

Amos 鐵人賽

Contents

鐵人 4 html 基本

鐵人 5 網頁裝潢 CSS 基本

鐵人 6 css 撰寫位置

鐵人 7 css 基本學習

鐵人 8 區塊尺寸計算

鐵人 9 Float 浮動

鐵人 10 網頁兩大主角 inline 、 block --- inline-block

鐵人 11 box-sizing (對比鐵人八內容)

鐵人 12 CSS reset

鐵人 13 flex 排版聖品

鐵人 14 fixed 固定定位

鐵人 15 relative 相對定位

鐵人 16 Absolute 絕對定位

鐵人 17 稀飯版

鐵人 18 transition 動畫

鐵人 19 animation 動畫

鐵人 20 媒體查詢 @media

鐵人 21 RWD 入門

鐵人 22 RWD 試排版

鐵人 23 RWD 選單製作

鐵人 24 Bootstrap 入門

鐵人 26 JQuery 入門

鐵人 27 JQuery 事件

鐵人 28 JQuery CSS 控制

鐵人 29 JQuery CSS 動畫

鐵人 4 html 基本 :

1. 標題

<h1> 主標題 <h2> 副標題 <h3>小標題

2. 無序清單 、 有序清單

```
<ul>
    <li>清單</li>
    <li>清單</li>
    <li>清單</li>
</ul>
<ol>
    <li>清單</li>
    <li>清單</li>
    <li>清單</li>
</ol>
```

3. 內文變化 :

超連結 / 斜體 、 <i> </i>
粗體

4. 圖片標籤

圖片
圖片
<figure>

 <figcaption>(描述文字)圖說</figcaption>
</figure>

5. 選單

```
<nav>
    <a href="">選單</a>
    <a href="">選單</a>
    <a href="">選單</a>
</nav>
```

6. 表格

```
<table border="2">
    <!-- border="2" 增加表格邊框線 -->
    <tr>
        <td>data</td>
        <td>data</td>
        <td>data</td>
        <td>data</td>
        <td>data</td>
    </tr>
    <tr>
        <td>data</td>
        <td>data</td>
        <td>data</td>
        <td>data</td>
        <td>data</td>
    </tr>
    <tr>
        <td>data</td>
        <td>data</td>
        <td>data</td>
        <td>data</td>
        <td>data</td>
    </tr>
    <tr>
        <td>data</td>
        <td>data</td>
        <td>data</td>
        <td>data</td>
        <td>data</td>
    </tr>
</table>
```

data	data	data	data	data
data	data	data	data	data
data	data	data	data	data

鐵人 5 網頁裝潢 CSS 基本

1. 標籤裡的樣式寫法

```
<p style="color:#aas"> mpfjsrgpiojpiwrgpiwpg </p>
```

2. CSS 樣式寫法 寫於<head>裡面

```
<style>
段落{
    文字色彩 : 紅色 ;
    字體大小 : 20px;
    背景色彩 : 淺灰色 ;
}
</style>
```

3. Class 選取器 :

h1 表示標籤的 h1

.h1 表示選取器(class)的 h1

4. 線上學習 MDN 、 W3C

Selector	Example	Example description
<u>class</u>	.intro	Selects all elements with class="intro"
<u>#id</u>	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element,element</u>	div, p	Selects all <div> elements and all <p> elements
<u>element element</u>	div p	Selects all <p> elements inside <div> elements
<u>element>element</u>	div > p	Selects all <p> elements where the parent is a <div> element
<u>element+element</u>	div + p	Selects all <p> elements that are placed immediately after <div> elements
<u>element1~element2</u>	p ~ ul	Selects every element that are preceded by a <p> element
<u>[attribute]</u>	[target]	Selects all elements with a target attribute
<u>[attribute=value]</u>	[target=_blank]	Selects all elements with target="_blank"

鐵人 6 CSS 撰寫位置

1. CSS 寫在外部 寫法

先在外部創建一個 CSS 檔案在 link 進去

```
<link rel="stylesheet" href="    .css">
```

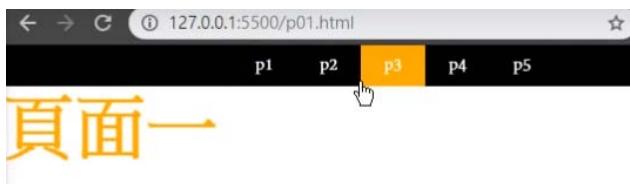
2. CSS reseat 設定：

```
*{  
margin : 0;  
padding : 0;  
}
```

3. 導覽列呈現

```
nav{  
    text-align: center; 選單置中  
}  
  
nav a {  
    color: white;  
    text-decoration: none; 字體裝飾：無  
    display: inline-block; 字體有間距出來  
    padding: 6px 20px;  
    font-size: 16px;  
    transition: 0.5s; (呈接下方 havor 摸到後延遲 0.5 秒)  
}
```

```
nav a:hover{ (havor:如果 nav 被碰到會有效果)  
    background-color: orange; 碰到後背景變橘色  
}
```



鐵人 7 CSS 基本學習

1.

```
div{  
    width: 320px;  
    border: 10px solid red;  
    padding: 20px;  
    background-color: yellow;  
    color: orange;  
    font-size: 24px ;  
    line-height: 1.5; 行高  
    height: 400px; 高度  
}  
  
div a{  
    text-decoration: none;  
    color: red;  
}
```

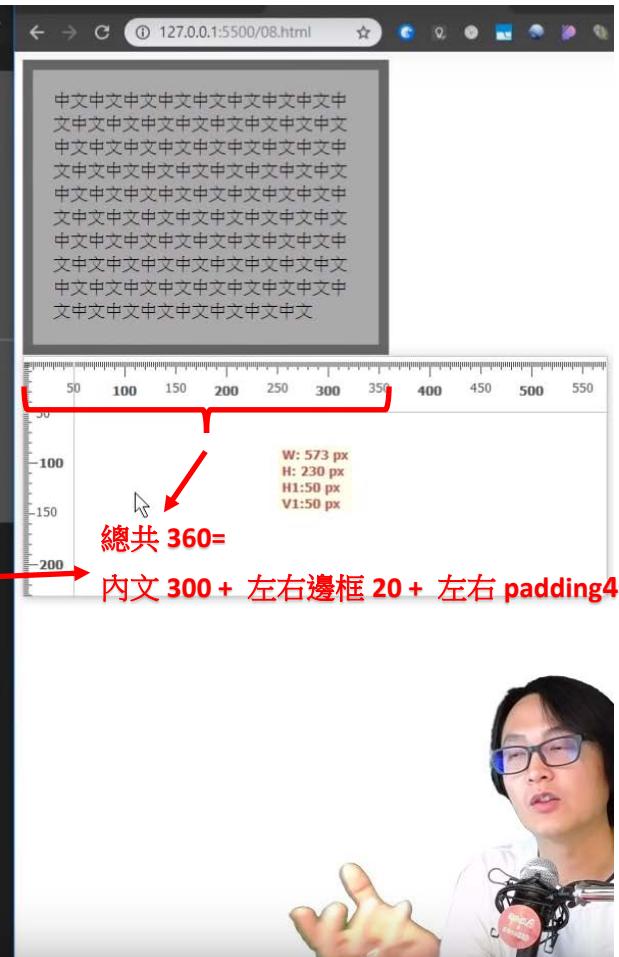
Lorem ipsum dolor dlfdlkjfdasit
amet consectetur adipisicing elit.
Voluptas eum ipsum quod
quibusdam! Sunt quasi quia velit
quod eum dolore, labore fuga
quo hic accusantium autem et
quidem eius soluta.

鐵人 8 區塊尺寸計算

1.

一般寬度的設定指的是內文的設定，並非整體寬度：

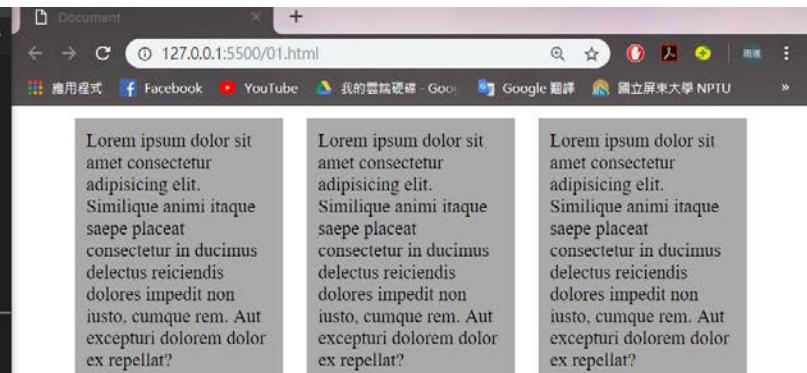
Div 裡的 width300 並不是整個 div300，而是內文 300



2. Float 使其排三列

```
① 14.html ② 01.html ●

1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, i
6       <meta http-equiv="X-UA-Compatible" content="ie=edge"
7       <title>Document</title>
8       <style type="text/css">
9       {
10           margin: 0;
11           padding: 0;
12       }
13       .box{
14           width: 600px;
15           background-color: #eee;
16           margin: auto;
17       }
18       .column{
19           width: 160px;
20           background-color: #aaa;
21           float: left;
22           margin: 10px;
23           padding: 10px;
24       }
25   </style>
26 </head>
27 <body>
28     <div class="box">
29         <div class="column">Lorem ipsum dolor sit am
30         <div class="column">Lorem ipsum dolor sit am
31         <div class="column">Lorem ipsum dolor sit am
32     </div>
33 </body>
34 </html>
```



鐵人 9 Float 浮動

Float：文繞圖等版面編排

粉紅色塊原先會看不見 (因黃色區塊浮動起來使其抓不到子層所以撐不開)

靠 clear 來撐開浮動 float 以下的父層物件(iii 類似介於 col 與 container 之間)，使 container 可以看的見(否則原本因為 col 浮起來的關係，container 將撐不開而不見)

```
7 <title>Document</title>
8 <style>
9   *{
10     margin: 0;
11     padding: 0;
12   }
13   .container{
14     width: 960px;
15
16     background-color: #rgb(233, 48, 125);
17     margin : auto;
18   }
19   .col{
20     width: 240px;
21     padding: 20px;
22     background-color: #rgb(223, 161, 27);
23     margin :20px;
24     float: left;
25
26   }
27   .col img{
28     width: 100%;
29   }
30   .iii{ /* 靠clear來撐開浮動以下的父層物件(他iii)
31     clear: both;
32   }
33
34 </style>
35 </head>
36 <body>
37
38   <div class="container">
39     <div class="col">
40     <h2>WWW</h2><p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae asperiores recusandae fugit maiores amet. Veniam, quas dolorem recusandae consequuntur facilis neque ut est, magnam quaerat aliquam repudiandae libero doloremque iure.
41     <div class="col">
42     <h2>WWW</h2><p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae asperiores recusandae fugit maiores amet. Veniam, quas dolorem recusandae consequuntur facilis neque ut est, magnam quaerat aliquam repudiandae libero doloremque iure.
43     <div class="col">
44     <h2>WWW</h2><p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae asperiores recusandae fugit maiores amet. Veniam, quas dolorem recusandae consequuntur facilis neque ut est, magnam quaerat aliquam repudiandae libero doloremque iure.
45     <div class="iii"></div>
46
47   </div>
48
49 </body>
```



鐵人 10 網頁兩大主角 inline、block --- inline-block

1. 網頁兩大主角 (float 可以使 block 的標籤整齊排排站)

inline：不可設定寬與高，但可與其他人並排，不會自己佔掉整行空間 ex:超連結

block：可設定寬與高，但會自己佔掉整個空間 ex: <h1>... <div>、<p>...

inline-block 結合兩者，可設定寬與高，也可與其他人並排

2. display:inline 可將原本是 block 的標籤轉換成 inline 屬性

3. 範例 – 原先都有 block 特性的標籤透過 display:inline 可轉成並排特性

The screenshot shows a browser window with the title "title". The page content includes a list of three items under an ul tag. The first item is styled with inline-block, causing it to break onto a new line. The second and third items are styled with inline, appearing side-by-side. The browser interface at the top shows various tabs and icons.

```
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <meta http-equiv="X-UA-Compatible" content="ie=edge">
7  <title>Document</title>
8  <style type="text/css">
9
10 </style>
11 </head>
12 <body>
13   <h1>title</h1>
14   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Iste, esse soluta. Aperiam repudiandae, animi inventore numquam deserunt sunt? Laudantium ipsum natus quisquam. Placeat in magni repellendus, atque eveniet nisi fugit!
15   <ul>
16     <li>fjfjifie</li>
17     <li>fjfjifie</li>
18     <li>fjfjifie</li>
19   </ul>
20   <div>
21     Lorem ipsum dolor sit amet consectetur adipisicing elit. Repellat quasi, nostrum velit iure nobis voluptatis quod sed, id in fuga placeat quos asperiores. Quae consectetur magnam ad assumenda officiis corrupti?
22   </div>
23   | ffjsljf
24 </body>
25 </html>
```

