

# 脫離靜態 PDF: UI/UX 驅動的現代數位內容 展示與傳播技術戰略

## 第一部分 導論: 從靜態資訊到動態體驗的轉變

在當今的數位環境中，傳統的靜態文件（如 PDF）在傳達複雜資訊和吸引大眾方面正迅速顯得力不<sup>1</sup>。使用者期望的不再僅僅是資訊的被動接收，而是一種沉浸式、互動式且美觀的體驗<sup>2</sup>。當一個開發項目進入面向大眾的階段時，其成功的關鍵不僅在於內容本身，更在於其「展示」和「傳播」的方式。

從用戶介面 (UI) 和使用者體驗 (UX) 的角度來看，這場轉變的核心是從「靜態」轉向「動態」。UI 負責產品的視覺外觀，如顏色、字體和佈局，確保其美觀且直觀<sup>3</sup>；而 UX 則關注使用者在整個互動過程中的流暢性、易用性和整體感受<sup>3</sup>。

本報告旨在深入探討當前最受歡迎的數位展示技術，分析它們如何超越 PDF，提供「美觀好用」的體驗。我們將解構真正能為溝通加分的圖表形式，剖析其背後的瀏覽器傳播技術（包括渲染引擎和動畫 API），並總結實現卓越 UI/UX 所需的核心設計原則。

## 第二部分 告別 PDF: 新一代「美觀好用」的數位出版技術

PDF 的核心問題在於其「靜態」和「非自適應性」。它本質上是數位化的紙張。而現代面向大眾的傳播技術，其核心在於「互動性」和「敘事性」。

### 2.1 互動式數位出版平台

超越 PDF 的第一步是採用專為數位體驗設計的出版平台。諸如 Readymag、Ceros、Joomag 和 DCatalog 等工具，允許創作者將內容製作成具有豐富互動性的網頁體驗，而不是靜態文件<sup>6</sup>。這

些平台的核心優勢在於能夠嵌入影片、動畫、可點擊的熱點，並提供分析功能<sup>1</sup>，同時許多平台（如 Ceros）專注於無需編碼即可實現高級互動<sup>6</sup>。

## 2.2 案例聚焦：Readymag 的無代碼敘事能力

Readymag 是一個突出的例子，它定位為一個無需編碼的網頁設計工具，特別擅長創建視覺震撼的數位出版物，如線上雜誌、作品集、簡報和登陸頁面<sup>8</sup>。

從 UI/UX 角度來看，Readymag 的核心優勢在於其「完全的創意自由」，它提供了一個類似 Sketch 或 Photoshop 的空白畫布，但直接在瀏覽器中運作<sup>11</sup>。設計師可以透過直觀的拖放介面<sup>8</sup>，結合其強大的排版功能（支援超過 5,000 種字體<sup>9</sup>）和精確的像素級控制，來打造高度客製化的佈局。

更重要的是，它內建了強大的動畫功能，這正是 PDF 所缺乏的<sup>9</sup>。設計師可以輕鬆為任何元素添加多步驟動畫，並設定四種觸發器：頁面加載時（On Load）、點擊時（On Click）、滑鼠懸停時（On Hover）或滾動時（On Scroll）<sup>13</sup>。這種能力使得靜態的報告或作品集<sup>17</sup> 得以轉化為引人入勝的互動故事。

## 2.3 沉浸式敘事：Scrollytelling（滾動敘事）

Scrollytelling（滾動式敘事）是目前公認在 UI/UX 上最「美觀」且強大的數位敘事技術之一，尤其適用於數據新聞和深度報告<sup>18</sup>。

其定義是將敘事（Storytelling）與滾動（Scrolling）行為相結合<sup>18</sup>。當使用者向下滾動頁面時，文字、視覺元素（如圖表、圖片、影片）會隨之動態變化、出現或產生動畫<sup>20</sup>。這種方式將使用者從被動的閱讀者轉變為「探索者」，讓他們能以自己的節奏體驗故事，從而極大地提高了參與感和資訊的吸收度<sup>18</sup>。

## 2.4 Scrollytelling 的核心 UI/UX 模式

從 UI/UX 結構來看，Scrollytelling 並非單一技術，而是一套設計模式。最常見的佈局變體包括<sup>23</sup>：

1. 全螢幕圖形模式：視覺元素（如地圖或 3D 模型）固定在整個瀏覽器視窗，而敘述文字區塊則以半透明圖層的形式在滾動時依序浮現。

2. 側邊欄文本模式：螢幕分為兩欄，一側（通常是左側）是滾動的文本敘述，另一側（通常是右側）是固定的視覺化窗口。當文本滾動到特定段落時，會觸發右側視覺化圖表的狀態更新（例如，地圖縮放、圖表高亮或播放動畫）。
3. 文本與圖形的關係<sup>23</sup>：
  - 一對一 (One-to-one)：每個文本區塊對應一個獨特的圖表狀態。
  - 多對一 (Many-to-one)：多個文本段落用來詳細解釋同一個圖表狀態。
  - 一對多 (One-to-many)：一個文本區塊在滾動時，觸發圖表的多個連續狀態變化。

## 2.5 典範案例分析：從數據中講述故事

Scrollytelling 的成功在於它無縫整合了多媒體和數據視覺化。

- 新聞業的標竿：雖然《紐約時報》2012 年的 "Snow Fall"<sup>19</sup> 被認為是開創者，但近年的作品更為成熟。例如，《金融時報》的 "How China Is Tearing Down Islam"<sup>24</sup> 結合了照片、影片和數據視覺化，以極具衝擊力的方式記錄了文化變遷<sup>24</sup>。
- 數據的呈現：彭博社 (Bloomberg) 經常使用 Scrollytelling 來解釋複雜的經濟數據，例如他們在分析中國民族主義對品牌影響時，使用了「水平滾動」的獨特變體<sup>23</sup>。
- 社會議題的深化：《衛報》(The Guardian) 在 2022 年的最佳數據故事中，利用熱圖 (Heatmap) 視覺化紐約市的「氣候不平等」，展示了不同社區的溫差如何與財富相關聯<sup>25</sup>。

這些案例表明，Scrollytelling 是面向大眾進行深度溝通的黃金標準，它利用滾動這個最自然的網頁互動行為，將複雜數據轉化為引人入勝且易於理解的敘事<sup>18</sup>。

