column-count 優先於 column-width
 - column-rule: line between columns
- ps: 不建議對長度超過螢幕的文字使用多欄位，此時多欄更不利閱讀

多欄版型 - 使用 multiple columns

- e.g. [layout/layoutColumn.html](#)

```
div#container {  
    column-count: 3;  
    column-gap: 1.5em;  
}  
nav,div#content,div#sidebar {height:300px;}  
...  
<div id="container">  
    <nav>...</nav>  
    <div id="content">... </div>  
    <div id="sidebar">...</div>  
</div>
```

display 屬性

- display 設定元素顯示的方式

- Syntax

display : inline | block | inline-block | list-item | box | grid | none ...;

- <http://www.w3.org/TR/css3-box/#types>

- e.g.

```
li { display: inline; }
```

- e.g. [layout/display.html](#)

多欄版型 - 使用 display:box

- e.g. [layout/layoutBox.html](#)

- 重要樣式碼

```
div#container {  
    display: box;  
    box-orient: horizontal;  
}
```

...

```
<div id="container">  
    <nav>...</nav>  
    <div id="content">... </div>  
    <div id="sidebar">...</div>  
</div>
```

多欄版型 - 使用 display:grid

- e.g. [layout/layoutGrid.html](#)

- 重要樣式碼

```
div#container {display: grid;}
```

```
nav {grid-column:1; grid-row:1;}
```

```
div#content {grid-column:2; grid-row:1;}
```

```
div#sidebar {grid-column:3; grid-row:1;}
```

...

```
<div id="container">
```

```
    <nav>...</nav>
```

```
    <div id="content">... </div>
```

```
    <div id="sidebar">...</div>
```

```
</div>
```

Navigation Bar

- Vertical Navigation Bar
 - 可利用 list 建立 Navigation Bar
- Horizontal Navigation Bar
 - 將 以 inline element 方式顯示
 - e.g. <li style="display: inline-block">
 - 使用浮動
 - e.g. <li style="float: left">
 - 使用 CSS3 的 box-orient 設定

Horizontal Navigation Bar - inline list

- e.g. [layout/menuInline.html](#)

- 重要樣式碼

```
ul#menu { list-style-type: none; } /* 不顯示項目符號 */  
ul#menu li { display: inline-block; }
```

```
<ul id="menu">  
  <li><a href="aboutme.html">關於我</a></li>  
  <li><a href="hw1.html">homework 1</a></li>  
  ...  
</ul>
```

Horizontal Navigation Bar - floated list

- e.g. <layout/menuFloat.html>

- 重要樣式碼

```
ul#menu { list-style-type: none; } /* 不顯示項目符號 */  
ul#menu li { float: left; ... }
```

```
<ul id="menu">  
  <li><a href="aboutme.html">關於我</a></li>  
  <li><a href="hw1.html">homework 1</a></li>  
  ...  
</ul>
```

Horizontal Nav. Bar - box-oriented

- e.g. <layout/menuOrient.html>

- 重要樣式碼

```
ul#menu {  
    /* CSS3 */  
    display: box;  
    display: -moz-box; /* Firefox */  
    display: -webkit-box; /* Chrome, Safari, Opera */  
    box-orient: horizontal;  
    -moz-box-orient: horizontal;  
    -webkit-box-orient: horizontal;  
}  
ul#menu li { margin: 0 5px; }
```


元素置中

- 要讓**固定寬度**的元素置中顯示，作法：
 - 指定 `margin-left` 與 `margin-right` 屬性為 `auto`
 - 使用絕對定位
- e.g. [layout/center.html](#)
- 註: 指定元素的 `text-align` 屬性為 `center` 是讓網頁內容置中，不是元素本身置中

指定 margin-left 與 margin-right 屬性

- 指定 margin-left 與 margin-right 屬性為 auto

```
div#content {  
    width: 500px;  
    margin-left: auto;  
    margin-right: auto;  
}
```

```
<div id="content">content area</div>
```

使用絕對定位

- 使用絕對定位

```
div#content {  
    position: absolute;  
    left: 50%;  
    width: 500px;  
    margin-left: -200px;  
}
```

```
<div id="content">內容區</div>
```

