重新認識 Pixel、DPI / PPI 以及像素密度

2021-04-07

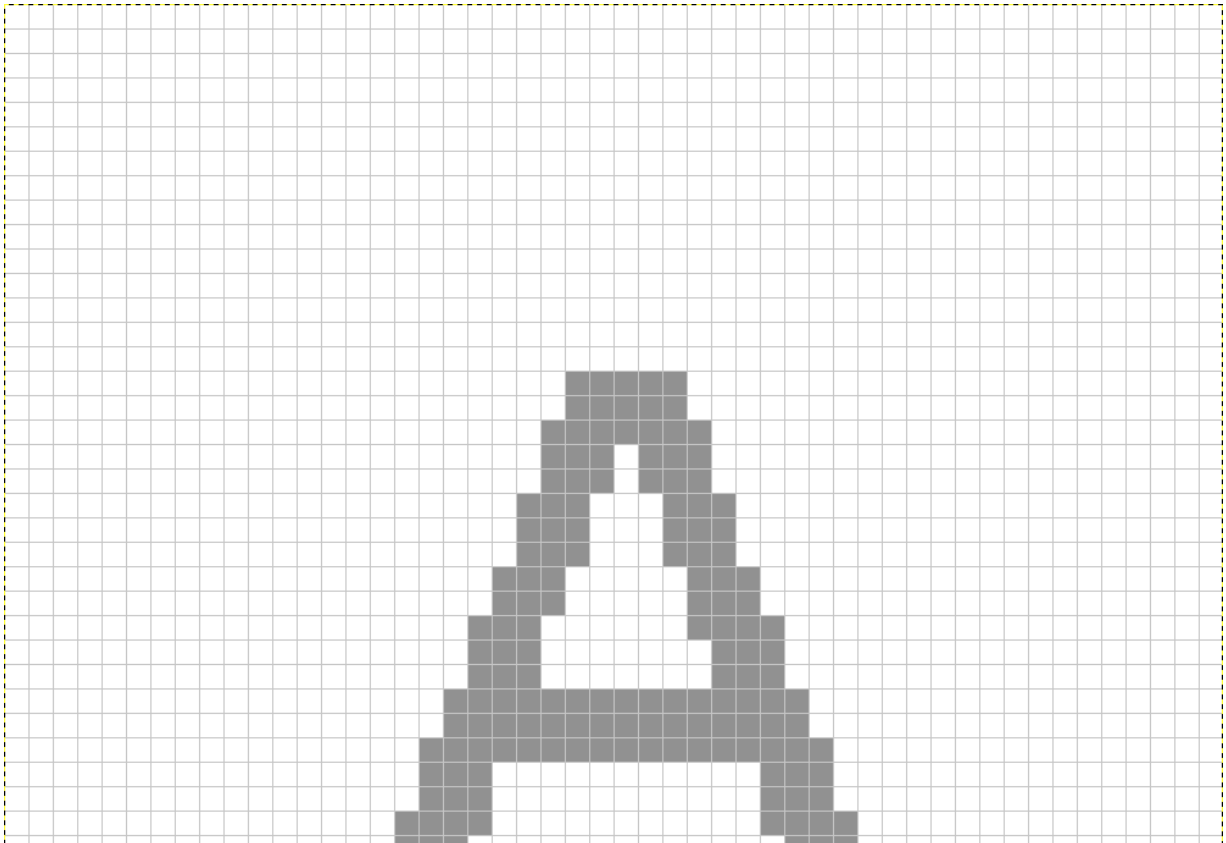
本文轉載自 Leon 的網誌

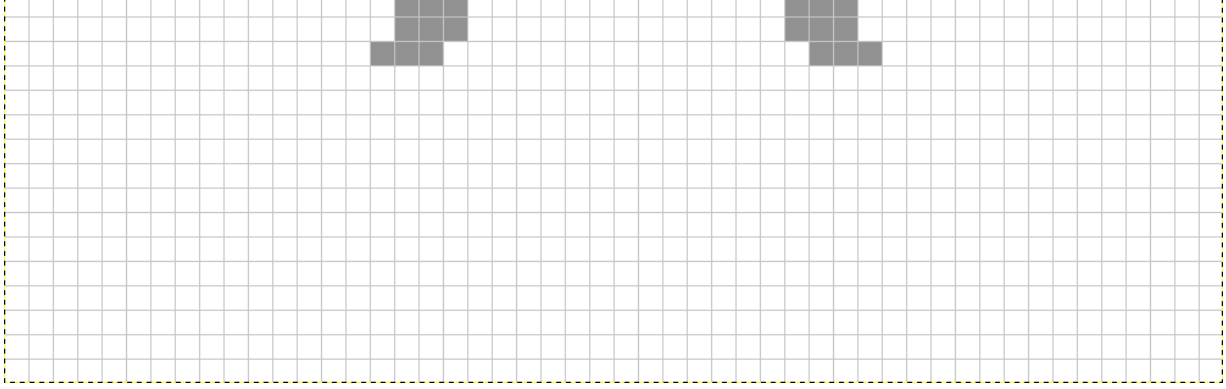
之前 Kenneth 曾經寫過〈React Native Pixel Ratio〉談到 pixel ratio 的觀念，而這篇則是更廣泛的從 pixel 為起點，談談什麼是 DPI、PPI、DRP 這幾個與 pixel 有關的詞彙，試圖釐清這些「解析度」與媒體（螢幕、紙張、感光元件）之間的混亂關係。