## 第三部分 真正加分的圖表：以「溝通目的」選擇視覺化形式

面向大眾傳播時，圖表的選擇不應追求花俏或複雜，而應絕對地服務於「溝通目的」<sup>28</sup>。一個常見的誤區是使用 3D 圖表或過度設計的圖表<sup>29</sup>，這些圖表會扭曲數據的真實比例，反而阻礙理解<sup>31</sup>。

清晰的視覺化設計應優先考慮數據的準確性、清晰度和完整性<sup>32</sup>。選擇正確的圖表類型，是確保資訊被準確傳達的第一步<sup>28</sup>。

### 圖表選擇的溝通目的矩陣

為了幫助開發團隊做出正確選擇，下表依據「溝通目的」總結了最有效且最應避免的圖表類型：

溝通目的 (Goal)	最佳圖表類型 (Best Choice)	說明與 UI/UX 建議	應避免的陷阱 (Avoid)
隨時間變化 (Trend)	折線圖 (Line Chart) 面積圖 (Area Chart)	最清晰的時間序列呈現方式 [28, 32]。面積圖可強調總量的變化 [34]。	餅圖 (Pie Chart) (完全無法表達趨勢)
類別比較 (Comparison)	柱狀圖 (Column Chart) 條形圖 (Bar Chart)	用於比較不同類別的量級 [28, 31]。類別名稱較長時，使用條形圖 (Bar) [32]。	3D 柱狀圖 (3D Column) (透視會扭曲高度，使比較困難 <sup>31</sup> )
部分佔比 (Part-to-Whole)	餅圖 (Pie Chart) 環圈圖 (Donut Chart) 樹狀圖 (Treemap)	展示靜態比例 <sup>28</sup> 。 【關鍵】餅圖和環圈圖的切片不應超過 5 塊 <sup>29</sup> ，否則人類肉眼難以比較角度 [35]。	多個餅圖 (多個餅圖並列比較，效率極低 [35])
數據關聯性 (Relationship)	散點圖 (Scatter Plot) 氣泡圖 (Bubble Chart)	顯示兩個變量 (X/Y 軸) 之間的相關性 [36]。氣泡圖可增加第三個維度 (氣泡大小) [37]。	折線圖 (Line Chart) (除非 X 軸是時間，否則連線會產生誤導)
數據分佈 (Distribution)	直方圖 (Histogram) 箱型圖 (Box Plot)	直方圖顯示數據的頻率分佈 [36, 37]。箱型圖濃縮了數據的關鍵統計值 (中位數、四分位數等) [37, 38]。	散點圖 (Scatter Plot) (數據點過多時會重疊，看不出分佈)
地理空間 (Spatial)	地圖 (Map) 熱圖 (Heatmap)	展示地理相關數據 [36]。熱圖可用於顯示密度或強度，如《衛報》的案例 <sup>25</sup> 。	在非地理數據上濫用 (例如，用世界地圖)

			顯示無關的銷售數據)
--	--	--	------------

## 第四部分 瀏覽器傳播技術 (一): 渲染引擎的抉擇

當圖表選定後，開發者面臨的第一個關鍵技術決策是：使用哪種瀏覽器技術來「繪製」它？這個選擇將深刻影響效能、互動性和無障礙性。

### 4.1 SVG vs. Canvas: 核心差異

網頁圖形渲染主要有兩種技術：SVG (Scalable Vector Graphics) 和 Canvas<sup>39</sup>。

- **SVG (向量)**: 這是一種基於 XML 的標記語言<sup>39</sup>。每個圖形元素(如 \$lt;circle\$、\$lt;path\$)都是一個 DOM 節點<sup>40</sup>。
- **Canvas (點陣)**: 這是一個 HTML 元素，提供一個空白畫布，開發者使用 JavaScript API (如 getContext("2d")) 在上面繪製像素<sup>39</sup>。

### 4.2 SVG 的優勢: 互動與清晰度

SVG 採用「保留模式」(Retained Mode)<sup>40</sup>。

- 優點：
  1. **解析度無關 (Resolution Independent)**: 作為向量圖形，SVG 可以無限放大而不失真或模糊<sup>40</sup>。
  2. **內建互動性 (Interactive)**: 由於每個元素都是 DOM 節點，可以像操作 HTML 元素一樣，輕易地為其添加 CSS 樣式和 JavaScript 事件(如 hover, click)<sup>40</sup>。
  3. **無障礙性 (Accessible)**: 螢幕閱讀器可以讀取 DOM 結構，使其比 Canvas 更易於實現無障礙訪問<sup>42</sup>。
- 缺點：
  1. **效能瓶頸 (Performance Bottleneck)**: 當 DOM 元素數量過多時(例如，繪製 10,000 個以上的數據點)，瀏覽器需要維護一個龐大的 DOM 樹，導致渲染和互動變得非常緩慢<sup>43</sup>。
- 適用場景：絕大多數的標準數據圖表(如折線圖、柱狀圖)、Logo 和圖標，尤其是在需要高

清晰度和豐富互動性的儀表板中<sup>40</sup>。

### 4.3 Canvas 的優勢 : 原始效能

Canvas 採用「即時模式」(Immediate Mode)<sup>40</sup>。

- 優點：
  1. 高效能 (**High Performance**)：系統在繪製像素後就會「忘記」它們<sup>40</sup>。這使得 Canvas 在處理極端大量的數據點時(例如 10 萬到 100 萬個點)表現極佳<sup>41</sup>。AG Charts 團隊就明確指出，他們選擇 Canvas 的原因就是為了處理 AG Grid 帶來的大型數據集(超過 10 萬筆)<sup>41</sup>。
- 缺點：
  1. 解析度相關 (**Resolution Dependent**)：放大時會失真和像素化<sup>42</sup>。
  2. 缺乏互動性 (**Not Interactive**)：它只是一張像素圖。如果需要 hover 效果，開發者必須手動編寫程式碼，去計算滑鼠座標是否落在某個繪製的圖形區域內<sup>44</sup>。
  3. 無障礙性差 (**Inaccessible**)：對螢幕閱讀器而言，Canvas 是一個「黑盒子」<sup>42</sup>。
- 適用場景：大規模數據視覺化(如熱圖、大型散點圖)、網頁遊戲、即時動畫<sup>41</sup>。