The screenshot shows a browser window with the title "title". The page content includes a list of three items under an ul tag. The first item is styled with inline, causing it to break onto a new line. The second and third items are styled with inline-block, appearing side-by-side. The browser interface at the top shows various tabs and icons.

```
7  <title>Document</title>
8  <style type="text/css">
9  div{
10    |   display:inline;
11  }
12  h1{
13    |   display: inline;
14  }
15  p{
16    |   display: inline;
17  }
18  </style>
19 </head>
20 <body>
21   <h1>title</h1>
22   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Iste, esse soluta. Aperiam repudiandae, animi inventore numquam deserunt sunt? Laudantium ipsum natus quisquam. Placeat in magni repellendus, atque eveniet nisi fugit!
23   <ul>
24     <li>fjfjifie</li>
25     <li>fjfjifie</li>
26     <li>fjfjifie</li>
27   </ul>
28   <div>
29     Lorem ipsum dolor sit amet consectetur adipisicing elit. Repellat quasi, nostrum velit iure nobis voluptatis quod sed, id in fuga placeat quos asperiores. Quae consectetur magnam ad assumenda officiis corrupti? ffjsljf
30   </div>
31   | ffjsljf
32 </body>
33 </html>
```

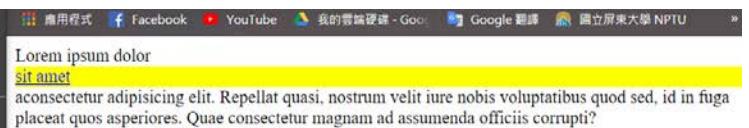
4. Inline 屬性無法給予 寬與高 (沒效果)

The screenshot shows a browser window with the title "title". The page content includes a single link with inline styling. The link's width is set to 900px, and its background color is yellow. The browser interface at the top shows various tabs and icons.

```
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <meta http-equiv="X-UA-Compatible" content="ie=edge">
7  <title>Document</title>
8  <style type="text/css">
9  a{
10    |   display: inline;
11    |   width: 900px;
12    |   background-color: yellow;
13  }
14  </style>
15 </head>
16 <body>
17   <div>
18     Lorem ipsum dolor <a href="#">sit amet</a>consectetur adipisicing e
19   </div>
20
21 </body>
22 </html>
```

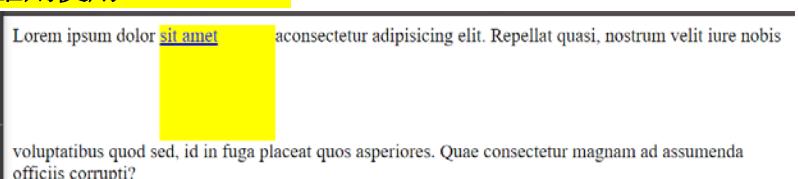
5. block 屬性則可以 純予寬與高

```
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Document</title>
8     <style type="text/css">
9       a{
10         display: block;
11         width: 900px;
12         background-color: #yellow;
13       }
14     </style>
15   </head>
16   <body>
17     <div>
18       Lorem ipsum dolor <a href="#">sit amet</a>aconsectetur adipisicing e
19     </div>
20
21   </body>
22 </html>
```



6. 那當需要與他人並排，又想要有寬高的話則使用 inline-block

```
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, i
6   <meta http-equiv="X-UA-Compatible" content="ie-edge">
7   <title>Document</title>
8   <style type="text/css">
9     a{
10       display: inline-block;
11       width:100px;
12       height: 100px;
13       background-color: #yellow;
14     }
15   </style>
16 </head>
17 <body>
18   <div>
19     |   Lorem ipsum dolor <a href="#">sit amet</a>aconsec
20   </div>
21
22 </body>
23 </html>
```



7. Inline-block 應用 : (1)原先的超連結加上 padding 後是撐不開的，但設定 inline-block 後即可撐開 (2)加入 font-size : 0 解決空隙問題

```
9   <style>
10  .menu{
11    background-color: #orange;
12    text-align: center;
13    font-size: 0; ←
14  }
15  .menu a{
16    color: #white;
17    text-decoration: none;
18    padding: 10px 20px;
19    background-color: #red;
20    display: inline-block;
21    font-size:16px ; ←
22  }
23  .menu a:hover{
24    background-color: #black;
25  }
26  </style>
27 </head>
28 <body>
29   <div class="menu">
30     <a href="">Link</a>
31     <a href="">Link</a>
32     <a href="">Link</a>
33     <a href="">Link</a>
34     <a href="">Link</a>
35   </div>
36 </body>
37 </html>
```



因此在 .menu 加入
font-size:0
使 menu 不要有字間距
(剛加入時會使<a>全部不見)

再來只需在 .menu a 加入
font-size:16px
即可將超連結撐開而顯示回來
便解決 link 之間空隙問題

在 .menu a 之中加入
transition: 1s;
即可控制摸到後的顯示時間

鐵人 11 box-sizing (對比鐵人八內容)

1. 盒模型計算模式 box-sizing :

控制寬度值(width)要指定給誰 (省去原本計算 width+padding+border 的麻煩)

*Box-sizing : border-box : 指我要把寬度(width)值指定給 border(邊框到邊框)，padding 和 border 會自動往內縮

*Box-sizing : content-box : 指我要把寬度(width)值指定內容物(content)

```
6  <meta http-equiv="X-UA-Compatible" content="ie=edge">
7  <title>Document</title>
8  <style>
9  div{
10     width: 300px;
11     background-color: orange;
12     border: 10px double black;
13     padding: 10px;
14     box-sizing: border-box; /* box-sizing 代表要將width(寬度值)設定給誰 */
15     /* border-box 代表根據邊框的部份去指定300px作為寬度 */
16 }
17 </style>
18 </head>
19 <body>
20 <div>
21     Lorem ipsum dolor sit amet consectetur adipisicing elit. Quia delectus rep
22     Quisquam natus nam aliquam dicta quibusdam minus sed provident id.
23 </div>
24 </body>
```

Lorem ipsum dolor sit amet consectetur
adipisicing elit. Quia delectus
repellendus nulla unde laudantium
inventore provident assumenda porro
aspernatur doloribus! Quisquam natus
nam aliquam dicta quibusdam minus
sed provident id.

內文 + padding20 + border20 = 300

border-box

將寬度指定給邊框

```
6  <meta http-equiv="X-UA-Compatible" content="ie=edge">
7  <title>Document</title>
8  <style>
9  div{
10     width: 300px;
11     background-color: orange;
12     border: 10px double black;
13     padding: 10px;
14     box-sizing: content-box;
15 }
16 </style>
17 </head>
18 <body>
19 <div>
20     Lorem ipsum dolor sit amet consectetur adipisicing elit. Quia delectu
21     Quisquam natus nam aliquam dicta quibusdam minus sed provident id.
22 </div>
23 </body>
```

Lorem ipsum dolor sit amet consectetur
adipisicing elit. Quia delectus repellendus
nulla unde laudantium inventore provident
assumenda porro aspernatur doloribus!
Quisquam natus nam aliquam dicta quibusdam
minus sed provident id.

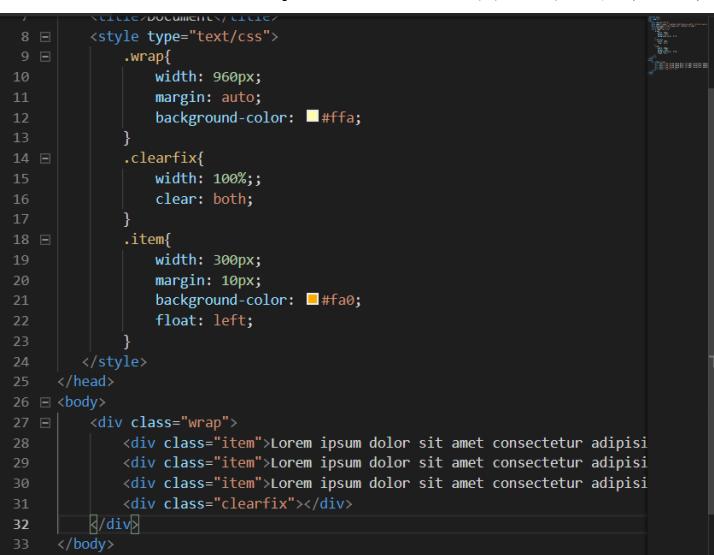
內文 300

content-box

將寬度指定給內文

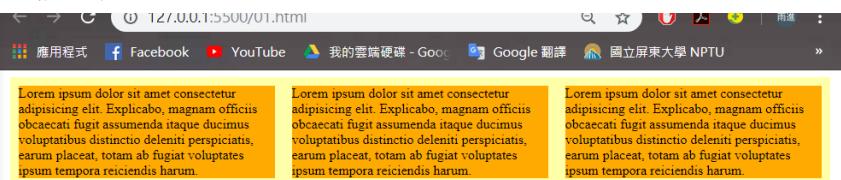
2. 範例:

原先都得計算好 $960px = 300px * 3 + 10px * 6$ 此時如果我要加入 padding 進去就得重新計算使整體等於 960px，否則版面會跑掉，變成不是三欄並排

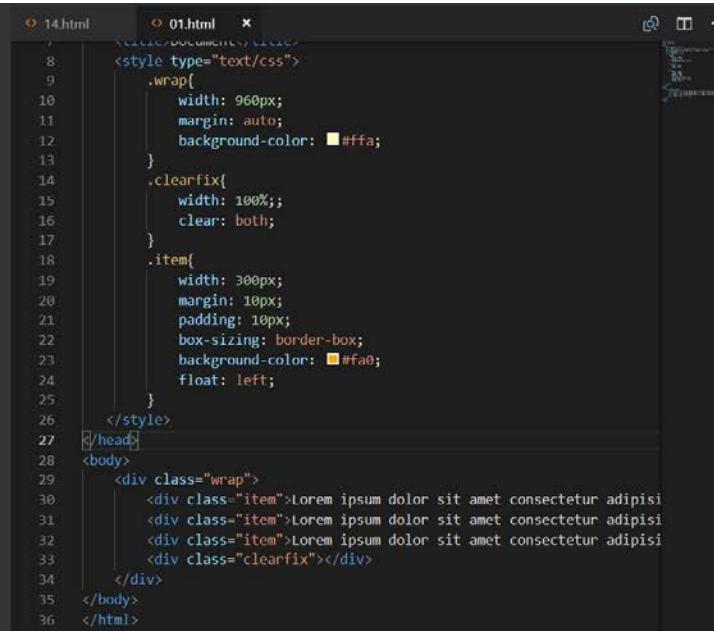


```
<html><head>
<style type="text/css">
.wrap{
    width: 960px;
    margin: auto;
    background-color: #ffa;
}
.clearfix{
    width: 100%;
    clear: both;
}
.item{
    width: 300px;
    margin: 10px;
    background-color: #fa0;
    float: left;
}
</style>
</head>
<body>
    <div class="wrap">
        <div class="item">Lorem ipsum dolor sit amet consectetur adipisci...
        <div class="item">Lorem ipsum dolor sit amet consectetur adipisci...
        <div class="item">Lorem ipsum dolor sit amet consectetur adipisci...
        <div class="clearfix"></div>
    </div>
</body>

```



但使用 box-sizing 不用，假如指調 width 給 border，則其他屬性將自動內縮，因此我再加入個 padding:10px 進去.item 也能繼續保持三欄



```
<html><head>
<style type="text/css">
.wrap{
    width: 960px;
    margin: auto;
    background-color: #ffa;
}
.clearfix{
    width: 100%;
    clear: both;
}
.item{
    width: 300px;
    margin: 10px;
    padding: 10px;
    box-sizing: border-box;
    background-color: #fa0;
    float: left;
}
</style>
</head>
<body>
    <div class="wrap">
        <div class="item">Lorem ipsum dolor sit amet consectetur adipisci...
        <div class="item">Lorem ipsum dolor sit amet consectetur adipisci...
        <div class="item">Lorem ipsum dolor sit amet consectetur adipisci...
        <div class="clearfix"></div>
    </div>
</body>

```



鐵人 12 CSS reset

CSS reset：解決各家瀏覽器版型不同之狀況，因此加入 **CSS Reset** 讓所有值重置。

讓開發者可以制定所有督興都按照自己的距離行高等等都自行設定(否則原先會被瀏覽器自訂)

