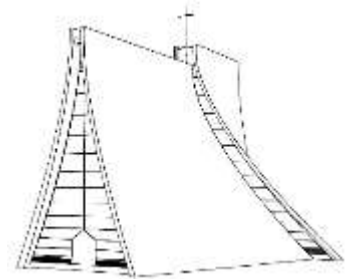




運算思維與程式設計

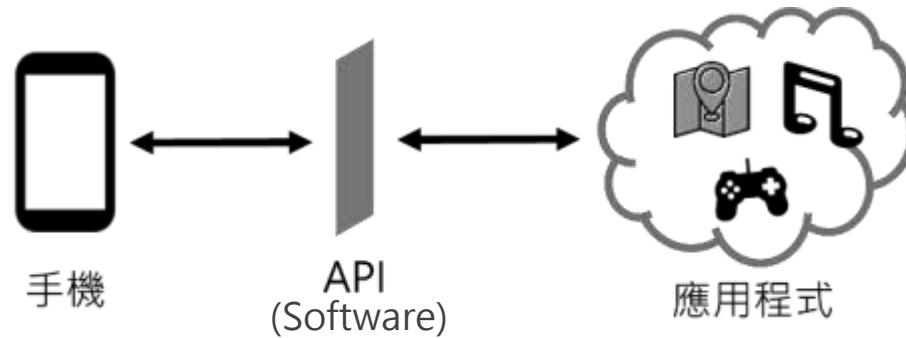
CH14 亂數與API



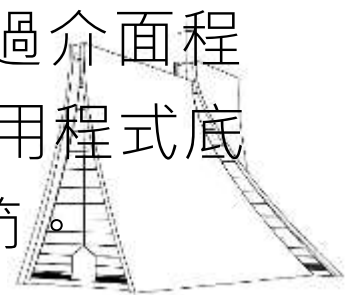


什麼是 API ？

API 為 Application Programming Interface 的縮寫，中文名稱為：應用程式介面。



API的中心思想是讓應用程式開發人員可以透過介面程式碼，取得想要的應用程式，而無須考慮此應用程式底層的原始碼為何、或理解其內部工作機制的細節。



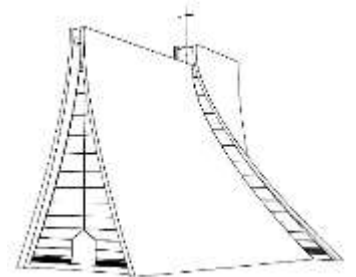


API 應用實例

1. 線上購物，信用卡帳號遠端查核

2. Google

- Google Maps API
 - 寶可夢探險尋寶 (Pokemon quest)
- Calendar API
- Cloud Vision
- Cloud Speech-to-Text






Google Cloud Vision API

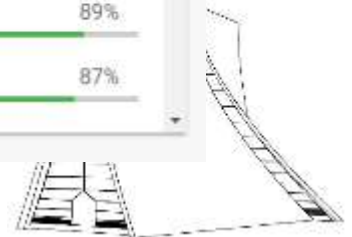
Google Cloud Why Google Products Solutions Pricing Security Documentation Customers Partners Supp >

AI & Machine Learning Products



The screenshot shows the Google Cloud Vision API interface. On the left, there is a photograph of a ginger tabby kitten lying on its back with its mouth open. Below the image is the filename 'cat.jpg'. On the right, a list of labels is displayed with corresponding confidence percentages. The labels are: Cat (99%), Facial Expression (94%), Whiskers (93%), Small To Medium Sized Cats (92%), Skin (92%), Fauna (90%), Dragon Li (89%), and Cat Like Mammal (87%). Each label is accompanied by a green horizontal bar representing the confidence level.

Label	Confidence
Cat	99%
Facial Expression	94%
Whiskers	93%
Small To Medium Sized Cats	92%
Skin	92%
Fauna	90%
Dragon Li	89%
Cat Like Mammal	87%



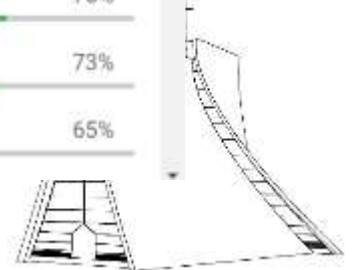


AI & Machine Learning Products



cat2.jpg

Cat	99%
Small To Medium Sized Cats	91%
Whiskers	85%
Cat Like Mammal	84%
Dragon Li	83%
Fur	75%
Kitten	73%
Snout	65%