### 4.4 Canvas 效能最佳化實踐

如果面向大眾的項目確實需要處理海量數據，而必須選擇 Canvas，開發者應採用以下最佳實踐來維持效能<sup>45</sup>：

1. 分層畫布 (**Layered Canvases**)：將複雜場景分解到多個 `$lt;canvas$` 元素上(例如，一個畫布放靜態背景，另一個畫布放動態元素)，這樣在更新時只需重繪動態層<sup>45</sup>。
2. 離線渲染 (**Off-screen Canvas**)：對於可重複使用的複雜圖形，先在一個記憶體中的「離線畫布」上預先繪製好，然後再將這個離線畫布的「快照」一次性繪製到主畫布上<sup>45</sup>。
3. 最小化重繪 (**Minimal Redraw**)：避免在每一幀都使用 `clearRect()` 清除整個畫布<sup>47</sup>。只清除和重繪發生變動的區域<sup>45</sup>。
4. 避免浮點數座標：盡可能使用整數座標 (e.g., `ctx.fillRect(10, 10, 100, 100)`)。使用浮點數座標 (e.g., `ctx.fillRect(10.5, 10.5, 100, 100)`) 會觸發瀏覽器的「次像素渲染」(Sub-pixel rendering)，增加額外的運算負擔<sup>45</sup>。
5. 批次處理 (**Batch Calls**)：集中處理畫布的狀態變更(如 `fillStyle`, `strokeStyle`)，避免在繪製迴圈中頻繁切換狀態<sup>45</sup>。

## 第五部分 瀏覽器傳播技術 (二): WebGL 與高效能運算

當 SVG 和 2D Canvas 的效能都達到極限時，面向大眾的頂級視覺化技術將轉向 WebGL (Web Graphics Library)。

### 5.1 什麼是 WebGL？

WebGL 是一個 JavaScript API，它允許網頁瀏覽器直接存取電腦的圖形處理單元 (GPU) 進行 2D 和 3D 圖形渲染<sup>49</sup>。

SVG 和 Canvas 2D 主要是由 CPU 驅動的，而 GPU 專為大規模並行運算 (如圖形渲染) 而設計<sup>51</sup>。WebGL 的核心優勢是將繁重的圖形計算任務從 CPU 轉移到 GPU，從而實現了以往在網頁上無法想像的效能<sup>51</sup>。

### 5.2 為什麼 WebGL 對數據展示很重要？

對於面向大眾的數據展示，WebGL 解鎖了兩大能力：

1. **超大規模數據集 (Large Datasets)**: WebGL 能夠流暢地即時渲染數十萬甚至數百萬個數據點<sup>52</sup>。例如，金融機構使用 WebGL 來即時視覺化股票市場數據<sup>49</sup>。
2. **沉浸式 3D 體驗 (3D Rendering)**: 它允許超越 2D 圖表，以 3D 形式展示多維數據<sup>52</sup>，例如 3D 散點圖、3D 網路圖<sup>53</sup>，或 AR/VR 體驗<sup>54</sup>。

### 5.3 實戰：WebGL 框架生態

直接編寫原生 WebGL (包含 GLSL 著色器語言) 非常複雜<sup>50</sup>。在實務中，開發團隊幾乎總會使用建立在 WebGL 之上的框架：

- **通用 3D 框架**: Three.js<sup>55</sup> 和 Babylon.js<sup>55</sup> 是最流行的 3D 函式庫，適用於遊戲、產品展示和自定義 3D 數據藝術。
- **數據視覺化框架**: 許多現代圖表庫已將 WebGL 作為其底層渲染器之一，以應對效能挑戰。
- **地理空間框架**: Deck.gl<sup>52</sup> 是專為大規模地理空間數據視覺化而設計的頂級 WebGL 框架。

## 5.4 案例 : Deck.gl 和 ECharts-GL 的高效能實踐

- Deck.gl<sup>57</sup>: Deck.gl 的範例庫展示了 WebGL 的強大能力。它不僅能繪製基本的點和線，還提供了專門的「圖層」(Layers)，如 HexagonLayer(六邊形熱圖)、PointCloudLayer(光達點雲數據)和 Tile3DLayer(3D 城市模型)<sup>57</sup>。這使其成為處理大規模 3D 和地理數據的首選。
- ECharts-GL<sup>58</sup>: Apache ECharts 提供了 ECharts-GL 擴展，允許開發者將特定圖表切換到 WebGL 渲染模式。其範例庫中包含了「1000 萬 GPS 點」的 ScatterGL 圖表和「100 萬條紐約街道」的 LinesGL 圖表<sup>58</sup>，這些都是傳統 SVG 或 Canvas 2D 難以實現的。

## 第六部分 瀏覽器傳播技術 (三): 主流圖表函式庫 (Charting Libraries) 評析

對於大多數「面向大眾」的專案，從零開始手動編寫 SVG 或 WebGL 是不切實際的。開發團隊通常會選擇一個成熟的圖表函式庫。這個選擇是一個在「易用性」、「客製化能力」和「效能」之間的權衡。

### 6.1 核心喻: 從零烹飪 vs. 現成醬料

一個廣為流傳的比喻可以幫助我們理解不同函式庫的定位<sup>59</sup>:

- D3.js (Data-Driven Documents): 這是\*\*「從零開始烹飪」\*\*<sup>59</sup>。D3 本身並不是一個圖表庫<sup>60</sup>，而是一個強大的 JavaScript 函式庫，用於基於數據來操作 DOM (主要是 SVG)<sup>60</sup>。
  - 優點：提供了無與倫比的靈活性和控制力，可以用來創建任何你能想像到的、高度客製化的視覺化作品<sup>61</sup>。
  - 缺點：學習曲線極其陡峭<sup>60</sup>。
  - 適用場景：需要高度客製化和獨特視覺敘事的 Scrolltelling 專案<sup>59</sup>。
- Chart.js: 這是\*\*「使用現成的醬料包」\*\*<sup>59</sup>。
  - 優點：非常簡單易用<sup>64</sup>，檔案輕量，提供 8 種核心圖表類型，並有流暢的預設動畫<sup>63</sup>。它使用 Canvas 渲染<sup>60</sup>。
  - 缺點：客製化能力有限，難以實現標準圖表之外的複雜視覺化<sup>63</sup>。
  - 適用場景：快速開發、標準化的圖表需求、內部儀表板<sup>59</sup>。

