

香港中學文憑試
資訊及通訊科技
SBA

電腦應用自學網站

余承熹

目錄

1. 背景.....	4
1.1 主題和目的	4
1.2 內容概述	4
2. 網站檔案結構概述.....	5
3. 具體網站檔案結構.....	6
3.1 網站大體架構.....	6
3.2 主頁 (index.html)	7
網站主頁頁首	7
網站風格	7
頁首	7
3.3 網站主頁內容	8
輪播圖.....	8
3.4 網站主頁頁尾.....	9
網站地圖	9
聲明.....	9
3.4 導航欄.....	10
靠南懸浮響應式動態垂直導航欄.....	10
頁首傳送按鈕	10
3.5 各教程傳送港口 (xx-link.html)	11
電腦軟硬件常識教學 (待完成)	11
編程教學 (待完成)	11
遊戲常識教學 (半完成)	12
Cheat Engine 的基本操作指南 (已完成):	12
Nexus 模組教程 (待完成):	12
免越獄安裝絕版遊戲的方法 (待完成):	12
解決 Xbox 遊戲服務下載問題的方法 (已完成):	12
4. 網站重點技術詳解.....	13
4.1 整體樣式設定	13
4.2 頁首設定	14
頁首難點/實現方法.....	14
4.3 輪播圖設定	15
輪播圖難點/實現方法.....	15
4.4 “返回頁首” 按鈕.....	16
“返回頁首” 按鈕難點/實現方法.....	16
4.5 教程表格	17

教程表格難點/實現方法	17
4.7 較佳的教程體驗（我認為的）	18
亮點.....	18
自動對齊系統難點/實現方法	19
窗口遮罩系統難點/實現方法	19
引導動畫難點/實現方法	20
Spoiler 補充效果	21
影片封面（persona 4）	22
自動評分系統	23
自動評分系統 2 號	25
滾動按鈕	28
4.5 靠南懸浮響應式動態垂直導航欄.....	29
亮點.....	29
旋轉瀏覽器圖標彩蛋.....	30
運作邏輯（抽屜效果）	32
運作邏輯（響應式）	33
4.6 網站響應式設計	34
5. 結論.....	35
5.1 網站瑕疵	35
5.2 網站製作時間（大概長度）	37
6. 參考資料/工具	38

1. 背景

該自學網站致力於提供電腦軟硬件、編程以及遊戲相關知識的學習資源。目標是幫助電腦用戶學習和掌握這些領域的核心概念和技能。

1.1 主題和目的

網站主題涵蓋了多個方面，包括電腦軟硬件、編程和遊戲相關知識。這些主題基於我們的經驗和興趣，並且我相信這些領域對於現代電腦用戶來說非常重要。

網站的目的是提供一個易於理解和學習的平台，讓使用者能夠在自己的舒適範圍內自主學習。我們相信自學是一種強大的學習方式，它可以讓人們根據自己的節奏和興趣來學習，同時培養解決問題和自我學習的能力，這就是該網站誕生的理由。

1.2 內容概述

網站涵蓋了多個領域，以下是它主要提供的內容：

1. 電腦應用：網站提供了如電腦管理員的電腦基礎應用教學，目標是讓用戶獲得對電腦最基本的認識，並且有效的管理、監控，與操控自己電腦的軟硬件。（待完成）
2. 編程教學：網站提供如 HTML、CSS 和 JS 的編程教學，目標是讓用戶可以通過這些教學學習如何建立簡單的網頁，並獲得對網頁開發的基礎理解。（待完成）
3. 遊戲教學：網站提供遊戲的修改與模組教程，即如何利用 Cheat Engine 來修改遊戲數據，以及透過模組給遊戲作出優化或新增功能的方法，最終獲得更高的遊戲性。此外，爲了消除世俗對遊戲的偏見，並且為遊戲修改去污名化，該部分會夾帶作者對遊戲的想法，希望能夠帶來啟蒙，引發用戶對遊戲的思考。（半完成）

2. 網站檔案結構概述

網站採用簡單直觀的檔案結構，以便讓我和其他可能的“程序猿”可以清晰地理解和管理網站的資源和內容。 以下是網站的主要檔案結構介紹：

- 1 index.html: 這是網站的首頁，包含了網站的簡介，並且不論在網頁頂部、中部，或是頁尾也能找到鏈接到網站內其他網頁的超鏈接，以提供用戶流暢的瀏覽體驗。
- 2 css/: 在這個資料夾中，存放了所有應用在不同 html 檔的樣式表，方便統一管理，以下是主要的樣式表：
 - 2.1 stylesheet.css： 整個網站樣式的骨幹，統一了所有網站中網頁的基本佈局與風格。
 - 2.2 swiper-bundle.min.css： 鑒於作者我沒能力利用已知 JS 知識獨立製作輪播圖，故引用外部[工具](#)來製作，而該樣式表正是該工具的樣式表。
 - 2.3 gaming-link.css： 為教程內容作排版的 css，原本預計只作用於遊戲相關教程鏈接的頁面，故取名 gaming-link。後來爲了統一全教程鏈接頁面樣式，並避免作用重疊的樣式表，故引入在所有教程鏈接頁面。
 - 2.4 navi.css： 定義“靠南懸浮響應式動態垂直導航欄”的樣式表，蘊含作者我一整個星期心血的巔峰之作！靈感出自於那一部[影片](#)。
- 3 js/: 在這個資料夾中，存放所有與網站互動和功能相關的 JavaScript 檔案。這些檔案：
 - 3.1 ce-quiz.js： 遊戲教程測驗小程序的代碼，只利用了 if/else、prompt 和 alert 等簡單的 function，也是所有 JS 中唯一我能獨立應付的項目…
 - 3.2 rotate.js： “靠南懸浮響應式動態垂直導覽列”的小彩蛋。
 - 3.3 typing.js： 作者介紹的彩蛋，靈感出至於“喪坤危機”。
- 4 pics\my_pics： 這個文件夾存放網站所需的圖片檔案。 你可以在這裡組織和管理網站使用的圖片資源，例如圖標、背景圖片和內容中使用的圖片。
- 5 videos\my_video： 在這個資料夾中，存放了網站使用的所有影片檔案。

3. 具體網站檔案結構

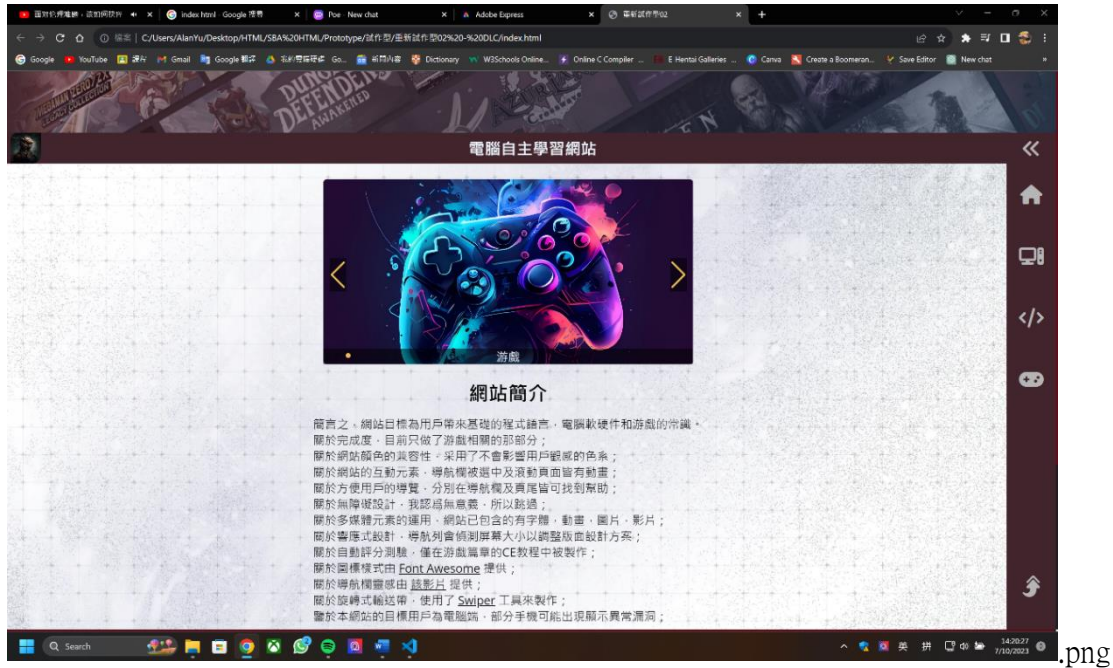
3.1 網站大體架構

如下：

```
.
├── index.html (網站主頁)
├── html/
│   └── (.html x11)
├── css/
│   └── (.css x8)
├── js/
│   └── (.js x10)
├── pics/
│   └── my_pics/
│       └── (圖片 x29)
├── videos/
│   └── my_video/
│       └── (影片 x3)
└── readme.docx (該報告)
```