解析度？畫素？像素？Pixel？到底什麼是什麼？

如同物質組成基本的粒子——原子，組成數位影像的基本單位我們稱為畫素或像素，英文是 pixel，一般會簡寫為 px。

下面的示意圖是一張尺寸 50 * 50 px 的圖片，裡面的格線一格代表一個 pixel，可以很明確地看到裡面的「A」是由許多的小 pixel 組合而成：

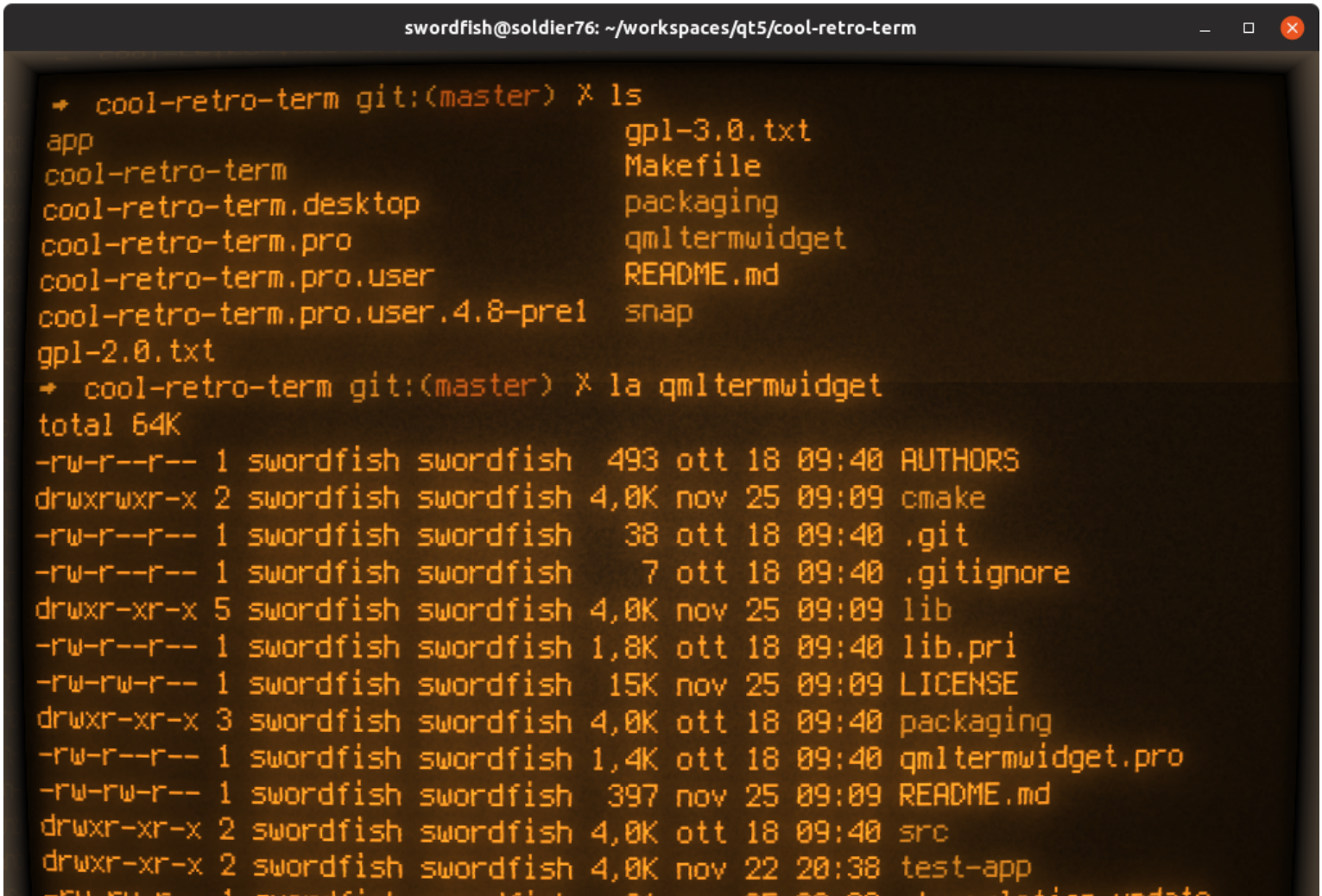


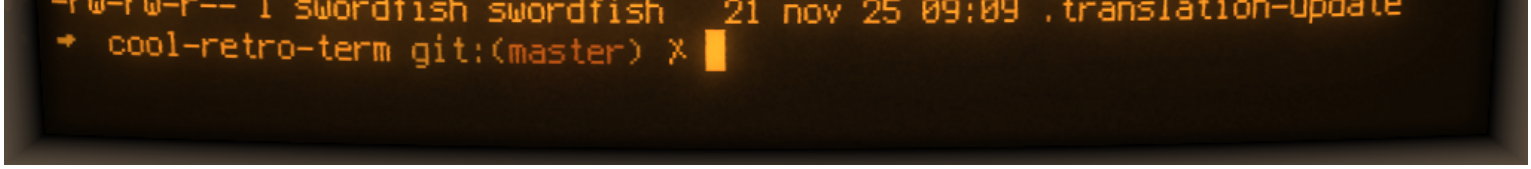


上面的 50 * 50 px 圖片是經過模擬放大數倍的效果，裡面的「 A 」也因此充滿鋸齒感，如果不刻意放大看，它應該是這樣的：

而這張 50 * 50 px 的圖片，我們稱它為 2,500 像素的圖片，因為裡面共有 50 * 50 = 2,500 個像素。

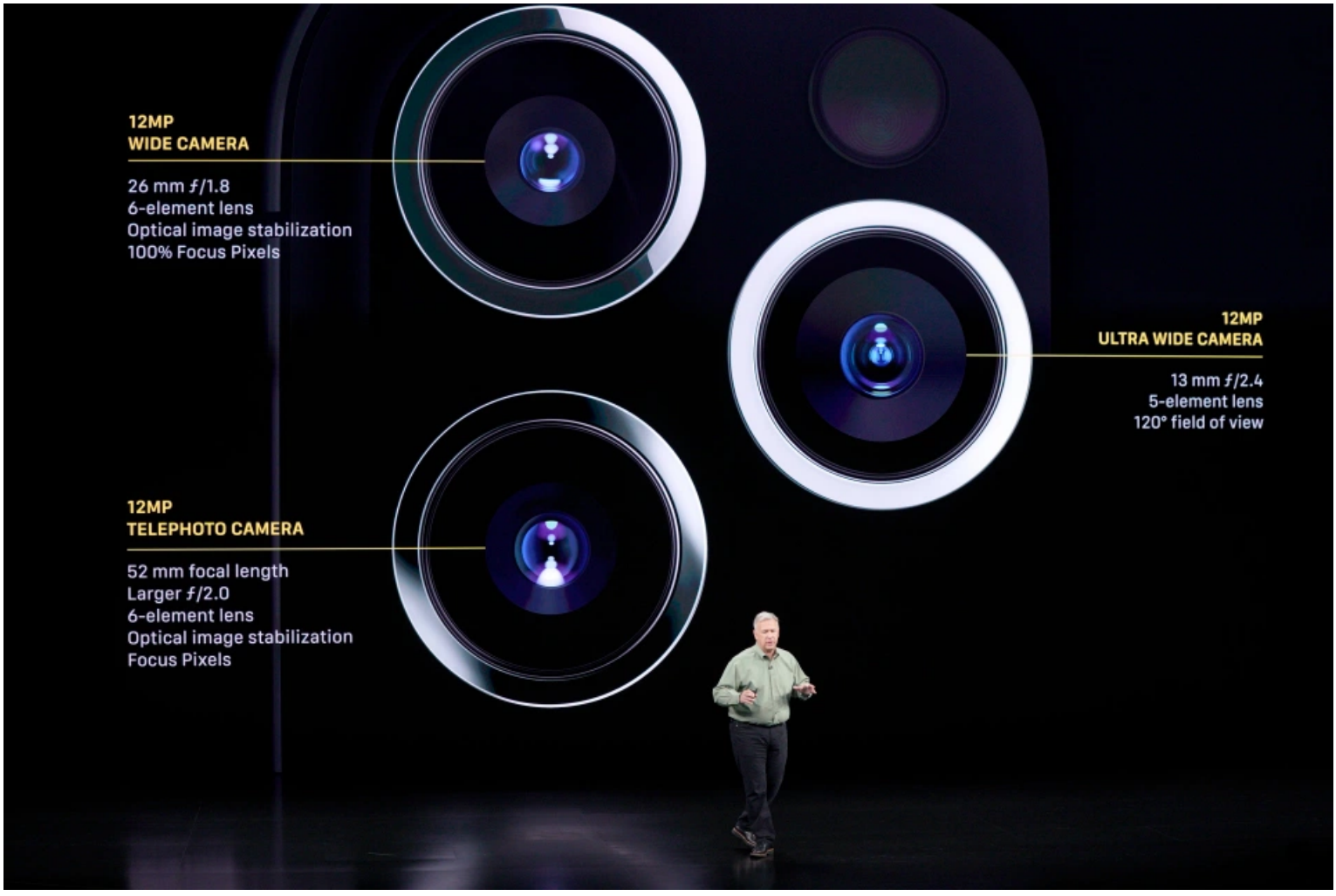
我們一般口語的「解析度」，指的是 50 * 50 px 這樣**寬×高**的表達方式，解析度一般會用來表示圖片、螢幕的寬、高畫素數，例如上面的 50 * 50 px 的圖片，或者是 1920 * 1080 px 的螢幕，如果把時間往回推三十年，當時的螢幕解析度只有 640 * 480 px 或更低，在當時的螢幕上，像素點是清晰可見的：





舊式單色 CRT 螢幕模擬圖
來源：cool-retro-term

而 2,500 像素，指的是像素的總數量，一般會用來表示相片或相機感光元件的像素數量，例如一張 2,500 像素的相片，或者一台百萬像素的相機，而「百萬像素」（mega pixel）又被縮寫成 MP，所以 iPhone 的相機是 12MP，意即 iPhone 的感光元件的有效面積大約是一千兩百萬像素，在不裁切或縮放的情況下它可以輸出 4,032 * 3,024 ≈ 1,200 萬像素的照片。



