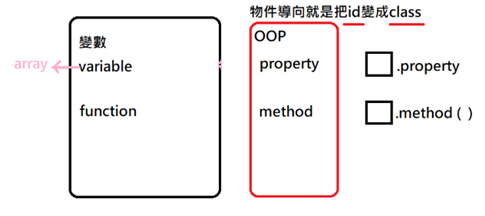
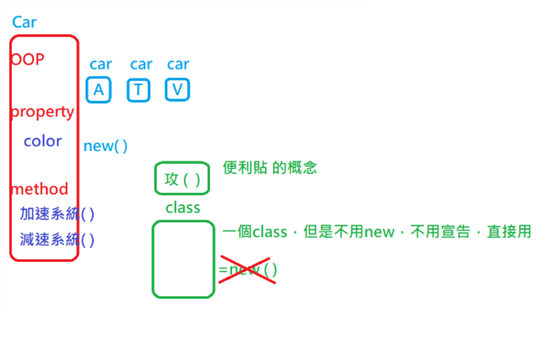
**一、定義**



Variable就像是property的概念

function就像是 method的概念

**二、解說**



物件導向就是，宣告一種類別，叫做汽車，擁有color的property跟加減速系統的method

就像是有一張藍圖、設計圖，告訴我們車子要怎麼做，有哪些功能、特色

有了這張圖，我們就可以跟著做出都是會跑但不同品牌的汽車，可以調整不同顏色、馬力強弱等，

另外抽象的意思就是便利貼，如果要增加汽車有攻擊性的功能，就加個便利貼，完成新增攻擊功能

ChatGPT 4o mini

*一、以樂高玩具為例*：

1. **類別（Class）**：就像樂高積木的設計圖，定義了積木的屬性（顏色、形狀）和方法（組合、拆卸）。也就是說，**類別就像是樂高積木的設計圖或說明書**，它描述了積木的形狀、顏色、功能等。每一種樂高積木的設計都可以視為一個類別，並且這個設計圖決定了積木應該如何被組合和使用。
2. **物件（Object）**：具體的積木物件，從類別創建出來（例如一塊紅色的 2x2 積木）。換句話說，物件是根據類別創建出來的具體實體，代表一個具體的積木。在換句話說，有一塊紅色的2x2積木，那麼這塊積木就是一個物件，它是從「樂高積木」這個類別創建出來的。
3. 可以有不同種類的積木（如基礎積木、輪子、特殊形狀積木），它們都可以繼承自「基本樂高積木」這個類別，但是每種積木也會有自己獨特的屬性和行為。

*二、以動物為例：*

1. **類別（Class）**：是一個模板，定義動物的共性（例如 Animal 類別），這個類別包含所有動物的一般特徵和行為。。
2. **物件（Object）**：是從類別創建的具體實體，例如一隻具體的狗或貓是從「動物」類別創建出來的具體物件。
3. **繼承（Inheritance）**：子類別可以繼承父類別的屬性和方法，並擴展或修改它們（例如 Dog 和 Cat 類別繼承自 Animal 類別）。
4. **封裝（Encapsulation）**：將物件的內部細節隱藏，只暴露必要的方法（例如封裝動物的年齡）。
5. **多型（Polymorphism）**：允許不同類型的物件使用相同的方法名稱，但實現的具體方式可能不同。這樣，我們可以讓不同的動物對「叫聲」這個行為做出不同的反應。即使每個動物的 make\_sound 方法不同，我們依然可以統一地使用相同的方法名稱。換句話說，不同的物件可以使用相同的接口方法，但會有不同的實現（例如不同動物的”叫聲”）。