Coding Style

- 程式設計的作文課 -

code（程式碼）是程式設計師與電腦溝通的語言，當然讓電腦理解你寫的code是很重要的事情，不然連編譯都沒辦法編譯。然而讓"人類"能夠理解你所寫的code也同樣重要，甚至可以說，讓人看得懂code遠比讓電腦看懂來的重要。

你可能會想，這程式只是為了解題目，寫完以後就再也不會去碰，寫得醜又有什麼關係，反正又沒有人會看到。但一個必須認清的事實是，你的code最少會有一個人會看到，那就是自己——**永遠不要相信未來的自己會懂你**。小編就有過被自己背叛的慘痛經驗。故事是這樣的，有天晚上小編花了好幾個小時在寫期末project，就在已經寫了數百行程式碼、馬上就要完成的時候因為一些事情暫時離開，等到重新回到電腦前、看到自己寫的程式碼時，只看到了一堆不明所以的變數名稱："i", "ii", "iii", "ab", "abc"...，我完全看不懂我是在寫三丁....毀！當下的心情大概就跟~~只會說台語的Elsa聽到國語~~挖出甲骨文的農夫差不多吧...

由小編慘痛的經驗我們可以知道，寫出漂亮的code不只賞心悅目，更重要的是方便事後的修改或除錯。以下就列出了幾點基本的注意事項，只要了解其中的原則，相信人人都能~~成為食神~~寫出漂亮的code！

※當然這些都只是一些常見的技巧，如果有更加適合當時情境的寫法也可以不用嚴格遵守，就像獨孤九劍的最高境界「無招勝有招」一樣。只要記得把握「讓人能夠看懂」的大原則即可。

一、變數命名

就像平常，我們會為每個東西取上方便記憶、了解的名稱，而不會把書包叫做a，鉛筆叫做aa，橡皮擦取作aaa之類。因此，一個好的變數名稱，應該會讓人一眼就可以猜出他的功能。像假如要計算某些元素的和，取作 SumOfThings 一定比 aabbcc 來的易懂：

有意義的名稱	沒意義的名稱

int SumOfThings = 0;
for(int i = 0;i < 5 ; i = i + 1)
  SumOfThings = SumOfThings + Things[i];
	int aabbcc;
	for(int i = 0;i < 5 ; i = i + 1)
	  aabbcc = aabbcc + abc[i];

取作aabbcc明顯還需要繼續讀下面的幾行Code，才有辦法了解aabbcc到底是甚麼功能。

有時，並不只用字詞意思區隔功能，"形狀"也是個好用的辨識方法。先來假想一種情境，有一個程式，需要用到常數，這時Const就顯得累贅，但為了不失"**常數**"意義，可以改成MAX_SIZE，搭配前面的命名法，可以一眼就看出來哪個是變數，哪個是常數。

原本命名法	使用大寫表示常數
int ConstMaxSize = 100;	int MAX_SIZE = 100;

一些可以參考的格式：

- SumOfThings
  把Sum of things 合起來後將首字大寫，這種命名法稱作駝峰式命名法
- SUM_OF_THING
  全部字母大寫，然後為了區隔單字，在間格加入"_"符號，通常會來表示常數
- strUserName
  str代表string，強調strUserName是字串，有時會把型別名稱小寫後接在前面以利辨識

二、縮排

code是層次分明的，像是for迴圈、if判斷是一整個區塊，這時候就可以用適當的縮排來強調這些層次的關係。

比較以下程式碼，可以發現同樣是找到1-100之中所有不能被3, 5, 7整除的數字，右邊的程式碼明顯比左邊來的好讀：

無縮排	有縮排

#include <cstdio>
int main(){
for(int i = 1; i <= 100; i = i+1){
if(i % 3 != 0 && i % 5 != 0 && i % 7 != 0){
printf(“%d ”, i);
}
}
}
	#include <cstdio>
	int main(){
	  for(int i = 1; i <= 100; i = i+1){
	  if(i % 3 != 0 && i % 5 != 0 && i % 7 != 0){
	  printf(“%d ”, i);
	  }
	  }
	}

除了美觀之外，縮排還有不少好處：

1. 容易檢查括號有沒有成對
2. 容易確定程式在什麼情況下才會進到那一層
3. 更多好處等你自己來發現XD

那麼到底要往內縮多少才好呢？這就跟個人的習慣有關係了，一般常用的有4格空白（最多人使用）、2 or 3格縮排（也有少數強者使用）或是8格(or tab鍵)（linux之父Linus Torvalds的習慣），這些習慣沒有哪個比較好的問題，網路上甚至會不時針對縮排的議題展開激烈的辯論，但最重要的是在同一份code裏面務必要統一，**長短不一的縮排反而會誤導自己**。

三、換行

C/C++的指令是以”行”為單位的，在每個分號之前就是一行指令。**確實把行斷開**可以比較容易看出程式的流程，就像是在寫作文的時候不可能都不換段落或是寫標點符號，相信一篇完全沒有段落與標點符號的文章是沒有人會想看的。

