selector { declaration } => selector { property:value; }

declaration 內容: property:value;

1. inline 行內樣式

<h1 style="property:value;"> ==> inline

2. embeded style sheet 嵌入式

<head>

<style>

==> embeded style sheet

</style>

</head>

3. external style sheet

.css file

使用方式

<link></link>

HTML <link href="" rel="stylesheet"> 標籤 (tag) - 載入 CSS 樣式表

驗證網頁

<https://html5.validator.nu/>

**font**

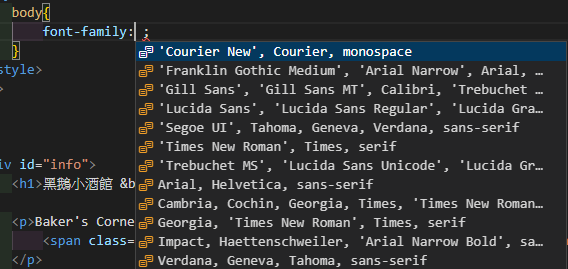
**CSS font 文字屬性表**

|  |  |
| --- | --- |
| **屬性** | **說明** |
| [font-size](https://www.wibibi.com/info.php?tid=93) | 設定文字大小，屬性值採用 "數字 + 單位" 呈現，數字越大通常表示字體越大。 |
| [font-weight](https://www.wibibi.com/info.php?tid=108) | 設定文字粗細，屬性值設為 bold 即為粗體字，取代傳統 [HTML](https://www.wibibi.com/c.php?cid=9) <b> tag 或 <strong> tag。 |
| [font-style](https://www.wibibi.com/info.php?tid=CSS_font-style_%E5%B1%AC%E6%80%A7) | 設定文字樣式，有兩種斜體字屬性值可用，分別為 italic 及 oblique，取代傳統 [HTML](https://www.wibibi.com/c.php?cid=9) <i> tag。 |
| [font-family](https://www.wibibi.com/info.php?tid=67) | 設定文字字型，可設定多個不同字型依序排列，取代傳統 [HTML](https://www.wibibi.com/c.php?cid=9) [face](https://www.wibibi.com/info.php?tid=149) tag。 |

Body

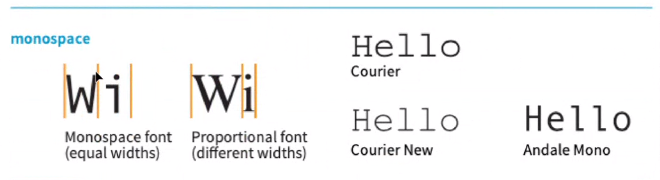
**CSS font-family 基本語法**

font-family:第一種字型,第二種字型,第三種字型;



* 內建的字型, monospace適合程式設計, 不適合網頁美化排版
* 老師通常使用最為為 sans-serif 或 serif

monospace => 基本字型 , 空格間距一致



sans-serif => 黑體字, 沒有修飾的

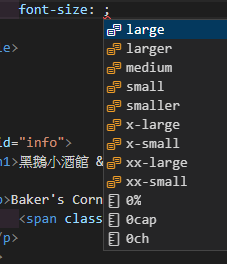
serif => 明體字, 有修飾



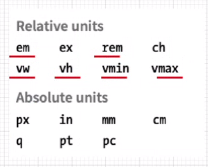


**Font-size 使用的單位:**

**系統預設是 medium**



尺寸單位:



rem : 相對root文件預設大小 => 會根據

px 的特性是屬於 **絕對數值**，他不受外圍的單位影響，只要固定是 12px，就會以 12 像素呈現。以下方的範例來說，他並不會受到外圍文字大小所影響，內部的文字大小還是依據 CSS 設定所呈現。

font-size: large

font-size:12px

font-size:13px

font-size:14px

font-size:15px

## 相對單位: EM

另一個常見的文字單位是 em， em 是相對的的數值單位，它會受到外圍的文字大小所影響，而 1em 即是 1 的文字大小， 1.5em 也就是 1.5 倍的文字大小。