Flexible Web Design

Fixed Width Layouts 固定寬度

- usually specify in pixel
- 固定寬度：不隨螢幕畫面大小改變版面寬度
 - 對 1024 x 768 螢幕，通常使用 950 pixels wide
 - 對 800 x 600 螢幕，通常使用 760 pixels wide，置中對齊，兩邊留下一點空白
- e.g. <http://www.htmlandcssbook.com/code-samples/>
 - chapter 15. e.g.
 - ```
.column1, .column2, .column3 {
 width: 300px; float: left; margin: 10px;
}
```

# Liquid Layouts 流動寬度

- 隨螢幕畫面大小改變版面寬度，通常是設 auto，由 browser 調
- 利用比率來設計。例如使用百分比或 em 值來設定寬度
  - em 值會隨字體大小改變，進而調整畫面寬度
- adaptive web design
  - using a *predefined set* of layout sizes based on device screen size
  - media queries
- responsive web design
  - fluid grids
  - flexible images
  - media queries

# Responsive versus Adaptive



source: <https://css-tricks.com/the-difference-between-responsive-and-adaptive-design/>

# Media Query

- 實作 Responsive Web Design Layout 主要有兩點技巧
  - Grid
    - e.g. Bootstrap 預先劃分為 12 欄，每欄 60 px，間距 20 px
  - Media Query
- e.g. 當螢幕寬度小於 480px 時應用此規則

```
@media screen and (max-width: 480px) {
 aside { width: 400px; }
}
```
- e.g. 螢幕大於 1024px 時(大螢幕) 應用此規則

```
@media screen and (min-width:1024px) {
 aside { ... }
}
```



# @media

- `<link rel="stylesheet" href="" type="text/css" media="..">`
- `<style media="screen, projection, print, aural">`
  - aural: 語音合成器輸出
- `@import url("") screen, print, ...;`
- `@import url(""); /* for all media */`
- `@media screen { ... }`  
`@media print { ... }`  
`@media screen, print { ... }`
- e.g.  
`@media print { body { font-size: 10pt; } }`

# media query for cellphone

■ e.g.

```
@media screen and (max-device-width: 320px) {..}
```

```
@media screen
```

```
and (max-device-width: 480px)
```

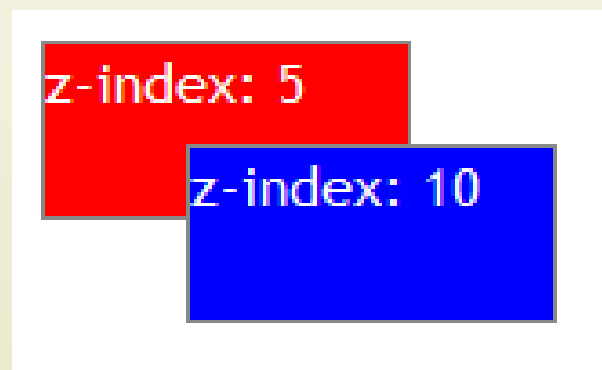
```
and (orientation: landscape)
```

```
{..}
```