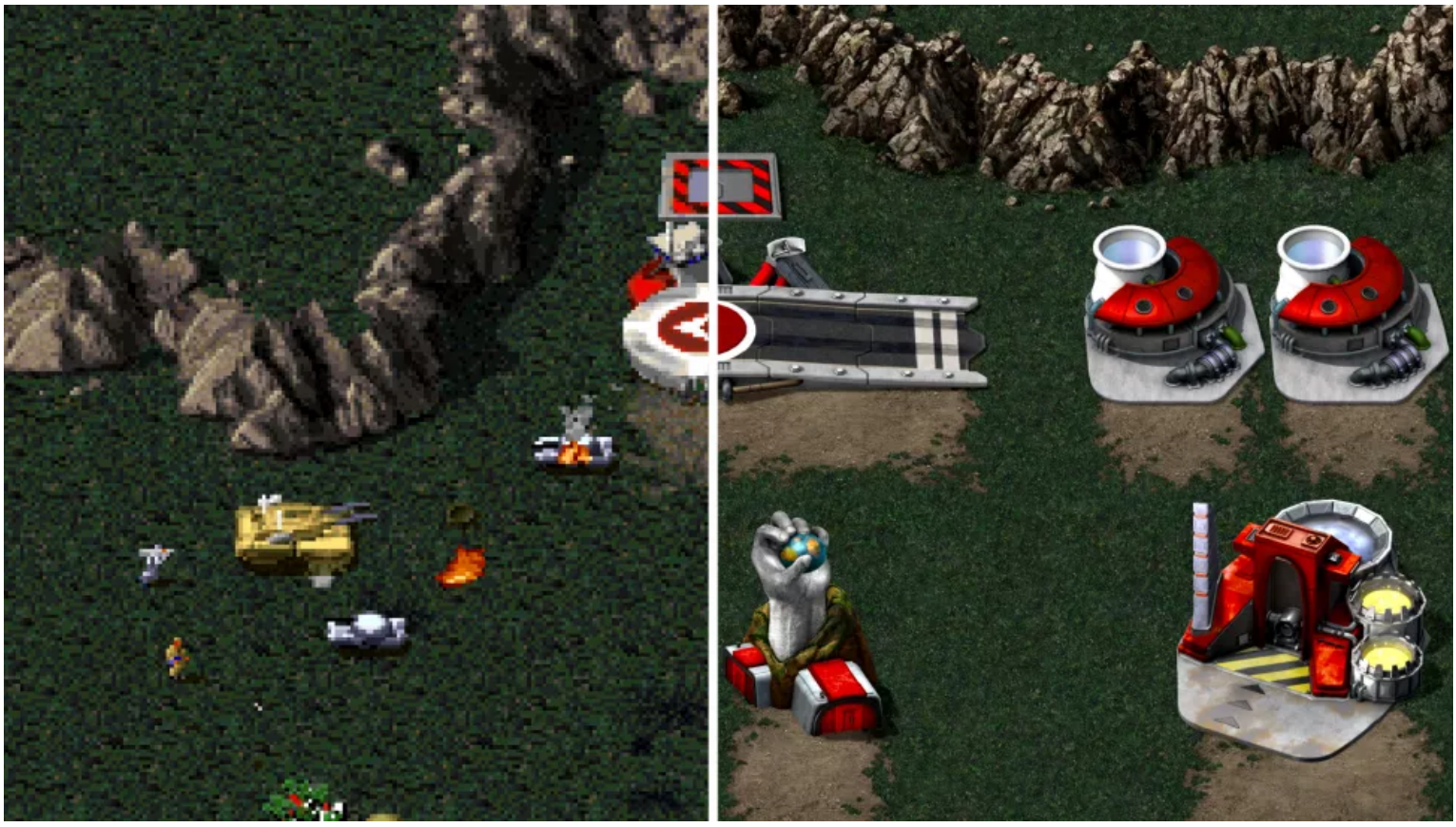
來源：Apple

「百萬像素」、「千萬像素」這類的表示方式大多用在相機上，而在非相機領域時我們比較慣用的是**寬×高**的表達方式。

像素與螢幕 PPI

像素是構成影像的基本單位，在一般人的認知，解析度越高通常也意味著畫質越高，同樣是 15 英吋大小的螢幕，一個的解析度是超古老的 640 * 480 px，另一個是 1,920 *

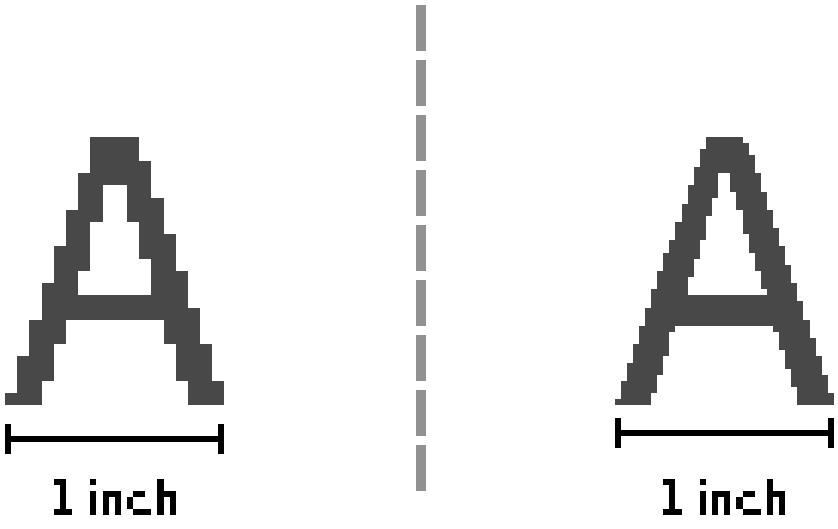
1,440 px，兩者的 X、Y 方向的像素數各差了三倍，這之間的差異可以用不同年代的同一款遊戲的畫面來感受：



左：1995 年《終極動員令》、右：2020 年《終極動員令》重製典藏版
來源：EA

透過上面的示意圖可以看到，在相同尺寸的螢幕上，因為解析度的差異，帶給我們視覺上畫面精細度的差異感受。

同樣 15 吋的兩個螢幕，卻可以有著差異數倍的解析度，真實世界的 1 英吋，對應在這兩個 15 吋螢幕上會得到不同的像素數，我們把這件事簡化成下面的示意圖：



上面的示意圖中，左右代表兩個相同尺寸，但解析度不同的兩個螢幕，在左邊低解析度

的螢幕上，1 英吋相當於 18 px；而右邊的高解析度螢幕上，1 英吋則是相當於 36 px，由此我們可以得知 pixel 具有相對單位的性質。

像上面這樣以「1 英吋內有多少 pixel」的表示法我們用於表示**像素密度**，英文是 pixels per inch，簡稱 PPI，上面的圖分別是 18 PPI 的螢幕與 36 PPI 的螢幕，當然生活中的螢幕不可能有如此低的 PPI，除非你活在麥塊世界：

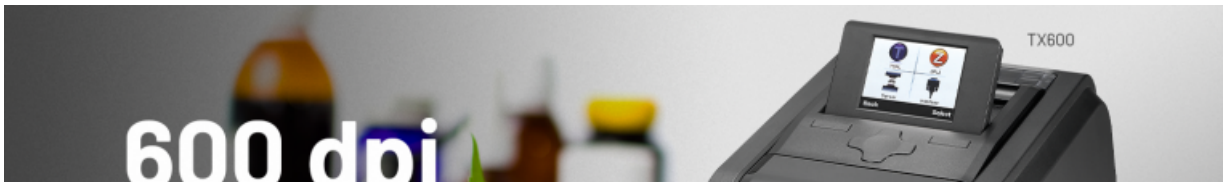
來源：TSMC

而自從蘋果開始行銷 Retina display 後，PPI 也開始成各家手機競逐的規格之一，根據蘋果的公式，只要手機螢幕大於 300 PPI，人眼在正常使用距離下就感受不到顆粒感，但對爾等 app 開發者來說，app 內的圖檔素材也必須跟上這麼高的解析度才能得到最佳的觀看效果，關於 PPI 與 app 的關係，後面會再提到，在此先點到為止，先談談常常容易和 PPI 搞混的「DPI」。