## 6.2 企業級與高效能的選擇

- **Highcharts**: 這是\*\*「企業級的精緻套餐」\*\*<sup>67</sup>。
  - 優點: 在企業應用中極受歡迎, 全球百大公司中有 80 家是其客戶<sup>68</sup>。它以其極為出色和詳細的文件<sup>68</sup>、穩定的 API 和強大的無障礙設計 (**Accessibility**) 模組<sup>69</sup> 而聞名。它提供了完整的產品線, 包括 Core、Stock、Maps 和 Gantt<sup>68</sup>。
  - 缺點: 商業使用需要付費授權<sup>67</sup>。
  - 適用場景: 需要長期維護、高度可靠且必須符合無障礙標準 (a11y) 的企業級應用。
- **Apache ECharts**: 這是\*\*「高效能的自助餐」\*\*<sup>67</sup>。
  - 優點: 由 Apache 基金會維護, 開源免費。它以其卓越的效能著稱<sup>67</sup>, 並且內建了對 WebGL 渲染器 (ECharts-GL) 的支援<sup>56</sup>, 使其在處理大數據集時優於 Highcharts 或 Chart.js。
  - 缺點: 學習和配置的複雜度可能高於 Chart.js。
  - 適用場景: 需要處理大量數據(數萬至數百萬點)的儀表板, 或需要 3D 視覺化效果的專案。
- **Vega-Lite**: 這是\*\*「未來的聲控食譜」\*\*。
  - 優點: 它是一種「視覺化語法」(Visualization Grammar)<sup>71</sup>。開發者不需要編寫繪製邏輯, 而是以 JSON 的形式「聲明」你想要的視覺化<sup>72</sup>。例如, 聲明 X 軸綁定到 "IMDB\_Rating" 欄位, Y 軸綁定到 "count"<sup>72</sup>。
  - 缺點: 抽象層意味著犧牲了 D3.js 那樣的底層控制力<sup>72</sup>。
  - 適用場景: 需要快速迭代、動態生成圖表(例如由用戶自定義儀表板)的應用程式。

## 第七部分 UI/UX 核心原則：從「能看」到「好懂」

選擇了正確的技術和圖表庫, 僅僅完成了工作的一半。面向大眾的成功關鍵, 在於遵循以使用者為中心的 UI/UX 原則。

### 7.1 儀表板 (Dashboard) 設計原則：清晰度至上

31

一個「好用」的儀表板, 其首要原則是「清晰」<sup>31</sup>。

1. 避免資訊過載：儀表板不應試圖一次顯示所有內容。應專注於最關鍵的指標 (KPI) <sup>31</sup>。
2. 建立視覺層級 (Visual Hierarchy)：遵循「F型」或「Z型」掃視模式，將最重要的資訊放在頁面的左上角 <sup>31</sup>。使用字體大小、粗細和顏色來引導用戶的注意力 <sup>31</sup>。
3. 簡化視覺元素 (Simplify Visuals)：堅決移除不必要的「圖表垃圾」(Chartjunk) <sup>29</sup>。這包括過多的網格線、邊框或裝飾性圖標 <sup>31</sup>。
4. 提供互動而非雜亂 (Interaction over Clutter)：與其顯示一個包含 100 個數據點的擁擠圖表，不如顯示一個總覽圖表，並提供「過濾器」(Filters) <sup>73</sup> 和「下鑽」(Drill-Downs) 功能 <sup>31</sup>，讓用戶能主動探索他們感興趣的細節。

## 7.2 暗黑模式 (Dark Mode) 的正確實踐

78

暗黑模式因其能減少眼部疲勞 <sup>74</sup> 和突顯內容 <sup>75</sup>，在數據密集的儀表板中廣受歡迎 <sup>76</sup>。但「面向大眾」的暗黑模式設計絕非簡單地反轉顏色 <sup>74</sup>。

錯誤的暗黑模式(例如純黑背景配純白文字)會產生過高的對比度，反而比亮色模式更刺眼 <sup>78</sup>。

專業儀表板的暗黑模式 UI 規則：

- 規則一：避免純黑背景。永遠不要使用 #000000 作為背景。應選擇深灰色或深海軍藍(例如 Google Material Design 推薦的 #121212) <sup>77</sup>。這能減少眩光並提供更豐富的深度感。
- 規則二：避免純白文字。純白文字 (#FFFFFF) 在深色背景上會產生「光暈」效應 <sup>78</sup>。應使用灰白色或去飽和的淺色(例如 80% 透明度的白色) <sup>78</sup>。
- 規則三：降低顏色飽和度。顏色在深色背景上看起來會更「亮」、更飽和 <sup>77</sup>。因此，在圖表中使用的主色調(如藍色、綠色)應降低其飽和度 (Desaturate)，使其在暗黑模式下看起來與在亮色模式下具有相同的視覺權重 <sup>77</sup>。
- 規則四：保持品牌一致性。確保品牌的視覺標識在兩種模式下都能被清晰識別 <sup>74</sup>。

## 7.3 響應式設計 (RWD)：為所有裝置設計圖表

面向大眾意味著使用者將在各種尺寸的螢幕上(從手錶到桌面電腦)查看您的內容 <sup>80</sup>。圖表的響應式設計 (RWD) 不是簡單地將整個圖表等比例縮小 <sup>80</sup>。一個在桌面上清晰的複雜折線圖，縮小到手機螢幕上將會變成一團無法閱讀的線條 <sup>80</sup>。

真正的「圖表 RWD」是一種\*\*「資訊自適應」(Adaptive Information Design)\*\* 策略 <sup>81</sup>：

1. 採用 Mobile-First 策略<sup>83</sup>:
  - 優先排序 (Prioritize): 在最小的螢幕上 (Mobile), 只顯示最核心的 1-2 個 KPI<sup>80</sup>。
  - 簡化數據 (Simplify): 減少數據的粒度 (Level of Detail)。例如, 將「每日趨勢」聚合為「每月趨勢」<sup>81</sup>。
  - 改變佈局 (Change Layout): 將寬幅的水平條形圖轉換為適合垂直滾動的柱狀圖<sup>81</sup>。
2. 漸進式增強 (Progressive Enhancement):
  - 在平板電腦上, 可以顯示圖表和簡單的數據表格。
  - 在桌面上, 才顯示完整的、具有複雜互動(如下鑽、過濾)的儀表板<sup>81</sup>。

