Cascading Style Sheets (I)

CSS 樣式表

Getting Start

What is CSS

- CSS (Cascading Style Sheets)
- A style sheet is a set of stylistic CSS rules that tell a browser how the HTML document are presented.
- CSS is a means to separate the presentation from the HTML content by controlling the appearance of content
- demo: http://www.w3schools.com/css/demo_default.htm
- 特點
  - 豐富的樣式, 容易設定
  - 網頁文件簡潔
  - reusability 重覆使用
  - cascade 階層性

CSS Version

- http://www.w3.org/Style/CSS/
- CSS Level 2 (CSS 2)
  - 1998-May-12 W3C Recommendation
  - 2008-April revised 11
- CSS Level 2 Revision 1 (CSS 2.1)
  - 2009-April-23 W3C Candidate Recommendation
  - http://www.w3.org/TR/CSS21/
- CSS Level 3
  - builds on CSS Level 2 module by module, using the CSS2.1 specification as its core. adds functionality and/or replaces part of the CSS2.1 specification.

Where you define the CSS rules

- There are three ways to add CSS rules to web pages
  - inline
    - use the *style* attribute in the HTML tag
    - e.g. <p style="font-size: 24pt;">
  - embedded (internal style)
    - <style…>…</style>
  - linked from a separate CSS style sheet (external style)
    - e.g. <link rel="stylesheet" type="text/css" href="file.css" />
    - e.g. <style…>@import url("file.css")</style>

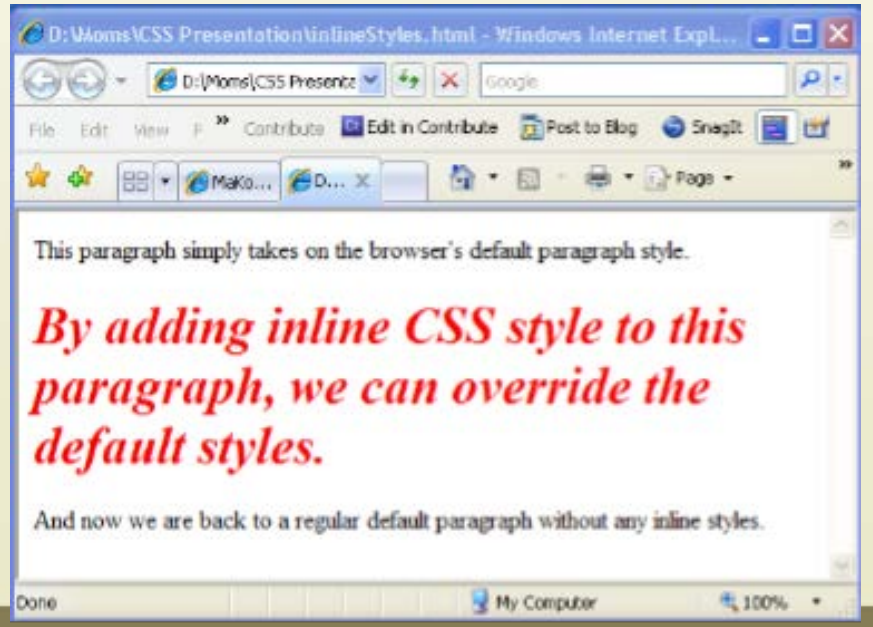
Simple Example

- <html><head> <title>Example</title> </head>
  <body style="background-color: #FF0000;">
  This is a red page
  </body>
  </html>
- <html>
  <head> <title>Example</title>
  <style type="text/css">
  body {background-color: #FF0000;}
  </style>
  </head>
  <body> This is a red page </body>
  </html>

Inline Style

- use the style attribute in the HTML tag
- e.g. c0-inline.html

```
<p>This paragraph simply takes on the browser's default paragraph
   style</p>
<p style="font-size: 24pt; font-weight:bold; font-style:italic;
   color:red;">By adding inline CSS style to this paragraph, we can
   override the default styles.</p>
<p>And now we are back to a regular default paragraph without
   any inline styles.</p>
```



Embedded Style

- embed the styles in the head of the HTML document
- The scope of embedded styles is limited to the page that contains the styles
- e.g. c0-embed.html
  定義 <h2> 標籤樣式為 36px 的藍色粗體字
  - <style type="text/css">
    h2 { font-size: 36pt; font-weight:bold; color: blue; }
    </style>

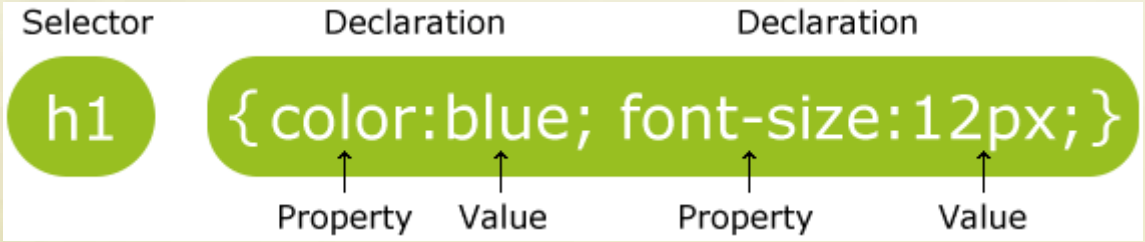
Linked Style

- Place styles in a separate document that links to multiple pages so that the styles can be shared.
- e.g. link.html
  引用儲存於 link.css 檔中的 CSS 定義
  - <link rel="stylesheet" type="text/css" href="link.css" />
- e.g.
  <style type="text/css">
  @import url('link.css'); // @import 必須出現在其他 CSS rule 之前
  ....
  </style>

CSS Rules

CSS Rule 1/2

- A CSS rule is made up of two parts:
  - selector: states which tag the rule selects
  - declaration: states what happens when the rule is applied
- The declaration itself is made up of two elements:
  - property: states what is to be affected
  - value: states what the property is set to



source:www.w3schools.com

CSS Rule 2/2

- multiple declarations can be contained within a rule.
- each declaration ends with a semicolon.

- e.g.

  p { color:red; font-size:12px; line-height:15px;}

- comment
  - /* …. */

Selector Types

1. Type/Element/Tag Selector
2. Grouped Selector
3. Universal Selector
4. Class Selector
5. ID Selector
6. Attribute Selector
7. Descendant Selector
8. Child Selector
9. Adjacent Sibling Selector
10. General Sibling Selector

T1: Type/Element/Tag Selector

- be applied to document language element type (such as HTML tag)
- e.g. The following rule matches all <h1> elements
  - h1 { font-family: "標楷體" }

T2: Grouped Selector

- selectors can be grouped
- e.g.

```
h1 { font-family: "標楷體"; }
h2 { font-family: "標楷體"; }
h3 { font-family: "標楷體"; }
h4 { font-family: "標楷體"; }
h5 { font-family: "標楷體"; }
h6 { font-family: "標楷體"; }
```

equivalent to

```
h1, h2, h3, h4, h5, h6 { font-family: "標楷體"; }
```

T3: Universal Selector

- The universal selector, written "*", matches any single element in the document.
- e.g.