# Misc.

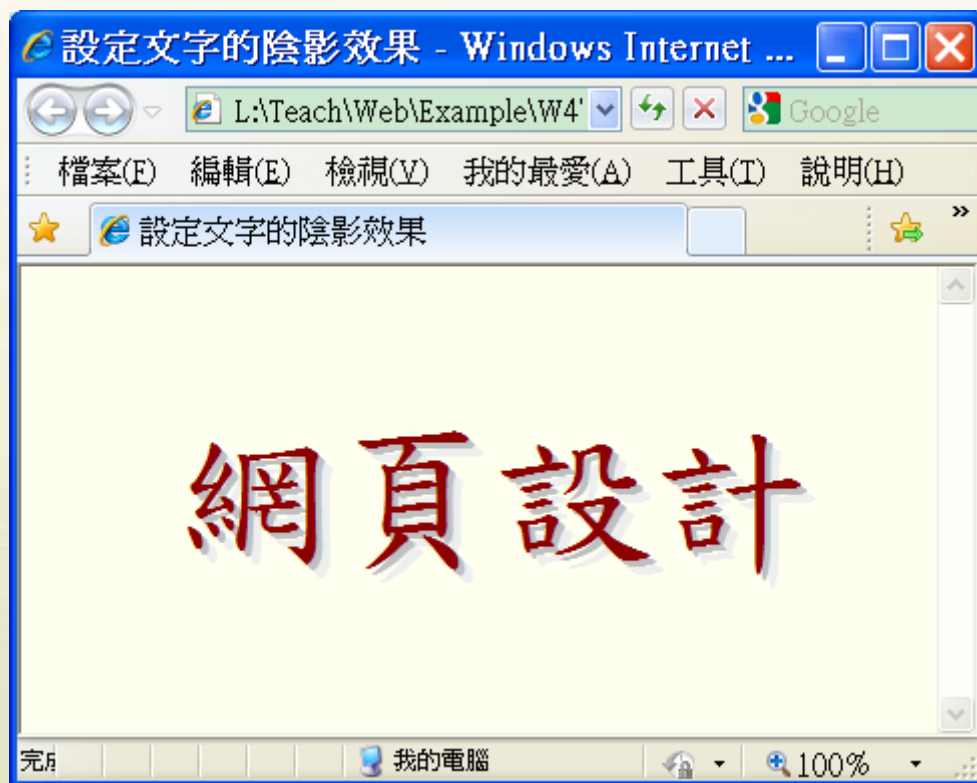
# z-index 屬性

- 設定元素的堆疊順序
- 語法： `z-index: <number>;`
- e.g. [float/z-index.html](http://float/z-index.html)  
`<p style="z-index: 5;"></p>`  
`<p style="z-index: 10;"></p>`  
`<p style="z-index: -1;"></p>`



# z-index 應用 - Text Shadow

- 利用 z-index 變化，產生字體陰影效果
- e.g. [float/z-index-text-shadow.html](http://float/z-index-text-shadow.html)



# Text Overflow

- If text overflows the block available space, the overflow property defines what happens.
- `overflow : visible | hidden | scroll | auto ;`
  - `visible`: (default) 可顯示於框外
  - `hidden`: 超出部分不顯示
  - `scroll`: 一律顯示 scroll bar
  - `auto`: browser 決定處理方式
    - e.g. 內容超出時顯示 scroll bar
- `text-overflow: clip | ellipsis | string | initial | inherit`
  - `"ellipsis"`: render an ellipsis ("...") to represent the clipped text
- e.g. <layout/overflow.html>

# visibility 屬性

- 顯示或隱藏元素
- 語法： `visibility : visible | hidden | collapse | inherit;`
  - `visible` : show the element
  - `hidden`: hide the element, but reserve the layout space
  - `collapse`: remove the rows, columns of table element (hide and not reserve the space). If used on elements other than rows, columns, 'collapse' has the same meaning as 'hidden'.
- e.g. [visibility.html](#), [visibility-table.html](#)  
`<span style="visibility: visible;">Text</span>`

# visibility:hidden versus display:none

- `display: none;`
  - 元素不但不顯示，也不佔空間
- `visibility: hidden;`
  - 元素不可見，但是畫面上會留下該元素的空間



# Reset Default Style

- Because of browser differences, it's a good idea to "reset" the default formatting for HTML tags. All you have to do is set up some basic styles at the beginning of your style definition. For example:
  - reset padding and margins of block-level elements
  - define default font sizes. ( 1em is good choice )
  - remove underlines from links.
    - In fact, most of the links in our site look more like buttons, or you use hover effects, then start off by removing the underlines. You can later selectively add underlines when you want them.

# Reset Default Style

- reset margin, padding  
html, body, div, h1, h2, h3, h4, h5, h6, p, ol, ul, dt, dl, dd,  
table, th, td, blockquote, form, fieldset, input  
{ padding: 0; margin: 0; }
- h1, h2, h3, h4, h5, h6 { font-size: 1em; } /\* font-size for  
headers \*/
- ol, ul { list-style: none; } /\* removes list-style from lists \*/
- a { text-decoration: none; } /\* removes underline from  
links \*/
- a img { border: none; } /\* removes border from img \*/

# Validation

- Markup Validation 檢查 HTML 正確性
  - [validator.w3.org](http://validator.w3.org)
- CSS Validation
  - [jigsaw.w3.org/css-validator/](http://jigsaw.w3.org/css-validator/)