## 7.4 無障礙設計 (Accessibility, a11y): 面向「所有」大眾的關鍵

面向「大眾」就意味著必須包含殘障人士。無障礙設計 (a11y) 不僅是法律和道德要求<sup>84</sup>, 它更是一種「設計乘數」(Design Multiplier): 為殘障人士解決的設計問題, 最終會改善所有使用者的體驗<sup>85</sup>。

將 a11y 變為優勢的實踐:

- 情境: WCAG (網頁內容無障礙指南) 規定「不能單獨依賴顏色來傳達資訊」(Guideline 1.4.1)<sup>86</sup>。這是為了幫助色盲或色弱使用者<sup>86</sup>。
- A11y 解決方案: 「雙重編碼」(Dual Encoding)<sup>85</sup>。
  - 例如, 在折線圖中, 除了使用不同顏色, 還應使用不同的線條樣式(實線、虛線、點線)<sup>86</sup>。
  - 在餅圖中, 除了顏色, 還應使用圖案或紋理<sup>87</sup>。
- 帶來的「普遍紅利」:
  1. 非色障使用者也受益: 一個使用「顏色 + 直接標籤」<sup>85</sup> 的圖表, 比使用「顏色 + 圖例 (Legend)」<sup>29</sup> 的圖表更易讀。使用者無需在圖表和圖例之間來回掃視。
  2. 情境受益: 該圖表在被黑白影印後, 依然保持了可讀性<sup>86</sup>。

面向開發者的 A11y 行動方案<sup>69</sup>:

- 鍵盤導覽 (Keyboard Navigation): 確保所有互動元素——包括圖表上的每一個數據點、過濾器和按鈕——都可以僅透過 "Tab" 鍵和箭頭鍵來訪問和操作<sup>69</sup>。Highcharts 的 a11y 模組<sup>70</sup> 在這方面是行業標準。
- ARIA 角色 (ARIA Roles): 使用 WAI-ARIA 屬性(如 role="img")和 aria-label 來為螢幕閱讀器提供圖表的「上下文」<sup>87</sup>。例如: aria-label="柱狀圖: 2024 年各季度銷售額, 第一季 100 萬..."<sup>70</sup>。
- 提供文字替代方案 (Text Alternatives):
  - Alt 文字: 為簡單圖表提供摘要<sup>87</sup>。
  - 數據表格 (Data Table): 對於複雜圖表, 最佳實踐是在圖表附近提供一個「跳轉至數

據表格」的連結<sup>86</sup>。這個表格(一個標準的 \$lt;table\$)包含了圖表中的所有原始數據。這不僅對螢幕閱讀器用戶至關重要，也對那些希望複製數據進行自己分析的「高階用戶」極為友好。

## 第八部分(進階)動畫的未來:Web Animations API (WAAPI)

在 Scrollytelling 和儀表板中實現流暢的動畫，是提升 UI/UX 質感的關鍵。傳統上，開發者在兩種技術間掙扎：

1. **CSS 動畫/過渡**: 效能極佳，因為瀏覽器可以將其交給 GPU 在合成器線程上執行(不阻塞主線程)<sup>91</sup>。但其控制力很弱，通常是「一次性」的(如 hover 效果)，難以編排複雜的序列<sup>92</sup>。
2. **JavaScript 動畫(如 requestAnimationFrame)**: 提供了完整的編程控制力(可暫停、倒帶、動態改變值)<sup>91</sup>。但它在主線程上運行，如果運算過於複雜(例如，同時計算 Scrollytelling 的滾動位置和動畫狀態)，很容易導致主線程阻塞，造成頁面卡頓和掉幀(Jank)<sup>91</sup>。

### 8.1 WAAPI 的核心優勢:效能與控制力的結合

Web Animations API (WAAPI)<sup>93</sup> 是 W3C 的新瀏覽器標準，它旨在提供\*\*「JavaScript 的控制力」和「CSS 的效能」\*\*<sup>91</sup>。

WAAPI 允許開發者使用 JavaScript 語法(如 element.animate()<sup>94</sup>)來定義動畫。但動畫的實際執行則被交給瀏覽器的合成器線程，從而實現了\*\*「主線程外的動畫」(Off-main-thread animations)\*\*<sup>91</sup>。

### 8.2 應用:編排複雜的 SVG 圖表動畫

WAAPI 的真正威力在於「編排」(Orchestrate)<sup>92</sup>。在複雜的 Scrollytelling 中，您可能需要：「當用戶滾動到 B 點時，A 圖表縮放，同時 B 圖表淡入，然後 C 標籤移動到新位置」。

使用 WAAPI，您可以精確控制這些 SVG 元素<sup>95</sup> 動畫的播放、暫停、時間軸，而無需擔心效能瓶

頸。這是現代瀏覽器傳播技術的尖端，也是實現電影般數據敘事的底層基礎。

## 第九部分 結論：給開發團隊的戰略路線圖

從靜態 PDF 邁向面向大眾的動態、美觀且好用的數位內容，是一場涉及技術選型、設計思維和開發實踐的系統性轉變。不存在「最好」的技術，只有「最適合」您溝通目標、團隊技能和數據規模的「技術堆疊」(Tech Stack)。

以下三大戰略劇本，可作為開發團隊的行動指南：

### 9.1 戰略劇本 1：行銷敘事 (Marketing Narrative) 堆疊

- 目標：快速產出「美觀」、具有故事性的內容(如年度報告、產品發布、深度報導)。
- 技術棧：
  - 無代碼通道：Readymag<sup>8</sup> 或 Shorthand<sup>21</sup>。利用其內建的滾動觸發動畫<sup>15</sup> 和精美排版<sup>9</sup>，快速交付。
  - 客製化通道：Scrollytelling 框架<sup>96</sup> + D3.js<sup>61</sup> / Vega-Lite<sup>72</sup>。D3.js 用於實現獨一無二的、與敘事緊密結合的視覺化效果。
- UI/UX 重點：Mobile-First (資訊自適應)<sup>80</sup>，Scrollytelling 佈局模式 (如側邊欄文本)<sup>23</sup>。