  * { color: red; } /* the color of all text is red. */
- If the universal selector is not the only component of a selector, the "*" may be omitted

Text Style

Text Font Style

- font 設定字型的全部屬性
  - [[<'font-style'> || <'font-variant'> || <'font-weight'>]? <'font-size'> [/ <'line-height'>]? <'font-family'>] | caption | icon | menu | message-box | small-caption | status-bar | inherit
  - font-stretch 字型寬窄 (deprecated)
  - font-size-adjust 字型高度(deprecated)

font-family 字型

- font-family: [[<family-name> | <generic-family>] [, <family-name>| <generic-family>]*] | inherit
  - e.g. body { font-family: Gill, Helvetica, sans-serif }
  - e.g. h1 {font-family: "Lucida Calligraphy", Verdana, serif;}
  - e.g. p {font-family: "Monotype Corsiva", Arial;}
  - e.g. h2, h3 {font-family: sans-serif;}

Generic Font 通用字型

- 為了保證瀏覽裝置能夠正確地顯示您的字型，您應該在 font-family 屬性值的最後面加上一個通用字型。通用字型不需以引號包含起來。
  - serif 附有襯線的比例字型
    - e.g. Times, Garamond, New Century, Schoolbook
  - sans-serif 沒有襯線的比例字型
    - e.g. Helvetica, Geneva, Verdana, Arial, Univers
  - monospace 非比例字型(字元寬度一致)
    - `Courier,` Courier New, Andale Mono
  - cursive 模仿手寫的字型 (草寫字)
    - e.g. Chancery, Author, Comic Sans
  - fantasy 其他 (創意字)
    - e.g. Western, Woodblock, Klingon

中英文不同字體

英文字體寫在中文字體前面

e.g. h1 { font-family: "Times New Roman", "標楷體"; }

Google Free Fonts - Noto

- **noto**: **no** more **to**fu(豆腐)
- 支援所有 Unicode 編碼字元
- https://www.google.com/get/noto/
  - Noto Sans CJK (思源黑體 Pan-CJK)
    - comprehensively covers Simplified Chinese, Traditional Chinese, Japanese, and Korean in a unified font family.
      - 正體中文 Noto Sans CJK TC
    - 含七種粗細
    - https://www.google.com/get/noto/#sans-hant

font-size

- 您使用font-size屬性來指定字型的大小，例如：
  - h1 {font-size: x-small;}
  - h2, h3 {font-size: larger;}
  - p {font-size: 3.5em;}
  - p {font-size: 220%;}

absolute font size 絕對字型大小

- [xx-small | x-small | small | medium | large | x-large | xx-large]
- default: medium
- 這些字型大小沒有精確的定義，而是取相對於彼此的比例而定。
  - CSS 1 使用1.5的比例值
  - CSS 2.1使用1.0~1.2的比例值。
  - 如果medium是16像素的話，large大約會是19像素(16像素的1.2倍)，small大約會是13像素。請參考abs-font-size.html。
- e.g. text/font-size-abs.html

relative font size 相對字型大小

- relative to the font size of the parent element
- [larger | smaller]
- 使用與絕對字型大小相同的比例值
- e.g. 父物件的字型大小是 medium
  - 子物件設定 larger，子物件的大小就是 large
  - 子物件設定 smaller，子物件的大小就是 small
- e.g. 父物件的字型大小是 large
  - 子物件設定 larger，子物件的大小就是 x-large
  - 子物件設定 smaller，子物件的大小就是 medium
- e.g. text/font-size-rel.html

explicit size - 絕對單位

直接明確指定字型大小文字

可用絕對長度單位

- pt: point，傳統上的印刷用單位，1點等於1/72英吋(72ppi)，但目前螢幕少則96ppi，多則甚至達到120ppi
  - e.g. p { font-size: 12pt }
- pc: pica, 1pc = 12pt
- cm (centimeters)
- mm (millimeters)
  - e.g. p { font-size: 15mm }
- in (inches)
- e.g. text/font-size-explicit.html

explicit size - 相對單位

可用相對長度單位

- px: pixel, CSS 規格文件建議像素大小為 96ppi
  - e.g. p { font-size: 16px }
- em
  - 字型中大寫字母M的高度(也等於寬度)
  - 1em = font-size 值
  - e.g. p { font-size: 1.5em }
- ex
  - 字型裏小寫字母 x 的高度，不同字型，高度會不同
  - 1ex大約是1em的一半

explicit size - 百分比例

- %: 相對於父物件的字型百分比例
  - e.g.

```
<style type="text/css">
  body {font-size: 32px;}
  ul{ font-size: 75% }
</style>
```

- e.g. text/font-size-percent.html

font-weight 字型粗細

- font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit
  - 'normal' is synonymous with '400', and 'bold' is synonymous with '700'
  - The 'bolder' and 'lighter' values select font weights that are relative to the weight inherited from the parent
  - e.g. text/font-weight.html

```
<span style="font-weight: normal">normal</span>
<span style="font-weight: bolder">bolder</span>
<span style="font-weight: lighter">lighter</span>
<span style="font-weight: 100">100</span>
```

font-style 字型格式

- font-style: normal | italic | oblique | inherit
  - italic (真)斜體字
    - Italic designs are not just the slanted version of the regular (roman) style
  - oblique (偽)斜體字:
    - a roman type that has been skewed a certain number of degrees (8-12 degrees)
    - it uses the same glyphs as roman type
    - usually change on the fly
  - e.g. text/font-style.html

```
<span style="font-style: normal">normal</span>
<span style="font-style: italic">italic<span>
<span style="font-style: oblique">oblique</span>
```

font Shorthand 字型速寫法

- 使用 font 屬性來指定字型的全部屬性
- e.g. text/font.html

```
h1 {
    font-family: Helvetica, sans-serif;
    font-size: 12px;
    font-weight: bold;
    font-style: italic;
}
```

may be rewritten with a single shorthand property

```
h1 { font: bold  12px  italic  Helvetica, sans-serif; }
```

擷取作業系統字型

- caption 用在有標題的控制項上(例如按鈕或是選單)
  - e.g. button {font: caption; font-size: 1em;}
    讓網頁上的按鈕看起來跟作業系統的按鈕一模一樣
- icon 用於文字標籤的小圖示
- menu 用於選單上
  - e.g. text/font.html
- message-box 用於對話視窗上
- small-caption 用於呈現小寫
- status-bar 用於視窗的狀態列上

