

網頁小遊戲_抽紅包

李賢達

課程大綱：

- ➡ ChatGPT生成專案：
- ➡ HTML+CSS+Javascript
- ➡ 紅包外觀：CSS樣式設定
- ➡ 遊戲控制：Javascript

ChatGPT:指令

- 我是一名小型電子商務網站的店家，為了提昇客戶光臨本店，擬開發小型網路抽獎遊戲讓消費者提供優惠，以增加回客率
- 提供「紅包抽獎 (開獎按鈕)」的HTML+JS

一、專案構想簡介

目標：吸引新顧客、活化舊會員、刺激下單。

方式：在網站首頁或活動頁面嵌入一個小型 HTML5 抽獎遊戲（例如：轉盤、刮刮卡、搖一搖、抽紅包）。

參加條件：

- 訪客登入會員即可抽一次。
- 購物滿額、分享活動頁面可再抽一次。
- 抽中可獲折價券或購物金（由後端資料庫控制數量與有效期限）。

模組	功能	技術建議
前端 (Web)	抽獎動畫、互動介面	HTML5 + CSS3 + JavaScript (或使用 Phaser.js)
後端伺服器	控制抽獎邏輯、防止作弊、 記錄中獎名單	PHP / Python Flask / Node.js
資料庫	儲存會員、抽獎紀錄、獎項 設定	MySQL / MariaDB

💡 三、遊戲類型建議

類型	特點	適合場合
🎯 幸運轉盤 (Lucky Wheel)	視覺效果佳、操作簡單、最常見	購物節、會員日活動
🗑️ 刮刮樂 (Scratch Card)	有真實互動感、可藏驚喜	實體店推廣活動
🎲 搖骰子抽獎	遊戲性強、可搭配關卡	線上活動、節慶
💰 紅包抽獎 (開獎按鈕)	適合節慶活動	春節、週年慶
📦 即開式抽獎 (點擊開箱)	模仿開箱感覺、刺激分享	電商新品上市時

提供「紅包抽獎 (開獎按鈕)」的HTML+JS

以下是一個簡單又好用的「紅包抽獎 (開獎按鈕)」HTML + JavaScript 範例，適合放在你的電子商務活動頁面中。

玩家按下「開紅包」按鈕後會隨機顯示中獎結果。

📁 紅包抽獎範例程式 (HTML + JS)

html

複製程式碼

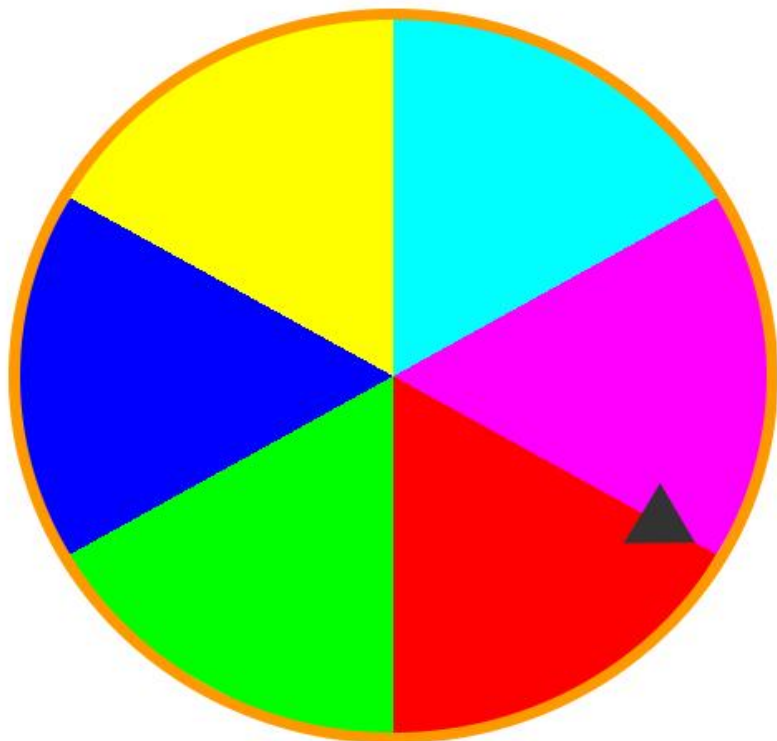
```
<!DOCTYPE html>
<html lang="zh-TW">
<head>
```



5-



幸運轉盤抽獎



開始抽獎



恭喜抽中：折價券10%！

6-



開紅包抽獎活動



開紅包

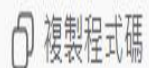
點我開獎



恭喜您抽中：🏆 超值購物金 \$200 元！

這一段是 CSS 樣式程式碼，用來設定某個網頁元素（例如 `body` 或某個區塊）的外觀。雖然你貼的是大括號內的內容，實際上它通常會長這樣：

CSS



```
body {  
  font-family: "微軟正黑體";  
  text-align: center;  
  background: linear-gradient(to bottom, #ffcccc, #ff9999);  
  height: 100vh;  
}
```