<https://meyerweb.com/eric/tools/css/reset/>

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>Document</title>
<link rel="stylesheet" href="reset.css">
<style type="text/css">
h1{
    font-size: 32px; /*手動加入字型大小*/
    line-height: 1; /*行高1倍*/
    margin-bottom: 1em; /*h1 底部距離*/
}
p{
    font-size: 16px;
    line-height: 1.2;
    margin-bottom: 20px;
}
</style>
</head>
<body>
    <h1>ffwwlorem20dddsddddddwwwpsum, dolor sit amet consectetur adipisicing elit. Architecto, optio corporis dolorum ratione, omnis vel quibusdam, nobis officia nulla praesentium quasi? Est dolorum quos omnis, incidunt aspernatur magni numquam dolores! lestiae consectetur laborum aliquam ex errorw
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Architecto, optio corporis dolorum ratione, omnis vel quibusdam, nobis officia nulla praesentium quasi? Est dolorum quos omnis, incidunt aspernatur magni numquam dolores! lestiae consectetur laborum aliquam ex error repudiandae quae, neque a nemo nostrum! Explicabo placeat numquam dolorum accusantium nobis eum ullam.
    <ol><li>fefef</li>
    <li>fefe</li>
    <li>fef</li>
    <li>fefe</li>
    <li>eff</li></ol>
    <nav>
        <a href="#">1212</a>
        <a href="#">dcd</a>
        <a href="#">dcdc</a>
    </nav>
</body>
```

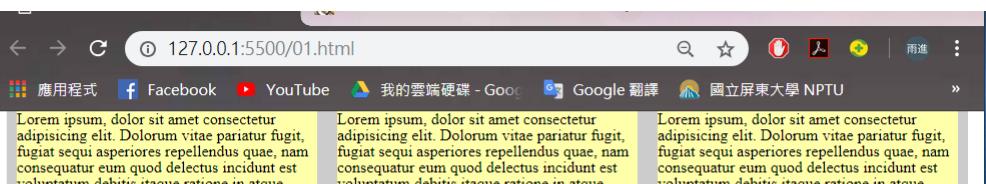
ffwwlorem20dddsddddddwwwpsum, dolor sit amet consectetur adipisicing elit. Architecto, optio corporis dolorum ratione, omnis vel quibusdam, nobis officia nulla praesentium quasi? Est dolorum quos omnis, incidunt aspernatur magni numquam dolores! lestiae consectetur laborum aliquam ex error repudiandae quae, neque a nemo nostrum! Explicabo placeat numquam dolorum accusantium nobis eum ullam.

fefef
fefe
fef
fefe
eff
1212 dcd dcdc

鐵人 13 flex 排版聖品 (不一定適用於傳統的瀏覽器 ex: IE 8...)

Wrap：父層 item：子層

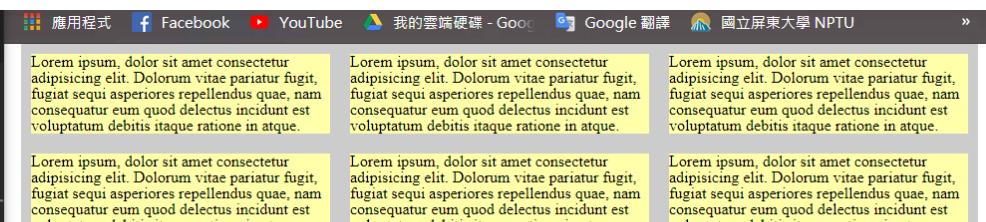
- 原先得計算好寬度比例再分配，並在 item 裡加上 float。現在只需直接在 wrap 裡加上 display:flex 將自動排版完成，也不會受 item 寬度設定錯誤而影響版面排版。



The screenshot shows a browser window displaying a web page with three columns of text. The browser interface includes a toolbar with icons for back, forward, search, and refresh, and a status bar indicating the URL is 127.0.0.1:5500/01.html.

```
<style type="text/css">
.wrap{
    width: 960px;
    margin: auto;
    background-color: #ccc;
    display:flex;
}
.item {
    width: 300px;
    margin: 0 10px;
    background-color: #ffa;
}
</style>
</head>
<body>
    <div class="wrap">
        <div class="item">
            Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolorum vitae pariatur fugit, fugiat sequi asperiores repellendus quae, nam consequatur eum quod delectus incidunt est voluptatum debitis itaque ratione in atque.
        </div>
        <div class="item">
            Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolorum vitae pariatur fugit, fugiat sequi asperiores repellendus quae, nam consequatur eum quod delectus incidunt est voluptatum debitis itaque ratione in atque.
        </div>
        <div class="item">
            Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolorum vitae pariatur fugit, fugiat sequi asperiores repellendus quae, nam consequatur eum quod delectus incidunt est voluptatum debitis itaque ratione in atque.
        </div>
    </div>
</body>
```

- (如有更多 div，且需要換行，則在 wrap 多加個 flex-wrap:wrap 即可根據 item 的寬度配合 wrap 換行)



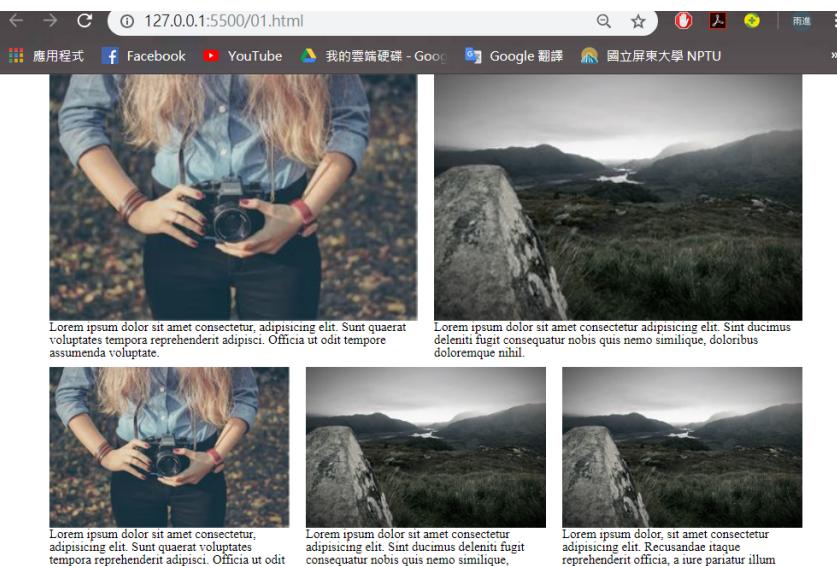
The screenshot shows a browser window displaying a web page with four columns of text. The browser interface includes a toolbar with icons for back, forward, search, and refresh, and a status bar indicating the URL is 127.0.0.1:5500/01.html.

```
.wrap{
    width: 960px;
    margin: auto;
    background-color: #ccc;
    display:flex;
    flex-wrap: wrap;
}
.item {
    width: 300px;
    margin: 10px 10px; /*加入flex-wrap後  
需要加個上下10px才看  
的出來有換行*/
    background-color: #ffa;
}

</style>
</head>
<body>
    <div class="wrap">
        <div class="item">
            Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolorum vitae pariatur fugit, fugiat sequi asperiores repellendus quae, nam consequatur eum quod delectus incidunt est voluptatum debitis itaque ratione in atque.
        </div>
        <div class="item">
            Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolorum vitae pariatur fugit, fugiat sequi asperiores repellendus quae, nam consequatur eum quod delectus incidunt est voluptatum debitis itaque ratione in atque.
        </div>
        <div class="item">
            Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolorum vitae pariatur fugit, fugiat sequi asperiores repellendus quae, nam consequatur eum quod delectus incidunt est voluptatum debitis itaque ratione in atque.
        </div>
        <div class="item">
            Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolorum vitae pariatur fugit, fugiat sequi asperiores repellendus quae, nam consequatur eum quod delectus incidunt est voluptatum debitis itaque ratione in atque.
        </div>
    </div>
</body>
```

3. 如果需要換行，但不想計算 item 寬度，則只需要多加幾個 wrap 即可用不同.wrap 來換行

```
10 .wrap{  
11     margin: auto;  
12     width: 960px;  
13     display: flex;  
14 }  
15 .item{  
16     width: 100%;  
17     margin: 0 10px 10px;  
18 }  
19 .item img{  
20     width: 100%;  
21     vertical-align: top;  
22 }  
23 </style>  
24 </head>  
25 <body>  
26 <div class="wrap">  
27     <div class="item">  
28         <p>Lorem ipsum dolor sit amet consectetur, adipisci<br/></p>">  
29         <p>Lorem ipsum dolor sit amet consectetur, adipisci<br/></p>  
30     </div>  
31     <div class="item">  
32         <p>Lorem ipsum dolor sit amet consectetur adipisci<br/></p>">  
33         <p>Lorem ipsum dolor sit amet consectetur adipisci<br/></p>  
34     </div>  
35 </div>  
36 <div class="wrap">  
37     <div class="item">  
38         <p>Lorem ipsum dolor sit amet consectetur, adipisci<br/></p>">  
39         <p>Lorem ipsum dolor sit amet consectetur, adipisci<br/></p>  
40     </div>  
41     <div class="item">  
42         <p>Lorem ipsum dolor sit amet consectetur adipisci<br/></p>">  
43         <p>Lorem ipsum dolor sit amet consectetur adipisci<br/></p>  
44     </div>  
45     <div class="item">  
46         <p>Lorem ipsum dolor, sit amet consectetur adipisci<br/></p>">  
47         <p>Lorem ipsum dolor, sit amet consectetur adipisci<br/></p>  
48     </div>
```



鐵人 14 fixed 固定定位 – 依據視窗範圍做定位，永遠固定在那個地方

position:fixed 可將其獨立抽出 (他會固定在同樣的位置，就算是滾動卷軸也一樣)，常用於蓋板廣告

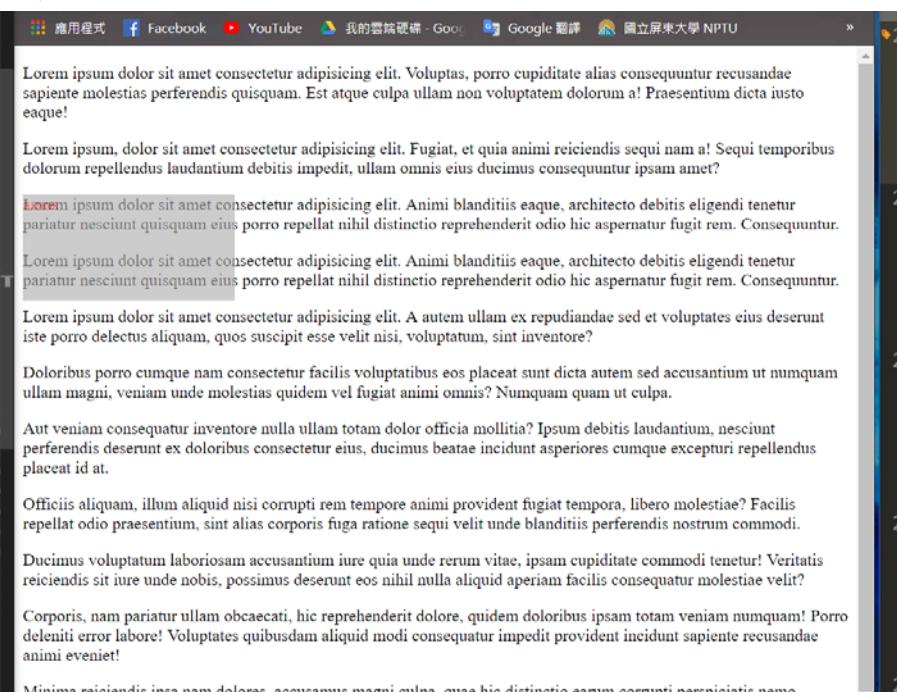
1. 不設定其詳細位置，則會依據本來的位置做故地定

```
10  

.amos{  
11    width: 200px;  
12    height: 100px;  
13    color: red;  
14    background-color: #aaa;  
15    position: fixed;  
16    opacity:0.6; /*使區域半透明*/  
17  }  
18