行高－ line-height 屬性

- 行高是從一行中最高的行內方框的頂端，到最低的行內方框的底端。
- line-height 屬性設定如下：

  normal | <number> | <length> | <percentage> | inherit
- e.g. text/line-height.html

  body {line-height: 14px; }

- 可在設定 font 時, 同時設定 font-size 與 line-height
  - e.g. font: 12/14;

水平對齊－ text-align 屬性

- 整個段落首行的縮排，若設定負數，則為凸排
- 語法(CSS2.1)：text-align: left|center|right|justify|inherit
  - e.g. p {text-align : justify} 分散對齊
- e.g. text/text-align.html

垂直對齊

- 語法：vertical-align: baseline|sub|super|top|text-top|middle|bottom|text-bottom|<lenght>|<percentage>|inherit
  - e.g. <span stype="vertical-align: super;"} 上標
- http://www.w3schools.com/cssref/pr_pos_vertical-align.asp

文字縮排

- 整個段落首行的縮排，若設定負數，則為凸排
- 語法：text-indent: <length>|<percentage>|inherit
  - e.g. p {text-indent: 2em}
  - e.g. p {text-indent: 10%}
- e.g. text/text-indent.html

文字裝飾

- text-decoration-line: none | underline | overline | line-through | inital | inherit
- text-decoration-color
- text-decoration-style: solid|double|dotted|dashed|wavy|initial|inherit;
- shorthand: text-decoration
- e.g. text/text-decoration.html

- e.g. 超連結通常會加上底線，如果希望移除，可設 a {text-decoration: none}

foreground color

- 設定前景顏色 (定義文字的顏色)
- 語法：color: <color> | inherit
- e.g. text/font-color.html

```
body { color: green; }
```

color 表示法

- 色彩名稱
  - CSS 3 定義了 16+1 種 Basic color keywords
    - aqua, black, blue, fuchsia, gray/grey, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow
    - http://www.w3.org/TR/css3-color/#html4
  - 及 130 Extended color keywords
    - http://www.w3.org/TR/css3-color/#svg-color

color 表示法

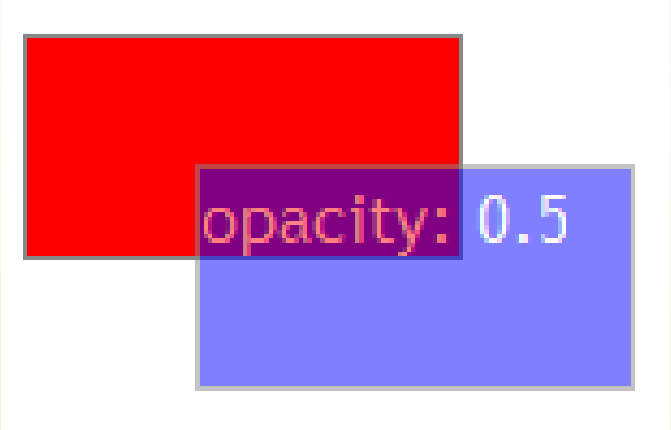
- RGB 色
  - 16進位表示法：
    - #RrGgBb:
    - #RGB: 會自動複製成 #RRGGBB
- rgb(r, g, b)函數
  - 數值：e.g. rgb(180,100,100)
  - 百分比：e.g. rgb(80%,50%,50%)
- rgba(r,g,b,alpha) 函數
  - alpha: 透明度, 0.0 (全透明) ~ 1.0 (不透明)
  - e.g. rgba(255,0,0,0.5)

color 表示法

- hsl(h, s, l)函數
  - h: Hue (色調/色相) 0~360
  - s: Saturation (飽和度/濃度) 0% ~ 100%
  - l: lightness (亮度) 0% ~ 100%
  - e.g. hsl(125, 38%, 82%)
  - http://www.w3.org/TR/css3-color/#hsl-examples
- hsla(h, s, l, alpha)

opacity 透明度

- CSS3 新增 opacity 來指定透明度
  opacity: 0.0 (全透明) ~ 1.0 (不透明);
- e.g. text/opacity.html



- IE 8 以前則需使用
  filter: alpha(opacity=0~100);
  來定義。

文字陰影

- Syntax: text-shadow: *h-offset v-offset blur color*;
- e.g.

  p {text-shadow: 0.1em 2px 4px green;}
- e.g. text/text-shadow.html
  - text-shadow: 0.1em 2px 4px green;

- reference

  http://www.w3.org/Style/Examples/007/text-shadow

Background-color 背景顏色

- background-color: <color> | transparent | inherit
- e.g. background/background-color.html

  background-color: aqua;

  background-color: #C0C0C0;

  background-color: #FF0;

  background-color: rgb(0,255,255);

  background-color: rgba(75%,75%,75%);

  background-color: rgba(255,0,0,50%);

background-image 背景圖片

- background-image 背景圖片
  - 語法：<uri> | none | inherit
  - e.g. background/background-image.html

    body {background-image: url(bg.gif) }
  - url() 內的位址，是與 CSS 檔案相對的路徑，如果開頭加上 / 則為整個網站的相對路徑。
-

background-repeat

- background-repeat 背景重覆方式
  - 語法：repeat | repeat-x | repeat-y | no-repeat | inherit;
- background-position 背景位置
  - 語法：[left | center | right] or
    [<length> | <percentage> | left | center | right]
    [<length> | <percentage> | top | center | botom] | inherit
- background-attachment 背景是否會跟著捲動
  - 語法： scroll | fixed
  - e.g. background/background-attachment.html

background-size

- Specify the size of a background image
- background-size: contain | cover | <percentage> | <x, y>
  - contain:等比例縮小至可放入 content area
  - cover: 等比例放大至可填滿 content area
- e.g. background-size: 100px 60px;

background-origin, background-clip

- background-origin: border-box | padding-box | content-box
  - 背景顯示的範圍
- background-clip: border-box | padding-box | content-box
  - 背景圖示的裁切
- e.g. background/background-origin.html

background shorthand 背景速寫法

- background: <background-color> <background-image> <background-repeat> <background-attachment> <background-position [/background-size]> <background-clip> <background-origin>
- e.g.

body{background: green url(bg.gif) repeat-y fixed top left;}

相當於

```
body {background-color: green;
        background-image: url(bg.gif);
        background-repeat: repeat-y;
        background-attachment: fixed;
        background-position: top left;
}
```