3.2 主頁（index.html）

● 網站主頁頁首



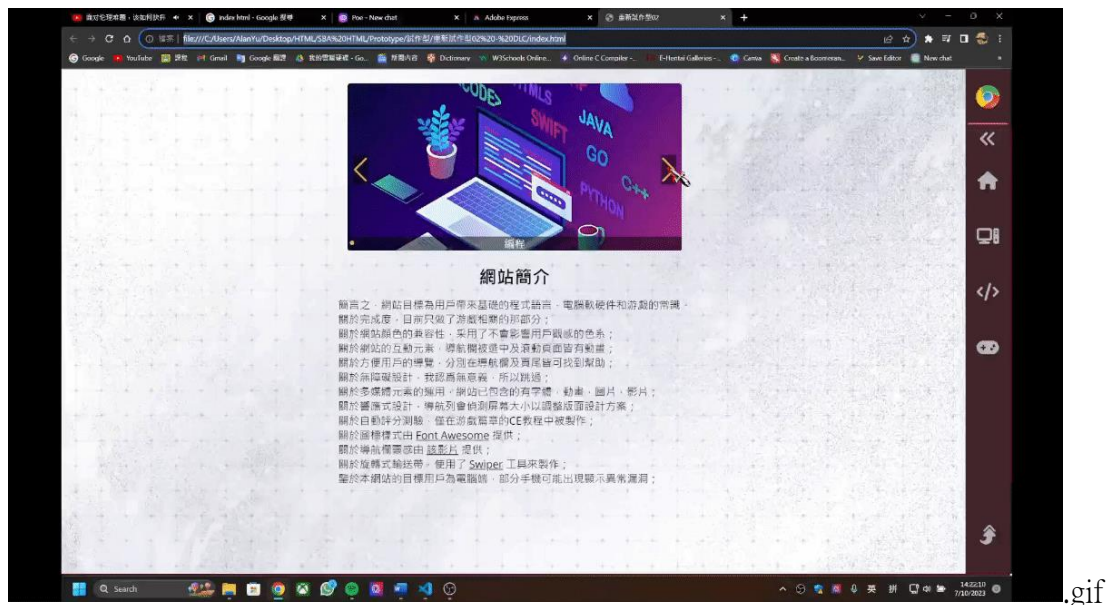
● 網站風格

以酒紅色為網站界面主色調。紅色引人注目，藍色象徵沉穩，兩者結合即成為富含皇家氣息的酒紅色，也能突出作為本網站的重點方向——電腦作為生產力與遊戲平臺的王者風範。

● 頁首

1. 使用以酒紅色為主色調的遊戲相關橫幅，以暗示網頁主力製作的方向，也反映著網站作者的個人興趣。
2. 主題橫板右側以 Midjourney 生成的肖像取代網站作者真實樣貌，以避免不必要的麻煩，並提高私隱係數。
3. 主題橫板中間設有標題以提醒用戶目前所在網站位置及網頁內容。

3.3 網站主頁內容



● 輪播圖

利用[外部工具](#)製作的“垂直自動循環切換輪播圖”配有三張對應教程圖片；配有分頁器；配有導航按鈕；沒有滾動條；每三千毫秒自動切換。網站簡介表簡單概括陳述了網站一切，即：

初衷：網站目標為用戶帶來基礎的程式語言，電腦軟硬件和遊戲的常識。

完成度：只做了遊戲相關的那部分（一小部分）。

色系：採用了不會影響用戶觀感的色系作為背景圖片；酒紅色作為界面色系。

互動元素：導航欄被選中及滾動頁面皆有動畫。

導航欄：分別在導航欄及頁尾皆可找到幫助。

無障礙設計：作者我無意義，所以跳過。

多媒體元素的運用：已包含的有字體，動畫，圖片，影片。

響應式設計：導航列會偵測屏幕大小以調整版面設計方案。

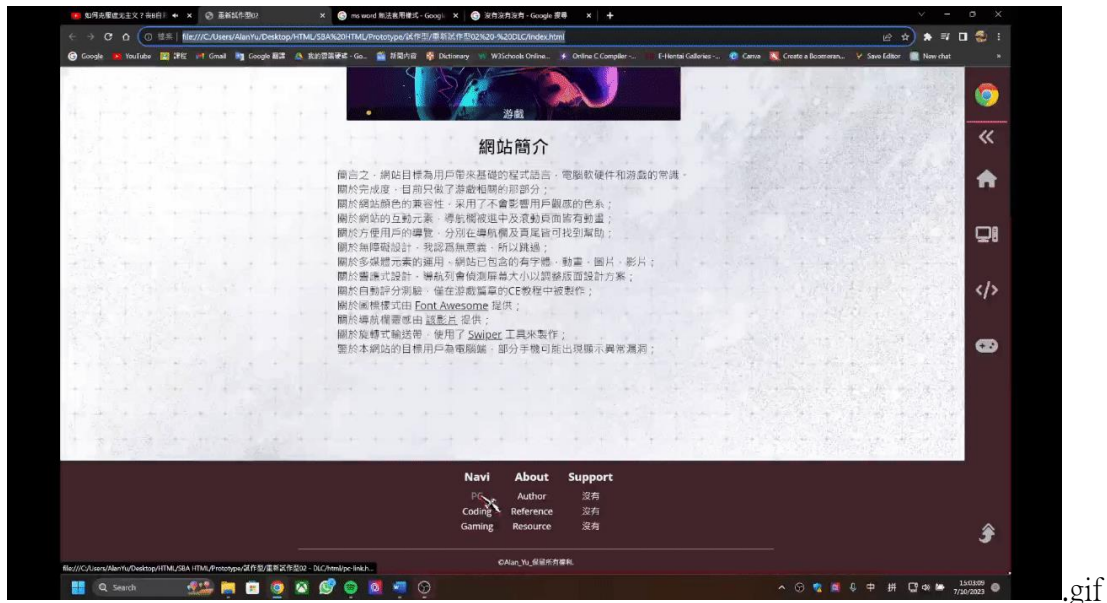
自動評分測驗：僅在遊戲篇章的 CE 教程中被製作。

圖標樣式來源：[Font Awesome](#)

旋轉式輸送帶：[Swiper](#)