像素與列印 DPI

DPI 全文為 dots per inch，和 pixels per inch 只差在最前面的「dot」，我們知道 pixel 是像素，那 dot 是什麼？

對印表機來說，DPI 的 dot 指的是「墨點」，一台規格為 600 DPI 的印表機表示它最多能在 1 英吋的尺度內印出 600 個間距相當的墨點：

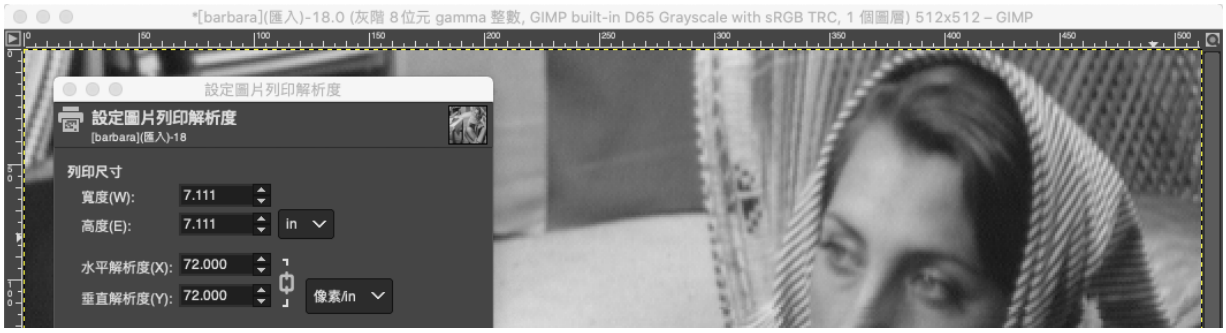


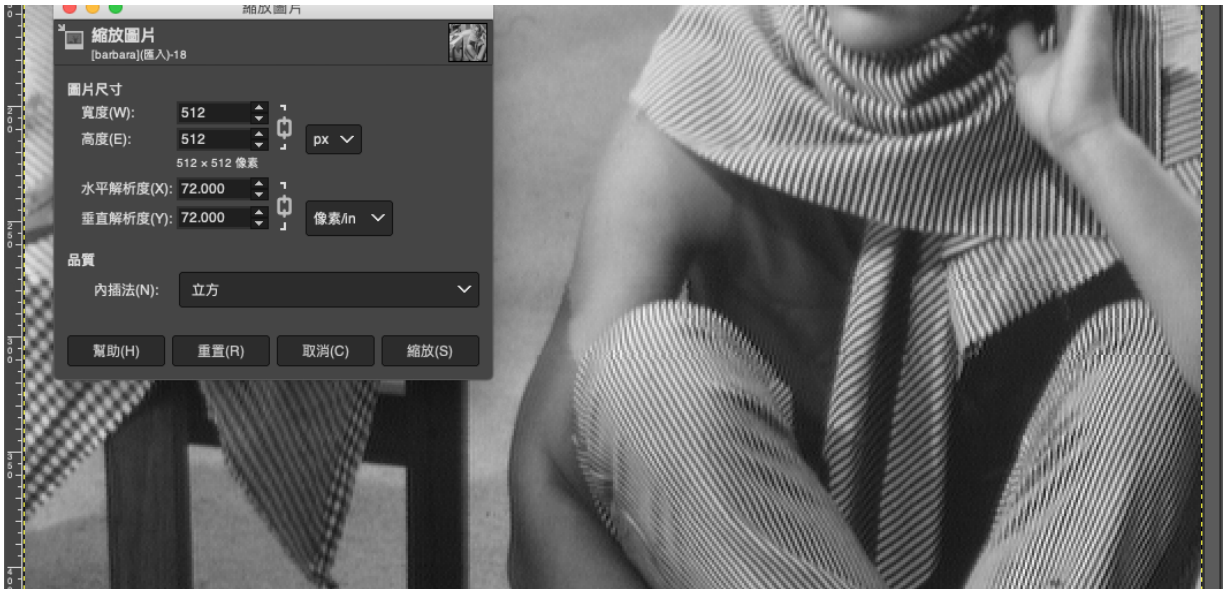
來源：鼎翰科技（TSC）

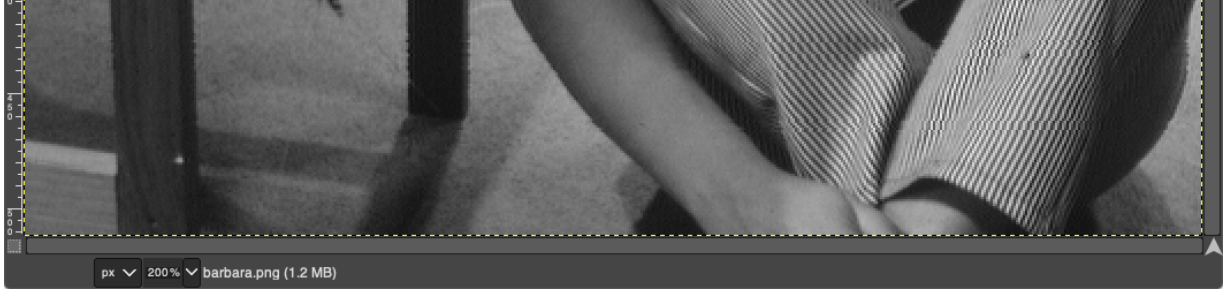
印表機的 DPI 是由它的機械結構決定的，以噴墨印表機來說，是由 X 與 Y 方向的兩組步進馬達的解析度來定義這台印表機的解析度，X 方向的步進馬達負責驅動噴墨頭；Y 方向的步進馬達負責帶動紙張前進，這也是之所以我們看普通印表機的規格表上常常是寫成 600 * 600 DPI 的原因，前後兩個數字分別代表 X 方向與 Y 方向的 DPI。

對於非噴墨型的印表機，例如上圖的熱感式標籤列印機或雷射印表機，他們雖然不存在 X 軸的步進馬達，但他們的成像元件也是有密度概念的，上圖為 TSC TX600，它的成像元件密度就有達到一英吋 600 個點，密度越高代表越高的製程能力，價錢當然也是越貴。而雷射印表機則是由 Galvano 反射鏡（振鏡）的轉動解析度來決定 X 方向的 DPI，不同形式的印表機成像原理不同，但輸出密度這個概念是通用的。

然而 DPI 的混亂之處在於這個「dot」在不同的媒體有著不同的定義，對印表機來說，dot 是墨點，但圖檔內卻也有著 DPI 的設定：







上面是一張 512 * 512 px 的圖片，它的 DPI 是 72 px/inch，這裡的 DPI 的「dot」指的是「pixel」，而不是「墨點」，但一旦送至印表機，DPI 的「dot」又會被解讀成「墨點」，因此以 X 方向來說，我們可以換算出這張影像的物理尺寸：

```
512 px / (72 px/inch) = 7.111 inch = 18.062 cm
```

把這樣的規格拿去列印，得到的是一張寬度為 18.062 公分的印刷品，因為圖檔的 DPI 是設定成 72 dot/inch，印表機因此會很聽話的在 1 英吋的尺度內印出 72 個墨點，而 X 方向總共有 512 個「點」，這裡的「點」既是 pixel 也是墨點，因此就會得到寬度 512 / 72 = 7.111 英吋的圖片，相當於 18.062 公分。

如果把這張圖檔的 DPI 改為 150，那 X 方向的輸出尺寸計算如下：

```
512 px / (150 px/inch) = 3.413 inch = 8.670 cm
```

比較一下兩種不同 DPI 設定下的輸出結果：

左邊為 72 DPI 輸出、右邊為 150 DPI 輸出

可以看到，512 * 512 px 的影像，在 72 DPI 下的輸出儘管尺寸較大，但也是較模糊的，而改用 150 DPI 輸出，結果銳利許多（因為更密集了），而在平面設計或排版界，往往是紙張與印表機的解析度先決，也就是說我的傳單或小冊已經注定是 A4，而輸出設備的解析度是 400 * 400 DPI，那麼我想印一頁 A4 寬度滿版的影像，我必須讓我的圖檔能夠撐到這麼大而不模糊，用上面的公式反向推算應有的 X 方向像素：

$$X \text{ px} / (400 \text{ px/inch}) = 8.268 \text{ inch} = 21 \text{ cm}$$

上面的 X 會等於 3,307 px，假設同樣是正方形的圖片，那 Y 方向也會是 3,307 px，因此這張要放到 A4 寬度滿版的相片，會是 3,307 * 3,307 ≈ 10 MP，需要千萬畫素的相機才能拍出。

隨著紙張的變大，一張正方形圖像想印成寬度滿版所需要的畫素也隨之變高，以 A3 來說，要維持 400 DPI 的解析度，大約需要兩千萬畫素的圖檔；A2 則需要約四千萬畫素圖檔；A1 則需要約九千萬畫素的圖檔，然而現在主流的專業全片幅相機，也大多是兩千萬畫素，因此實際上的做法會是把圖檔的 DPI 降低，只要能騙過人眼就好。

要騙過人眼，得利用觀看距離，對 A4 來說，一般的觀看距離是 30 公分，在這樣的距離下，大約只要 400 DPI 的列印解析度，就能讓肉眼感受不到墨點，對更大尺寸的媒體來說，觀看距離也會相對的拉遠，因此並不需要強求維持 400 DPI 的列印解析度，也能達到看不出墨點的品質，例如廣告牆：

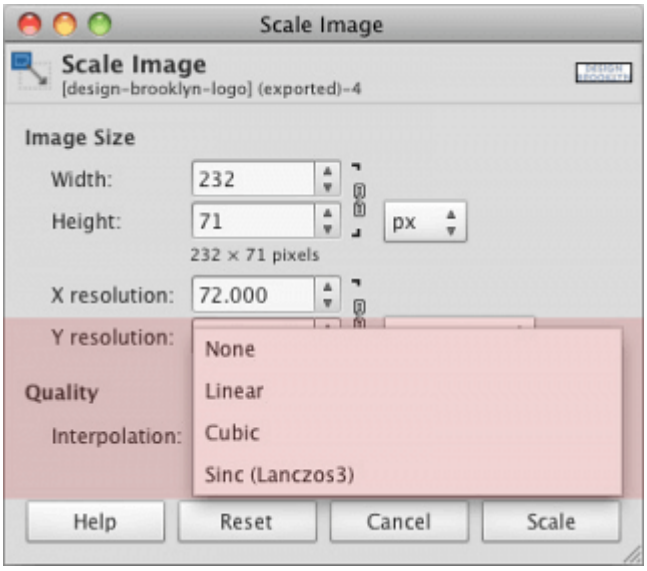
來源：小草

在這樣大面積的媒材上要以 400 DPI 輸出，顯然是不現實的，不僅電腦處理不了這麼大的圖檔，也沒有實際上的益處，廣告牆的觀看距離遠至數十公尺，在這樣的距離下墨點的密度顯然無關緊要，因此實務上只要 150 DPI 就足以滿足這麼大面積、長距離觀看特性的媒體。