Gradients Background 漸層色

- (CSS3) background:
  linear-gradient(*direction*, *color-stop1*, *color-stop2*, ...);
  repeating-linear-gradient(…);
  radial-gradient(*shape size* at *position*, *start-color*, ..., *last-color*);
  repeating-radial-gradient(…);
- e.g.
  background: -webkit-linear-gradient(left, red, blue);
  background: -moz-linear-gradient(left, red, blue);
  background: linear-gradient(left, red, blue);

- http://www.w3schools.com/css/css3_gradients.asp

Selector Types

1. Type/Element/Tag Selector
2. Grouped Selector
3. Universal Selector
4. Class Selector
5. ID Selector
6. Attribute Selector
7. Descendant Selector
8. Child Selector
9. Adjacent Sibling Selector
10. General Sibling Selector

T4: Class Selector 1/3

- The Class Selector be applied to the element that has an "class" attribute with a value of class-name
- Syntax: element.class-name {rule-declarations}
- e.g. selector-class.html

```
p.right {text-align:right}
p.center {text-align:center}
…
<p class="right"> right-aligned paragraph</p>
<p class="center"> center-aligned paragraph</p>
```

T4: Class Selector 2/3

- omit the element name, applied to all elements
  - *.class-name or .class-name
  - e.g. selector-class.html

```
.center {text-align:center}
…
<h1 class="center">center-aligned</h1>
<p class="center">center-aligned too</p>
<div class="center">center text</div>
```

T4: Class Selector 3/3 – Multiple Class

- To match a subset of "class" values
- e.g. selector-class-multiple.html
  - the following rule matches any P element whose "class" attribute has been assigned a list of space-separated values that includes "pastoral" and "marine":
    p.marine.pastoral { color: green }
    <p class="pastoral blue aqua marine“>
  - This rule matches when class="pastoral blue aqua marine" but does not match for class="pastoral blue"

T5: ID Selector

- The ID Selector will be applied to the element that has an “id” attribute with a value of id
- Syntax: element#identifier {rule-declarations}
- e.g. selector-id.html

```
#green {color:green}
…
<span id="green">green text</span>
```

Class Selector versus ID Selector

- No two element can have the same id.
  - In reality, browser 通常不會檢查 ☹
  - 使用多次 id 可能會造成一些困擾，不建議如此用。例如 getElememtByID() 函數就必須依賴 unique ID
- id 不像 class 屬性可以設 multiple class
  - e.g. 不會有

T6: Attribute Selector 1/2

- Syntax 1: elem[attr] {…}
  - matches all elem elements that specify the attr attribute
- Syntax 2: elem[attr *op* "value"] {…}
  - e.g. input[type="text"] {font-family: sans-serif;}
  - where *op* could be =, ^=, $=, *=, ~=, |=,
    - = : 完全相同
    - ^= : begin with "value"
    - $= : end with "value"
    - *= : include substring "value"
    - ~= : ⊃ include item "value"
    - |= : value either is exactly "value" or beginning with "value" immediately followed by "-"

T6: Attribute Selector 2/2

- e.g. selector-attribute.html
  - attach a pdf icon to the right of any hyperlink who's URL ended in '.pdf'
  - a[href $='.pdf'] { padding-right: 24px;
    background: url(icon_pdf.gif) no-repeat center right; }

- Multiple attribute selectors can be used to refer to several attributes of an element
  - e.g. elem[attr1][attr2] {...}

Selector Types

1. Type/Element/Tag Selector
2. Grouped Selector
3. Universal Selector
4. Class Selector
5. ID Selector
6. Attribute Selector

Combinators (Contextual Selector)

7. Descendant Selector
8. Child Selector
9. Adjacent Sibling Selector
10. General Sibling Selector

T7: Descendant Selector

- A descendant selector is made up of two or more selectors separated by white space. A descendant selector of the form "A B" matches when an element B is an arbitrary descendant of some ancestor element A.
- e.g. selector-descendant.html
  this example will match the in the following content, “very” will be blue:
  h1 em { color: blue }
  …
  <h1>This headline is veryimportant</h1>

T7: Descendant Selector Example

- e.g.

```
ul li { color: blue; }
<ul>
  <li>item 1-1</li>
  <li>item 1-2
  <ol>
     <li>item 2</li>
  </ol>
  </li>
</ul>
```

  - item 1-1, item 1-2, item 2 的顏色都是blue, 因為它們都在 ul 之下。

T8: Child Selector

- A child selector matches when an element is the child of some element
- e.g. selector-child.html 若前例改為
  ul > li { color: blue; }
  - 則只有 item 1-1, item 1-2 的顏色會被改成 blue。item 2 是在第二層，所以沒被選擇到。

- PS: IE 系列，IE 7 才開始支援。

T9: Adjacent Sibling Selector

- Syntax: E1 + E2
  - where E2 is the subject of the selector.
- The selector matches if E1 and E2 share the same parent in the document tree and E1 immediately precedes E2
- e.g. selector-sibling-adjacent.html
  only paragraph 1 is red.

```
h1 + p { color: red; }
<h1>subject</h1>
<p>Paragraph 1</p>
<p>Paragraph 2</p>
```

T10: General Sibling Selector

- new selector defined in **CSS 3**
- e.g. selector-sibling-general.html

  上個例子中若改用 General Sibling Selector

  ```
  h1 ~ p { color: red; }
  <h1>subject</h1>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  ```

  則 Paragraph 1 及 Paragraph 2 都會是紅色

Pseudo-Classes, Pseudo-Elements

- Style is normally attached to an element in the document tree. It is insufficient for some scenarios.
- CSS 2.1 introduces the concepts of pseudo-classes and pseudo-elements to permit formatting based on information that lies outside the document tree.

Pseudo-Classes, Pseudo-Elements

- Pseudo-Classes classify elements on characteristics other than their name, attributes or content.
  - Structural pseudo-classes
    :firstchild :lastchild :nth-child() :nth-last-child() …
  - link pseudo classes :link :visited
  - dynamic pseudo classes :hover :active :focus
  - language pseduo class :lang
- Pseudo-Elements
  - ::first-line ::first-letter ::before ::after
    - (CSS 2.1 用 : CSS3 改用 ::)
- http://www.w3.org/TR/selectors/

Pseudo Classes :first-child

- :first-child pseudo-classes
  - matches an element that is the first child element
- link pseudo-classes :link, :visited
  - applies for links that have not yet or have been visited
  - The two states are mutually exclusive.
- e.g. selector-pseudoClass.html

- http://www.w3.org/TR/css3-selectors/

Pseudo Classes :hover :active

- dynamic pseudo-classes
  - :hover
    - while the user designates an element (with some pointing device), but does not activate it. For example, the cursor (mouse pointer) hovers over a box generated by the element
  - :active
    - applies while an element is being activated by the user. For example, between the times the user presses the mouse button and releases it.
  - :focus
    - applies while an element has the focus (accepts keyboard events or other forms of text input).
  - An element may match several pseudo-classes at the same time.

Pseudo Classes :hover :active

- [Note] you must specify the links pseudo-classes in a particular order :link, :visited, :hover, and :active.
  - the a:hover must be placed after the a:link and a:visited rules, since otherwise the cascading rules will hide the 'color' property of the a:hover rule.
  - Similarly, because a:active is placed after a:hover, the active color will apply when the user both activates and hovers over the <a> element
  - combining dynamic pseudo-classes
    - e.g. a:focus:hover { background: white ;}

Pseudo Classes :target

- :target
  style the current active target element(目的元素), for URLs with bookmark
- e.g. selector-pseudoClass-target.html

Pseudo Classes :lang

- language pseudo-class :lang
  - If the HTML document specifies the language attribute (e.g. <meta http-equiv="Content-Language" content="zh">) It is possible to write selectors in CSS that match an element based on its language
  - e.g. selector-pseudoClass-lang.html