目標平臺：目標用戶為電腦端，部分手機可能出現顯示異常漏洞。

3.4 網站主頁頁尾



● 網站地圖

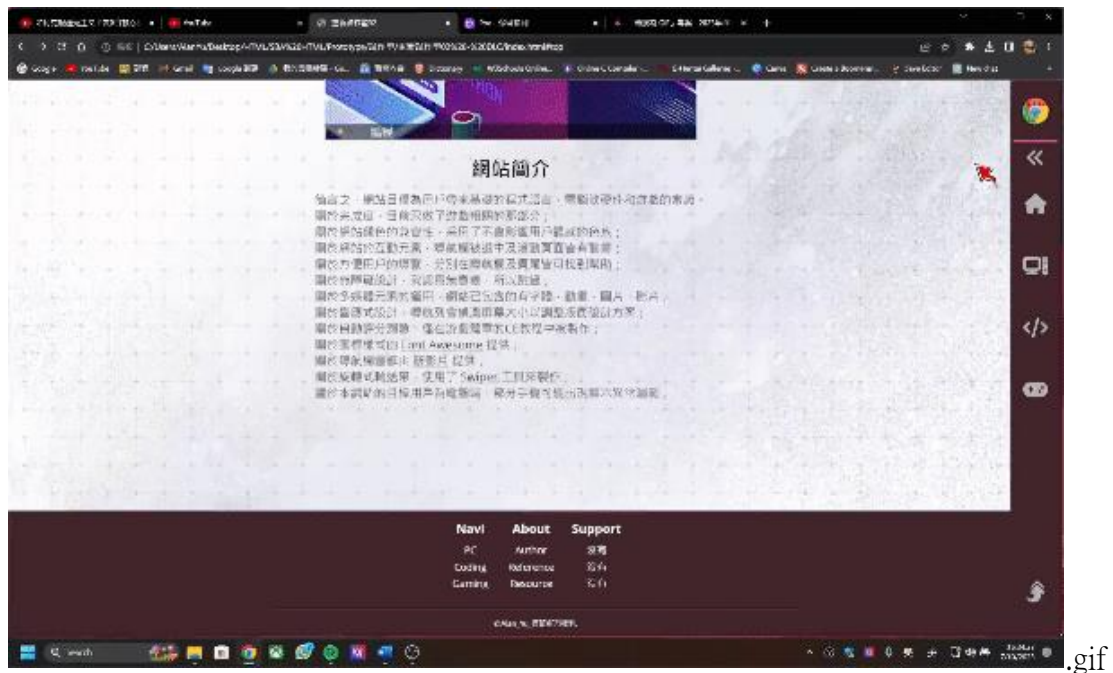
使用了懸停效果的網站地圖傳送點，方便用戶快速傳送到網站中不同的網頁：

- 1 Navi（導航）
 - 1.1 PC：傳送至軟硬件教程頁面
 - 1.2 Coding：傳送至編程教程頁面
 - 1.3 Gaming：傳送至游戲教程頁面
- 2 About（關於我）
 - 2.1 Author：傳送至作者介紹頁面
 - 2.2 Reference：傳送至參考資料頁面
 - 2.3 Resource：傳送至網站資源頁面
- 3 Support（幫助）
 - 3.1 沒有：沒有用戶支援服務

● 聲明

用於表明我網頁上的內容受到版權保護或其他法律保護，提醒用戶在未經授權的情況下不得複製、修改、分發或使用網頁上的內容。

3.4 導航欄



● 靠南懸浮響應式動態垂直導航欄

拆解“靠南懸浮響應式動態垂直導航欄”的含義，即：

1. 靠南：默認靠在右邊的靠南導航欄
2. 懸浮：懸浮在網頁窗口絕對位置而非網頁相對位置的懸浮導航欄
3. 響應式：因應視窗大小（用戶裝置）而變化的響應式導航欄
4. 動態：當鼠標用戶懸浮在按鈕時回有響應動畫的動態導航欄
5. 垂直：默認以垂直形態出現的垂直導航欄
6. 導航欄：以示其導航欄的本質

除此之外，還有一個彩蛋，即：

當用戶在頁首時，隱藏在頁首橫板背後；向下滾動後，出現用戶瀏覽該網站的瀏覽器的圖標並隨頁面進度滾動。

● 頁首傳送按鈕

實質是獨立於導航欄外的產物，但鑒於它預設被設定固定在相對於瀏覽器視窗的導航欄內的下方位置，因此分類在導航欄處。

3.5 各教程傳送港口 (xx-link.html)



● 電腦軟硬件常識教學（待完成）

- 1 任務管理員的應用
 - 1.1 任務管理員介紹
 - 1.2 開機啟動項開關教學
 - 1.3 進程暴力關閉教學
- 2 其他
 - 2.1 實用應用程序推介

● 編程教學（待完成）

1. HTML 教學
2. CSS 教學
3. JS 教學

● 遊戲常識教學（半完成）

➤ Cheat Engine 的基本操作指南（已完成）：

Cheat Engine 是一個強大的遊戲修改工具，它可以讓玩家修改遊戲中的數據和參數。在這個部分的教程中，我們將介紹 Cheat Engine 的基本操作，包括如何安裝和使用它；同時也會包含作者對修改器的見解，希望能夠消除玩家對“外挂”與“作弊”二詞語的偏見；以及最後的如何進行常見與基本的遊戲修改操作。以期望能夠幫助玩家在遊戲能自由的根據需求修改遊戲數值，以避免因為廠商不合理的數值設定與過高的肝度導致不良的遊戲體驗。最後，因應某某的請求，即便我認為沒必要，我還是勉為其難的再加多了一個制動評分測驗項目。

➤ Nexus 模組教程（待完成）：

Nexus 是目前最流行的遊戲模組分享平台，它提供各種用戶創建和分享遊戲模組的能力。在這個部分的教程中，我將介紹如何利用 Nexus 的 Vortex 軟件下載，安裝與管理模組、手動下載並安裝模組、以及提供最基本的避免模組衝突及 debug 的能力。最後期望能幫助玩家利用模組為遊戲作優化或新增功能以提升可玩性與遊戲體驗，並創造出獨屬於自己的遊戲。

➤ 免越獄安裝絕版遊戲的方法（待完成）：

我發現一些空前絕後的絕版經典遊戲神作已經被世人遺忘，並且只能在越獄設備上安裝和遊玩。所以這次，我將以無盡之劍三部曲為例，介紹一種免越獄的方法來安裝和遊玩無盡之劍，以期望透過手游模範生，即能好好利用手機畫屏特性製作玩法的遊戲，喚醒世人何為“好遊戲”，而不是那些把電腦元素摻和一些網遊抽卡氪金農素材元素就將其稱之為神作的遊戲生態毒瘤。

➤ 解決 Xbox 遊戲服務下載問題的方法（已完成）：

最近，Xbox 遊戲服務平臺因為“西瓜皮”（Xbox Game Pass）的遊戲訂閱方案被廣為人知。然而，眾所周知，Xbox PC 平臺也是出了名的問題多，有時候連遊戲也下載不了。因此，我將會在眾玩家當中最常遇到的問題下手，即提供 MGS（Microsoft Gaming Service）的下載錯誤問題的解決方法。

4. 網站重點技術詳解

4.1 整體樣式設定

代碼	解釋/作用
<pre>*{ list-style-type: none; user-select: none; padding: 0; margin: 0; }</pre>	網站所有網頁所有元素的定義： 取消列表元素的默認樣式； 文本無法被選中； 所有元素無內邊距； 所有元素無外邊距；
<pre>:root { font-size: 16px; font-family: 'Open Sans'; --text-primary: #b6b6b6; --text-secondary: #ececec; --bg-primary: #42242c; --bg-secondary: #610c1b; }</pre>	根元素樣式與自定義變量定義： 定義整個網頁的字體大小為 16px； 統一字體樣式為“Open Sans”； 定義文本顏色； 定義文本顏色； 定義背景顏色； 定義背景顏色；
<pre>html, body { width: 100vw; height: 100%; background-attachment: fixed; scroll-behavior: smooth; }</pre>	頁面的定義： 讓頁面寬度貼合視窗； 讓頁面高度隨內容變化； 固定了背景圖片位置； 設置錨點為順滑滾動；
<pre>a { text-decoration: none; user-select: none; color: inherit; }</pre>	超鏈接的定義： 取消下劃綫； 無法被選取； 繼承文本顏色；

代碼	解釋/作用
<pre>div, span, nav { display: flex; }</pre>	容器的定義： 使用彈性佈局
<pre>ul, ol { list-style-position: inside; }</pre>	列表的定義： 使得列表標記在列表容器的內部

4.2 頁首設定

頁首的樣子：



頁首橫板的圖片



● 頁首難點/實現方法

把高解析度且長寬比與容器不同的圖片放進容器的同時不壓縮圖片。

解決方案：

難點/解釋/作用	解決代碼
爲了圖片不因適應容器大小而壓縮變形，只要讓其填充容器的同時，允許超出容器邊界即可。	<pre>.header-img { object-fit: cover; }</pre>