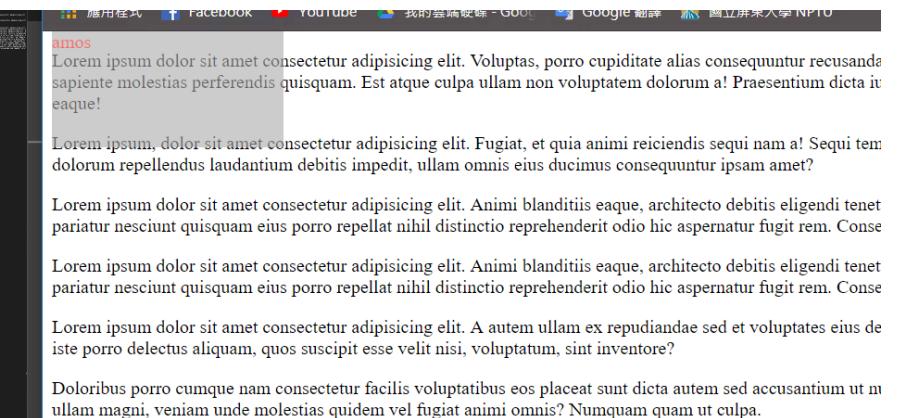

```

```
19 </style>  
20 </head>  
21 <body>  
22   <p>  
23     |   Lorem ipsum dolor sit amet consectetur adipisicing elit. v  
24   </p>  
25   <p>  
26     |   Lorem ipsum, dolor sit amet consectetur adipisicing elit.  
27   </p>  
28   <div class="amos">amos</div>  
29   <p>  
30     |   Lorem ipsum dolor sit amet consectetur adipisicing elit. A  
31     |<div style="border: 1px solid black; padding: 5px;">n  
32     <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ani  
33     <p> doloribus porro cumque nam consectetur facilis voluptatibus  
34     <p> Autem consequatur inventore nulla ullam totam dolor off  
35     <p> Officiis aliquam, illum aliquid nisi corrupti rem tempore an  
36     <p> Ducimus voluptatum laboriosam accusantium iure quia unde rer  
37     <p> Corporis, nam pariatur ullam obcaecati, hic reprehenderit de  
38     <p> Minima reiciendis ipsa nam dolores, accusamus magni culpa, o  
39     <p> Incidunt, ducimus veritatis porro eos fugit molestias simili  
40     <p> Soluta quia facilis est reprehenderit natus beatae pariatur  
41     <p> Ad maxime, iure ut asperiores recusandae sed impedit vero ma  
42 </body>  
43 </html>
```



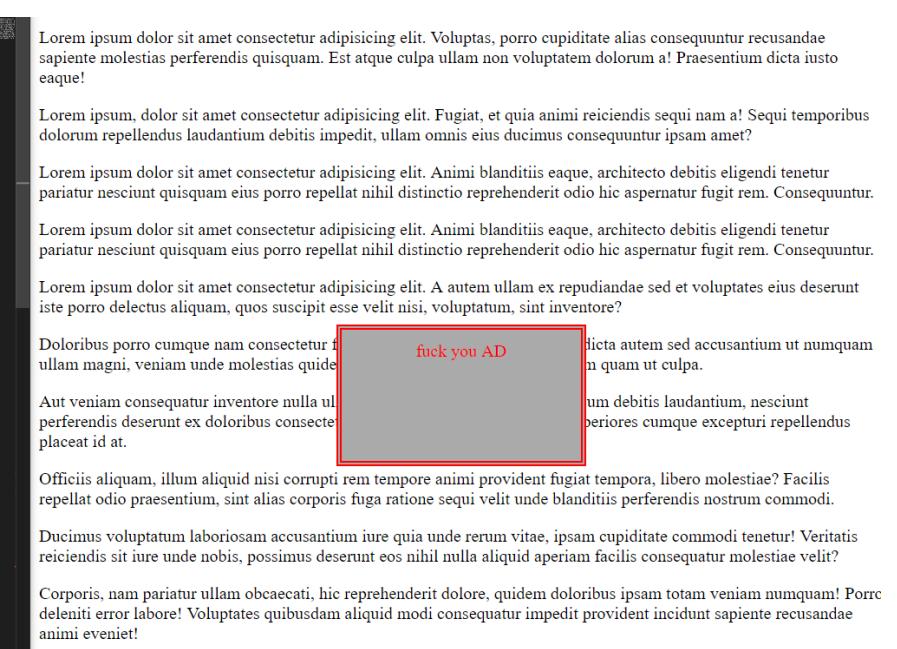
2. 如標明好詳細位置(上右下左) 則常用於 LOGO、導覽列、選單、頁尾...

```
<title>Document</title>
<!-- <link rel="stylesheet" href="reset.css"> -->
<style type="text/css">
    .amos{
        width: 200px;
        height: 100px;
        color: red;
        background-color: #aaa;
        position: fixed;
        opacity:0.6; /*使區塊半透明*/
        top: 0;
    }
</style>
</head>
<body>
    <p>
        |   Lorem ipsum dolor sit amet consectetur adipisicing elit. v
    </p>
    <p>
```



3. 正中間之蓋板廣告

```
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 <title>Document</title>
8 <!-- <link rel="stylesheet" href="reset.css"> -->
9 <style type="text/css">
10 .amos{
11     width: 200px;
12     height: 100px;
13     color: red;
14     background-color: #aaa;
15     position: fixed;
16     top: 0;
17     bottom: 0;
18     left: 0;
19     right: 0;
20     margin: auto;
21     text-align: center;
22     padding: 10px;
23     /* box-sizing: border-box; */
24     border: 5px double red;
25 }
26 </style>
27 </head>
28 <body>
29 <p>
30 | Lorem ipsum dolor sit amet consectetur adipisicing elit. V
31 </p>
32 <p>
33 | Lorem ipsum, dolor sit amet consectetur adipisicing elit.
34 </p>
```



4. 製作固定在上方的導覽列 (因為加入 position:fixed 後會使 nav 獨立抽出，導致其與段落 p 重疊，因此在 body 設定 padding-top 和 nav 設定 top:0、left:0、width:100%來撐開使版面完成)

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays a CSS file named 15.HTML with the following content:

```
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
```

The browser window on the right shows the rendered HTML page. The navigation bar (nav) is fixed at the top of the page, and the main content (p) is positioned below it. The browser's address bar shows the URL 127.0.0.1:5500/15.HTML.

鐵人 15 relative 相對定位 - 根據原有位置去做相對的偏移顯示 (未設定偏移值就不會有作用)

1. 範例：製作 50 格格子，使第 20 格偏移顯示 (其偏移方式是根據原本的位置做偏移)

```
<title>document</title>
<link rel="stylesheet" href="reset.css">
<style>
    ul{
        font-size: 0; /* 調整原本inline-block之間的空格問題 */
        /* 底下在li裡面用16px把它拉回來 */
    }
    li{
        display: inline-block; /* 改成inline-block即可並列
                                (否則本來block屬性會換行) */
        width: 100px;
        height: 100px;
        background-color: #aaa;
        margin: 10px;
        font-size: 16px; /* 拉回ul設定的font-size:0 */
        font-size: 50px; /* 使數字設定50px */
        text-align: center; /* 數字水平置中 */
        line-height: 100px; /* 數字垂直置中 */
    }
    .item20{
        position: relative; /* 設定成相對定位 */
        top: 30px; /* 冠上偏移值 */
        left: 30px; /* 冠上偏移值 */
        background-color: red;
        opacity: .5;
    }
</style>
</head>
<body>
    <ul>
        <li class="item1">1</li>
        <li class="item2">2</li>
        <li class="item3">3</li>
        <li class="item4">4</li>
        <li class="item5">5</li>
        <li class="item6">6</li>
        ...
        <li class="item19">19</li>
        <li class="item20" style="background-color: red; opacity: 0.5;">20</li>
        <li class="item21">21</li>
        <li class="item22">22</li>
        <li class="item23">23</li>
        <li class="item24">24</li>
        <li class="item25">25</li>
        <li class="item26">26</li>
        <li class="item27">27</li>
        <li class="item28">28</li>
        <li class="item29">29</li>
        <li class="item30">30</li>
        <li class="item31">31</li>
        <li class="item32">32</li>
        <li class="item33">33</li>
        <li class="item34">34</li>
        <li class="item35">35</li>
        <li class="item36">36</li>
        <li class="item37">37</li>
        <li class="item38">38</li>
        <li class="item39">39</li>
        <li class="item40">40</li>
        <li class="item41">41</li>
        <li class="item42">42</li>
        <li class="item43">43</li>
        <li class="item44">44</li>
        <li class="item45">45</li>
        <li class="item46">46</li>
        <li class="item47">47</li>
        <li class="item48">48</li>
        <li class="item49">49</li>
        <li class="item50">50</li>
    </ul>
</body>
</html>
```

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50				

2. z-index 應用

範例 假如格子 21 也給上相對定位：原先慣例是後方的物件會在前方物件之上，但設定 z-index 之後即可調整順序。

```
<title>document</title>
<link rel="stylesheet" href="reset.css">
<style>
    ul{
        font-size: 50px; /* 使數字設定50px */
        text-align: center; /* 數字水平置中 */
        line-height: 100px; /* 數字垂直置中 */
    }
    .item20{
        position: relative; /* 設定成相對定位 */
        left: 50px; /* 冠上偏移值 */
        background-color: red;
    }
    .item21{
        position: relative;
    }
</style>
</head>
<body>
    <ul>
        <li class="item1">1</li>
        <li class="item2">2</li>
        <li class="item3">3</li>
        <li class="item4">4</li>
        <li class="item5">5</li>
        <li class="item6">6</li>
        ...
        <li class="item19">19</li>
        <li class="item20" style="background-color: red; left: 50px; position: relative;">20</li>
        <li class="item21" style="position: relative;">21</li>
        <li class="item22">22</li>
        <li class="item23">23</li>
        <li class="item24">24</li>
        <li class="item25">25</li>
        <li class="item26">26</li>
        <li class="item27">27</li>
        <li class="item28">28</li>
        <li class="item29">29</li>
        <li class="item30">30</li>
        <li class="item31">31</li>
        <li class="item32">32</li>
        <li class="item33">33</li>
        <li class="item34">34</li>
        <li class="item35">35</li>
        <li class="item36">36</li>
        <li class="item37">37</li>
        <li class="item38">38</li>
        <li class="item39">39</li>
        <li class="item40">40</li>
        <li class="item41">41</li>
        <li class="item42">42</li>
        <li class="item43">43</li>
        <li class="item44">44</li>
        <li class="item45">45</li>
        <li class="item46">46</li>
        <li class="item47">47</li>
        <li class="item48">48</li>
        <li class="item49">49</li>
        <li class="item50">50</li>
    </ul>
</body>
</html>
```

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50				

加入 z-index:1 以後 即可跑到上層 (原先預設都是 z-index:0)

```
<title>document</title>
<link rel="stylesheet" href="reset.css">
<style>
    ul{
        line-height: 100px; /* 數字垂直置中 */
    }
    .item20{
        position: relative; /* 設定成相對定位 */
        left: 50px; /* 冠上偏移值 */
        background-color: red;
        z-index: 1; /* 加入z-index:1 後20格子即可往上跑了 */
    }
    .item21{
        position: relative;
    }
    .item22{
        position: relative;
        /* 預設的z-index為0 → z-index:0 */
        /* 如果在此加上z-index:1 則21即又照慣性跑到上層了 */
    }
</style>
</head>
<body>
    <ul>
        <li class="item1">1</li>
        <li class="item2">2</li>
        <li class="item3">3</li>
        <li class="item4">4</li>
        <li class="item5">5</li>
        <li class="item6">6</li>
        ...
        <li class="item19">19</li>
        <li class="item20" style="background-color: red; left: 50px; position: relative;">20</li>
        <li class="item21" style="position: relative;">21</li>
        <li class="item22" style="position: relative;">22</li>
        <li class="item23">23</li>
        <li class="item24">24</li>
        <li class="item25">25</li>
        <li class="item26">26</li>
        <li class="item27">27</li>
        <li class="item28">28</li>
        <li class="item29">29</li>
        <li class="item30">30</li>
        <li class="item31">31</li>
        <li class="item32">32</li>
        <li class="item33">33</li>
        <li class="item34">34</li>
        <li class="item35">35</li>
        <li class="item36">36</li>
        <li class="item37">37</li>
        <li class="item38">38</li>
        <li class="item39">39</li>
        <li class="item40">40</li>
        <li class="item41">41</li>
        <li class="item42">42</li>
        <li class="item43">43</li>
        <li class="item44">44</li>
        <li class="item45">45</li>
        <li class="item46">46</li>
        <li class="item47">47</li>
        <li class="item48">48</li>
        <li class="item49">49</li>
        <li class="item50">50</li>
    </ul>
</body>
</html>
```

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50				

3. 範例：使 20 格與 21 格左右互換 (往左往右 120px 原因: li 的 margin $10*2+width100=120px$)

```
ul{  
    font-size: 0; /* 調整原本inline-block之間的空格問題 */  
    /* 底下在li裡面用16px把它拉回來 */  
}  
  
li{  
    display: inline-block; /* 改成inline-block即可並列  
                           (否則本來block屬性會換行) */  
    width: 100px;  
    height: 100px;  
    background-color: #aaa;  
    margin: 10px;  
    font-size: 16px; /* 拉回ul設定的font-size:0 */  
    font-size: 50px; /* 使數字設定50px */  
    text-align: center; /* 數字水平置中 */  
    line-height: 100px; /* 數字垂直置中 */  
}  
  
.item20{  
    position: relative; /* 設定成相對定位 */  
    left: 120px; /* 往右跑120px */  
    background-color: red;  
}  
  
.item21{  
    position: relative;  
    left: -120px; /* 往左跑120px */  
}  
style>
```



鐵人 16 Absolute 絕對定位 (根據其父層做定位動作)

一般使用動作都會將父層設定成 **position:relative**，然後再將子層做 **position:absolute**，然後加上偏移值即可根據父層做其欲做的絕對定位

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>

<style>
div{
    padding: 40px;
    background-color: #aaa;
    border: 2px solid red;
    font-size: 20px;
}
.b{
    position: relative; /* 將b(父層)做相對定位 */
}
.c{
    position: absolute; /* 將c做絕對定位 */
    bottom: 0; /* 冠上偏移值 */
    left: 0;
    border: 2px solid blue; /* 另給框線區別 */
    padding: 5px; /* 將c變小 */
}
</style>
</head>
<body>
    <div class="a">a
        <div class="b">b
            <div class="c">c
            </div>
        </div>
    </div>
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Pariatur om<br>Repellat deleniti incident, quaerat corrupti autem tenetur id obcaecat<br>Nam odio a consequuntur quia quibusdam, provident, hic facere repellat</p>
```



Lorem ipsum dolor sit amet consectetur adipisicing elit. Pariatur omnis consectetur recusandae fugiat ex velit exercitationem, ipsum nam perferendis, esse, natus dolorum id aut in deserunt illo minus voluptas voluptatem!