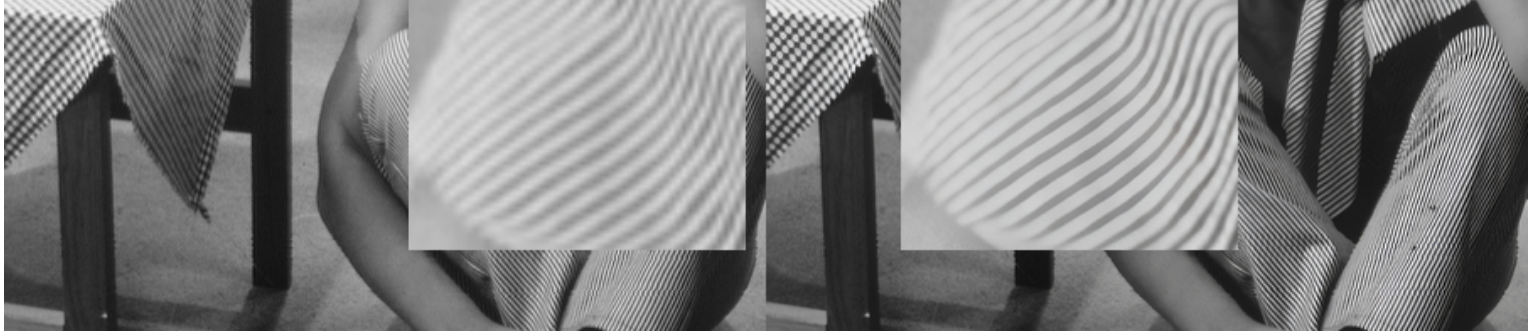
由上面我們可以得知，不論 DPI 怎麼設，圖檔的原始寬高像素是不會受 DPI 影響的，DPI 僅是用於決定輸出時要叫印表機把「點」印的多密集的一項參數。

話再說回來，那張 512 * 512 px 的圖片，我硬是要印成 A4 滿版，卻又想獲得好的畫質，那應該怎麼做？

我們可以用手動放大影像的方式把影像放大：



以 GIMP 來說，它就提供了四種放大演算法，但這樣基於補插點的放大只會得到變大也變模糊的影像，剛好近幾年 AI 演算法爆棚，我們可以利用 AI 演算法的工具把圖像放大，而不損失畫質：



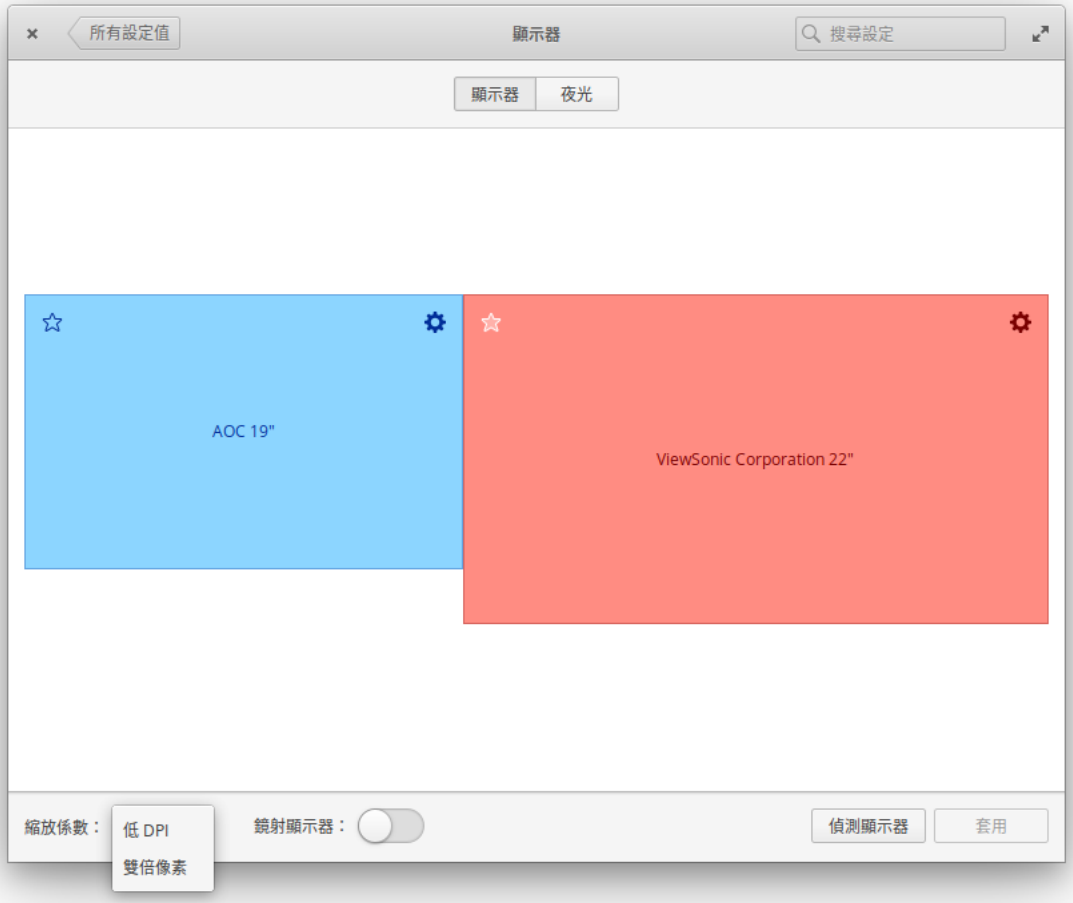
左邊為 512 * 512 px 原圖、右邊為放大至 3,000 * 3,000 px 的圖

上面左邊是原圖，右邊則是用 Smart Upscaler 放大的結果，比較頭巾的細節，很明顯的在黑科技的加持下，圖像放大了反而畫質還更好了，再用肉眼比較桌巾的格紋，也可以看到右邊的格紋更加銳利（但格紋形狀卻好像不那麼「方正」了，有一好沒兩好）。

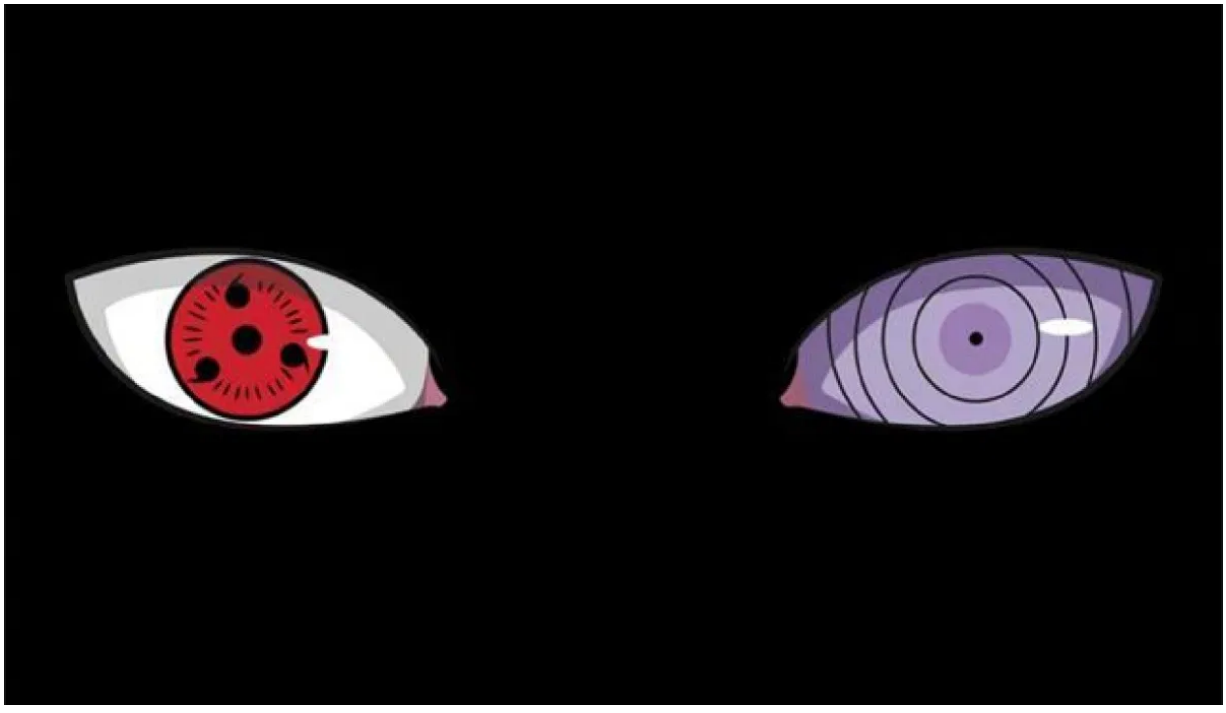
DPI 與 PPI

綜合以上，PPI 與 DPI 都用來表示「1 英吋內點的數量」，也就是密度的概念，不同的僅在於 PPI 的「點」很明確的定義為 pixel，因此 PPI 永遠都只與螢幕相關，而 DPI 的「點」有時候是 pixel，有時候是墨點，因此有時候會用於表示螢幕的點密度，有時候又會用於表示印表機的點密度，甚至 3D 印表機也是用 DPI 表示機台的解析度，因為 3D 印表機也是由步進馬達驅動，只不過多了 Z 軸的機構。

PPI / DPI 兩者混用的狀況時有所見，相當容易令人誤會，因而近代的作業系統也大多屏棄了 DPI 的字眼，改以更為通俗的方式讓用戶調整解析度與 UI 元件比例：