以下方的範例可以發現，如果外圍是以 15px，內層的文字第一層 1em 等同於 15px，隨後的依比例放大。

font-size: 15px

font-size: 1em

font-size: 1.2em

font-size: 1.4em

接下來改成使用 em 來做實驗，外層的文字大小直接從 1.4em 開始，可以看到內層的 1em 文字大小等同於外層 1.4em，隨後的逐漸放大。如果在加入一層在內部，文字就會以 1.4 \* 1.4 的倍數再放大，這是 em 的相對比例單位的特性，是優點也是缺點，如果無法掌握就容易失控。

font-size: 1.4em

font-size: 1em

font-size: 1.2em

font-size: 1.4em

font-size: 1.4em

font-size: 1em

font-size: 1.4em

## REM

rem 也是相對的文字尺寸，和 em 使用方法接近，不同的是他僅相對於 root 層級的文字大小(網頁中的 html)。

以這個範例來說，此網站的文字大小是 16px，rem的尺寸就是以 16px 為基準，故下方的 1rem 會與 16px 相同尺寸。所以無論外層是否有其他文字大小，皆不會影響 rem 的尺寸，唯一會影響 rem 比例的只有 **html** 的文字大小(可以打開 開發者工具 試試看喔)。

font-size: 1.4em

font-size: 1rem

font-size: 16px

font-size: 1.2rem

font-size: 1.4rem

font-size: 1.4em

使用 rem 時，可以先了解瀏覽器的限制， rem 是不支援 IE8 以前的瀏覽器的喔。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature** | **Chrome** | **Firefox (Gecko)** | **Internet Explorer** | **Opera** | **Safari** |
| rem | 4 (532.3) | [3.6](https://www.hexschool.com/en-US/Firefox/Releases/3.6) (1.9.2) | 9.0 | 11.6 | 4.1 |

rem 和 em 都是 CSS 的「相對」單位。

兩者最大的差別在於

rem 是乘以「根」元素的 px 數值作為運算的結果，如果沒有設定「根」元素的數值，default 值就是 16px。

<html class="root">

root

<div class="child">

child

<div class="grandson">

grandson

</div>

</div>

</html>

html.root {

font-size: 20px;

}

.child {

font-size: 2rem; */\* 20px \* 2 = 40px \*/*

}

.grandson {

font-size: 2rem; */\* 20px \* 2 = 40px \*/*

}

em 是乘以「父」元素的 px 數值作為運算的結果，如果沒有設定「父」元素的數值，default 值就是 16px。

<div class="father">

father

<div class="child">

child

<div class="grandson">

grandson

</div>

</div>

</div>

.father {

font-size: 20px;

}

.child {

font-size: 2em; */\* 20px \* 2 = 40px \*/*

}

.grandson {

font-size: 2em; */\* 40px \* 2 = (20px \* 2) \* 2 = 80px \*/*

}

### Why

為什麼要使用相對單位？使用 px 這種絕對單位不適合嗎？  
rem 和 em 是為了解決響應式設計下，px 這種絕對尺寸不夠彈性的問題，使用相對單位的話，這樣只要在不同尺寸的螢幕上改變「根」或「父」元素的 px 數值，就可以改變所有子元素的單位數值，讓不同的螢幕都有好的呈現，而不用改動所有 CSS 的 px 數值大小。

### How

既然 rem 和 em 都是相對單位，那我們要選擇使用哪一個？  
「根」元素是所有「父」元素的最上層元素，所以秉持著先動粗後調細的原則，大多數情況下都會先在 root 的 html 元素上設定 font-size，在底下的所有子元素上設定 rem，如果有細節的樣式需要調整的話，才在父元素上設定 font-size，子元素加上 em 做更細微的調整。

以我工作上遇到的案例來說明的話  
rem 會用於 RWD 的排版

<h1>title</h1>

<p>paragraph</p>

html {

font-size: 12px;

@media (min-width: 768px) {

font-size: 16px ;

}

}

h1 {

margin-bottom: 2rem;

}

em 則是會用於調整 RWD 下 svg icon 或 logo 的細節

<sapn class="icon">

<svg width="1em" height="1em">

...