Repellat deleniti incident, quaerat corrupti autem tenetur id obcaecati delectus unde in quas doloribus illum itaque fugit, sunt totam sint? Ea eum eos dolore fuga pariatur reprehenderit ratione quidem officia!

Nam odio a consequuntur quia quibusdam, provident, hic facere repellendus pariatur molestias eos eum iure saepe voluptatem, modi est reprehenderit in eveniet omnis inventore perspiciatibus eaque fugiat! Ut, necessitatibus voluptatem!

Aut adipisci corporis dolor recusandae officia tenetur error minus reprehenderit? Libero, recusandae cupiditate atque, ratione odit, excepturi molestias voluptatum eaque nobis nostrum impedit? Similique sunt velit nihil et, culpa suscipit?

Dignissimos asperiores eligendi pariatur fugiat deserunt perspiciatibus! Dignissimos odit iusto, porro labore quidem expedita voluptatem enim voluptas suscipit ex praesentium quasi doloribus dolores eum quos beatae consectetur impedit! Labore, autem.

Natus apariani enim, rerum nobis libero dolores eum, vero quasi iusto nesciunt, excepturi laboriosam sapiente quia dignissimos quod accusantium ea ratione praesentium molestiae voluptatum obcaecati sequi iste modif! Ab, maiores.

Placeat dolorum totam porro excepturi laudantium. Nostrum quo praesentium molestias cupiditate eveniet, illum rerum pariatur consequuntur eligendi explicabo. Molestias vel tempore odio voluptatum vero ab, est provident delectus neque id?

Assumenda ipsam nisi tempore tenetur corporis explicabo dolor. Sit, voluptatum, doloribus sequi ipsam quod unde provident suscipit repellat libero dignissimos, aut id possimus. Optio eligendi similique, minus veniam vero animi!

Voluptates quis assumenda fuga, animi illo itaque praesentium adipisci beatae corrupti molestiae nobis quam voluptate reiciendis commodi iste temporis perspiciatibus. Dolore rerum error hic quia itaque? Cumque doloribus quisquam ex.

Placeat consequuntur obcaecati nostrum ex dicta molestias omnis quasi voluptatem apariani. Vero inventore perspiciatibus molestias officia corrupti. Provident omnis maiores quasi accusamus. Est temporibus id maiores voluptas natus similius non.

Voluntate exercitationem furiat laboriosam, incident nihil impedit nesciunt, reiciendis delectus eorum necessitatibus horum. A autem nihil autemmodi modi

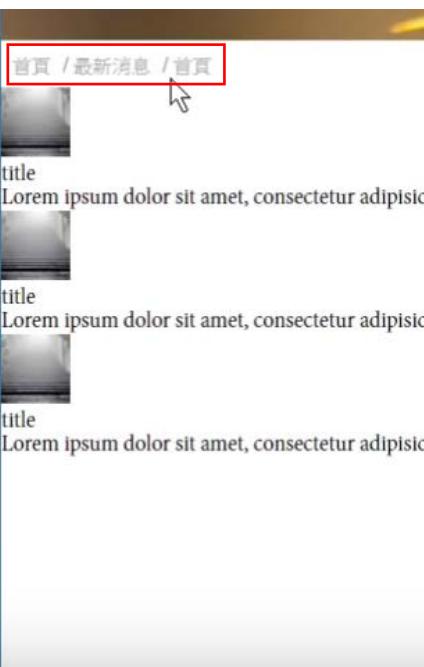
1. 範例：一般的 tag 或是蓋板廣告的關閉按鈕(X)

```
<link rel="stylesheet" href="reset.css">
<style>
.wrap{
    width: 100%;
    width: 960px;
    display: flex; /* 將wrap做flex來排版 */
    margin: auto;
}
.item {
    width: 300px;
    margin: 10px;
    border: 1px solid black;
    padding: 10px;
    position: relative; /* 將item(父層)做相對定位 */
}
.item .tag{
    background-color: red;
    color: white;
    padding: 6px 10px; /* 因設定成絕對定位所以被抽離，因此可撐開寬度 */
    position: absolute; /* 設定成絕對定位 */
    top: -5px; /* 冠上偏移值 */
    right: -5px;
}
</style>
</head>
<body>
    <div class="wrap">
        <div class="item">
            <div class="tag">hot</div> <!-- tag -->
            
            <h3>title</h3>
            <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit</p>
        </div>
        <div class="item">
            <div class="tag">hot</div> <!-- tag -->
            
            <h3>title</h3>
            <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit</p>
        </div>
    </div>
</body>
```



鐵人 17 稀飯版

1. 偽類用法 ex: 增加 "/"



```
.path{ display: flex; }
.path li{ position: relative; padding: 6px 8px; }
.path li a{ color: #aaa; text-decoration: none; font-size: 14px; }
.path li + li::before{ content: '/'; color: #aaa; position: absolute; left: 0; }
/style>
```

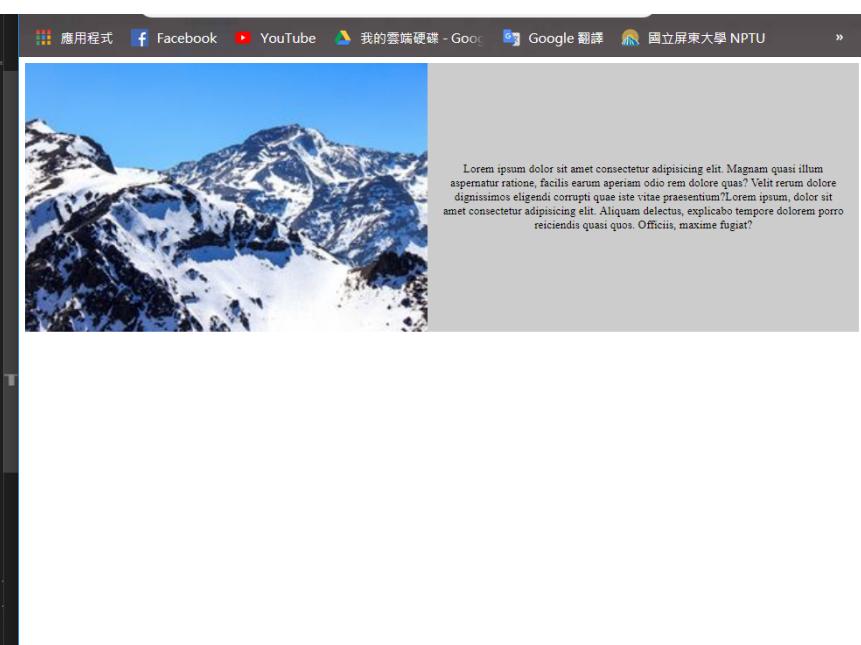
2. 加上 flex 屬性後，有些內容需要 text-align 或 left:0 等等控制位置的話不一定會正常作用，因此加上 width:100%即可正常運作

3. 在 .item img 中 加入 width: 100% 即可撐大圖片使其對稱 (如第 4 點↓)



```
.display: flex; /* 如原未設定 flex 只會垂直排列 */
.item{
  width: 50%; /* 使圖片與文字一半一半佔有 item */
}
.item img{
  vertical-align: bottom; /* 處理原先圖片底下的留白 */
}
/* 處理flex下的文字的置中設定 */
.text{
  display: flex;
  flex-direction: column;
  justify-content: center;
  text-align: center;
  padding: 0 20px;
}
```

4. 圖片與文字並排、細部設定 範例



```
<style>
.products{
  background-color: #ccc;
}
.products .wrap{
  display: flex; /* 如原未設定 flex 只會垂直排列 */
}
.item{
  width: 50%; /* 使圖片與文字一半一半佔有 item */
}
.item img{
  width: 100%;
  vertical-align: bottom; /* 處理原先圖片底下的留白 */
}
/* 處理flex下的文字的置中設定 */
.text{
  display: flex;
  flex-direction: column;
  justify-content: center;
  text-align: center;
  padding: 0 20px;
}
</style>
</head>
<body>
<div class="products">
  <div class="wrap">
    <div class="item pic">
    <div class="item text"><p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Magnam quasi illum aspernatur ratione, facilis earum aperiam odio rem dolore quas? Velit rerum dolore dignissimos eligendi corrupti quae iste vitae praesentium!Lorem ipsum, dolor sit amet consectetur adipisicing elit. Aliquam delectus, explicabo tempore dolorem porro reiciendis quasi quo. Officis, maxime fugiat?
```

5. 假如今天只需要對某 css 之 div 屬性底下的屬性做控制就好 則加上 ”>“即可

Ex: 指針對 .service 底下的 h2 或 p 做設定而已

```
.service > h2{  
    width: 960px;  
    margin: auto;  
    text-align: center;  
}  
.service > p{  
    width: 960px;  
    margin: auto;  
    text-align: center;  
}
```

6. 加入圓角 使用 border-radius: 50%; 即可圓角

```
padding: 40px 0 60px;  
}  
.about .wrap{  
    display: flex;  
}  
.about .item{  
    width: 300px;  
    margin: 0 10px;  
    text-align: center;  
}  
.about .item img{  
    border-radius: 50%;  
}  
.about > h2{  
    width: 960px;  
    font-size: 42px;  
}
```



鐵人 18 transition 動畫

transition: 屬性 轉換時間 延遲執行時間 速度 ;

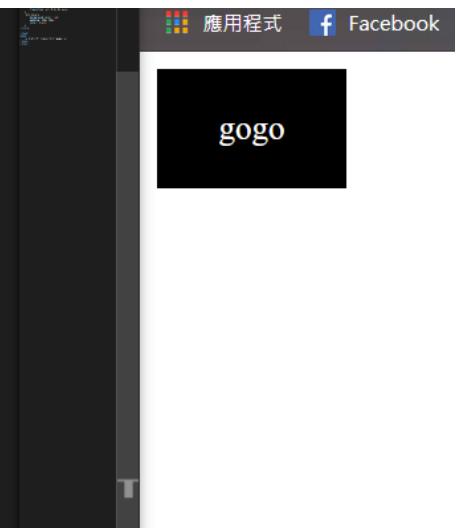
ex1: transition: all 1s 0s ease(預設值)

ex2: transition: background-color(只針對背景色做轉換) 1s 0s ease(預設值)

如需針對多組指定的話加入”，即可

範例：原為黑色按鈕白色字體，後來摸到後改為紅色按鈕黑色字體 (整體一起轉換)

```
<style>
    .btn {
        display: inline-block;
        background-color: black;
        padding: 20px 30px;
        text-decoration: none;
        color: white;
        /* transition: 屬性 轉換時間 延遲執行時間 速度 ; */
        transition: all 0.3s 0s ease;
    }
    .btn:hover{
        background-color: red;
        padding: 20px 40px;
        color: black;
    }
</style>
```



鐵人 19 animation 動畫

動畫需搭配 keyframes(動畫內容) 建立起始到結束的外型是如何轉變

ex : box1 的圓形會有動畫 動畫名稱: jia

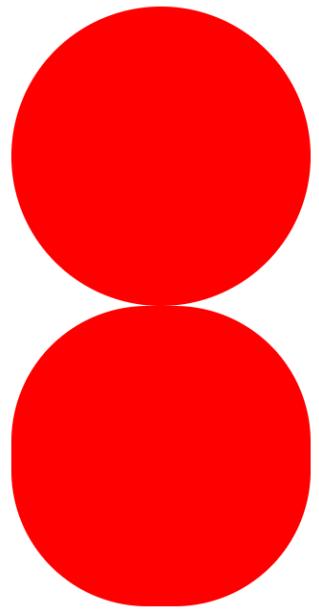
次數: infinite (重複) 方向 : alternate (來回播放)