```
:lang(fr) { color: green; }

我是法國人<q  lang="fr">Je suis Français.</q>
```

Pseudo Elements ::first-line

- ::first-line
  - applies special styles to the contents of the first formatted line of a paragraph.
  - e.g. p::first-line { text-transform: uppercase; }
    - the rule means "change the letters of the first line of every paragraph to uppercase“
  - e.g. selector-pseudoElement.html
- ::first-letter
  - select the first letter of the first line of a block

Pseudo Elements ::before ::after

- ::before, ::after
  - insert generated content before or after an element's content
  - e.g.

    ```
    p::before {content: “{開始嚕}”;}
    p::after {content: url(smile.gif);} /* 後面插入圖片 */
    a[href$=".pdf"]::after {content: url("pdf.ico");}
    ```
  - e.g. `div::after { content: 'the end'; }`
  - e.g.在 a 元素後面顯示網址

    ```
    a[href]::after {content: " (" attr(href) ")";}
    ```
- selector 中只能存在一個 pseudo element，且必須是最後一個元素

Cascading Order

- What style will be used when there is more than one style specified for an HTML element?
- priority: inline > embedded > linked > browser default
  - which means that inline style will override a style defined inside the <style> tag, or in an external style sheet
  - 註: browser 大都實作為依 code 的位置，後覆蓋前
- e.g.
  - selector/cascading.html

Specificity of Selector 1/2

- 明確性: 多個 selector 作用於同一元素時，Specificity 值愈大表示愈明確，愈優先應用
- Specificity 值由 (d, c, b, a) 四值組成，計算方式:
  - a: number of element/pseudo-element
  - b: number of attribute/class/pseudo-class
  - c: number of ID attribute
  - d: declared by "style=", 1 or 0
- e.g.
  - type selector
    - p {color: blue;} /* (0, 0, 0, 1) */
    - p em {color: red;} /* (0, 0, 0, 2) */
    - p::first-line {color: red;} /* (0, 0, 0, 2) */

Specificity of Selector 2/2

- class/attribute selector
  - .note {…} /* (0, 0, 1, 0) */
  - p.note {color: orange;} /* (0, 0, 1, 1) */
- Id selector
  - #aside {color: lightgreen;} /* (0, 1, 0, 0) */
  - div#aside p {color: gray;} /* (0, 1, 0, 2) */
- inline style
  - style="color:green;" /* (1, 0, 0, 0) */
- 全域 selector
  - * {color: black;} /* (0, 0, 0, 0) */
- Quiz:
  ul ol+li {}, h1 + *[rel=up]{}, ul ol li.red {}, li.red.level {}
- e.g. selector/specificity.html

CSS Box Model

CSS 的方框概念

Box Model

- CSS Box Model concept:
  "Each HTML element is in a box or is a box"
- Each box has a *content area* (e.g., text, an image, etc.) and optional surrounding *padding*, *border*, and *margin* areas;

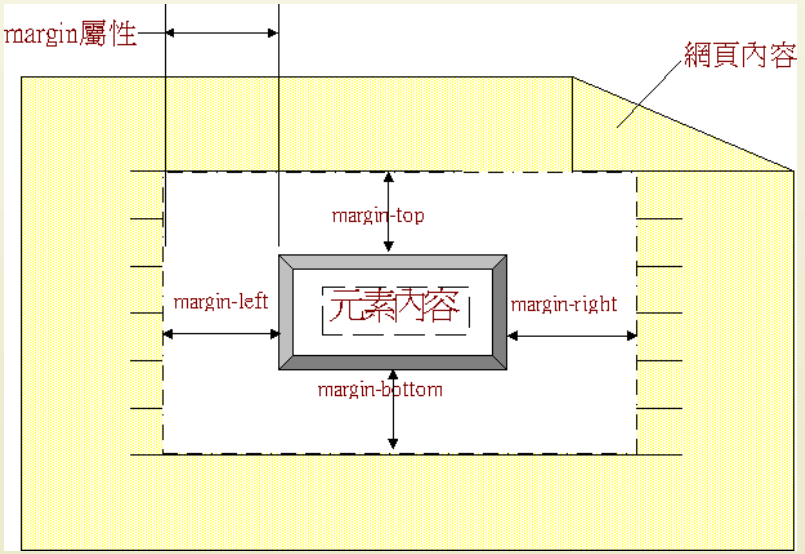
Inline Box versus Block Box

- inline box
  - generated by inline-level elements, e.g. , <b>, , …
- block box
  - generated by block-level elements, e.g. , , …
- When an inline box contains a block box, the inline box are broken around the block. The line boxes before the break and after the break are enclosed in anonymous block boxes, and the block box becomes a sibling of those anonymous block boxes.
  - e.g. box/box-inline-block.html

    text before the P. content of P. *text after the P.*
- Reference
  - http://www.w3.org/TR/CSS21/visuren.html#visual-model-intro

邊界的設定－margin 屬性

- margin　設定元素邊界與其他元素的間距
  - margin-top　設定元素的上間距
  - margin-right　設定元素的右間距
  - margin-bottom　設定元素的下間距
  - margin-left　設定元素的左間距
  - 四邊界的設定語法均相同
    <length>|<percentage>|auto
- 直接使用 margin
  一次設定四邊間距，依序為
  上、右、下、左間距
- e.g. box/box-margin.html



邊界設定－center alignment

- 如果您要讓元素置中顯示，您可以將 margin-left 與 margin-right 的值都設定為 auto，瀏覽器會自動幫您調整位置。
- e.g. box/box-margin-auto.html