PPI / DPI 兩者的概念也大多是共通的，人的眼力有所極限，能否看到「點」與距離有關，想要騙過大腦讓它認為眼前的螢幕或紙張是高品質的，那就讓螢幕 / 紙張離眼睛遠一點，或者讓螢幕 / 紙張的 PPI / DPI 高一點，這個定律是適用於所有人的，除非他有瞳術那另當別論……。



來源：火影忍者

另一個涵義模糊的詞彙是「解析度」，「解析度」是個更廣泛的概念，不論是 512 * 512 px、300 PPI、150 DPI，他們都可以被「解析度」稱之，解析度的英文 resolution 意義也是同樣的粗略，因此看到解析度最好還是看它後面接的是什麼才能正確地理解前後文脈絡。

像素、DPR 與更多的單位

再把主題拉回像素與 PPI 的世界，前面提到過自從 Retina display 問世以來，手機的 PPI 有著飛躍式的成長，像軍備競賽一樣，隨便都破 400 PPI，下面是幾款 iPhone 的面板解析度與 PPI 表：

裝置	螢幕像素	螢幕對角線尺寸	PPI
iPhone 3	320 * 480 px	3.5 inch	163
iPhone 4	640 * 960 px	3.5 inch	326
iPhone SE2	750 * 1134 px	4.7 inch	326
iPhone 12 Pro Max	1284 * 2778 px	6.68 inch	458

上表中的 iPhon 3 是末代非 Retina display 的型號，從 iPhone 4 起，開啟了 Retina display 的時代，扣掉上面兩排已經停產的型號，看 iPhone SE2 與 iPhone 12 Pro Max 這兩款正在線上的機種，螢幕寬度的像素分別是 750 px 與 1,284 px，這樣巨大的寬度像素對於 app 或 RWD web 佈局是有問題的，試想 1,284 px 寬的螢幕，相當於一個 15 吋桌上型顯示器的像素寬度，但它的實際寬度卻只有 8 公分左右，這對 RWD 的佈局是不合理的，在 CSS media query 的判斷上 1,284 px 會落入桌機板的區間內，如此就無法為這樣高 PPI 的裝置正確的顯示出手機版的網頁，必須有一種機制來處理這樣的問題，於是在 web 方面引入了 CSS pixel 這個新尺度。

CSS Pixel

CSS pixel 是 web 世界的 pixel，它與螢幕像素之間有一個倍率關係，這個倍率關係有許多名字，有稱之為 device pixel ratio，簡稱 DPR，也有稱之為 dots per pixel，簡稱 DPPX，也有人稱為 pixel density，不論是 DRP 或 DPPX 或 pixel density，他們的意義都是一樣的——螢幕像素與 CSS pixel 之間轉換的倍率值。

在 CSS pixel 與 DPR 的引入下，我們把上面的表格擴充一下：

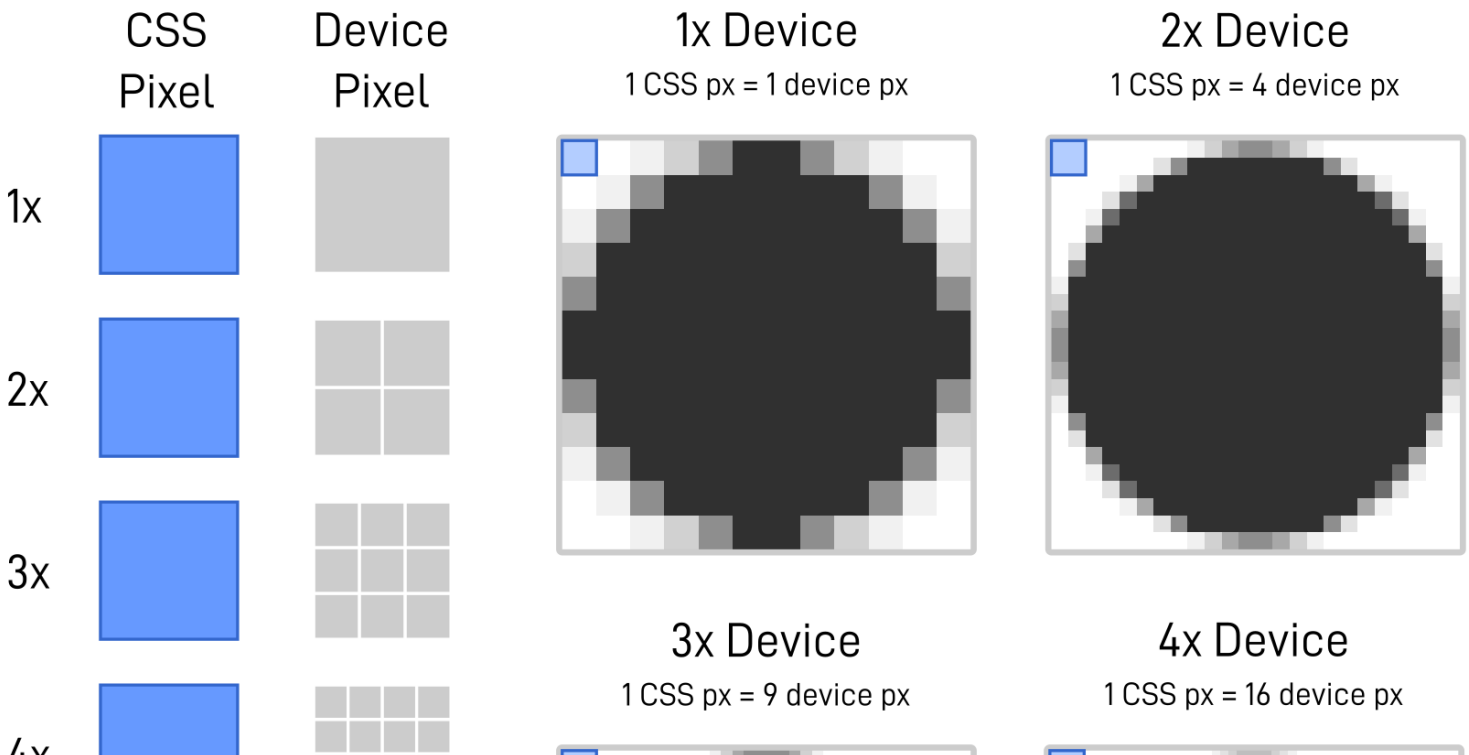
裝置	螢幕像素	螢幕對角線尺寸	PPI	CSS Pixel	DPR
iPhone 3	320 * 480 px	3.5 inch	163	320 * 480 px	1
iPhone 4	640 * 960 px	3.5 inch	326	320 * 480 px	2
iPhone SE2	750 * 1134 px	4.7 inch	326	375 * 667 px	2
iPhone 12 Pro Max	1284 * 2778 px	6.68 inch	458	428 * 926 px	3

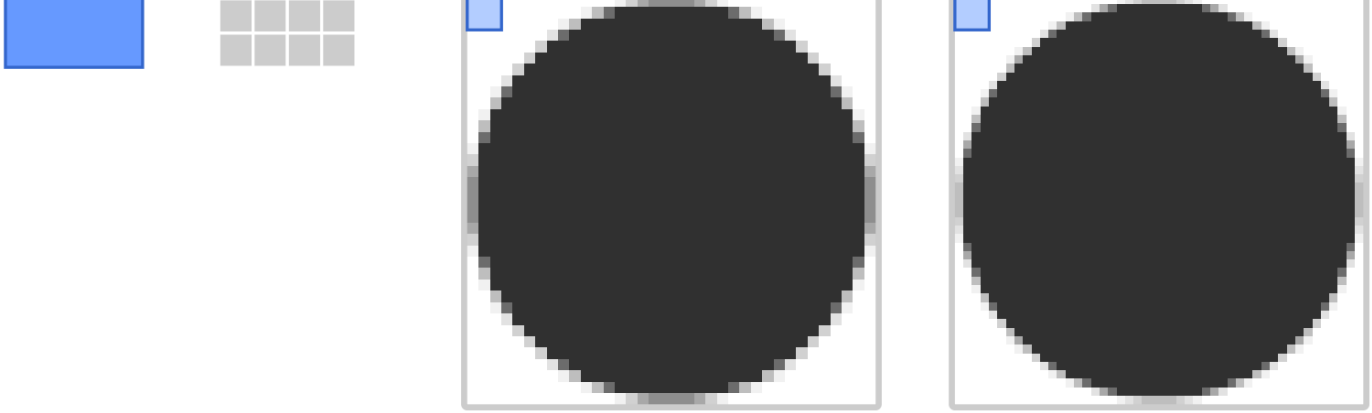
透過 DPR 與 CSS pixel 的換算，螢幕寬度從原本的 1,284、750 px 變成了少少的 428、375 px，完全可以在一個 CSS media query 區間內處理。