1. font-family: "微軟正黑體";

- 設定字型為「微軟正黑體」。
- 只要使用這個樣式的元素內的文字，預設都會以「微軟正黑體」顯示。
- 常見用法是放在 `body`，讓整個網頁的字體統一。

2. text-align: center;

- 設定文字對齊方式為「置中」。
- 影響的是這個元素「裡面的文字或 inline/inline-block 元素」，例如：
 - 若寫在 `body`，則 `body` 內直接的文字、`inline` 元素會水平置中。
 - 若是一個區塊 `<div>` 套用了這個樣式，則該 `<div>` 裡的文字都會置中。

3. `background: linear-gradient(to bottom, #ffc000, #ff9900);`

這行設定的是背景顏色為漸層：

- `background`：是背景的總設定屬性，可以設定顏色、圖片、漸層等。
- `linear-gradient(...)` 表示使用線性漸層。
- `to bottom`：漸層的方向是「由上往下」。
- `#ffc000`：上方的顏色（較淺的粉紅）。
- `#ff9900`：下方的顏色（較深的粉紅）。
- 合起來效果就是：畫面從上到下，顏色由淺粉紅漸層到深粉紅，看起來像粉紅色背景漸層。

RGB: 00~FF

這是一段 CSS 樣式，用來設計一個叫做 `#redpacket` 的區塊，看起來很像一個「紅包」卡片。通常會搭配 HTML 這樣寫：

html

複製程式碼

```
<div id="redpacket"></div>
```

來套用這個樣式。下面逐行說明每個屬性的意思和視覺效果：


```
#redpacket { ... }
```

- `#redpacket` 代表 id 為 `redpacket` 的元素。
- 也就是 `<div id="redpacket">...</div>` 這一個元素會套用這些樣式。
- `#` 開頭表示「id 選擇器」。

```
width: 200px; height: 260px;
```

- 設定這個元素的 寬度與高度：
 - `width: 200px;` → 寬度 200 像素。
 - `height: 260px;` → 高度 260 像素。
- 這樣的比例偏長，看起來像一個直立的紅包或卡片。

```
background: red;
```

- 設定 背景顏色為紅色。
- 因為是紅包，背景用 `red` 很符合主題。
- 這個紅色會填滿整個 200×260 的區塊。

```
border-radius: 20px;
```

- 設定 四個角都變成圓角。
- `20px` 表示圓角的弧度，數字越大，角越圓。
- 原本是直角矩形 → 變成有圓角的紅色卡片，更像實體紅包。

13

`margin: 60px auto;`

- `margin` 是外距（元素與外面其他元素之間的距離）。
- 這裡有兩個值：`60px auto;`
 - 上、下外距：`60px`
 - 左、右外距：`auto`
- `auto` 用在左右，搭配固定寬度（`200px`）時，會讓這個元素水平置中。
- 所以效果是：
 - 距離上方 `60px`
 - 元素在畫面中左右居中顯示
 - 距離下方也會有 `60px` 空白（視其他布局而定）。

14

`position: relative;`

- 設定定位方式為相對定位。
- 本身看起來位置不會改變，但後續若在 `#redpacket` 裡面放子元素（如：文字、金幣圖示），可以讓子元素用 `position: absolute;` 來相對這個紅包定位。
- 換句話說：這是為了方便在紅包內精準擺放內容。

`box-shadow: 0 5px 15px rgba(0,0,0,0.3);`

- 設定盒子陰影，讓紅包有立體感。
- 參數分別是：
 - `0`：水平位移（x 方向）→ 陰影不往左或右偏移。
 - `5px`：垂直位移（y 方向）→ 陰影向下偏移 `5px`。
 - `15px`：模糊半徑 → 數字越大，陰影越柔和、範圍越大。
 - `rgba(0,0,0,0.3)`：陰影顏色 → 黑色、不透明度 `0.3`（30% 透明）。
- 視覺效果：紅包好像浮在背景上，不會太扁平。

15

`cursor: pointer;`

- 當滑鼠移到這個元素上時，游標會變成「手指形狀」。
- 代表「這個東西可以點」，通常用在按鈕或可點擊的紅包區塊。
- 使用者看到就會直覺以為「可以點來抽獎/開紅包」。