```
h1 {
    margin-left: auto;
    margin-right: auto;
    width: 140px;
}
```

邊界設定－margin collapsing

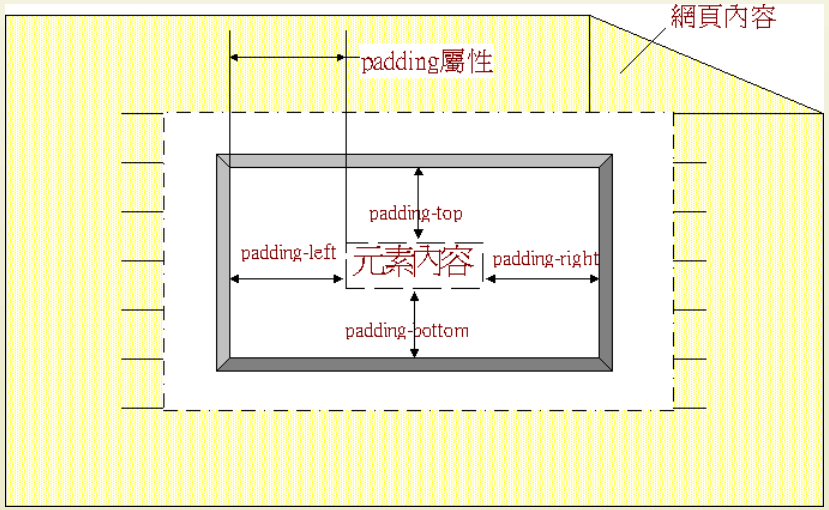
- Two or more adjoining vertical margins of block boxes collapse in the normal flow.
- so that the bottom margin of one element overlaps with the top margin of the element below it.
- The resulting margin width is the maximum of the adjoining margin widths.
- e.g. box/box-margin-collapse.html

- margins of inline-block elements do not collapse.

- http://www.w3.org/TR/CSS21/box.html#margin-properties

邊界的設定－padding 屬性

- padding 設定元素的上，右，下，左內距
- padding-top 設定元素的上內距
- padding-right 設定元素的右內距
- padding-bottom 設定元素的下內距
- padding-left 設定元素的左內距
- e.g. box/box-padding.html



框線的寬－ border-width 屬性

- border-width　設定元素的上，右，下，左邊框厚度
  - border-width屬性可以設定的值如下：

    border-width: thin | medium | thick | <length>
- border-width 屬性，共有四種設定方法：
  - 設定一個值：四框線厚度均使用同一個設定值。
  - 設定二個值：上與下框線厚度套用第一個值，右與左框線厚度套用第二個值。
  - 設定三個值：上框線厚度套用第一個值，右與左框線厚度套用第二個值，下框線厚度套用第三個值。
  - 設定四個值：四框線厚度的套用順序，依序為上、右、下、左。

框線的寬－ border-width 屬性

- 若您想設定元素中某一邊的框線厚度時，可利用下列四種框線屬性來設定：
  - border-top-width　設定元素的的上邊框厚度
  - border-right-width　設定元素的的右邊框厚度
  - border-bottom-width　設定元素的的下邊框厚度
  - border-left-width　設定元素的的左邊框厚度
- e.g. div { border-top-width: thin;　border-right-width: medium; border-bottom-width: thick; border-left-width: 12px; }

框線的顏色－ border-color 屬性

- border-color 屬性用於控制元素框線的顏色，在設定時跟 border-width 屬性一樣，有四種設定方法：
  - 設定一個值：四框線顏色均使用同一個設定值。
  - 設定二個值：上與下框線顏色套用第一個值，右與左框線套用第二個值。
  - 設定三個值：上框線顏色套用第一個值，右與左框線顏色套用第二個值，下框線顏色套用第三個值。
  - 設定四個值：四框線顏色的套用順序，依序為上、右、下、左。

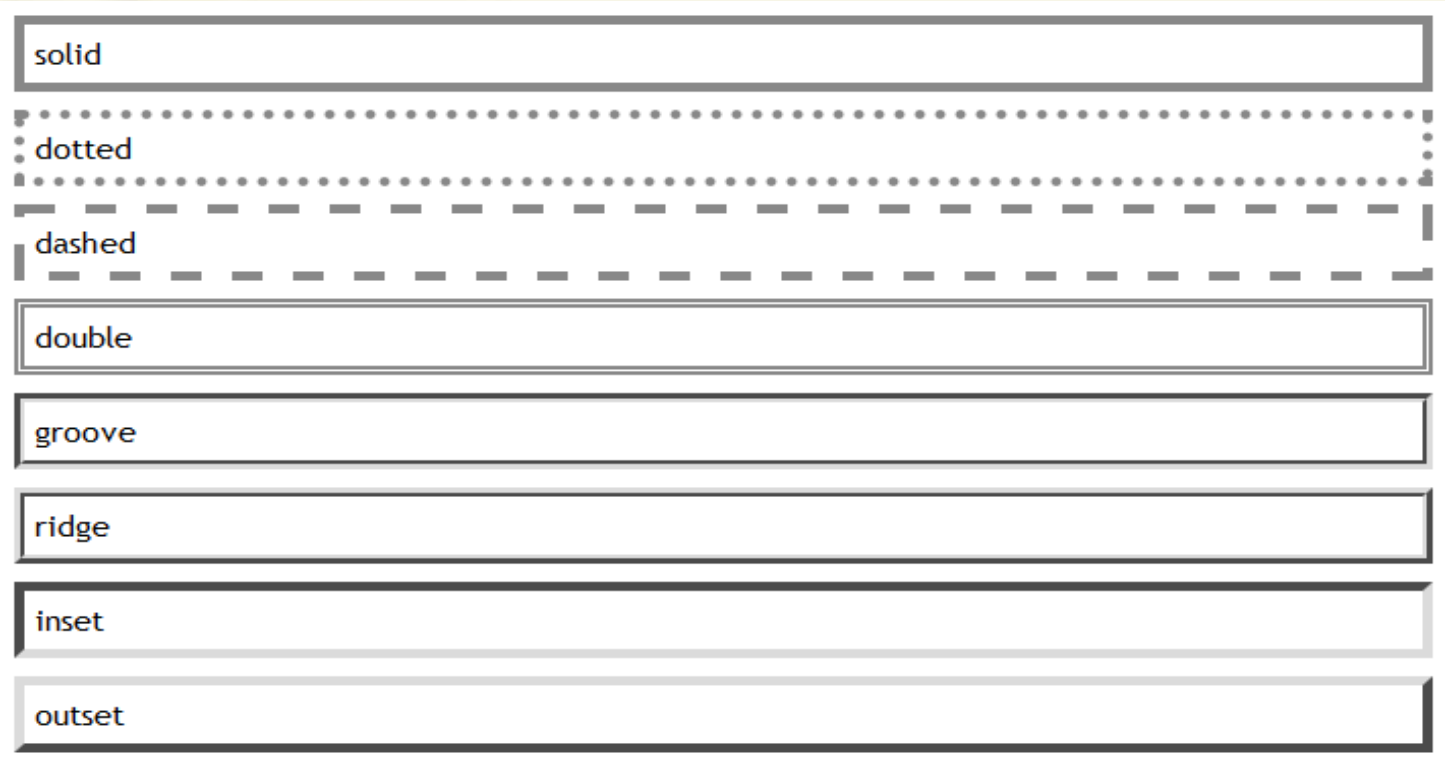
框線的顏色－ border-color 屬性

- 若您想設定某一邊的框線顏色時，可利用下列四種屬性來設定：
  - border-top-color 設定元素的的上邊框顏色
  - border-right-color 設定元素的的右邊框顏色
  - border-bottom-color 設定元素的的下邊框顏色
  - border-left-color 設定元素的的左邊框顏色
- e.g. div {
  border-top-color: red; border-right-color: #00FF00;
  border-bottom-color: #00F; border-left-color:
  rgb(128,0,128);
  }

框線的樣式－ border-style 屬性

- border-style框線樣式屬性主要是設定元素框線的樣式語法設定值如下：

  border-style: none | dotted(小點虛線) | dashed(大點虛線) | solid(實線) | double(雙直線) | groove(3D凹線) | ridge(3D凸線) | inset(3D框入線) | outset(3D隆起線)



框線的樣式－ border-style 屬性

- 若您僅想設定某一邊的框線樣式時，可利用下列四種框線屬性來設定：
  - border-top-style　設定元素的的上邊框格式
  - border-right-style　設定元素的的右邊框格式
  - border-bottom-style　設定元素的的下邊框格式
  - border-left-style　設定元素的的左邊框格式
- e.g. div {
  border-top-style: solid;
  border-right-style: dashed;
  border-bottom-style: dotted;
  border-left-style: double;
  }

快速設定－border 屬性

- 語法如下：

  border: <border-width> || <border-style> || <color>

- 屬性清單
  - border 設定上,右,下,左邊框的格式，厚度，與顏色
  - border-top 設定上邊框的格式，厚度，與顏色
  - border-right 設定右邊框的格式，厚度，與顏色
  - border-bottom 設定下邊框的格式，厚度，與顏色
  - border-left 設定左邊框的格式，厚度，與顏色

快速設定－border 屬性

- e.g. div {
  border-top: solid thin red;
  border-right: medium dashed #00FF00;
  border-bottom: rgb(128,0,128) dotted;
  border-left: double;
  }
- If the border is set to 0 it effectively disappears and the border edge is the same as the padding edge.

Rounded Corner 圓角的方框

- 瀏覽器顯示方框的方式，預設上是使用直角
- CSS3 標準下，設定 border-radius 屬性
  e.g. border-radius: 32px;
- 語法 border-radius: 左上 右上 右下 左下 / 左上 右上 右下 左下 (前半水平半徑, 後半垂直半徑)
- 如果不支援 CSS3
  - Gecko 系列的瀏覽器 (e.g. Firefox, Flock)
    自定的屬性 -moz-border-radius: 32px;
  - WebKit 系列 (e.g. Chrome,Safari)
    -webkit-border-radius: 32px;
  - 為了相容性，可三個同時設
- e.g. box/box-border-radius.html

綜合練習 - Button Style

- e.g.