### 9.2 戰略劇本 2：商業智慧 (BI Dashboard) 堆疊

- 目標：打造「好用」、清晰、高互動性的內部或外部數據儀表板。
- 技術棧：Highcharts<sup>68</sup> (若預算充足且重視無障礙) 或 Apache ECharts<sup>67</sup> (若重視開源與效能)。
- UI/UX 重點：嚴格遵循儀表板設計原則 (清晰層級、避免雜亂)<sup>31</sup>，提供專業的暗黑模式 (避免純黑純白)<sup>78</sup>，並完整實施無障礙 (A11y) 規範 (鍵盤導覽、ARIA)<sup>69</sup>。

### 9.3 戰略劇本 3：高效能數據科學 (Data Science) 堆疊

- 目標：展示「大數據集」(數十萬至數百萬點)或 3D/地理空間數據。

- 技術棧：
  - 選項 A (2D/3D 大數據)：ECharts<sup>58</sup> 並開啟其 WebGL 渲染模式 (ECharts-GL)<sup>56</sup>。
  - 選項 B (地理/3D)：Deck.gl<sup>57</sup>。
- UI/UX 重點：核心是效能。必須實施 Canvas 效能最佳化(如離線渲染、最小化重繪)<sup>41</sup>，並充分利用 GPU 加速<sup>52</sup>。

## 9.4 最終建議：從「開發者」到「溝通者」的思維轉變

面向大眾的傳播，其最終的衡量標準不是技術的複雜度，而是溝通的清晰度 (Clarity) 與體驗的同理心 (Empathy)。

開發團隊必須轉變思維：您不僅是在「開發功能」，更是在「設計一場對話」。您選擇的 Scrolltelling 技術是為了引導用戶的視線，您選擇的 WebGL 渲染器是為了流暢地呈現真相，您實施的無障礙規範 (a11y) 則是為了確保這場對話對所有人開放。技術，是實現清晰溝通與同理心的最終手段。

### Works cited

1. Interactive Magic: Best Digital Publishing Platforms for Engagement - Kitaboo, accessed November 5, 2025, <https://kitaboo.com/interactive-digital-publishing/>
2. 10 best digital publishing platforms for 2025 - Shorthand, accessed November 5, 2025, <https://shorthand.com/the-craft/best-digital-publishing-platforms/index.html>
3. UI、UX是什麼？了解UI UX 差別在哪裡 - EG網頁設計, accessed November 5, 2025, <https://www.eg-creative.com/what-is-uiux/>
4. UI/UX 分別是什麼？它們有什麼差異？ - 網頁設計公司, accessed November 5, 2025, <https://www.wakeup.com.tw/uiux-difference/>
5. 網頁設計和UI怎麼協作提升UX？技術數據帶你深入了解 - 從心設計F5 Design, accessed November 5, 2025, <https://www.f5design-tw.com/post/how-do-web-design-and-ui-work-together-to-improve-ux>
6. The Best Digital Publishing Platform Comparison for 2025 - StoryChief, accessed November 5, 2025, <https://storychief.io/blog/digital-publishing-platform-comparison>
7. 17 Best Digital Publishing Platforms in 2024, accessed November 5, 2025, <https://www.stateofdigitalpublishing.com/digital-platform-tools/best-digital-publishing-platform/>
8. Create outstanding websites without coding - Readymag, accessed November 5, 2025, <https://readymag.com/readymag/4208284/>
9. Readymag Review 2025: Is This No-Code Website Builder Worth It for Designers?, accessed November 5, 2025,

<https://www.myfirstwebsite.com/readymag-review-2025>

10. Guide to presentation design by Readymag, accessed November 5, 2025,  
<https://readymag.com/webspecials/presentation-guide/tools/>
11. Is Readymag the Ultimate Portfolio Tool for Designers? | by Ales Nesetril - Medium, accessed November 5, 2025,  
<https://medium.com/design-insights/is-readymag-the-ultimate-portfolio-tool-for-designers-f4418dde4ead>
12. From concept to launch: How to build a website from scratch - Readymag Blog, accessed November 5, 2025,  
<https://blog.readymag.com/how-to-build-a-website-from-scratch/>
13. No-code design—the ultimate choice for swift creative processes - Readymag Blog, accessed November 5, 2025,  
<https://blog.readymag.com/no-code-design-the-ultimate-choice-for-swift-creative-processes/>
14. Animation | Readymag, accessed November 5, 2025,  
<https://readymag.com/designs/animation/>
15. Interactivity with Readymag: 6 features to catch all eyes, accessed November 5, 2025, <https://blog.readymag.com/interactivity-features-to-catch-all-eyes/>
16. Animation - Readymag Help, accessed November 5, 2025,  
<https://help.readymag.com/hc/en-us/articles/360020527292-Animation>
17. Underrated design portfolio formats that can help get you hired - Readymag Blog, accessed November 5, 2025,  
<https://blog.readymag.com/underrated-design-portfolio-formats/>
18. What is ScrollTelling? [A Complete Guide with 15 Great Examples] - KrishaWeb, accessed November 5, 2025,  
<https://www.krishaweb.com/blog/what-is-scrolltelling/>
19. How social media is transforming the way we tell stories, accessed November 5, 2025,  
<https://thefix.media/2024/10/09/how-social-media-is-transforming-the-way-we-tell-stories/>
20. 12 engaging scrolltelling examples to inspire your content - Shorthand, accessed November 5, 2025,  
<https://shorthand.com/the-craft/12-engaging-scrolltelling-examples-to-inspire/>
21. Evolution of UI/UX Design and ScrollTelling in UI/UX - PAYODA, accessed November 5, 2025,  
<https://www.payoda.com/evolution-of-ui-ux-design-and-scrolltelling-in-ui-ux/>
22. Scrolltelling Implementation for Data Narrative Visualization - Dev3lop, accessed November 5, 2025,  
<https://dev3lop.com/scrolltelling-implementation-for-data-narrative-visualisation/>
23. Scrolltelling design variations - data.europa.eu, accessed November 5, 2025,  
<https://data.europa.eu/apps/data-visualisation-guide/scrolltelling-design-variations>
24. 10 Outstanding Data Projects Win the 2024 Sigma Awards – Global ..., accessed November 5, 2025,