4.3 輪播圖設定



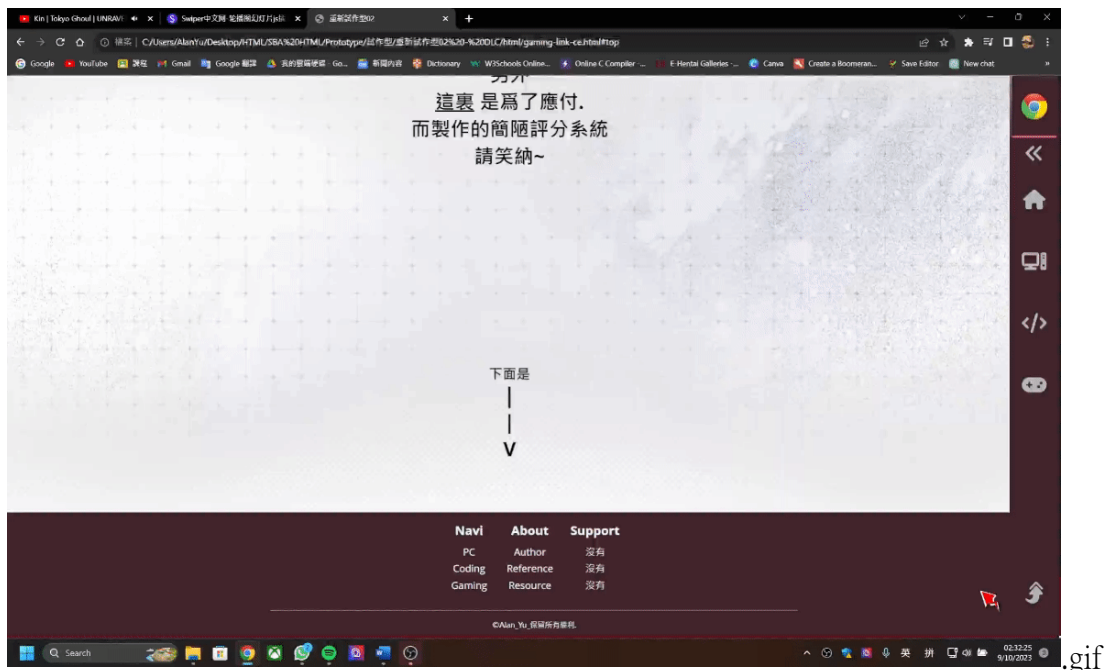
● 輪播圖難點/實現方法

製作輪播圖，在設定容器大小後，要把圖片放進去。但是，要把不同解析度的圖片以同樣的大小與位置正確的展示在輪播圖容器內是其中一個難點，以下是解決方案：

難點/解釋/作用	代碼
爲了圖片不因適應容器大小而壓縮變形，只要讓其填充容器的同時，允許超出容器邊界即可。同時，設定圓角，增加質感。 最後，flex-shrink 用來使得圖片不因收縮改變尺寸，但因為容器已經被設定長寬，因此在這個部分實質是無關緊要的。但是，作者我那時研究了好一會，所以捨不得刪除。	<pre>.swiper-slide img{ /* 其實這個無關緊要 */ flex-shrink: 0; object-fit: cover; border-radius: 5px; }</pre>

如果你問我爲啥不分析輪播圖 JS 的話，那是因爲我使用了外部工具（[Swiper](#)）來製作，沒啥好討論的，就改改參數而已。沒辦法啊，我 JS 技術還沒到那種程度。

4.4 “返回頁首” 按鈕



● “返回頁首” 按鈕難點/實現方法

實質上目的的實現並沒有難點，就只是為每個頁面設置一個錨點而已，即：

```
<a id="top"></a>
```

但是後來細節打磨無聊的加了點陰影與互動效果，所以就說說吧。

解決方案：

難點/解釋/作用	代碼
設定當游標（Cursor）懸浮在按鈕時的反應，即左上角外部與右下角外部出現陰影，並且左上角內部出現光源，以模擬立體的效果。	<pre>.top-portal:hover { background-color: var(--bg-secondary); box-shadow: inset 4px 4px 4px rgba(255, 255, 255, 0.5), -4px -4px 4px rgba(0, 0, 0, 0.25), 4px 4px 4px rgba(0, 0, 0, 0.5); }</pre>

4.5 教程表格



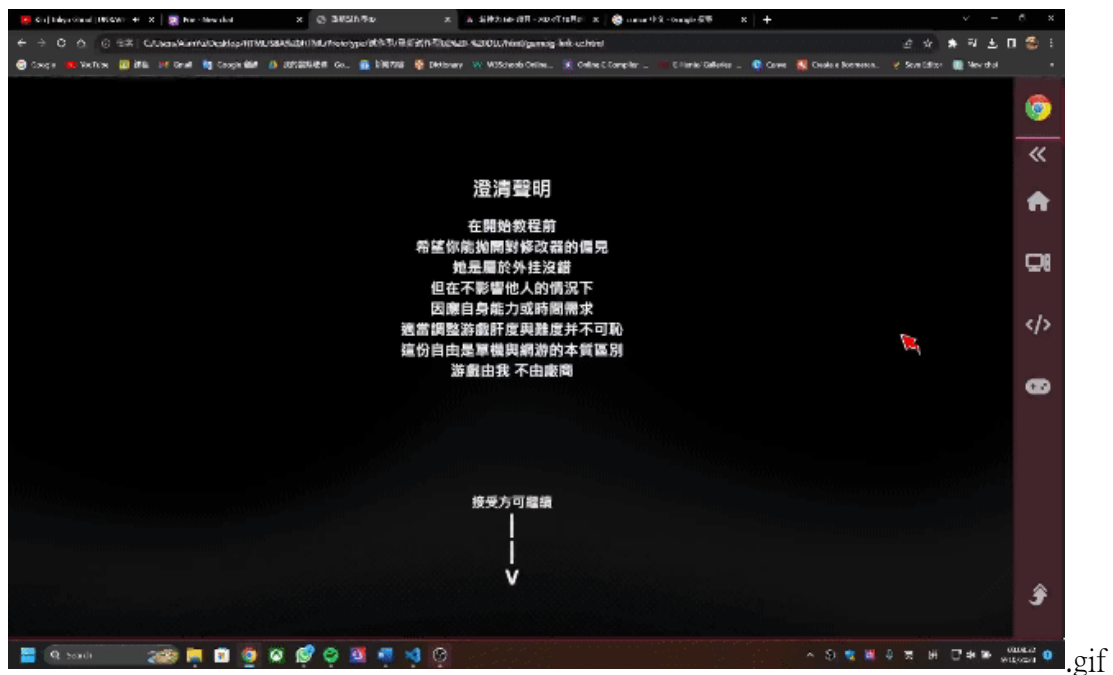
● 教程表格難點/實現方法

本來想用 div 或 span 來實現，但是如要逐一調整會非常耗時（因為數量不同），所以這次用 grid 來弄。

解決方案：

難點/解釋/作用	代碼
採用網格佈局，以更方便快捷的因應內容數量調整欄數。當中 fr 代表每個列的水平比例，并且設定每個框的空隙為 5px，實心邊框。 遺憾的是，當我照著 Chrome 完成後，到其他瀏覽器如 Edge, Firefox 測試時，竟然錯位了！我也懶得重做，就這樣吧！	<pre>.gaming-link { display: grid; grid-template-columns: 1fr 1fr; grid-template-rows: 100px 100px ; grid-gap: 5px; border-style: solid; border-color: var(--bg-primary); margin: 15px; padding: 5px; }</pre>

4.7 較佳的教程體驗（我認爲的）



● 亮點

Cheat Engine 教程，畢竟是花了我最長時間打磨的網頁，細節不能少，包括：

1. 自動對齊系統
2. 窗口遮罩系統
3. 引導動畫
4. Spoiler 補充系統
5. 影片封面（Persona 4）
6. 自動評分系統
7. 自動評分系統 2 號
8. 滾動按鈕