```
.btn {
        padding: 0 0.5em;
        font: bold 10px verdana, sans-serif;
        color: #FFF;  background: #F60;
        text-decoration: none;
        border: 1px solid;
        border-color: #FC9 #630 #330 #F96;
        border-radius: 16px;
        font-size: 2em;
}
<a href="" class="btn">Button</a>
```

border-image

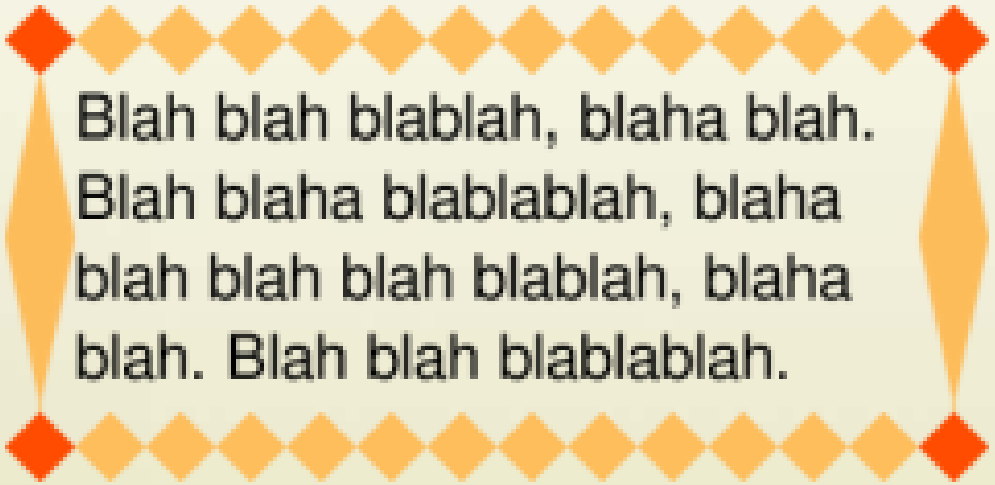
- 使用圖片作為 border
  - border-image-source
    path to the image to be used as a border
  - border-image-slice inward offsets of the image-border
  - border-image-width The widths of the image-border
  - border-image-outset
    The amount by which the border image area extends beyond the border box
  - border-image-repeat
    Whether the image-border should be repeated, rounded or stretched
- For border-image to work, the element also needs the border property set!

border-image

- shorthand property

  border-image: *source slice width outset repeat*;
- e.g. box/box-border-image.html

  border-image:url("border.png") 27 round stretch;



- http://www.w3.org/TR/css3-background/#border-images

outline 1/2

- 設定與 border 類似，但是四邊都一樣，無法個別設定
- outline: [<'outline-color'> || <'outline-style'> || <'outline-width'>]| inherit
  - outline-witdh
  - outline-style:
    - 不能使用 hidden
  - outline-color: <color> | invert | inherit
- outline 總是在 box 上面，不影響 box 的大小與位置，不影響版面編排
- outline 不一定是個矩形
- http://www.w3.org/TR/CSS21/ui.html#propdef-outline

outline 2/2

- e.g. box/box-outline.html
- outline 常與 :focus, :active 同時使用，提醒使用者已選用某項物件
  - button:focus {outline: thick solid black }
    button:active {outline: thick solid red }
- outline-offset:
  - 與 outline 搭配，設定 outline 與 border 的間距

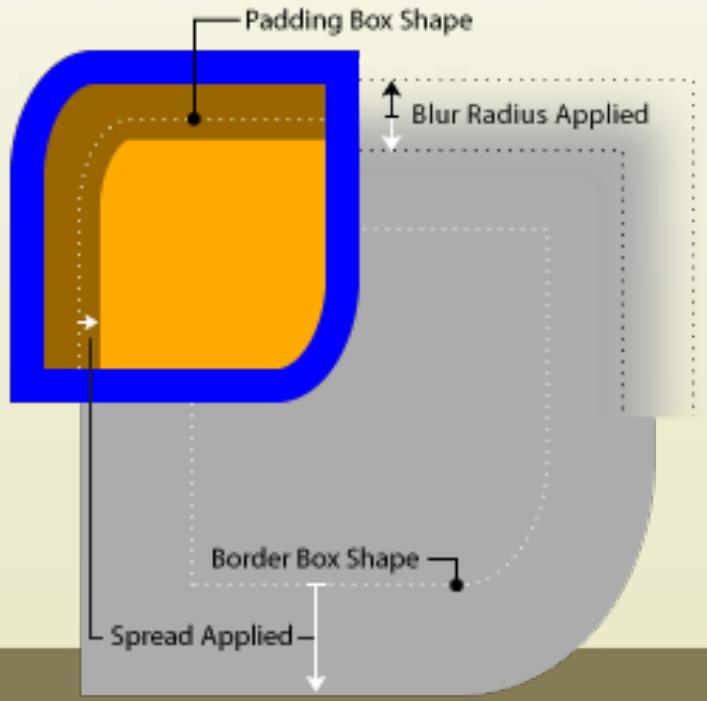
box-shadow

- box-shadow:

  h-shadow v-shadow blur-radius spread color inset;

  - inset: 內陰影
- 可同時設定多重 shadow