<https://gijn.org/stories/data-projects-sigma-awards-2024-competition/>

25. Our favorite data stories: how Guardian US visualized 2022, accessed November 5, 2025,  
<https://www.theguardian.com/us-news/2022/dec/30/guardian-data-stories-visualized-2022>
26. 11 examples of impressive digital stories - Shorthand, accessed November 5, 2025, <https://shorthand.com/the-craft/examples-of-impressive-digital-stories/>
27. Top Creative Data Visualization Examples of 2024 - Infogram, accessed November 5, 2025,  
<https://infogram.com/blog/top-data-visualization-examples-2024/>
28. 怎样选择合适的图表类型？确保数据有效呈现。 - FineReport报表知识库, accessed November 5, 2025,  
<https://www.finereport.com/blog/article/68592ab5d2527e0eb72bed32>
29. Designing Charts — Principles Every Designer Should Know | by Ryan Bales, accessed November 5, 2025,  
<https://uxdesign.cc/designing-charts-principles-every-designer-should-know-5bd3969a0150>
30. Data Visualization – How to Pick the Right Chart Type? - eazyBI, accessed November 5, 2025, <https://eazybi.com/blog/data-visualization-and-chart-types>
31. Effective Dashboard Design Principles for 2025 | UXPin, accessed November 5, 2025, <https://www.uxpin.com/studio/blog/dashboard-design-principles/>
32. Data visualization - Material Design, accessed November 5, 2025,  
<https://m2.material.io/design/communication/data-visualization.html>
33. The Ultimate Data Visualization Handbook for Designers | by UX Magazine - Medium, accessed November 5, 2025,  
<https://uxmag.medium.com/the-ultimate-data-visualization-handbook-for-designers-efa7d6e0b6fe>
34. SVG versus Canvas: Which technology to choose and why? - JointJS, accessed November 5, 2025, <https://www.jointjs.com/blog/svg-versus-canvas>
35. Canvas vs SVG: Choosing the Right Tool for the Job - SitePoint, accessed November 5, 2025, <https://www.sitepoint.com/canvas-vs-svg/>
36. Optimising HTML5 Canvas Rendering: Best Practices and Techniques - AG Grid Blog, accessed November 5, 2025,  
<https://blog.ag-grid.com/optimising-html5-canvas-rendering-best-practices-and-techniques/>
37. SVG Vs. Canvas: A Comparison - Medium, accessed November 5, 2025,  
<https://medium.com/stackanatomy/svg-vs-canvas-a-comparison-1b58e6c84326>
38. 【前端可视化】Canvas 和SVG 区别原创 - CSDN博客, accessed November 5, 2025, <https://blog.csdn.net/IAIPython/article/details/130571681>
39. SVG vs Canvas / David B. - Observable, accessed November 5, 2025,  
<https://observablehq.com/@bumbeishvili/svg-vs-canvas>
40. Improving HTML5 Canvas performance | Articles - web.dev, accessed November 5, 2025, <https://web.dev/articles/canvas-performance>
41. Optimizing canvas - Web APIs | MDN, accessed November 5, 2025,  
[https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API/Tutorial/Optimizin](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Optimizin)

## g\_canvas

42. HTML5 Canvas Performance and Optimization Tips, Tricks and Coding Best Practices, accessed November 5, 2025,  
<https://stackoverflow.com/questions/8205828/html5-canvas-performance-and-optimization-tips-tricks-and-coding-best-practices>
43. Improving performance of HTML5 canvas with thousands of rectangles drawn per frame, accessed November 5, 2025,  
<https://stackoverflow.com/questions/77943338/improving-performance-of-html5-canvas-with-thousands-of-rectangles-drawn-per-frame>
44. How to Use WebGL for Interactive Animations - PixelFreeStudio Blog, accessed November 5, 2025,  
<https://blog.pixelfreestudio.com/how-to-use-webgl-for-interactive-animations/>
45. Getting started with WebGL - Web APIs | MDN, accessed November 5, 2025,  
[https://developer.mozilla.org/en-US/docs/Web/API/WebGL\\_API/Tutorial/Getting\\_started\\_with\\_WebGL](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Getting_started_with_WebGL)
46. WebGL and Data Visualisation. Curious how you can give your data data... | by Alexander Vasbotn Røyne-Helgesen | Bekk, accessed November 5, 2025,  
<https://blogg.bekk.no/webgl-and-data-visualisation-379d8252ea51>
47. How to Use WebGL for Data Visualization and 3D Charts - PixelFreeStudio Blog, accessed November 5, 2025,  
<https://blog.pixelfreestudio.com/how-to-use-webgl-for-data-visualization-and-3d-charts/>
48. Visualizing Graphs in 3D with WebGL | by Michael Hunger | Neo4j Developer Blog | Medium, accessed November 5, 2025,  
<https://medium.com/neo4j/visualizing-graphs-in-3d-with-webgl-9adaaff6fe43>
49. Top 10 digital publishing trends for 2024 - Blockthrough, accessed November 5, 2025, <https://blockthrough.com/blog/top-10-digital-publishing-trends-for-2024/>
50. WebGL: 2D and 3D graphics for the web - Web APIs - MDN Web Docs, accessed November 5, 2025,  
[https://developer.mozilla.org/en-US/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API)
51. 7 Powerful Open-Source WebGL Data Visualization Tools for 2025 - Cybergarden, accessed November 5, 2025,  
<https://cybergarden.au/blog/7-powerful-open-source-webgl-data-visualization-tools-2025>
52. deck.gl Script Gallery | WebGL visualization, accessed November 5, 2025,  
<https://deck.gl/gallery>
53. Examples - Apache ECharts, accessed November 5, 2025,  
<https://echarts.apache.org/examples/en/index.html>
54. D3 or Chart.js for Data Visualisation?, accessed November 5, 2025,  
<https://www.createwithdata.com/d3js-or-chartjs/>
55. Comparison between d3.js and chart.js (only for charts) [closed] - Stack Overflow, accessed November 5, 2025,  
<https://stackoverflow.com/questions/27347798/comparison-between-d3-js-and-chart-js-only-for-charts>
56. D3 by Observable | The JavaScript library for bespoke data visualization,