最後，還有教化系統（誤

● 自動對齊系統難點/實現方法

即滾動頁面後自動填滿整個窗口的系統：

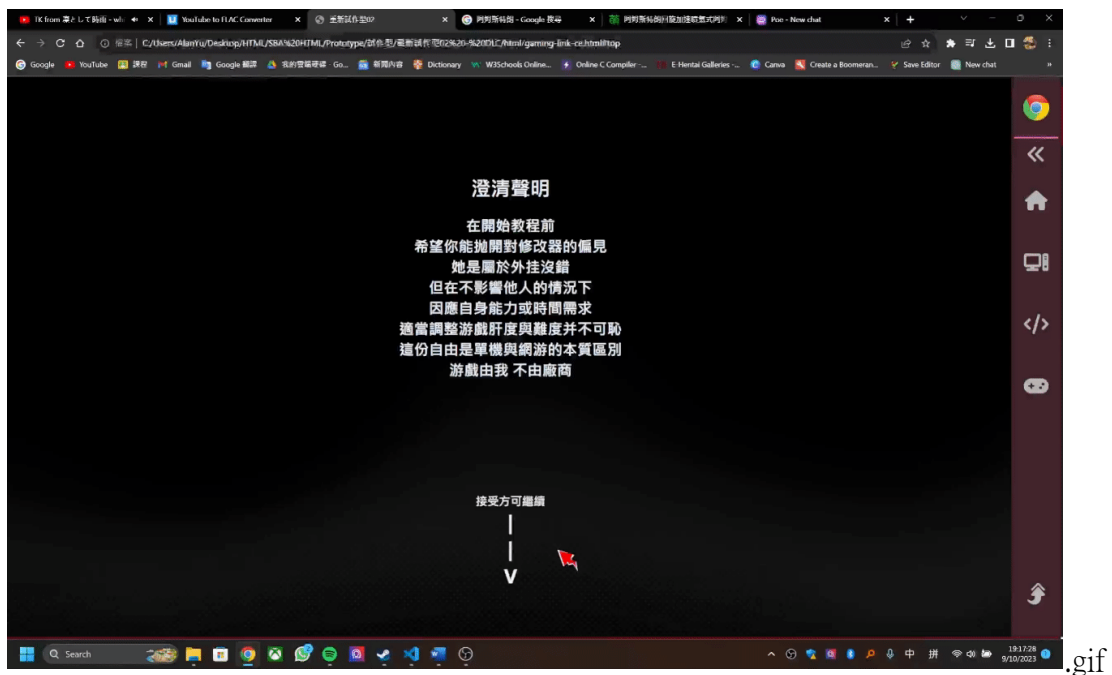
難點/解釋/作用	代碼
首先啓用網頁的垂直滾動捕捉，並強制將窗口自動對齊至最近不做點。滾動行為實質上是不用設定的，但是也捨不得刪，因此留著。	<pre>html { scroll-snap-type: y mandatory ; /* scroll-behavior: smooth; */ }</pre>
接著為每一個部分都加上捕捉點就行啦！	<pre>.header { scroll-snap-align: start; } .footer { scroll-snap-align: end; } .beginner-lv > div { scroll-snap-align: start;</pre>

● 窗口遮罩系統難點/實現方法

即教程內容上下部分的半透明漸變過渡效果：

難點/解釋/作用	代碼
爲了更好看，以及安撫原崩玩家，我特地從崩鐵網站參考來了一個遮罩的圖片。 其實不難實現，就把他們設定成固定位置，把一個固定在一個部分的底部，然後把上邊的倒過來固定在頂部，如此質感就上來啦（我覺得）。	<pre>.beginner-lv .img-transition-top { transform: rotate(180deg); position: absolute; top: 0px; width: 100%; } .beginner-lv .img-transition-bottom { transform: rotate(0deg); position: absolute; bottom: 0px; width: 100%; }</pre>

● 引導動畫難點/實現方法

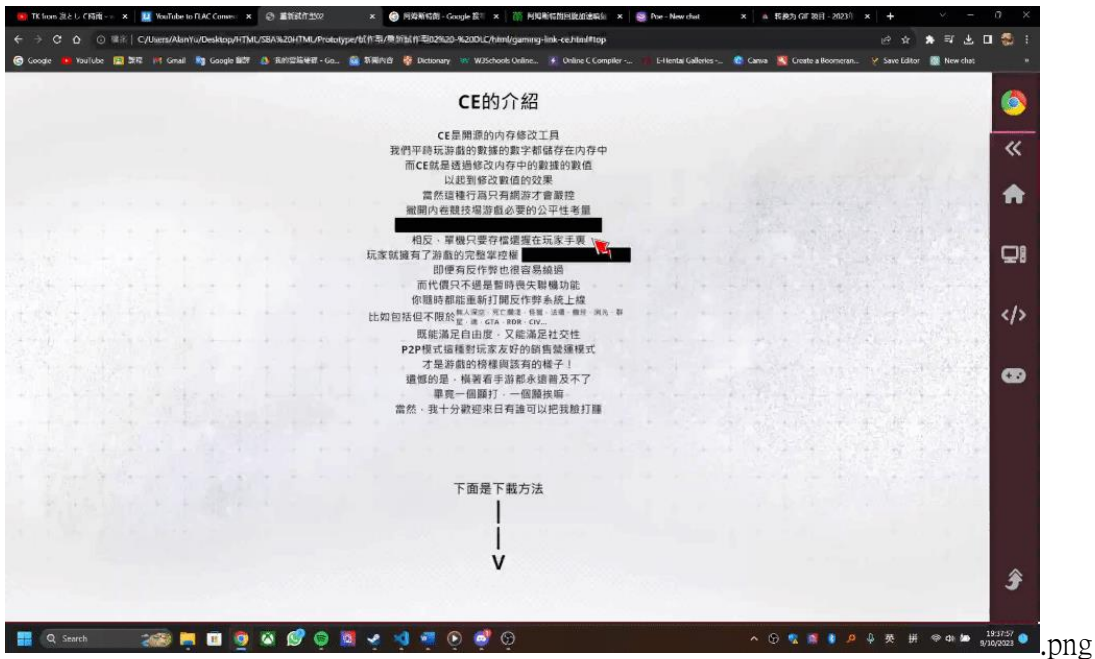


即波浪形的引導效果：

難點/解釋/作用	代碼
首先定義整個動畫的名字，然後設定動畫總時間為三秒，最後設定中心點為動畫的原點。	<pre>.anime-1 { animation-name: anime-1; animation-duration: 3s; transform-origin: center center; }</pre>
為每一個動畫設定一個延遲，以營造出波浪效果。	<pre>.anime-1-1 { animation-delay: 1.25s; } .anime-1-2 { animation-delay: 1.5s; } .anime-1-3 { animation-delay: 1.75s; } .anime-1-4 { animation-delay: 2.0s; }</pre>

最後爲了讓動畫在播完後有停頓效果，讓動畫的後半段時間靜止不動。	<pre>@keyframes anime-1 { 0% {transform: scale(1);} 25% {transform: scale(1.2);} 50% {transform: scale(1);} 100% {transform: scale(1);} }</pre>
---------------------------------	---

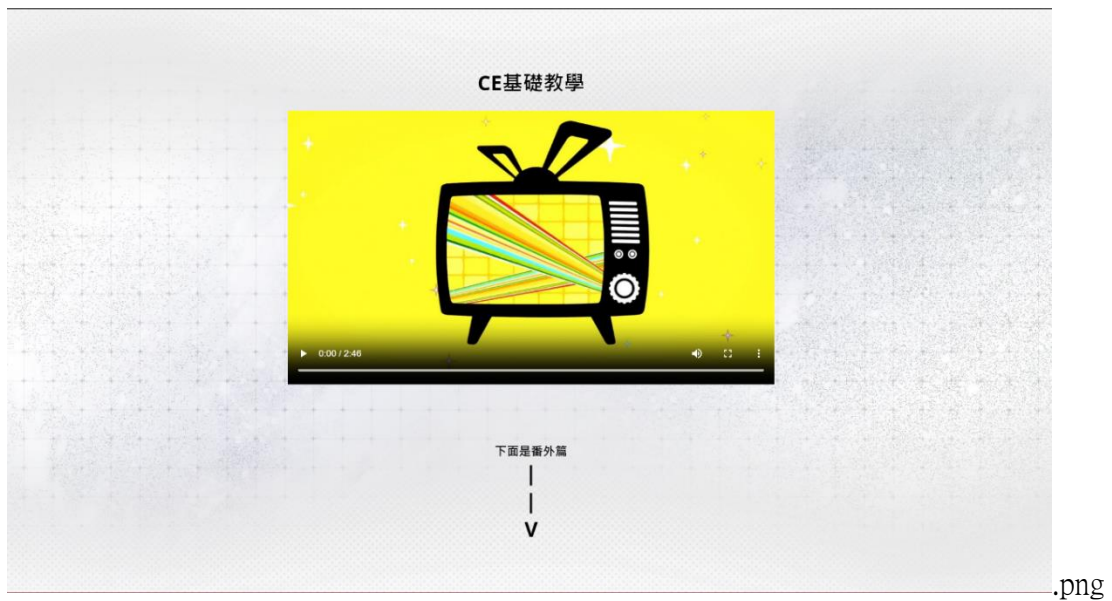
● Spoiler 補充效果



即所謂的防劇透效果：

難點/解釋/作用	代碼
<p>雖然說是防劇透效果，但我拿來當內容補充用的效果了，當然，也有防誤傷原友崩友的作用。</p> <p>其實就只是先設定背景為黑色，當光標懸浮在其上時，字體由黑變白而已。</p>	<pre>.spoiler { background-color: black; } .spoiler:hover { color: aliceblue; }</pre>