實際上用 CSS media query 與 JavaScript 查出來的寬度，也都會是經過 DPR 換算後的寬度，CSS pixel 與 DPR 的計算一切都是隱式發生的，就算對 CSS Pixel 與 DPR 完全沒概念的人，也不影響他開發 web，我輩 web 開發者不需考慮設備的真實解析度，一律都以 CSS pixel 的尺度去設計即可。

DPR / DPPX / Pixel Density

在 Retina Dispaly 出現以前，CSS pixel 與螢幕 pixel 是很間單的 1:1 對應，一顆像素就是一顆像素，並沒有比例關系的概念，Retina display 問世後，一個 CSS pixel 對應到螢幕像素的關係呈一種比例關係，可以用下面的圖示理解這個比例關係：





螢幕寬度固定在 8 公分左右，在 8 公分內塞入越多的螢幕像素，對人眼的感受越細緻，前面提過我們是用 PPI 來衡量單位距離內塞入的螢幕像素數，而以 160 PPI 做基準設為 1x，320 PPI 則視為 2x，以此類推。

而圖中的 CSS pixel 是被定義為 1/160 inch，因此 CSS pixel 與真實世界的關係是穩定的，因此 CSS pixel 與螢幕像素之間也會有這樣的比例關係存在，關於 CSS pixel 的定義下面會再談到。

回顧我們賦予這個比例關係的三個名字，各有其意義在：

- Device Pixel Ratio / DPR：描述螢幕像素與 CSS 像素間的**比例關係**
- Dots Per Pixel / DPPX：「dot」指的是螢幕像素、「pixel」指的是 CSS pixel，描述兩者間的**數量比**
- Pixel Density：描述**像素密度**

儘管面向略有不同，但三者指涉的的確是同一個概念——CSS 像素與螢幕像素的**比例關係**。

這邊又接回像素密度的概念，前面有提過 PPI 的意義——1 英吋內有多少 pixel，越高的 PPI 表示越精細的螢幕，而 CSS pixel 又是因應高 PPI 螢幕下的產物，CSS pixel 的原始定義是這樣的：

```
1 CSS pixel ≈ 1/96 inch
```

但為了因應不同大小的螢幕，觀看距離也有所不同，而在原始定義之外，有其它的參考定義：

```
Modern laptop with LCD:
1 CSS Pixel ≈ 1/125 inch

Smartphones / Tablets:
```

$$1\ \text{CSS Pixel} \approx 1/160\ \text{inch}$$

在手機上引用的是第三個參考定義，把公式翻轉一下：

$$160\ \text{CSS Pixel} \approx 1\ \text{inch}$$

因此：

- 在 160 PPI（每英吋有 160 顆螢幕像素）的螢幕：1 CSS pixel = 1 螢幕像素（160 / 160 = 1），DPR = 1
- 在 320 PPI 的螢幕：1 CSS pixel = 2 螢幕像素（320 / 160 = 2），DPR = 2
- 在 163 PPI 的螢幕：DPR = 1.02 ≈ 1
- 在 326 PPI 的螢幕：DPR = 2.03 ≈ 2
- 在 458 PPI 的螢幕：DPR = 2.86 ≈ 3

CSS pixel 的定義公式僅為原則，現實上還要考慮到螢幕面板的製程能力與切割最佳化因素，又或者是手機廠牌自己對人機工程的考慮，所以實際上各廠牌還是有自己的一套 CSS pixel 的決定法則，以 iOS 來說，它的 CSS pixel 實際上是以 1/163 inch 為基礎的，這是為了與他們定義的 iOS point 一致，而 iOS point 是以 iPhone 3 的 163 PPI 為基礎的，但還是偶有例外 。

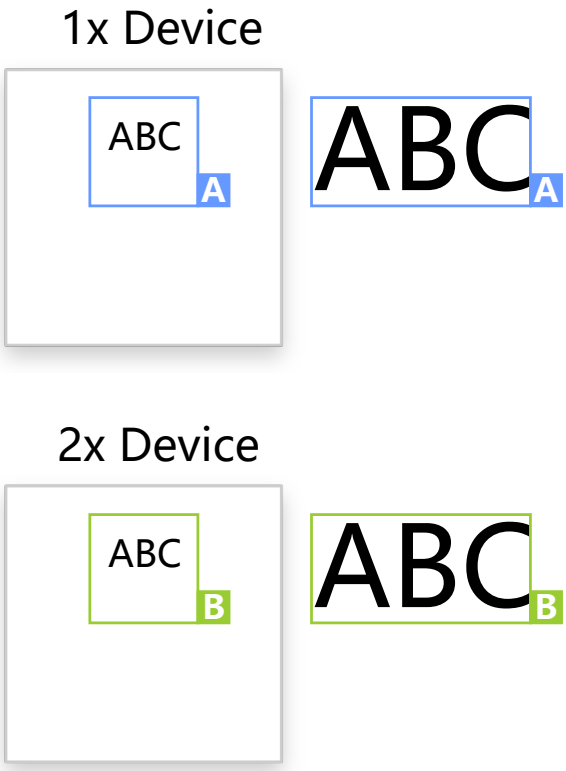
Android 陣營也有類似的狀況，Android 的 dp 單位是以 1/160 inch 為基礎，但一樣偶有例外，關於 iOS point 與 android dp 兩個單位，後面會再提到。

不過就如同前面說的，站在開發者的角度，CSS media query 查出來的已經是 CSS pixel 的尺度，我們的佈局可以不用管真實解析度是多少、不用管 DPR 是多少、不用管設備的型號，直接在 media query 用最直觀的 pixel 做 RWD 的分區間設計就好，render 引擎會幫我們處理好這些底層換算的複雜工作，我們唯一要做的是加上這句標籤讓瀏覽器 render 出符合裝置寬度，且未縮放的頁面：