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>

<style>
    .box{
        width: 200px;
        height: 200px;
        background-color: red;
        border-radius: 50%;
    }
    .box1{
        width: 200px;
        height: 200px;
        background-color: red;
        border-radius: 50%;
    }
    /* animation: 動畫名稱 播放時間 延遲執行時間(進入網站後) 速度 次數(infinite:重複) 方向(alternate:來回播放) 填充模式 播放狀態 ; */
    animation: jia .5s .5s ease infinite alternate;
}
/* 動畫內容 */
@keyframes jia{
    0%{border-radius:50%;/*初始狀態*/}
    100%{border-radius:0%;/* 結束 */}
}

</style>

</head>
<body>
    <div class="box"></div>
    <div class="box1"></div>
</body>
```

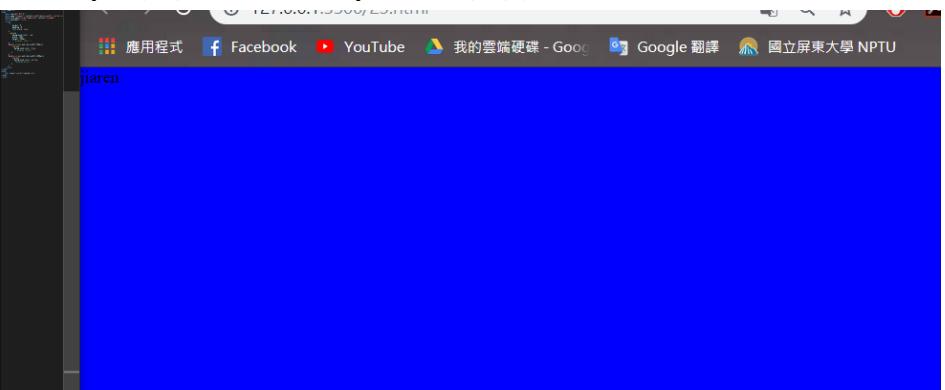


鐵人 20 媒體查詢 @media

max- 表示這個數字以下（包含）的都適用。

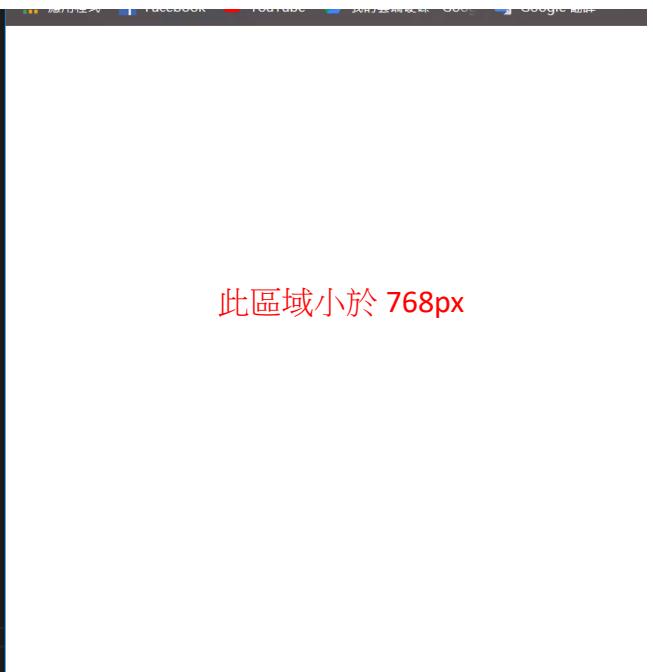
min- 表示這個數字以上（包含）的都適用。

1. 範例：768px 以下顯示紅色，768px-1200px 顯示藍色，1200px 以上顯示黃色。



```
<style>
  *{
    margin: 0;
    padding: 0;
    list-style: none;
  }
  .jiaren{
    background-color: red;
    width: 100%;
    height: 300px;
    /* display: none; */
  }
  @media screen and (min-width:768px){
    .jiaren{
      background-color: blue;
      /* display: block; */
    }
  }
  @media screen and (min-width:1200px){
    .jiaren{
      background-color: yellow;
      /*display:block;*/
    }
  }
</style>
</head>
<body>
  <div class="jiaren">jiaren</div>
</body>
```

2. 範例 2：將 768px 以下設為無色顯示



```
<title>Document</title>
<style>
  *{
    margin: 0;
    padding: 0;
    list-style: none;
  }
  .jiaren{
    background-color: red;
    width: 100%;
    height: 300px;
    display: none;      /*使768px以下無顏色*/
  }
  @media screen and (min-width:768px){
    .jiaren{
      background-color: blue;
      display: block;   /*因jiaren整體被設定成display:none
                           |所以在此重新設定display:block將其呼喚出來*/
    }
  }
  @media screen and (min-width:1200px){
    .jiaren{
      background-color: yellow;
      display: block;  /*因jiaren整體被設定成display:none
                           |所以在此重新設定display:block將其呼喚出來*/
    }
  }
</style>
</head>
<body>
  <div class="jiaren">jiaren</div>
</body>
```

此區域小於 768px

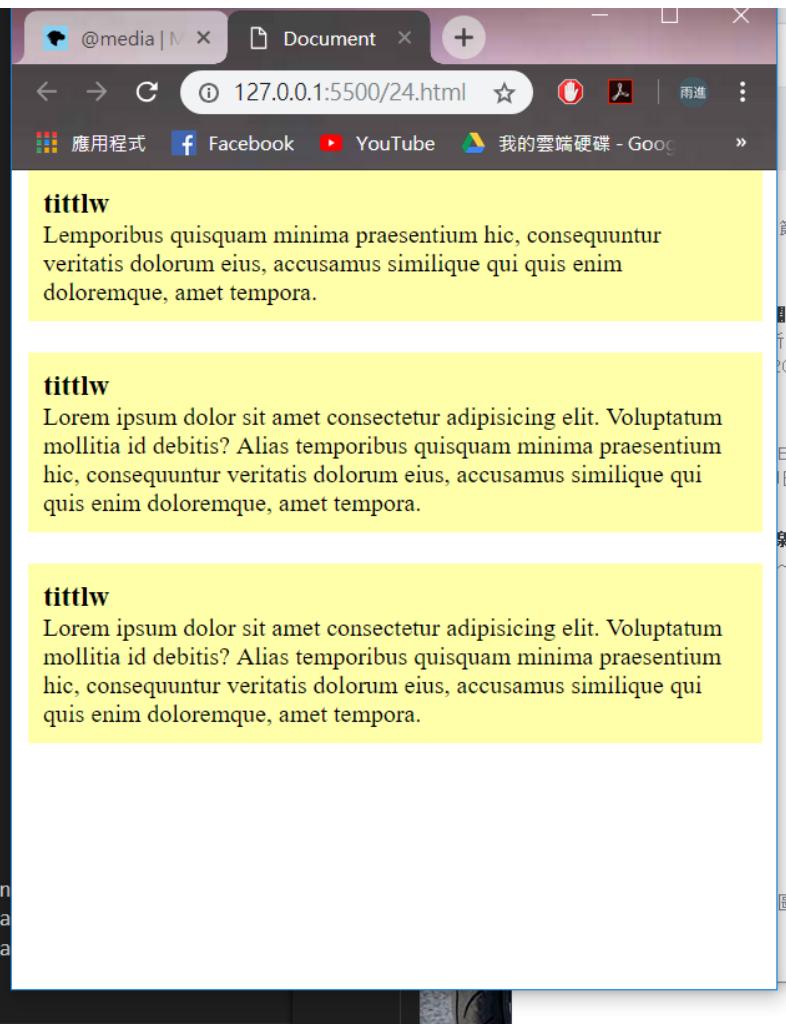
鐵人 21 RWD 入門 現今多以手機為優先開發其網頁配置

1. 範例：以手機優先設計，當超過 768px，則 flex

(其中設定了 **max-width** 這項最大寬度，限制如遇到超長螢幕的話有些區塊如果繼續維持寬度 100% 則會不美觀，因此限制了最大寬度，使其不超過最大寬度)

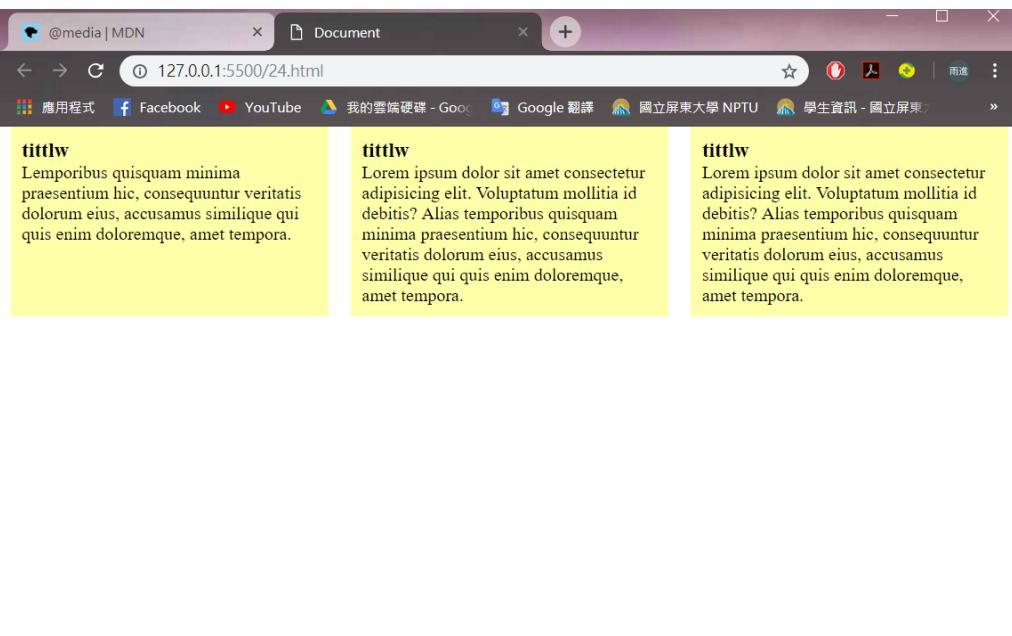
(手機顯示 (小於 768px) 螢幕畫面 ↓)

```
<style>
  *{
    margin: 0;
    padding: 0;
    list-style: none;
  }
  .wrap{
    width: 100%;
    max-width:1400px;
    margin: auto;
  }
  .item{
    margin: 0px 10px 20px;
    background-color: #ffa;
    padding:10px ;
  }
  @media screen and (min-width:768px){
    .wrap{
      display: flex;
    }
    .item{
      width:600px; /* 隨便給個超過 "視窗寬度/3" 的值
                     |即可撐開 (33.333% 也可以) */
    }
  }
</style>
</head>
<body>
  <div class="wrap">
    <div class="item"><h3>tittlw</h3>Lemporibus quisquam min
    <div class="item"><h3>tittlw</h3>Lorem ipsum dolor sit a
    <div class="item"><h3>tittlw</h3>Lorem ipsum dolor sit a
  </div>
</body>
```



(視窗顯示(超過 768px)螢幕畫面 ↓)

```
<style>
  *{
    margin: 0;
    padding: 0;
    list-style: none;
  }
  .wrap{
    width: 100%;
    max-width:1400px;
    margin: auto;
  }
  .item{
    margin: 0px 10px 20px;
    background-color: #ffa;
    padding:10px ;
  }
  @media screen and (min-width:768px){
    .wrap{
      display: flex;
    }
    .item{
      width:600px; /* 隨便給個超過 "視窗寬度/3" 的值
                     |即可撐開 (33.333% 也可以) */
    }
  }
</style>
</head>
<body>
  <div class="wrap">
    <div class="item"><h3>tittlw</h3>Lemporibus quisquam min
```



鐵人 22 RWD 試排版

範例

```
/* 原先因為.item{margin: 0 10px 40px;} 關係而撐開的圖片底部空間收起來 */
.service .item{
    margin:0 10px 0;
}
```

100%

title

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Dicta querat, maxime expedita doloremque vel quia aliquid soluta ipsa maiores laboriosam sit, quo perspicaci, cumque aspernatur animi blanditiis dignissimos ratione similique!

關於我

Lorem ipsum dolor sit amet consectetur adipisicing elit. Molestiae doloremque, excepturi natus molestias corrupti nobis asperiores saepe ratione corporis voluptate!

(code 放大版本)