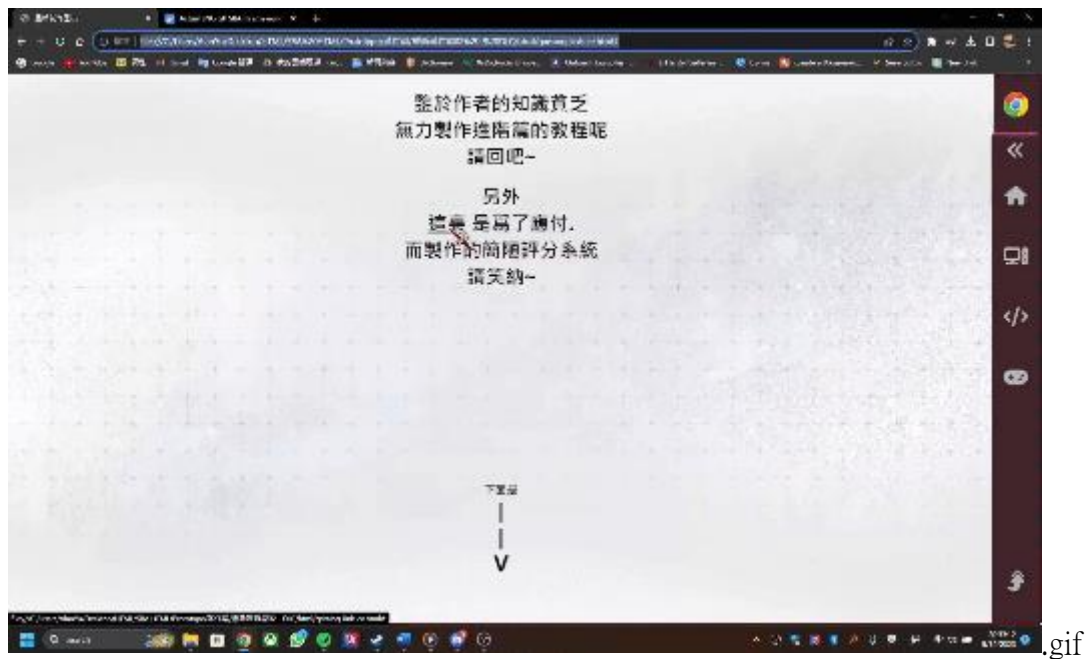
- 影片封面（persona 4）



其實這個更簡單，就是去 persona 4 截開場動畫的圖而已，其餘的甚至都不用 css，所實話，截圖比寫這代碼更難：

難點/解釋/作用	代碼
是的，單單 html 碼就實現到，首先指定影片路徑，後用 poster 指定封面圖路徑，後用 controls 顯示影片控制面板，後設定 preload 讓影片預加載。以上!	<pre> <video src="../videos/my_video/ce-recalculate- new-address.mp4" controls preload="auto" poster="../pics/my_pics/persona4- tv.jpg"> </video> </pre>

● 自動評分系統



整個 js 項目中最簡單的一個，極少函數：
（下一頁）

難點/解釋/作用	代碼
<p>函式的執行流程如下：</p> <ol style="list-style-type: none"> 1. 初始化變數 a 為 1，代表問題編號。 2. 初始化變數 b 為 0，代表分數。 3. 進入一個無窮迴圈，直到使用者輸入 '2' 停止測試。 4. 在每個迴圈中，使用 prompt 函式顯示提示訊息，詢問使用者的操作。 5. 如果使用者輸入 '1'，表示開始測試。 6. 出現第一個問題，使用 prompt 函式詢問使用者答案。 7. 根據使用者的答案判斷是否正確，如果正確則分數 b 加一。 8. 重複步驟 6 和 7，出現其他問題，直到所有問題都回答完畢。 9. 如果使用者輸入 '2'，表示停止測試，跳出迴圈。 10. 迴圈結束後，使用 alert 函式顯示最終得分。 	<pre> function quiz_1() { let a = 1; // 問題編號 let b = 0; // 分數 for (i = 0; i < 1; i++) { let d = prompt(`這裡是自動評分測試 你現在的分數為\${b} 請按照提示輸入答案或操作： 輸入 1 開始測試 輸入 2 停止測試 `); if (d === '1') { // 第一個問題 let answer1 = prompt(` 問題\${a}: 以下哪個最符合對 Cheat Engine (CE) 的描述？ A) 作弊工具 B) 內存修改工具 `); if (answer1 === 'b' answer1 === 'B') { alert(`你答對了！`); b++; // } else if (answer1 === null) { // break; } else { alert(`你答錯了！`); } a++; // 這邊重複第一個問題 } else if (d === '2') { break; } else { alert(`請輸入有效的操作！`); } } alert(`智商測試完畢！你的最終得分 為 \${b} 分！`); } </pre>

● 自動評分系統 2 號



鑑於網站之前的評分系統採用彈窗方式填寫，考慮到可能對手機用戶而言不方便及不美觀，於是在原來問卷超鏈接位置的下面增添了符合網站風格界面的問卷：

難點/解釋/作用
以下是 HTML 的其中一條問題的示例，當中 name 的設定能防止同一條問題被進行多次回答；id 的設定用於讓 JavaScript 識別正確的答案有否被選擇；當中 button 標籤的 onclick 使得點擊能觸發 JavaScript 的 Funtion。
代碼（HTML）
<pre> <input type="radio" name="question1" id="true1">內存修改工具
 <input type="radio" name="question1">作弊工具
 <button id="butt1" onclick="count_mark1(this)">點我</button></pre>

（下一頁）

難點/解釋/作用

以下是 CSS 樣式表，計分器設定為 inline 能讓 span 與 input 標籤能水平排列；按鈕設定內容超出邊界的隱藏功能；開啓動畫時長；利用貝塞爾曲線改變動畫效果，以增加碰撞效果。

按鈕設定偽元素，偽元素的内容為空，背景亮度為按鈕的 1.5 倍，層級在按鈕之下，在鼠標懸浮之前在按鈕的容器之上，並因為超出邊界被隱藏，在鼠標懸浮後，沿 y 軸延申，最終以動畫改變按鈕的效果。

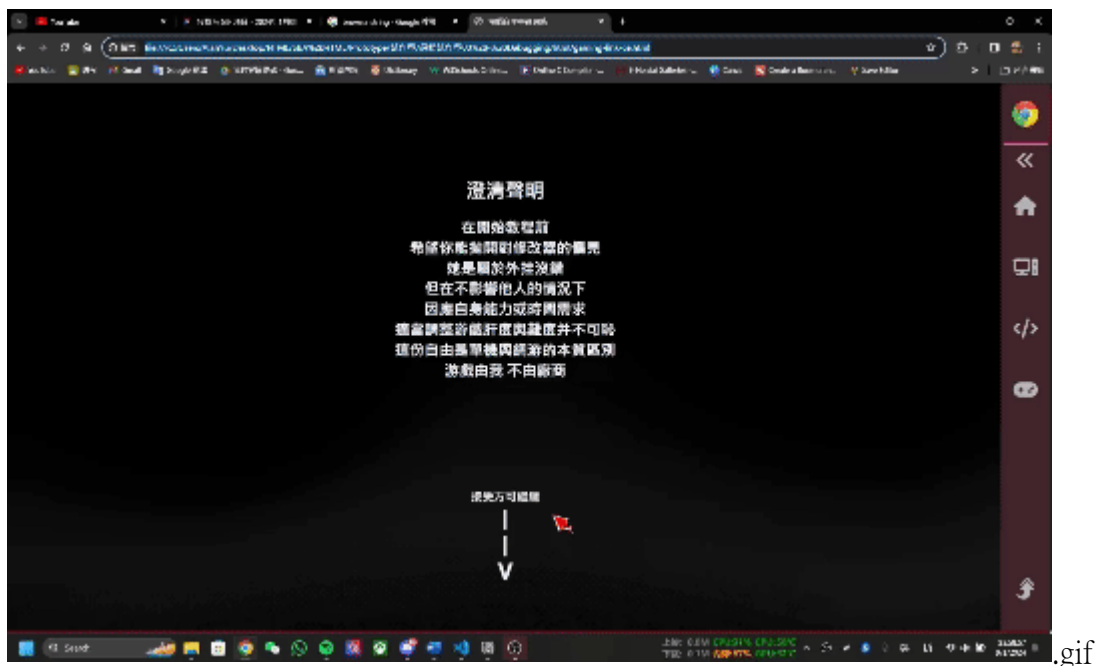
代碼 (CSS)

```
.ce-quiz span {
  display: inline
}
#butt1 {
  z-index: 10;
  overflow: hidden;
  transition: 300ms ease;
  transition-timing-function: cubic-bezier(0.5, 1.6, 0.4, 0.7);
}
#butt1::before {
  position: absolute;
  content: "";
  top: 0;
  left: 0;
  filter: brightness(150%);
  z-index: -1;
  transform-origin: 0 0;
  transform: scaleY(0);
}
#butt1:hover::before {
  /* transform: scaleX(1); */
  transform: scaleY(1);
}
```

