常見 CSS 選擇器、並以自己理解的方式去重新筆記：

* 在看選擇器之前，先來看一下 CSS 的結構：  
    
  沒錯，就是這麼簡單，就三件事情、三個關鍵名詞。

白話一點，選擇器就是你指定「網頁上的哪些元素」起來做樣式定義。所以你也可以把「選擇器」的語法理解為「取網頁上特定元素的篩選條件。」

ps. 下列表格簡寫:

* X / el <= element / selector
* attr <= attribute
* val <= value

**依元素 (tag / class / id)**

| **Syntax** | **Mean** | **Selector** |
| --- | --- | --- |
| el | 以 html tag選取 | p |
| . class | class 值為\_\_\_的元素 | .active |
| # id | id 值為\_\_\_的元素 | #slider |
| \* | 所有元素 |  |

**依父子/同輩關聯**

| **Syntax** | **Mean** | **Selector** |
| --- | --- | --- |
| elA, elA | A 跟 B | h1, h2 |
| elAelB | A 裡的所有 B (含子層 / 子子層 ...) | p span |
| elA > elB | A 的下一子層 的所有 B (不含子子層) | article > h2 |
| elA + elB | 緊接在 A 後面的 B (同輩) | h1 + p |
| elA ~ elB | 接 A 後面的所有 B（同輩） | h1 ~ p |

**依屬性與屬性值**

| **Syntax** | **Mean** | **Selector** |
| --- | --- | --- |
| X[ attr ] | 有特定屬性的元素 | [target] |
| X[ attr = val ] | 有\_\_\_屬性且屬性值為\_\_\_的 X | a[target=\_blank] |
| X[ attr ~= val ] | 有\_\_\_屬性且屬性值"包含"為\_\_\_的 X | article[data-type~="news"] |
| X[ attr ^= val ] | 有\_\_\_屬性且屬性值"開頭"為\_\_\_的 X | a[src^="https"] |
| X[ attr $= val ] | 有\_\_\_屬性且屬性值為\_\_\_"結尾"的X | a[src$=".pdf"] |
| X[ attr \*= val ] | 有\_\_\_屬性且屬性值"包含"\_\_\_的X | img[src\*="index"] |

**偽類**

| **Syntax** | **Mean** | **Selector** |
| --- | --- | --- |
| :link | 連結預設樣式 | a:link |
| :visited | 已點過的連結 | a:visited |
| :hover | 游標移到Ｘ時 | button:hover |
| :active | 點擊當下(按下去到放開之間) | a:active |
| :focus | X focus 狀態 | input:focus |

◎如果同時定義<a>的四種偽類，一般狀況應依此順序指定 a:link→a:visited→a:hover→a:active，不然會有樣式覆蓋的問題（除非特殊設計考量）。

**偽元素**

| **Syntax** | **Mean** | **Selector** |
| --- | --- | --- |
| X::before | 在 X 的內容最前面插入內容 | article::before |
| X::after | 在 X 的內容最後面插入內容 | article::after |

◎ 其預設 display 屬性為 inline-block  
◎ 一定要指定 content 屬性才會作用，不想放內文可設為空 content: ''

**偽元素－首行/首字/被選中的**

| **Syntax** | **Mean** | **Selector** |
| --- | --- | --- |
| X::first-letter | X 裡第一個字母 | p::first-letter |
| X::first-line | X 裡第一行 | p::first-line |
| X::selection | 被 user 選取的元素 |  |

◎ [關於::selection｜MDN](https://developer.mozilla.org/zh-CN/docs/Web/CSS/::selection)

**子元素歷遍**

| **Syntax** | **Mean** | **Selector** |
| --- | --- | --- |
| Ｘ:first-child | 同層(其父元素的子層)裡的第一個元素且為Ｘ | p:first-child |
| Ｘ:last-child | 同層(~)裡的最後一個元素且為Ｘ | p:last-child |
| Ｘ:nth-last-child(n) | 同層(~)裡的倒數第 N 個、且為Ｘ | p:nth-last-child(2) |
| Ｘ:nth-child( n ) | 同層(~)裡的第 N 個且為Ｘ | p:nth-child(2) |
| Ｘ:nth-child( 2n ) | 同層(~)裡的第偶數個、且為Ｘ | p:nth-child(2n) |
| Ｘ:nth-child( 2n+1 ) | 同層(~)裡的第奇數個、且為Ｘ | p:nth-child(2n+1) |
| Ｘ:only-child | 同層(~)裡只有一個，且為Ｘ | p:only-child |

◎ 是該層的「所有子元素」，若該層第Ｎ個非指定的元素，將不會生效。(備註1)

**特定子元素歷遍**

| **Syntax** | **Mean** | **Selector** |
| --- | --- | --- |
| X:first-of-type | 同層(~)裡的第一個Ｘ | p:first-of-type |
| X:last-of-type | 同層(~)裡的最後一個Ｘ | p:last-of-type |
| X:nth-last-of-type | 同層(~)裡的倒數第 N 個Ｘ | p:nth-last-of-type |
| X:nth-of-type( n ) | 同層(~)裡的第 N 個Ｘ | p:nth-of-type(2) |
| X:nth-of-type( 2n ) | 同層(~)裡的每一偶數個Ｘ | p:nth-of-type(2n) |
| X:nth-of-type( 2n+1 ) | 同層(~)裡的每一奇數個Ｘ | p:nth-of-type(2n+1) |
| X:only-type | 同層(~)裡只有Ｘ時的所有 Ｘ | p:only-type |

◎ 是該層的「第Ｎ個特定元素」(備註1)

**root / 反選偽類 not / empty**

| **Syntax** | **Mean** | **Selector** |
| --- | --- | --- |
| X:empty | 沒有子元素的 X | div:empty |
| :not( selector ) | 不符合特定 selector |  |
| :root | 文檔根元素 | :root |

◎ [關於 :root｜Amos](https://ithelp.ithome.com.tw/articles/10228111)  
◎ [關於 :not｜MDN](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:not) - 不能:not(:not / :before / :after)

**狀態**

| **Syntax** | **Mean** | **Selector** |
| --- | --- | --- |
| X:enabled | 可作用中的 | input:enabled |
| X:disabled | 禁用中的 | input:disabled |
| X:checked | 被勾選的 | input:checked |
| :target | 當前的 target (#) | :target |

備註:

1. -child & of-type  
   -child 為什麼寫「第Ｎ個、且為Ｘ元素」呢？  
   試想當你的同層裡有 <div>、 <p> 混排，此時你指定了 p:first-child，但在該層的第一個元素是 div 而不是 p 時，會選不到哦！  
   因為 "-child" 選擇器指定的第 Ｎ 個是指該層的「所有元素」，而不是所有的 <p>裡的第 N 個 <p>。  
   但我想要很直覺的 該層所有 <p> 裡的第 N 個 <p> 呢？  
   把 "-child" 改成 "of-type" 就可囉！  
   (更多參考：[偽類 child 和 of-type｜oxxo](https://www.oxxostudio.tw/articles/201405/css-selector.html))
2. 一個冒號(:before)、還是兩個冒號(::before)？
   * 兩個冒號(::before)是CSS3的寫法，為區分偽類與偽元素：
     + 偽元素 (pseudo element )：雖不是真正的元素，但有著和元素相同的特性
     + 偽類 (pseudo classes)：特殊狀態（例如:hover）
   * 但目前瀏覽器兼容 CSS2 寫法，偽元素不用兩個冒號(:before)也是可以的。  
     (除了::selection 語法是固定兩個冒號)