```
}
```

```
@media screen and (min-width:768px){
    .item-group{
        display: flex;
    }
    .service .item{
        display: flex;
    }
    .service .item .pic, /*讓圖片與文字在桌機螢幕顯示時各占50%*/
    .service .item .text{ /*讓圖片與文字在桌機螢幕顯示時各占50%*/
        width: 50%;
        margin: 0;
    }
    .service .item .text{
        display: flex;
        flex-direction: column; /* 使flex下的內容-text與內文換行 */
        justify-content: center; /* 使flex下的內容垂直置中 */
        padding: 0 20px;
        box-sizing: border-box; /* 因上一行padding的關係會使內文篇一位置(由下方關於我標題可發現)
        因此使用box-sizing把寬度(width)值指定給border(邊框到邊框),
        padding和border 即會自動往內縮 --參閱鐵人11筆記*/
    }
    /* 將原先因為.item{margin: 0 10px 40px;} 關係而撐開的圖片底部空間收起來 */
    .service .item{
        margin:0 10px 0;
    }
}
```

鐵人 23 RWD 選單製作

RWD 收合選單製作：先設定一 **label**，用其控制 **input-checkbox** 開關

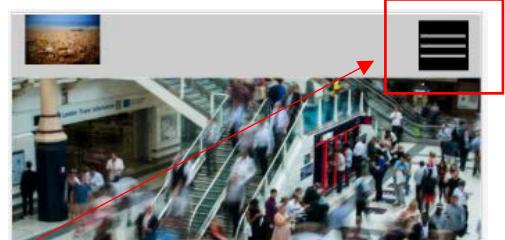
Html : ↓

```
<body>
    <!-- menu_control -->
    <input type="checkbox" name="" id="menu_control">

    <div class="header">
        <h1 class="logo">
            <a href="#"></a>
        </h1>
        <!--漢堡選單製作-->
        <label for="menu_control" class="menu_btn">
            <!--span放文字使用-->
            <span>選單</span>
        </label>
        <nav>
            <a href="#">link</a>
            <a href="#">link</a>
            <a href="#">link</a>
            <a href="#">link</a>
            <a href="#">link</a>
            <a href="#">link</a>
        </nav>
    </div>
```

Css : ↓

```
23 .menu_btn{
24     width: 40px;
25     height: 40px;
26     background-color: #000;
27     display: block;
28     position: absolute;
29     top:5px;
30     right: 10px;
31 }
32 .menu_btn span{
33     opacity: 0;
34     width: 1px;
35     height: 1px;
36     display: block;
37     overflow:hidden;
38 }
39 /* 假類，製作三條橫 "—" */
40 .menu_btn::before{
41     /* 中間 "—" */
42     content:'';
43     position: absolute;
44     height: 2px;
45     width: 36px;
46     background-color: #aaa;
47     left: 2px;
48     top: 0;
49     bottom:0;
50     margin: auto;
51     /* 利用box-shadow設定另外兩條橫槓
52     | x軸 y軸 模糊 色彩 */
53     box-shadow: 0px 8px 0px #aaa, /* 下 "—" */
54     0px -8px 0px #aaa; /* 上 "—" */
55 }
```



關於我

Lore ipsum dolor sit amet consectetur adipisicing elit. Molestiae doloremque, excepturi natus molestias corrupti nobis asperiores saepe ratione corporis voluptate!



title

Lore ipsum dolor, sit amet consectetur adipisicing elit. Dicta quaerat, maxime expedita doloremque vel quia aliquid soluta ipsa maiores laboriosam sit, quo perspicatis, cumque aspernatur animi blanditiis dignissimos ratione similique!

鐵人 24 Bootstrap 入門

1. .container-fluid (fluid 滿版寬)

2. Column 使用與規則 (佔欄數預設 12 欄，因此由補齊或是超過 12 欄控制換行)

手機 直向 .col-佔欄數

手機 橫向 .col-sm-佔欄數

平板 直向 .col-md-佔欄數

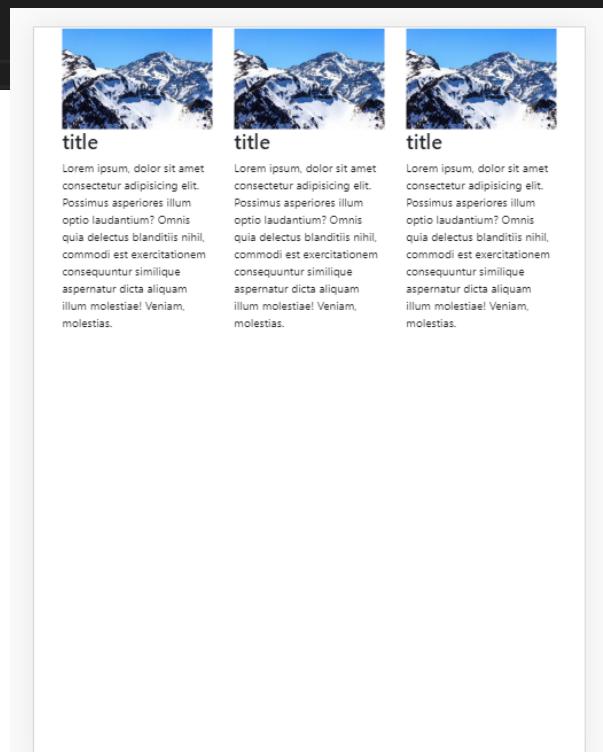
平板 橫向 .col-lg-佔欄數

電腦 .col-xl-佔欄數

使用範例 - 使手機使用時會跳行，平板以上使用時則並排

```
<style>
    .item .img-fluid{
        width: 100%;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="row"> <!-- col-12控制手機直向佔12欄 / col-md-4控制平板直向佔4欄 -->
            <div class="col-12 col-md-4">
                <div class="item"> <!-- 預設class="img-fluid"可使圖片滿版，但要注意其設定的是max-width:100% -->
                    
                    <h3>title</h3>
                    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Possimus asperiores illum optio laudantium? Omnis quia delectus blanditiis nihil, commodi est exercitationem consequuntur similique aspernatur dicta aliquam illum molestiae! Veniam, molestias.</p>
                </div>
            </div>
            <div class="col-12 col-md-4">
                <div class="item">
                    
                    <h3>title</h3>
                    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Possimus asperiores illum optio laudantium? Omnis quia delectus blanditiis nihil, commodi est exercitationem consequuntur similique aspernatur dicta aliquam illum molestiae! Veniam, molestias.</p>
                </div>
            </div>
            <div class="col-12 col-md-4">
                <div class="item">
                    
                    <h3>title</h3>
                    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Possimus asperiores illum optio laudantium? Omnis quia delectus blanditiis nihil, commodi est exercitationem consequuntur similique aspernatur dicta aliquam illum molestiae! Veniam, molestias.</p>
                </div>
            </div>
        </div>
    </div>
```

圖片 fluid 預設是 max-width:100% 意思是顯示大照片本身的大小
因此如需要圖片整個滿版，則需要自行修改成 width:100%即可。



鐵人 26 JQuery 入門

Google cdn :

<https://developers.google.com/speed/libraries/#jquery>

起手勢：先進入以上網址 → 選擇 3.X 版本，並下載下來即可套用 jQuery 函式庫

jQuery

3.x snippet:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

2.x snippet:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
```

1.x snippet:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
```

Ext Core
Hammer.JS
Indefinite Observable
jQuery
jQuery Mobile
jQuery UI
MooTools
Myanmar Tools
Prototype
script.aculo.us
Shaka Player
SPF

宣告 function

“當文件準備好了 ~ ”

```
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8 </head>
9 <body>
10
11
12 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
13 <script>
14   // 起手勢 //ready為jQuery的一種方法
15   $(document).ready(function(){
16
17     |  // ... 做的事
18
19   })
20 </script>
21 </body>
22 </html>
```

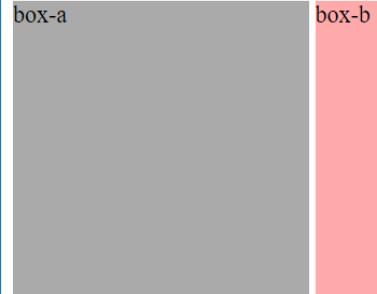
鐵人 27 JQuery 事件

Document API :

1. 範例 events -> mouse event

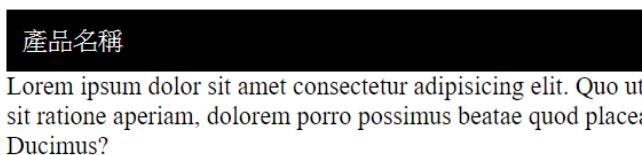
點擊 box-a 兩下 可使 box-b 隱藏或出現

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
    // 起手勢 //ready為jQuery的一種方法
    $(document).ready(function(){
        $(".box-a").dblclick(function(){ // click(點一下) / dblclick(點兩下)
            $(".box-b").toggle();           // 使用hide()只能隱藏(一次動作而已) ,
                                            // toggle則可重複
        })
    })
</script>
</body>
</html>
```



2. 點擊產品介紹時秀出內容

```
<style>
    .product-title {
        background-color: #000 ;
        color: #fff;
        padding: 10px;
        cursor:pointer; /* 使滑鼠游標變手 */
    }
    .content{
        display: none; /* 預設內容先不要看到，點擊後才秀出來 */
    }
</style>
</head>
<body>
    <div class="product-title">產品名稱</div>
    <div class="content">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo ut mollitia, accusantium hic consectetur sit ratione aperiam.
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script>
        $(document).ready(function(){
            $(".product-title").click(function(){
                $(".content").toggle();      //toggle會自動開關display:none
            })
        })
    </script>
</body>
```



範例：自製蓋板廣告，並且在點擊 close 按鈕後使其消失

```
<style>
/* 蓋板廣告區塊 */
.full-screen{
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    z-index: 10;
    background-color: □rgb(0,0,0,0.5); /* 使廣告底層呈現灰色透明 */
    /* opacity: 0.5; 不使用opacity原因是它會使整個區塊(連同廣告內文一起透明) */
    display: flex;
    justify-content: center;
    align-items: center;
}

/* 廣告內文 */
.full-screen .text{
    width: 90%;
    max-width: 500px;
    background: □#fff;
    padding: 20px;
    position: relative;
}

/* 自製close按鈕 */
.full-screen .text .btn-close{
    position: absolute;
    top: -5px;
    right: -5px;
    text-decoration: none;
    background-color: □red;
    border-radius: 50%;
}

</style>
```

The screenshot shows a dark gray full-screen overlay. In the center, there is a white rectangular box containing text. At the top right corner of this box, there is a small red circular button with a white 'X' inside, which is the close button mentioned in the code. The text inside the box is placeholder text (Lorem ipsum...) and is partially cut off at the bottom.

```
<p>Ullam accusantium eligendi facere, cumque dolorum molestiae ad dolorem minima id tempora esse i</p>
<p>Minus veniam aliquam accusantium porro pariatur in natus quisquam repudiandae magnam mollitia,</p>
<p>Minus in atque consectetur ea fugit necessitatibus officiis libero nostrum adipisci et aut expl</p>
<p>Inventore, voluptates? Nisi quis ex animi nobis. Amet obcaecati blanditiis impedit soluta odio,</p>

<div class="full-screen">
    <div class="text">
        <div>Lorem ipsum dolor sit amet consectetur adipisicing elit. Unde id doloremque vitae fugiat,</div>
        <a href="#" class="btn-close"> X </a>
    </div>
</div>

<!--<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>-->
<script src="js/jquery.min.js"></script>
<script>
    $(document).ready(function(){
        $(".btn-close").click(function(){
            $(".full-screen").toggle();
        })
    })
</script>
</body>
```

鐵人 28 JQuery CSS 控制

不同於鐵人 27 的方式，直接使用 `__Class →css` 控制器的方式執行效果（另外建立一 `css` 方法
Ex:預先多添加一個 `display:none` 之類的 `css` 屬性，爾後再 `JQuery` 中用函式來控制此 `css` 選取器裡的屬性，已達成效果。

Document API > css :

1. 添加一 `css` 選取器 `.full-screen.x{display: none;}` 來控制，使其被函式觸動、選取時，發生效果讓蓋板訊息消失。另外，其中 `.x` 以前不空格(因這是為了控制而產生，所以並非一般的父子層關係，所以不加空格) – **addClass & removeClass**