```
Width:100px; Height:100px;
Border: 12px solid blue; Background-color: orange;
Border-top-left-radius: 60px 90px;
Border-bottom-right-radius: 60px 90px;
Box-shadow: 64px 64px 12px 40px rgba(0,0,0,0.4),
            12px 12px 0px 8px rgba(0,0,0,0.4) inset;
```



http://www.w3.org/TR/css3-background/#the-box-shadow

Browser Prefixes for Extension

Browser	Prefix	Example
Firefox	-moz-	-moz-box-shadow
Chrome/Safari	-webkit-	-webkit-box-shadow
Opera	-o-	-o-box-shadow
Microsoft	-ms-	-ms-box-shadow

[note] IE8 使用 filter: dropshadow()

設定 box 寬度與高度

與 Box 相關的其他屬性

- height 設定元素的高度
- width 設定元素的寬度
- e.g. 指定div元素的寬度與高度：

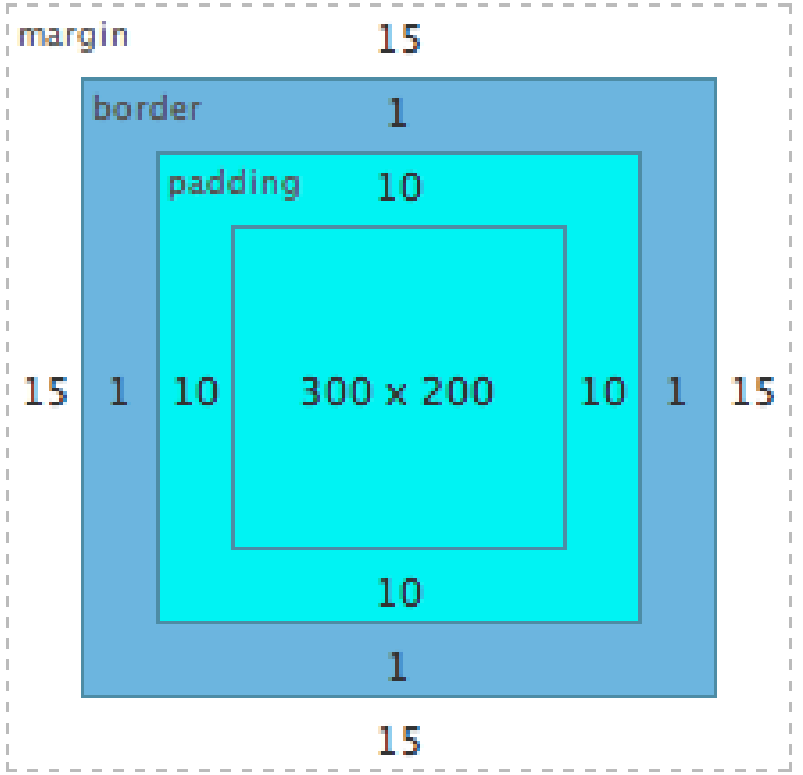
```
div#div1 {width: 120px; height: 120px;}
div#div2 {width: 160px; height: 60px;}
div#div3 {width: 60%; height: 15%;}
```

CSS Compatibility

- The main problem with using CSS has been a inconsistent browser support. Browsers can interpret CSS commands in different ways.
- e.g. CSS box 的 width and height
  - CSS box 的 width, height 在不同的 browser 會有不同的作法
  - 根據 W3C 的規格, width, height 指的只有內容區, 不含 padding, border, and margin
  - Internet Explorer 有些版本(e.g. 5.5)則包括 content area, padding, and border

Width and Height Issues of Box Model

- e.g. div {
  width: 300px;
  height: 200px;
  padding: 10px;
  border: 1px solid black;
  margin: 15px;
  }



- 若依 W3C 的規格，width 加上 padding, border，整個 box 的大小為 352 * 252
- 在 IE5.5 則因 width 已含 padding, border，整個 box 的大小為 330 * 230

box-sizing property

- CSS3 明訂 box 的計算方式
- box-sizing: content-box | padding-box | border-box | inherit
  - content-box: 內定值，同 W3C CSS 2.1 規格
  - padding-box: 長寬包括 padding
  - border-box: 長寬包括 padding 及 border

- http://www.w3.org/TR/css3-ui/#box-sizing

設定最大與最小寬度與高度

- max-height　設定元素的最大高度
- max-width　設定元素的最大寬度
- min-height　設定元素的最小高度
- min-width　設定元素的最小寬度
- 只能使用在區塊元素以及可替換元素(例如圖片)上，而且數值只應用在元素的內容區域。
- e.g. box/box-height-min.html

note

width, height, padding, border-width 都不可以是負值，margin 可以