(下一頁)

難點/解釋/作用
<p>以下是 JavaScript 的計分器：</p> <ol style="list-style-type: none"> 1. 當 button 被點擊，觸發 function 2. 定義 “result1” 等於按鈕內的數值 3. 定義 “mark” 等於 0 4. 當正確答案被選取，mark 自增 1 5. 計算總分 6. 替換按鈕的內容為 “總分為” +mark 變數+ “分”，假設總分為 5 分，按鈕的內容由原本的內容 “點我” 替換顯示為 “總分為 5 分”
代碼 (JavaScript)
<pre>function count_mark1(element) { var result1 = document.getElementById("butt1") var mark = 0; if (document.getElementById('true1').checked) { mark++; } if (document.getElementById('true2').checked) { mark++; } if (mark >= 0) { result1.innerText = '總分為' + mark + '分'; } }</pre>

● 滾動按鈕



點擊箭頭區域即可向下滾動：

難點/解釋/作用

首先選擇全部 class 為 scroll-button 的標籤 (div)，為每一個添加點擊監聽器。當被點擊，觸發 scroll 函數，以平滑滾動效果向下滾動 100px。程序當中，console.log 被用於測試點擊有否成功觸發函數。

代碼 (JavaScript)

```
document.querySelectorAll('.scroll-button').forEach(button => {
  button.addEventListener('click', function() {
    scroll();
  });
});

function scroll() {
  console.log('Scrolling...');
  window.scrollTo({
    top: 100,
    behavior: 'smooth'
  });
}
```

4.5 靠南懸浮響應式動態垂直導航欄



● 亮點

靠南懸浮響應式動態垂直導航欄，蘊含作者我一整個星期心血的巔峰之作，她理論上看是簡單，做起來卻十分繁瑣，樣式一環扣一環，先說裏頭用了哪些技術吧：

- 1 光標懸浮導航欄效果
 - 1.1 容器伸縮動畫
 - 1.2 引導標識旋轉效果
- 2 光標懸浮教程鏈接效果
 - 2.1 教程鏈接變色動畫
 - 2.2 教程鏈接陰影效果
- 3 列表的顯示邏輯
- 4 響應式設計

● 旋轉瀏覽器圖標彩蛋



平時隱藏在頁首 banner 之後，在頁面向下滾動後顯現，並且理論上會依照用戶目前瀏覽器轉換圖標：

難點/解釋/作用
<p>首先，圖標的圖層在 banner 之下，該技術曾解釋過，因此略過；圖標更換部分：首先，獲取圖片元素（用 id）。接著，獲取用戶代理訊息（瀏覽器的字串）。再接著，根據字串替換圖片路徑。當中，console.log 是用來測試檢測有否正確觸發。</p>
代碼 (JavaScript)
<pre> window.addEventListener('DOMContentLoaded', function() { var browserIMG = document.getElementById('browser-img'); var userAgent = navigator.userAgent; if (userAgent.indexOf('Firefox') != -1) { browserIMG.src = './pics/my_pics/firefox.png'; console.log('Firefox detected'); } else if (userAgent.indexOf('Chrome') != -1) { // 重複省略 // 重複省略 // 重複省略 } }); </pre>

難點/解釋/作用
<p>根據頁面的 y 軸滾動百分比達成滾動動畫：</p> <p>首先，獲取瀏覽器圖片元素。接著，獲取頁面滾動距離。再接著，計算滾動百分比。緊接著，將百分比轉換成旋轉角度。最後，應用旋轉角度於圖標。</p>
代碼（JavaScript）
<pre> window.addEventListener('scroll', () => { const image = document.querySelector('#browser-img'); const scrollTop = window.pageYOffset; const scrollPercentage = scrollTop / (document.body.scrollHeight - window.innerHeight); const rotation = scrollPercentage * 360; image.style.transform = `rotate(\${rotation}deg)` }); </pre>

悄悄話：我實際上用了兩個 js 檔來分別連結 index 以及另外在 html 文件夾內的網頁檔，因為路徑問題，兩個 js 都略有不同。後面本來想把 index.html 塞進那堆 html folder 裏，然後用 shortcut 放出來，但還是算了，懶得改鏈接。

● 運作邏輯（抽屜效果）

先說說思路吧：

- 1 創建列表（ul）
- 2 在列表內（li）填充教程圖標及描述

這個時候她看起來會是這樣：

(描述) (圖標)
(描述) (圖標)
(描述) (圖標)

那麼該如何隱藏呢？

邏輯：當光標沒懸浮在她身上時，（描述）的 display 設定為隱藏；當懸浮時，水平拓展容器的同時（描述）設定為顯示。

因為描述在左，圖標在右，所以 html 要這樣寫：

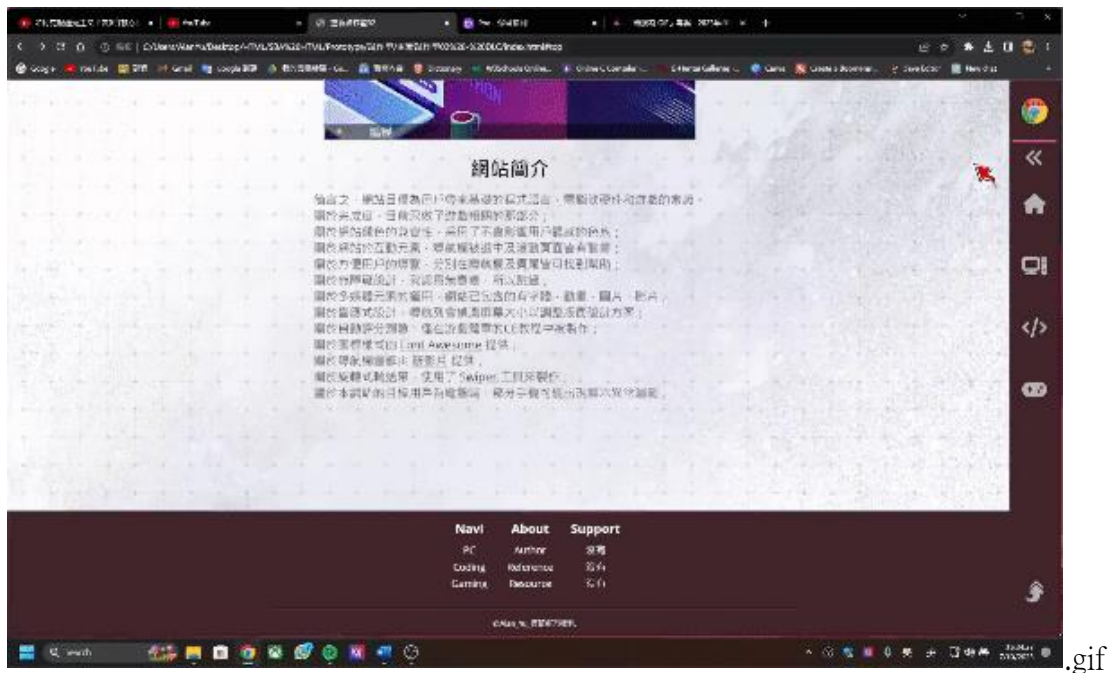
```
<li class="nav-item misc-item">
  <a href="./index.html" class="link-text">Home</a>
  <a href="./index.html" class="nav-link">
    <i class="fa-solid fa-house"></i>
  </a>
</li>
```