`transition: transform 0.3s ease;`

- 設定過渡動畫效果，這裡是針對 `transform` 屬性做動畫。
- 拆開來看：
 - `transform`：要產生動畫的屬性，例如之後加上 `transform: scale(1.05);` 或 `transform: rotate(...)`。
 - `0.3s`：動畫時間為 0.3 秒。
 - `ease`：速度曲線，一開始慢、中間快、最後再慢下來，看起來更自然。
- 意義是：

→ 當你對 `#redpacket` 做 `transform` 變化（例如 `hover` 時放大一點）時，會有平滑的動畫，而不是瞬間

16

複製程式碼

```
const btn = document.getElementById("drawBtn");  
const packet = document.getElementById("redpacket");  
const result = document.getElementById("result");
```

- `document.getElementById("drawBtn")`

去 HTML 裡找 `id="drawBtn"` 的元素，例如：

html

複製程式碼

```
<button id="drawBtn">立即抽獎</button>
```

找到後存進變數 `btn`，之後用 `btn` 就可以控制這個按鈕。

- `document.getElementById("redpacket")`

找到紅包外觀的區塊，例如：

html

```
<div id="redpacket"></div>
```

複製程式碼

存進 `packet`，用來做翻轉動畫 (transform)。

- `document.getElementById("result")`

找到顯示抽獎結果的地方，例如：

html

```
<div id="result"></div>
```

複製程式碼

存進 `result`，等等會把中獎文字寫進

2 設定「按鈕被點擊時」要做什麼

js

複製程式碼

```
btn.onclick = () => {
  ...
};
```

- `btn.onclick = ...` 表示：「當這個按鈕被點擊時，執行右邊這個函式」。
- `() => { ... }` 是 箭頭函式，等同於：

js

複製程式碼

```
btn.onclick = function() {
  // ...
};
```



```
btn.disabled = true;
```

```
packet.style.transform = "rotateY(180deg)";
```

- `btn.disabled = true;`

把按鈕設為「不可按」，避免使用者一直狂按，重複觸發抽獎。

- HTML 會變成有 `disabled` 屬性，通常會變灰：

```
html
```

```
<button id="drawBtn" disabled>...</button>
```

- `packet.style.transform = "rotateY(180deg)";`

讓 `#redpacket` 套用 CSS 變形：

- `rotateY(180deg)` 是沿 Y 軸旋轉 180 度，看起來就像紅包被翻面。
- 如果你之前在 CSS 有設定 `transition: transform 0.3s ease;`

這裡就會產生平滑的翻轉動畫。

```
const prizes = [
```

```
"💰 現金折價券 $50 元",
```

```
"🧧 滿千折 $100 優惠券",
```

```
"🍀 下次再接再厲!",
```

```
"🎁 神秘小禮物",
```

```
"🥤 免費飲品兌換券",
```

```
"🏆 超值購物金 $200 元"
```

```
];
```

- `prizes` 是一個陣列 (array)，每一個元素是一個字串，代表一種獎項。
- 索引從 0 開始：

- `prizes[0]` → 💰 現金折價券 \$50 元

- `prizes[1]` → 🧧 滿千折 \$100 優惠券

- ...

5 產生 0 ~ prizes.length-1 的隨機整數

js

複製程式碼

```
const index = Math.floor(Math.random() * prizes.length);
```

拆開來看：

- **Math.random()**

會產生一個「0 以上、1 以下」的亂數（例如 0.2345、0.9182...）。

- **Math.random() * prizes.length**

假設 `prizes.length` 是 6，這行會產生介於 0 ~ 6 之間的浮點數（不含 6）。

- **Math.floor(...)**

無條件捨去小數，例如：

- **Math.floor(2.9) → 2**

- **Math.floor(5.01) → 5**



6 延遲 1.5 秒後顯示結果 + 復原狀態

js

複製程式碼

```
setTimeout(() => {
  result.innerHTML = `🎉 恭喜您抽中：${prizes[index]}!`;
  packet.style.transform = "rotateY(0deg)";
  btn.disabled = false;
}, 1500);
```

這段做三件事，而且是「1.5 秒後」才執行。

- **setTimeout(() => { ... }, 1500);**

- **setTimeout**：設定一個「延遲執行」的動作。

- **1500**：延遲毫秒數 → 1500 毫秒：↓ 1.5 秒。

- 用意：給紅包翻轉、製造「抽獎等待中」的感覺，1.5 秒後才公布結果。

7 總流程整理一次

1. 使用者按下「抽獎」按鈕。
2. 按鈕被暫時鎖住（避免連點）。
3. 紅包執行 `rotateY(180deg)` → 做翻面動畫。
4. 程式隨機選出一個獎項。
5. 等待 1.5 秒（模擬抽獎過程、讓動畫有時間跑）。
6. 顯示「恭喜您抽中：某某獎項」，紅包翻回正面。
7. 按鈕重新啟用，可以再抽。