</svg>

</span>

.icon {

font-size: 16px;

@media (min-width: 768px) {

font-size: 20px ;

}

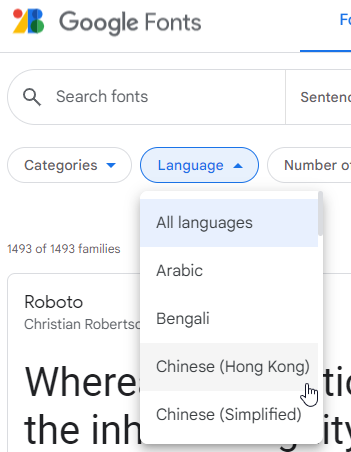
}

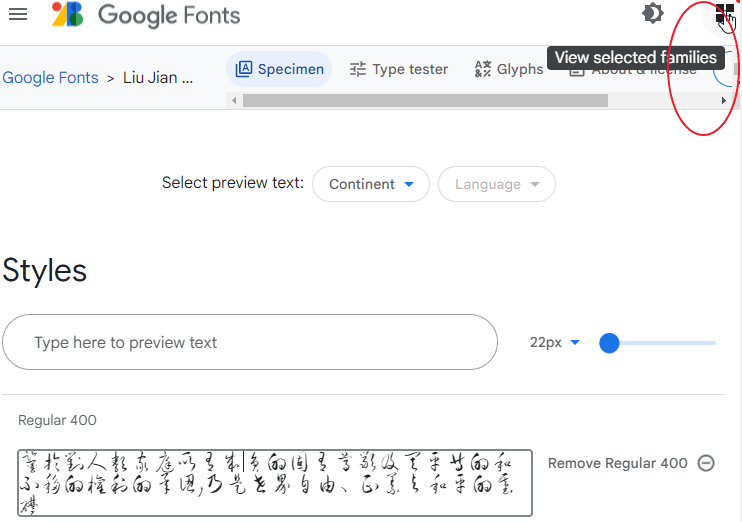
繼承來的優先順序是最小的

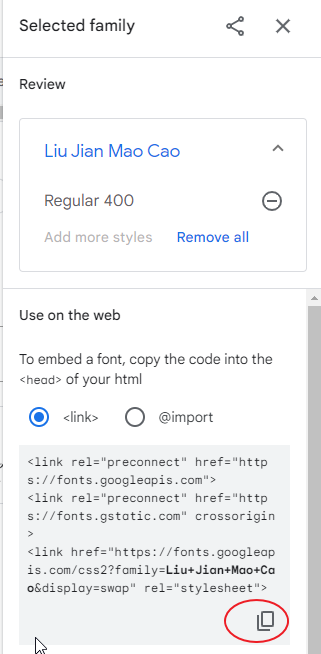
網路字型

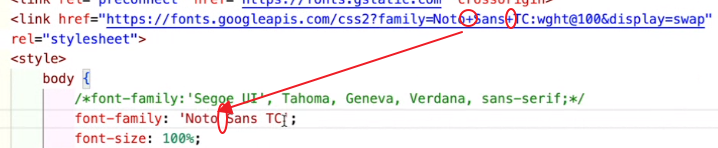
Google web font => 瀏覽器的字型

<https://fonts.google.com/>

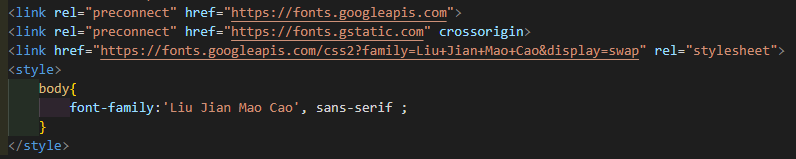




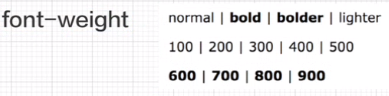




練習:

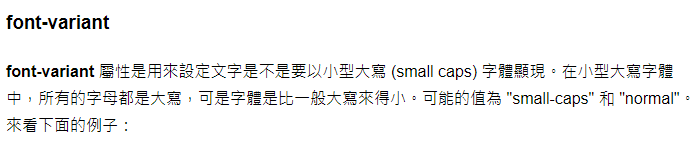


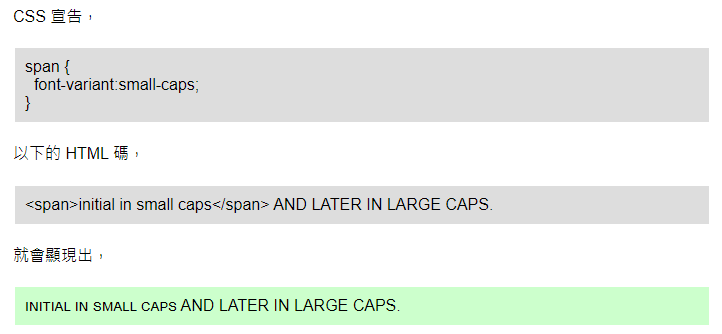
Font-weight





Italic/oblique 斜體

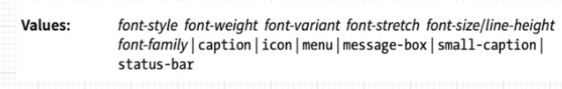


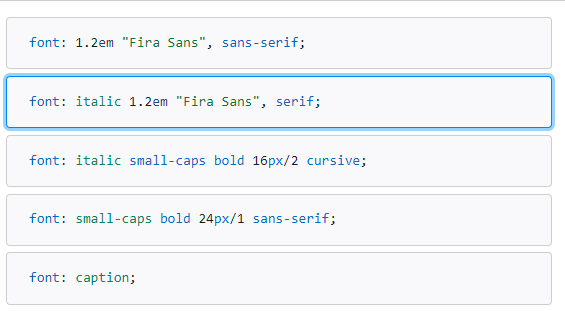


**~~font-stretch: 很多瀏覽器已不支援, 所以不用~~**



Font 縮行的順序





文字行距 line-height

數字後面沒有加單位, = em em 可省略

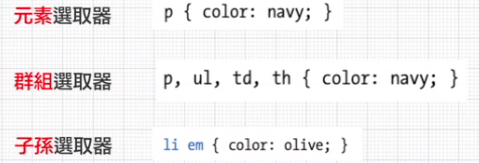


文字顏色: color

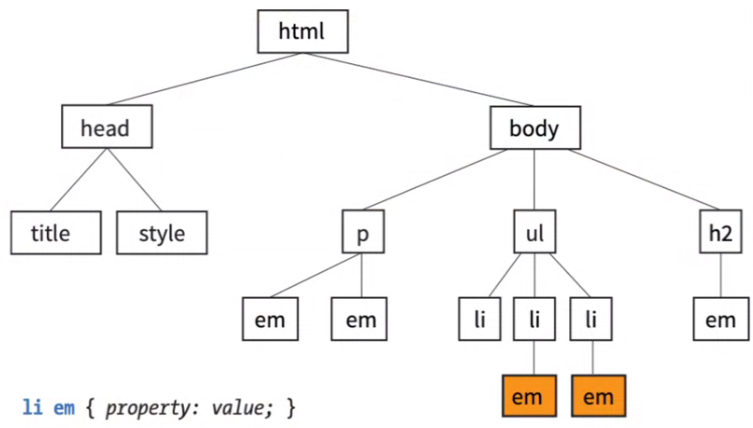


**群組選取器, 用逗號分格**

**子孫 (後裔) 選取器, 用空格分格**



li.class



**id 選取器 用 # ,即 id名稱前加 #**

**class 選取器 (類別選取器) 用 . 即 class名稱前加 .**

**注意: 後代用法時 li.class 中間要連在一起**