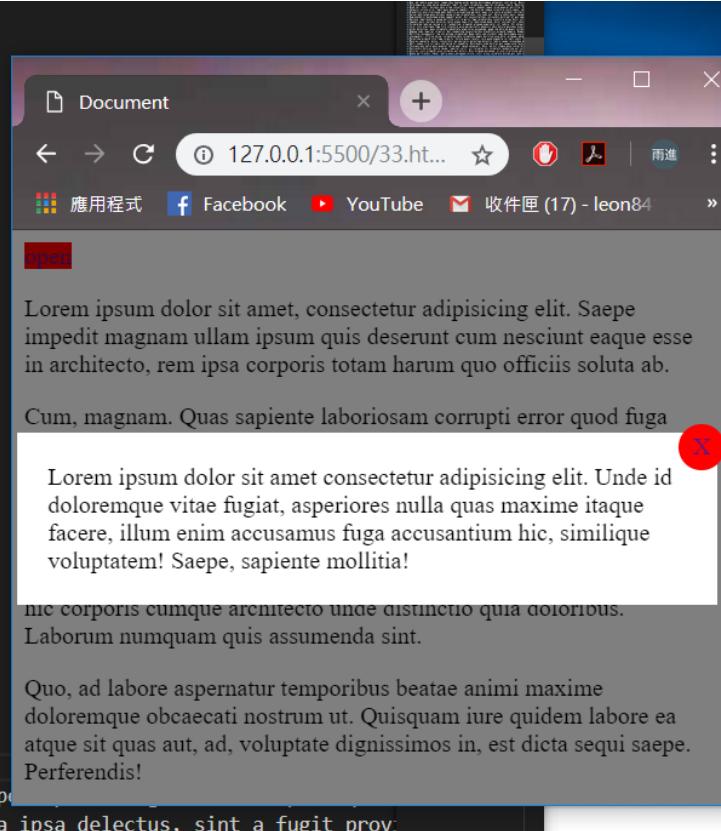
```
/*
 * 自製close按鈕 */
.full-screen .text .btn-close{
    position: absolute;
    width: 30px;
    height: 30px;
    line-height: 30px; /*使 X 垂直置中技巧*/
    text-align: center; /*使 X 水平置中*/
    top: -5px;
    right: -5px;
    text-decoration: none;
    background-color: ■ red;
    border-radius: 50%;
}

/* 建立一關閉蓋板訊息的 css控制器鑰匙 概念 底下用css控制器共控制他*/
.full-screen.x{
    display: none;
}

/* 自製open扭，將蓋板訊息叫回來 */
.btn-open{
    background: ■ red;
    text-decoration: none;
}

```

```
</style>
</head>
<body>
    <a href="#" class="btn-open">open <!-- 添加一個open按鈕 -->
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Saepe
    <p>Cum, magnam. Quas sapiente laboriosam corrupti error quod fuga ipsa delectus, sint a fugit prov.
```



```
<p>Inventore, voluptates? Nisi quis ex animi nobis. Amet obcaecati blanditiis impedit soluta odio,
```

```
<div class="full-screen">
    <div class="text">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Unde id doloremque vitae fugiat,
        <a href="#" class="btn-close"> X </a>
    </div>
</div>

<!-- <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script> -->
<script src="js/jquery.min.js"></script>
<script>
    $(document).ready(function(){
        $(".btn-close").click(function(){
            $(".full-screen").addClass("x"); //在此add增加.full-screen.x進去的意思
        })

        $(".btn-open").click(function(){
            $(".full-screen").removeClass("x");
            /*在此remove移除 原先增加的.full-screen.x
             *使蓋板廣告再次被啟動出來 */
        })
    })
</script>
```

2. 呈上範例的 CSS 控制原理，原先 full-screen 蓋滿了整個版面，但當觸及了關閉的 close 扭，即會動態的添加 .full-screen.x 進入。

open

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Saepe impedit magnam ullam ipsum quis deserunt cum nesciunt eaque esse in architecto, rem ipsa corporis totam harum quo officiis soluta ab.

Cum, magnam. Quas sapiente laboriosam corrupti error quod fuga ipsa delectus, sint a fugit provident cumque aut excepturi ab consequatur porro animi nihil saepe natus dicta? Vitae, dignissimos consequuntur.

Fugit non amet quos veniam illum cumque. Quod nobis accusantium magnam. Commodi dolor, laborum numquam voluptatem assumenda, hic corporis cumque architecto unde distinctio quia doloribus. Laborum numquam quis assumenda sint.

Quo, ad labore aspernatur temporibus beatae animi maxime doloremque obcaecati nostrum ut. Quisquam iure quidem labore ea atque si

Doloremque at ne voluptatem iusto quo?

Quod vel, nihil, d

X

```

▶ <p>...</p>
...
... ▶ <div class="full-screen">...</div> == $0
    <!-- <script
      src="https://ajax.googleapis.com/ajax/li
      y.min.js"></script> -->
      <script src="js/jquery.min.js"></script>
      <script>...</script>

```

蓋板訊消失後，添加了 X 進去

open

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Saepe impedit magnam ullam ipsum quis deserunt cum nesciunt eaque esse in architecto, rem ipsa corporis totam harum quo officiis soluta ab.

Cum, magnam. Quas sapiente laboriosam corrupti error quod fuga ipsa delectus, sint a fugit provident cumque aut excepturi ab consequatur porro animi nihil saepe natus dicta? Vitae, dignissimos consequuntur.

Fugit non amet quos veniam illum cumque. Quod nobis accusantium magnam. Commodi dolor, laborum numquam voluptatem assumenda, hic corporis cumque architecto unde distinctio quia doloribus. Laborum numquam quis assumenda sint.

Quo, ad labore aspernatur temporibus beatae animi maxime doloremque obcaecati nostrum ut. Quisquam iure quidem labore ea atque sit quas aut, ad, voluptate dignissimos in, est dicta sequi saepe. Perferendis!

Doloremque at nemo eaurum obcaecati fugit corrupti voluptates repellat voluptas libero. Maiores voluptas ad voluptatem iusto aliquid ducimus culpa possimus ut repudianda quis harum, nihil voluptatibus nisi deleniti beatae quo?

Quod vel, nihil, ducimus odio ullam consectetur, quasi commodi neque rerum quaerat mollitia! Deleniti eligendi

```

    > <p>...</p>
    ▶ <p>...</p>
    ...
... ▶ <div class="full-screen x">...</div> == $0
    <!-- <script
      src="https://ajax.googleapis.com/ajax/li
      y.min.js"></script> -->
      <script src="js/jquery.min.js"></script>
      <script>...</script>

```

3. 示範開與關在圖一按鈕發生 - toggleClass

open / close

Lorem ipsum dolor sit amet consectetur adipisicing elit. Libero ex aliquam, eum delectus amet fugit perferendis molestiae modi quo architecto. Iure iusto eaque ab molestiae impedit dolorem atque fugiat suscipit!

```

.controller{
  background-color: black;
  color: white;
  display: block;
  text-decoration: none;
  padding: 10px ;
  width: 300px;
}
.text{
  background-color: #ccc;
  width: 280px;
  padding: 20px;
}
.text.x{
  display: none;
}
</style>
</head>
<body>
  <a href="#" class="controller">open / close</a>
  <div class="text">
    Lorem ipsum dolor sit amet consectetur adipisicing elit
  </div>
  <script src="js/jquery.min.js"></script>
  <script>
    $(document).ready(function(){
      $(".controller").click(function(){
        $(".text").toggleClass("x");
        /* 使用toggleClass轉換，將自動分辨進行開關(開則觀，關則開) */
      })
    })
  </script>

```

鐵人 29 JQuery CSS 動畫

Document API > Effects

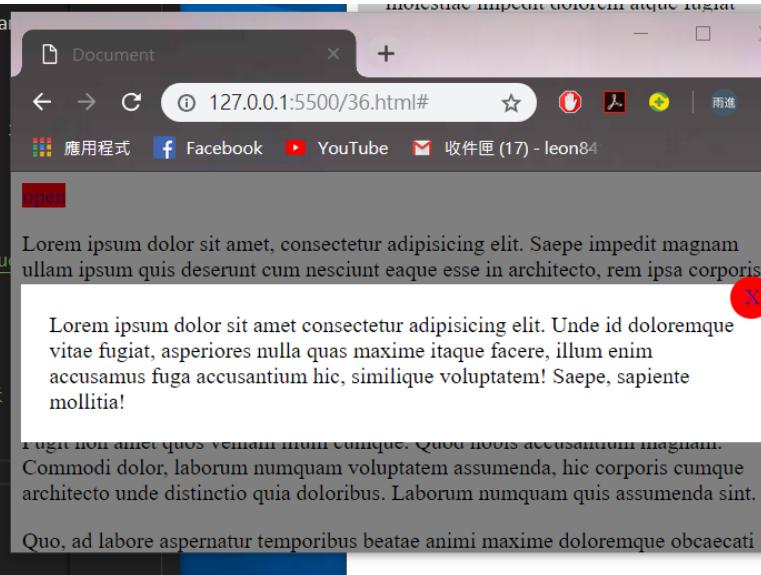
1. 微動畫的效果 - slideToggle / fadeToggle

```
</style>
</head>
<body>
    <a href="#" class="controller">open / close</a>
    <div class="text">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Libero ex aliquam, eum delectus amet fugit perferendis molestiae modi quo architecto. Iure iusto eaque ab molestiae impedit dolorem atque fugiat suscipit!
    </div>
    <script src="js/jquery.min.js"></script>
    <script>
        $(document).ready(function(){
            $(".controller").click(function(){
                $(".text").slideToggle();
            })
        })
    </script>
</body>
</html>
```



2. 微動畫的效果 - fadeOut

```
<p>Inventore, voluptates? Nisi quis ex animi nobis. Amet obcaecati bla</p>
<div class="full-screen">
    <div class="text">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Unde id doloremque vitae fugiat, asperiores nulla quas maxime itaque facere, illum enim accusamus fuga accusantium hic, similique voluptatem! Saepe, sapiente mollitia!
    </div>
</div>
<!--<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="js/jquery.min.js"></script>
<script>
    $(document).ready(function(){
        $(".btn-close").click(function(){
            $(".full-screen").fadeOut(); //fadeOut 效果使其緩慢消失
        })
    })
</script>
body>
html>
```



3. 呈上範例的動畫效果原理：原先 full-screen 蓋滿了整個版面，但當觸及了關閉的 close 扭，即會動態的修改蓋板訊息的不透明度 opacity，使其漸變至 display:none 為止。

am. Quas sapiente laboriosam corrupti error quod fuga ipsa delectus, sint a fugit provident cumque cum aut
o consequatur porro animi nihil saepe natus dicta? Vitae, dignissimos consequuntur.

net quis veniam illum cumque. Quod nobis accusantium magnam. Comodi dolor, laborum numquam
assumenda, hic corporis cumque architecto unde distinctio quia doloribus. Laborum numquam quis assumenda

ore aspernatur temporibus beatae animi maxime doloremque obcaecati nostrum ut. Quisquam iure quidem
que sit c
e at nem
il ducim
ihil, du
as, num
uta error fugit quos quaerat impedit, cum dolorem cumque inventore iusto sequi, ipsa asperiores neque
nis, nesciunt illo quam perspicatis dolore in iste iure eveniet quas obcaecati. Consequuntur!

xercitationem modi debitibus accusantium pariatur quas illo iusto provident veritatis veniam. Voluptatum rerum
trum, laboriosam soluta qui commodi consectetur veritatis exercitationem doloribus odit harum ullam eaque

蓋板訊息消失同時，漸變更改其不透明度，直至變成 display:none (消失為止)

```
> <p>...</p>
> <p>...</p>
> <p>...</p>
> <p>...</p>
... <div class="full-screen" == $0
  <div class="text">
    ...
      Lorem ipsum dolor sit amet consectetur
      adipisicing elit. Unde id doloremque vitae fugiat,
      asperiores nulla quas maxime itaque facere, illum enim
      accusamus fuga accusantium hic, similique voluptatem!
      Saepe, sapiente mollitia!
    ...
  </div>
<a href="#" class="btn-close"> X </a>
</div>
</div>
```

atur porro animi nulli saepe natus dicta? Vitae, dignissimos consequuntur.

veniam illum cumque. Quod nobis accusantium magnam. Comodi dolor, laborum numquam
da, hic corporis cumque architecto unde distinctio quia doloribus. Laborum numquam quis assumenda

natur temporibus beatae animi maxime doloremque obcaecati nostrum ut. Quisquam iure quidem
as aut, ad, voluptate dignissimos in, est dicta sequi saepe. Preferendis!

earum obcaecati fugit corrupti voluptates repellat voluptas libero. Maiores voluptas ad voluptatem
s culpa possimus ut repudiandae quis harum, nihil voluptatibus nisi deleniti beatae quos?

mus odio ullam consectetur, quasi commodi neque rerum quaerat mollitia! Deleniti eligendi aliquid
nam excepturi debitibus veniam soluta nam, quasi voluptate eos, nemo asperiores quo.

```
> <p>...</p>
> <p>...</p>
> <p>...</p>
> <p>...</p>
... <div class="full-screen" style="display: none;" == $0
  <div class="text">...</div>
</div>
<!-- <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script> -->
```