- accessed November 5, 2025, <https://d3js.org/>
- 57. Unlock 2024's Best JavaScript Libraries for Data Visualization - vizGPT, accessed November 5, 2025,  
<https://vizgpt.ai/docs/blog/top-5-data-visualization-javascript-libraries-in-2024>
  - 58. Chart.js vs D3.js - Medium, accessed November 5, 2025,  
<https://medium.com/@mintholic1/chart-js-vs-d3-js-45e9abe1f08c>
  - 59. Chart.js | Open source HTML5 Charts for your website, accessed November 5, 2025, <https://www.chartjs.org/>
  - 60. Animations | Chart.js, accessed November 5, 2025,  
<https://www.chartjs.org/docs/latest/configuration/animations.html>
  - 61. Top 5 Chart Libraries to use in Your Next Project - Strapi, accessed November 5, 2025, <https://strapi.io/blog/chart-libraries>
  - 62. JavaScript Chart Libraries In 2026: Best Picks + Alternatives | Luzmo, accessed November 5, 2025, <https://www.luzmo.com/blog/javascript-chart-libraries>
  - 63. Highcharts - Interactive Charting Library for Developers, accessed November 5, 2025, <https://www.highcharts.com/>
  - 64. How to make interactive charts accessible | Deque, accessed November 5, 2025,  
<https://www.deque.com/blog/how-to-make-interactive-charts-accessible/>
  - 65. Accessibility module - Highcharts, accessed November 5, 2025,  
<https://highcharts.com/docs/accessibility/accessibility-module>
  - 66. 20 Best JavaScript Data Visualization Libraries in 2024 - Fusioncharts.com, accessed November 5, 2025,  
<https://www.fusioncharts.com/blog/top-20-javascript-libraries-for-data-visualizat ion-for-2022/>
  - 67. Top 7 Data Visualization Libraries in 2024 [Open-source] - DataHen, accessed November 5, 2025,  
<https://www.datahen.com/blog/best-data-visualization-libraries-in-2024/>
  - 68. 20 Filter UI Examples for SaaS: Design Patterns & Best Practices - Arounda, accessed November 5, 2025, <https://arounda.agency/blog/filter-ui-examples>
  - 69. 10 UI/UX Design Trends That Will Dominate 2025 & Beyond - BootstrapDash, accessed November 5, 2025,  
<https://www.bootstrapdash.com/blog/ui-ux-design-trends>
  - 70. Data Visualization Design: Dark vs Light modes in HappyFox BI, accessed November 5, 2025,  
<https://blog.happyfox.com/data-visualization-design-dark-vs-light-modes-in-hap pyfox-bi/>
  - 71. ui-ux 2025 design trends - by Kashaf Maryam khan - Medium, accessed November 5, 2025,  
<https://medium.com/@kashafmaryamkhan/ui-ux-2025-design-trends-fb572555c057>
  - 72. Designing Power BI Dashboards in Dark Mode - Numerro, accessed November 5, 2025, <https://www.numerro.io/blog/designing-dashboard-in-dark-mode>
  - 73. Dark Mode Dashboard Design Principles - Qodequay, accessed November 5, 2025, <https://www.qodequay.com/dark-mode-dashboards>
  - 74. 2025 Inspiration UI Design: Top 10 Trends - Bookmarkify, accessed November 5,

- 2025, <https://www.bookmarkify.io/blog/inspiration-ui-design>
75. How to shape data visualisation for mobile | by Xwerx | UX Design by Xwerx - Medium, accessed November 5, 2025,  
<https://medium.com/xwerx-ideas/how-to-shape-data-visualisation-for-mobile-b7adb032f5d9>
76. Responsive Data Visualization: Techniques That Scale | by Think Design | Medium, accessed November 5, 2025,  
<https://medium.com/@marketingtd64/responsive-data-visualization-techniques-that-scale-e707a3e7c03c>
77. Responsive web design basics | Articles | web.dev, accessed November 5, 2025,  
<https://web.dev/articles/responsive-web-design-basics>
78. What is Mobile First? — updated 2025 | IxDF - The Interaction Design Foundation, accessed November 5, 2025,  
<https://www.interaction-design.org/literature/topics/mobile-first>
79. Guidance on Web Accessibility and the ADA - ADA.gov, accessed November 5, 2025, <https://www.ada.gov/resources/web-guidance/>
80. How Accessibility Standards Can Empower Better Chart Visual ..., accessed November 5, 2025,  
<https://www.smashingmagazine.com/2024/02/accessibility-standards-empower-better-chart-visual-design/>
81. Charts & Accessibility - Penn State | Accessibility, accessed November 5, 2025, <https://accessibility.psu.edu/images/charts/>
82. The Ultimate Checklist for Accessible Data Visualisations - The A11Y Collective, accessed November 5, 2025,  
<https://www.a11y-collective.com/blog/accessible-charts/>
83. Designing Accessible Visualizations for Screen Readers - Dev3lop, accessed November 5, 2025,  
<https://dev3lop.com/designing-accessible-visualizations-for-screen-readers/>
84. Accessible Rich Internet Applications (WAI-ARIA) 1.2 - W3C, accessed November 5, 2025, <https://www.w3.org/TR/wai-aria-1.2/>
85. Data Visualizations - Technology Accessibility - The University of Alabama, accessed November 5, 2025,  
<https://accessibility.ua.edu/accessibilityresources/accessible-data-visualizations/>
86. An intro to animating with the Web Animations API - Mainmatter, accessed November 5, 2025,  
<https://mainmatter.com/blog/2021/01/29/web-animations-intro/>
87. Orchestrating Complexity With Web Animations API - Smashing Magazine, accessed November 5, 2025,  
<https://www.smashingmagazine.com/2021/09/orchestrating-complexity-web-animations-api/>
88. Web Animations API - MDN Web Docs - Mozilla, accessed November 5, 2025, [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Animations\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Animations_API)
89. Using the Web Animations API - MDN Web Docs, accessed November 5, 2025, [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Animations\\_API/Using\\_the\\_Web\\_Animations\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Animations_API/Using_the_Web_Animations_API)

90. Animating SVGs in React with the Native Web Animations API (No Libraries Needed), accessed November 5, 2025,  
<https://dev.to/hexshift/animating-svgs-in-react-with-the-native-web-animations-api-no-libraries-needed-25km>
91. Creating data stories with ScrolllyTeller | by Ryan Shackleton | IHME Tech - Medium, accessed November 5, 2025,  
<https://medium.com/ihme-tech/creating-data-stories-with-scrolllyteller-601a34327545>