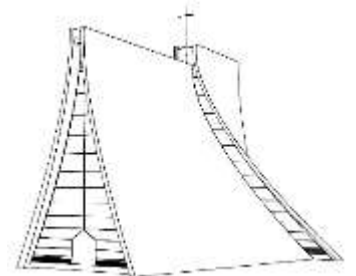




關於亂數

通常程式語言本身的標準程式庫內，多半會附上某個亂數API，但你曾有想過這些亂數產生器，有可能是「假」的嗎？！

當然，**不是那種真的「假」，而是不夠「亂」**！每經過一次的循環，之前出現過的「事件」又會再出現一次！

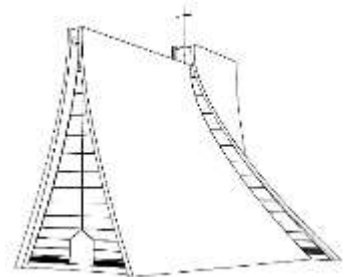




亂數種子

程式任意產生的兩個亂數之間應該沒有任何關聯，但實際上，只要設定相同的「亂數種子(Seed)」，接下來所產生的亂數序列就會相同。

許多程式的亂數函式皆是如此，這種以電腦自動生成的亂數，我們稱之為「**偽亂數** (Pseudorandom)」。

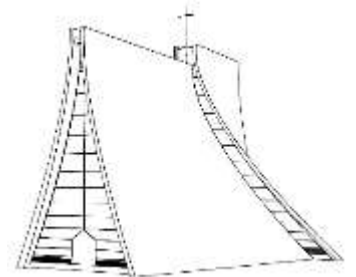




真亂數

為了追求真正的亂數 (True randomness), 有學者開始向自然界的物理現象尋求解決的方式。例如, random.org 的網站所提供的隨機亂數服務, 即是利用大氣雜訊(Atmospheric noise)來產生。

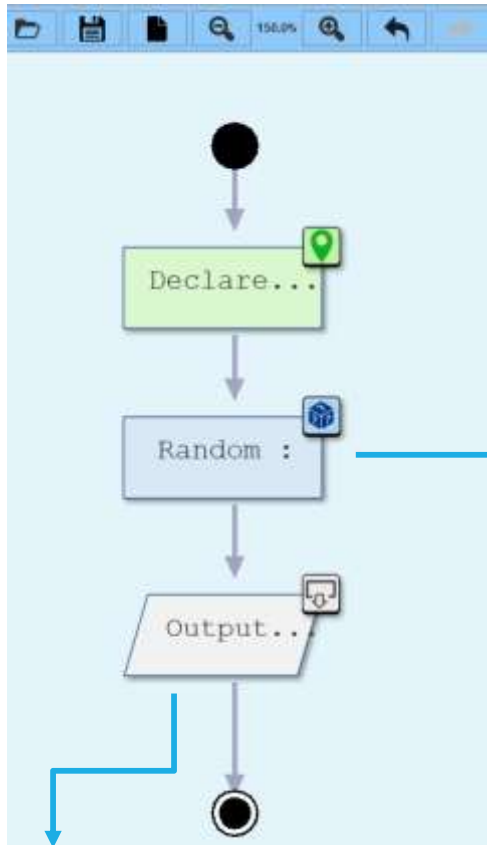
但最終所得到的亂數, 就是所謂的真正亂數嗎? 還是即使在多混沌的世界中, 都有其規律性存在。



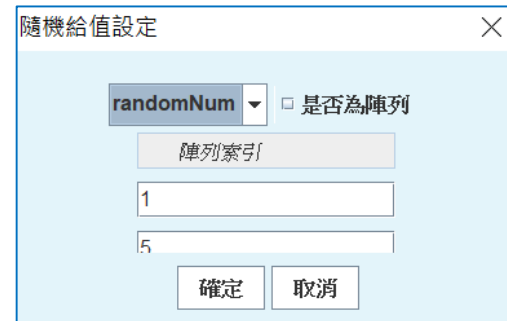


亂數 - CT2Code

```
int  
r1;
```

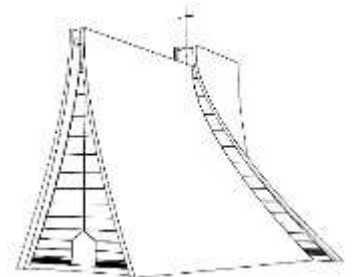


```
Random random = new Random();  
r1 = random.nextInt(5-1)+1;
```



隨機產生 1-5 之間的亂數

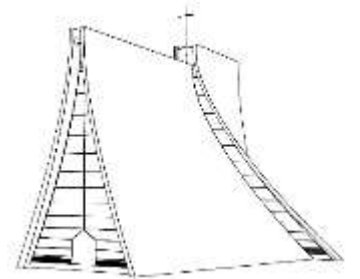
```
System.out.println("隨機亂數：" + r1);
```