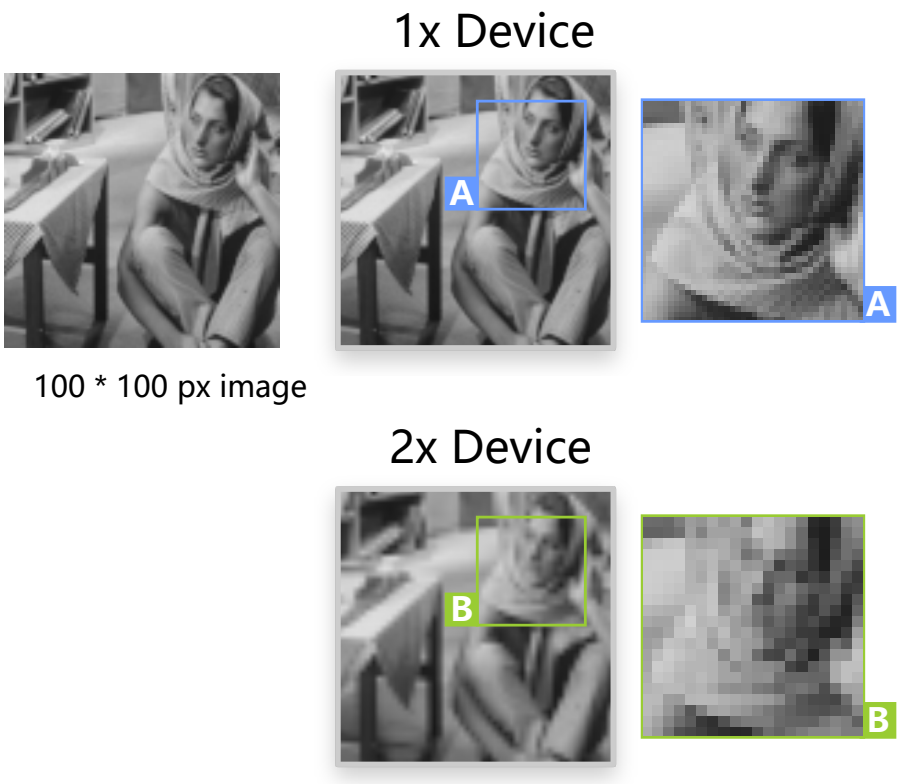
```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

DPR 沒那麼簡單

但事情沒有那麼簡單……對於文字、SVG、canvas、HTML 元素框線等向量元素，他們都是即時 render 下的結果，因此不論 DPR 為何，render 引擎都會給出最佳化的結果：

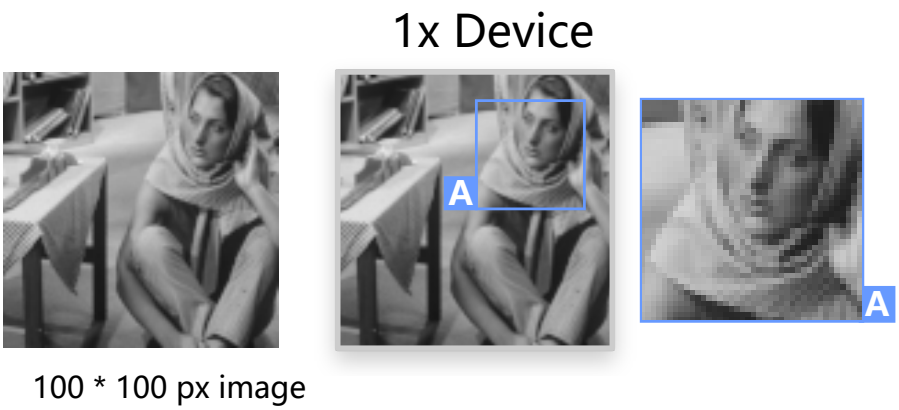


純文字區域放大看還是相當清晰，但對於點陣圖，就會有畫質劣化的感受：

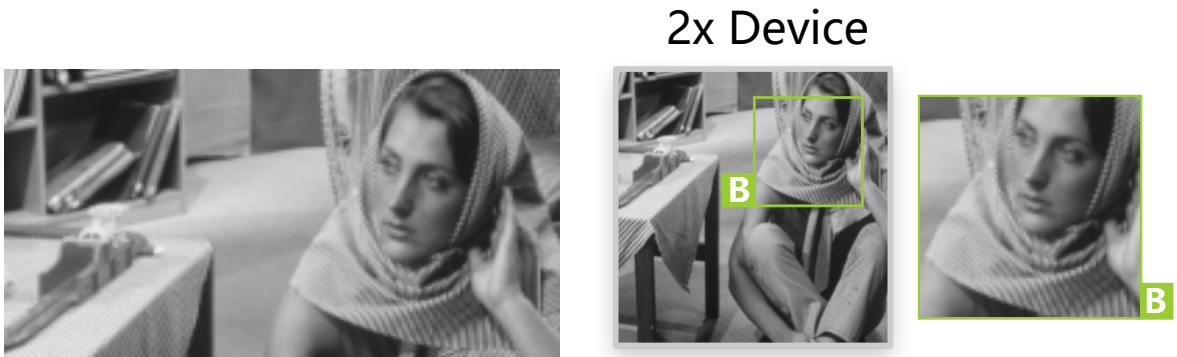


因為在 1x 設備上，圖像的 1 個像素對應到 1 個 CSS 像素，而 1 CSS 像素也對應到螢幕的 1 個像素，但在 2x 的設備，圖像的 1 個像素對應到 1 個 CSS 像素，但此時 1 CSS 像素卻是對應到 4 個螢幕像素，相當於這 100 x 100 px 的影像被投放到 400 x 400 px 的平面上，這讓 render 引擎只能把那 100 * 100 px 的圖像**粗暴的補插點放大**，因此使第 1x 設備

上，這讓 render 引擎只能把那 100 * 100 px 的圖像**粗暴的補插點放大**，因此儘管 1x 設備與 2x 設備他們的手持寬度都是 8 公分左右，但螢幕像素密度差異還是讓視覺感受到的品質下降，所以對於圖像，我們必須準備不同解析度的版本供應給不同 DPR 的設備：



100 * 100 px image



200 * 200 px image

綜合以上，在 web 開發上，我們可以利用 `<img>` 的 `srcset` 和 `sizes` 屬性來供應不同倍率的圖檔，而瀏覽器會自動根據**設備當前的 DPR** 來顯示最適合的圖檔。

App 的 pt 與 dp

在 app 方面，也是相同的概念，由於不同的手機也存在著各自的螢幕解析度與 PPI，因此一樣會面臨到 app 開發時需考慮窄至 320 px，寬至 1,284 px 的佈局呈現，因此 iOS 與 Android 也各自引進了類似 CSS pixel 的單位，在 iOS 用的是 point，簡稱 pt，不過這 pt 與 CSS 的 pt 又撞名，實際上兩者是不同的定義，為了分別起見我們稱為 iOS pt；而 Android 這邊用的單位是 device-independent pixel，簡稱 dp，不論是 CSS pixel、iOS pt、dp 都是類似概念下的產物。

回顧前面的 CSS pixel 定義：

$$1 \text{ CSS pixel} \approx 1/160 \text{ inch}$$

而這是 iOS pt 的定義：

$$1\ iOS\ pt \approx 1/163\ inch$$

而這是 dp 的定義：

$$1\ dp \approx 1/160\ inch$$

與 CSS pixel 相同的，這些公式都僅做為原則使用，僅供參考，手機上實際的 iOS pt 與螢幕解析度的比例關係是由人工定義出來的，理由與上面相同，因為面板製程能力或分割效率或廠牌對人機工程的考慮而決定，因此儘管公式不同，但結果卻是完全相同：

裝置	螢幕像素	PPI	CSS Pixel	iOS pt	DPR	iOS Sacle Fac
iPhone 3	320 * 480 px	163	320 * 480 px	320 * 480 pt	1	@1x
iPhone 4	640 * 960 px	326	320 * 480 px	320 * 480 pt	2	@2x
iPhone SE2	750 * 1134 px	326	375 * 667 px	375 * 667 pt	2	@2x
iPhone 12 Pro Max	1284 * 2778 px	458	428 * 926 px	428 * 926 pt	3	@3x

一模模一樣樣。

在 Android 方面，在 dp 與 DPI（PPI）的基礎下，直接制定了以倍率為基準的區間：

密度限定符	倍率	DPI (PPI)
ldpi	0.75x	~ 120 DPI
mdpi	1.0x 基準	~ 160 DPI
hdpi	1.5x	~ 240 DPI
xhdpi	2.0x	~ 320 DPI
xxhdpi	3.0x	~ 480 DPI
xxxhdpi	4.0x	~ 640 DPI

Android 的各廠牌手機都必然會落入上表的某一個組別內，因此 Android 的 app 開發者不用去考慮真實的螢幕解析度是多少，只要確保你的佈局可以適用於從 ldpi ~ xxxhdpi 而不跑版，並且提供從 0.75x 起至 4.0x 的各組倍率的圖檔，那 Android 會自動根據用戶手機的 DPI 去呈現佈局以及最適用的圖檔。

不論是 iOS 還是 Android 他們都引入了倍率的概念，儘管基準的 PPI（DPI）略有不同，在 iOS 的倍率從 @1x 到 @3x，在 Android 則從 0.75x 到 4.0x 這些倍率的用意也就是我輩開發者需要準備的圖檔倍率，而作業系統會根據用戶當前的設備去顯示最適合他的解析度的圖檔。

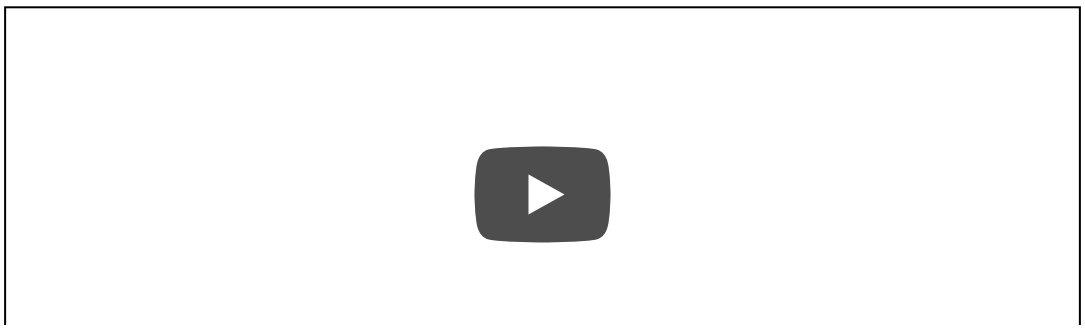
關於 React Native 的可以看 Kenneth 的〈React Native Pixel Ratio〉。

結語

綜合全文，對開發者而言，需要關注的點有三：

- 不用去管圖檔本身的 DPI 設定，它只在列印時有意義。
- 平台佈局的單位是什麼？web 用 CSS pixel、iOS 用 iOS pt、Android 用 dp，用正確的單位進行版面的佈局，不要去糾結螢幕的硬體解析度。
- 圖檔依照平台指定的倍率提供，iOS 有 @1x ~ @4x、Android 有 ldpi ~ xxxhdpi，確保 app 內的圖檔都有提供這些倍率的版本，web 的話則是用 `<img>` 標籤的 `srcset` 和 `sizes` 來定義各倍率的圖檔，瀏覽器會自行抓取最適合的做顯示。

這篇文章是本人寫過最長的一篇，在寫的途中參考了大量資料，自己也釐清了一些過往一知半解的認知，但這麼廣的議題（從像素講到 DPI / PPI 和印表機的機構再講到 app 的單位）應該是難免有錯，有錯誤的地方請大聲怒罵指責。



程式　設計　前端

本文由 INFOLINK 聯騰資訊股份有限公司提供，聯騰資訊專注於為零售與餐飲產業提供智慧化的系統解決方案，以 ERP 為核心為客戶開拓 E 化應用，與 POS、BI、EC 等應用實現無縫整合，我們在此分享我們對產業與技術的觀點，歡迎與我們交流或追蹤我們。