頂部的引導圖標則相反：

```
<li class="nav-item" id="drawer">
  <a style="color: var(--text-primary);" class="nav-link" id="logo">
    <i class="fa-solid fa-angles-left"></i>
  </a>
  <span class="link-text" id="logo-text">
    <span>這玩意真難搞</span>
  </span>
</li>
```

以上即初步思路。

● 運作邏輯（響應式）



即當頁面過小時，導航欄由垂直變為水平：

難點/解釋/作用	代碼
<p>當窗口水平尺寸小於 700px 時：</p> <ol style="list-style-type: none"> 1 導航欄移至底部 2 導航欄及列表寬度設定為 100% 3 高度為 $16 \times 6 = 96\text{px}$ 4 靠左 5 水平平均分佈 6 列表水平排序 7 懸浮無描述文本 8 懸浮無陰影 	<pre> @media only screen and (max-width:700px) { .navbar { bottom: 0px; width: 100%; height: 6rem; left: 0px; justify-content: space-around; } .navbar-nav { flex-direction: row; width: 100%; } .navbar: hover .link-text { display: none; } .navbar: hover .misc-item { box-shadow: 0 0 0 0 ; } } </pre>

4.6 網站響應式設計

這個部分的代碼也就那樣，大差不差的，就改改大小和開關效果而已，因此這部分就只講設計的部分，不解釋代碼了：

當窗口水平尺寸小於 700px 時：

- 1 縮小整個網站的文本大小
- 2 縮小整個網站的圖片大小
- 3 縮小輪播圖的大小
- 4 導航欄移至底部
- 5 導航欄及列表寬度設定為 100%
- 6 導航欄懸浮無描述文本
- 7 導航欄懸浮無陰影
- 8 教程無自動對齊系統

以上！

5. 結論

5.1 可供改善之處

● 網站瑕疵

說實話，網站大部分瑕疵以及錯誤都在完成網站後的一大段時間被我修復了。然而，除了在內容方面，挖出來的坑（教學）沒來得及逐一填上的遺憾之外，還有一個因為技術水平不足而始終無法修復的錯誤：瀏覽器檢測系統：

難點/解釋/作用
當中的程式我已在 旋轉瀏覽器圖標彩蛋 中解釋過，所以簡言之，該系統無法根據獲得的用戶代理訊息字串中檢測出正確的瀏覽器（只有 Edge 無法啣~）。檢測 Edge 的部分，無論在 <code>userAgent.indexOf()</code> 填寫什麼，例如 <code>Edge</code> , <code>Edg</code> , <code>edge</code> , <code>edg</code> , <code>Edg/</code> , <code>edg/</code> 等，都無法成功檢測 <code>Edge</code> ，反而檢測出了 <code>Chrome</code> 。
代碼 (JavaScript)
<pre>window.addEventListener('DOMContentLoaded', function() { var browserIMG = document.getElementById('browser-img'); var userAgent = navigator.userAgent; if (userAgent.indexOf('Firefox') != -1) { browserIMG.src = '../pics/my_pics/firefox.png'; console.log('Firefox detected'); } else if (userAgent.indexOf('Chrome') != -1) { browserIMG.src = '../pics/my_pics/google.png'; console.log('Google detected'); } else if (userAgent.indexOf('Edg') != -1) { //俺故意這麼寫的 browserIMG.src = '../pics/my_pics/edge.png'; console.log('Edge detected'); } else { browserIMG.src = '../pics/my_pics/openai.png'; console.log('None browser detected'); } });</pre>

● 報告瑕疵

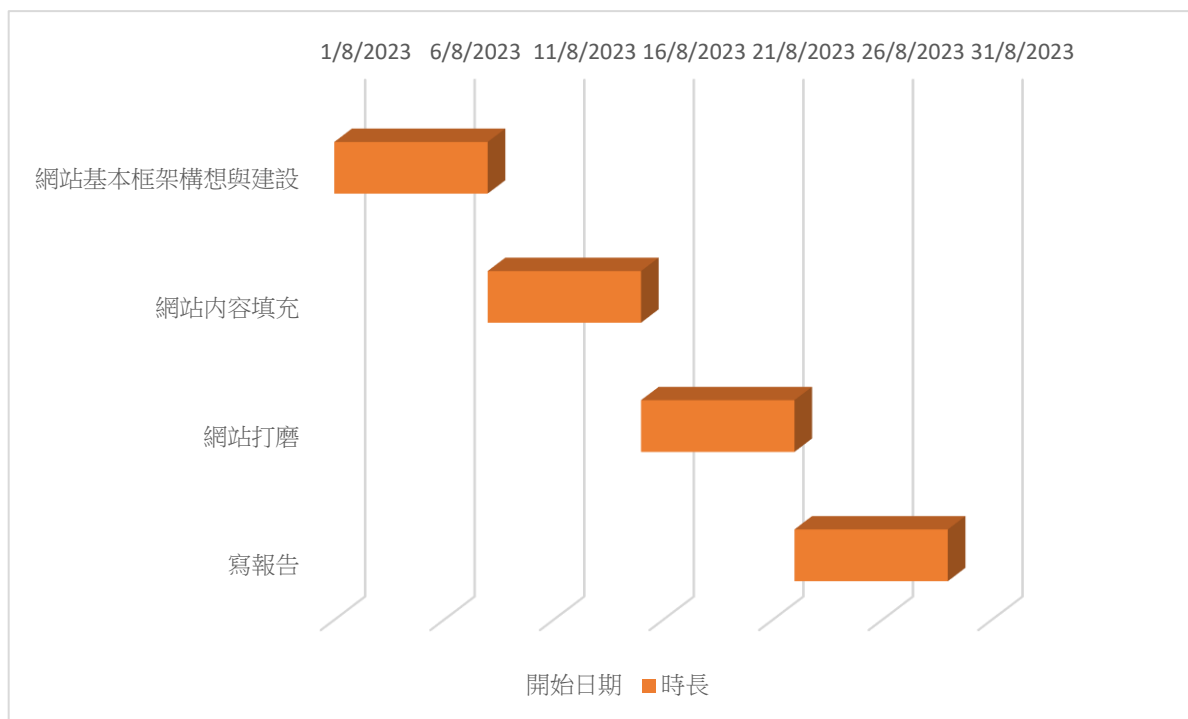
如圖：

<input type="checkbox"/>	教程表格難點/實現方法	17↵
4.7	較佳的教程體驗（我認為的）	18↵
<input type="checkbox"/>	亮點	18↵
<input type="checkbox"/>	自動對齊系統難點/實現方法	19↵
<input type="checkbox"/>	窗口遮罩系統難點/實現方法	19↵
<input type="checkbox"/>	引導動畫難點/實現方法	20↵
<input type="checkbox"/>	Spoiler 補充效果	21↵
<input type="checkbox"/>	影片封面（persona 4）	22↵
	自動評分系統	23↵
	自動評分系統 2 號	25↵
<input type="checkbox"/>	滾動按鈕	28↵
4.5	靠南懸浮響應式動態垂直導航欄	29↵
	亮點	29↵
<input type="checkbox"/>	旋轉瀏覽器圖標彩蛋	30↵
	運作邏輯（抽屜效果）	32↵
	運作邏輯（響應式）	33↵
4.6	網站響應式設計	34↵
5.	結論	35↵
5.1	網站瑕疵	35↵
5.2	網站製作時間（大概長度）	36↵
6.	參考資料/工具	37↵

明明加入項目符號的步驟全部都完全一致，為什麼有些就不能顯示呢？

為什麼啊！？

5.2 網站製作時間（大概長度）



網站基本框架構想與建設：一個星期

網站內容填充：一個星期

網站打磨：一個星期

寫報告：一個星期

6. 參考資料/工具

- 1 圖標 Font Awesome：<https://fontawesome.com/icons>
- 2 導航欄靈感：<https://www.youtube.com/watch?v=biOMz4puGt8>
- 3 按鈕動畫靈感：<https://www.youtube.com/watch?v=Nmaw6pBEsI8>
- 4 輪播圖搭建工具 Swiper：<https://www.swiper.com.cn/>
- 5 HTML、CSS、JS 教程：數不清的大雜燴影片
- 6 技術指導：Poe，Notion