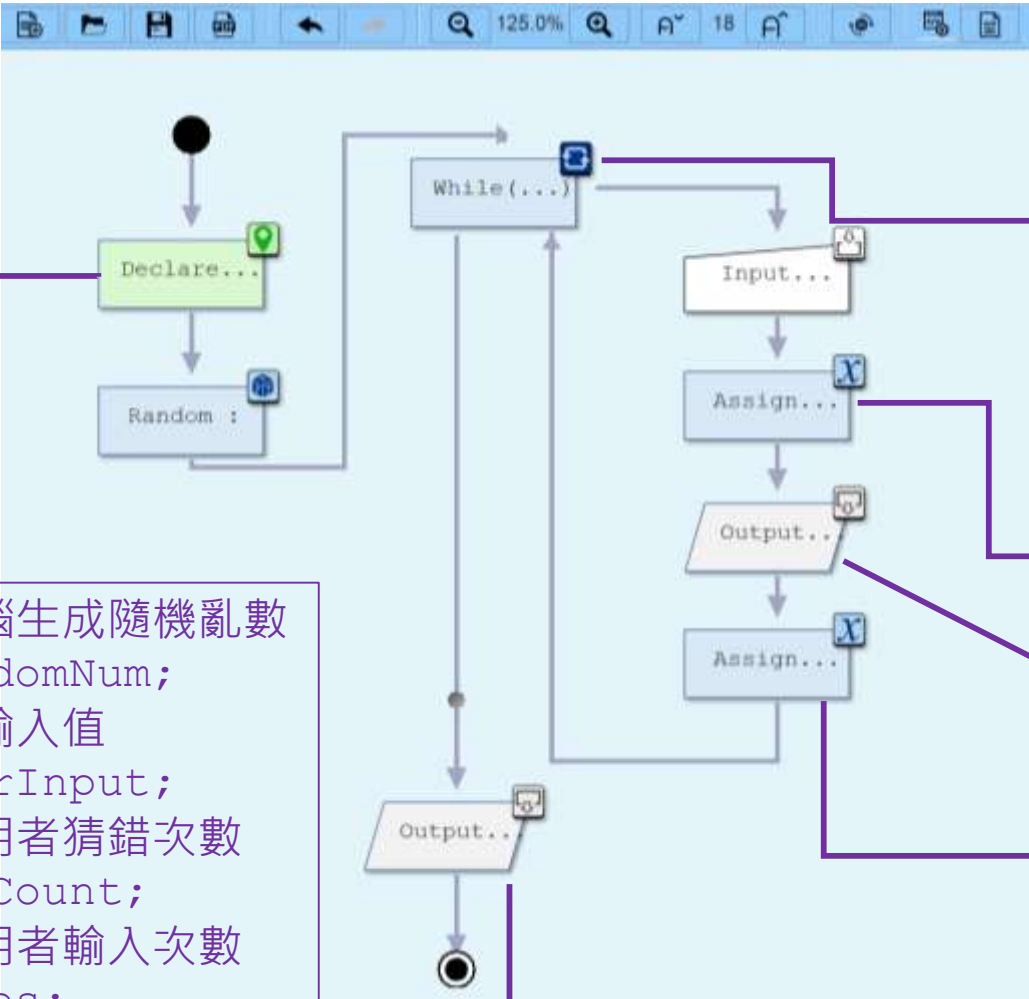




隨堂演練

電腦隨機產生一個**1-10**之間的亂數，試試手氣，看看你需要猜幾次，才能猜到電腦所選的神秘數字。





While迴圈設定

randomNum 是否為陣列

陣列索引

!=

userInput

// 註解內容

確定 取消

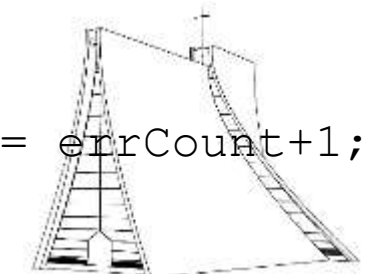
```
//暫存電腦生成隨機亂數
int randomNum;
//使用者輸入值
int userInput;
//計算使用者猜錯次數
int errCount;
//紀錄使用者輸入次數
int times;
```

```
times = times + 1;
```

```
System.out.println
("第"+times+"次輸
入："+userInput);
```

```
errCount = errCount+1;
```

"總共錯猜" + errCount + "次!"





請再仔細檢查一下，上一案例的程式邏輯是否有誤？
如果有請將錯誤地方進行改正。

執行結果：

第1次輸入：3

第2次輸入：5

第3次輸入：6

第4次輸入：7

第5次輸入：1

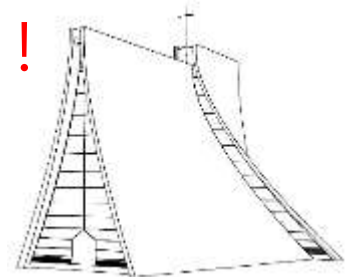
第6次輸入：2

第7次輸入：4

第8次輸入：8

第9次輸入：9 → 猜對啦！

總共錯誤 9 次！

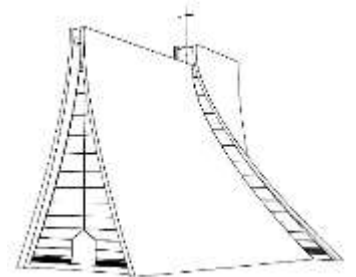




隨堂演練

Trying your luck !

修改課堂例子 (pp. 12) , 當玩家猜 3 次以下即猜中神秘數字, 則顯示「Today is your day !」; 猜4-6次, 顯示「今日運勢普普 !」; 若猜 7 次以上, 則顯示「今日的你有一點背啊, 凡事小心為妙 !」



Thank You!