以下是一個簡單（誇張）的比較：

不換行	有換行

#include <cstdio>int main(){for(int i = 1; i <= 100; i = i+1){if(i % 3 != 0 && i % 5 != 0 && i % 7 != 0){printf(“%d ”, i);}}}	#include <cstdio>
	int main(){
	  for(int i = 1; i <= 100; i = i+1){
	  if(i % 3 != 0 && i % 5 != 0 && i % 7 != 0){
	  printf(“%d ”, i);
	  }
	  }
	}

當然凡事都是過猶不及的，所以也儘量不要在不必要的地方隨意斷行，除非你立志要參加「C語言混亂代碼大賽*(註1)」。

最後一個建議是，當一行程式碼長到一定的程度(如80個字元)，可以考慮看看有沒有更好的寫法可以把它**拆成比較短的數行**。

比如你想印出小星星歌詞的前四句，下面兩種寫法會得到相同的輸出，但右邊顯然比較好讀。

寫成一行	拆成數行
printf(“Twinkle, twinkle, little star\nHow I wonder what you are\nUp above the world so high\nLike a diamond in the sky\n”);	printf(“Twinkle, twinkle, little star\n”); printf(“How I wonder what you are\n”); printf(“Up above the world so high\n”); printf(“Like a diamond in the sky\n”);

四、 註解

要盡量的提醒自己，當自己的程式碼無法自己說話時，就必須幫程式碼說話。在C++裡面，註解可以單行或多行。

單行	多行
// 這是單行註解	/* 多行註解 OAO QAQ */

以下是一些可以幫程式碼說話的TIPs：

1. 檔案前面的註解

可以建議在檔案前端置入一些關於這檔案的描述，如下：

```
/* program : Hello World Program
  …
*/
```

2. 分Blocks註解

可以把程式碼切成好幾個區塊，然後在前端加入該區塊在**做甚麼事？**

```
// 確認array裡面是否存在0
int CheckZero = 0;
for(int i = 0;i < 5;i ++)
{
        if(array[i] == 0)
                CheckZero = 1;
}
```

3. TODO 註解

倘若暫時沒有時間繼續完成一部分的Code，可以在該處註解：

```
// TODO : 尚未完成的事
```

4. 切忌濫用

最後需要注意，請不要寫出跟程式碼描述同一件事的註解。

以下是錯誤示範：

```
i = 1; //把變數i變成1
```

這只要是學過程式語言的人都看的出來，你需要說明的應該是”**為什麼**”要寫這行而不是這行在”**做什麼**”。

參考資料：
codingstyle - 程式設計風格對軟體開發的影響：http://mmdays.wordpress.com/2007/04/24/coding-style/
怨氣文(?)：http://blog.allenown.com/2009/01/coding-style.html
註解程式碼的13個建議(翻譯)：http://www.cnblogs.com/oomusou/archive/2008/04/26/1172208.html

***註1**:世界上真的有這樣的一個比賽，以下是一個2011年的得獎作品，它真的可以成功編譯！功能就如同它的形狀，是在一張圖片中加入近乎隱形的動漫角色"燈里" 的名字。

```
                                  /*
                                   +
                                  +
                                 +
                                +
                                +
                                [         >i>n[t
                                 */   #include<stdio.h>
                             /*2w0,1m2,]_<n+a m+o>r>i>=>(['0n1'0)1;
                           */int/**/main(int/**/n,char**m){FILE*p,*q;int        A,k,a,r,i/*
                         #uinndcelfu_dset<rsitcdti_oa.nhs>i/_*/;char*d="P%"   "d\n%d\40%d"/**/
                       "\n%d\n\00wb+",b[1024],y[]="yuriyurararayuruyuri*daijiken**akkari~n**"
          "/y*u*k/riin<ty(uyr)g,aur,arr[a1r2a82*y2*/u*r{uyu}riOcyurhiyua**rrar+*arayra*="
        "yuruyurwiyuriyurara'rariayuruyuriyuriyu>rarararayuruy9uriyu3riyurar_aBrMaPrOaWy^?"
       "*]/f]`;hvroai<dp/f*i*s/<ii(f)a{tpguat<cahfaurh(+uf)a;f}vivn+tf/g*`*w/jmaa+i`ni("/**
      */"i+k[>+b+i>++b++>l[rb";int/**/u;for(i=0;i<101;i++)y[i*2]^="~hktrvg~dmG*eoa+%squ#l2"
      ":(wn\"1l))v?wM353{/Y;lgcGp`vedllwudvOK`cct~[|ju {stkjalor(stwvne\"gt\"yogYURUYURI"[
      i]^y[i*2+1]^4;/*!*/p=(n>1&&(m[1][0]-'-'||m[1][1]  !='\0'))?fopen(m[1],y+298):stdin;
       /*y/riynrt~(^w^)],]c+h+a+r+*+*[n>)+{>f+o<r<(-m]    =<2<5<64;}-]-(m+;yry[rm*])/[*
        */q=(n<3||!(m[2][0]-'-'||m[2][1]))?stdout /*]{     }[*/:fopen(m[2],d+14);if(!p||/*
        "]<<*-]>y++>u>>+r >+u+++y>--u---r>++i+++"  <)<      ;[>-m-.>a-.-i.++n.>[(w)*/!q/**/)
    return+printf("Can "  "not\x20open\40%s\40"    ""       "for\40%sing\n",m[!p?1:2],!p?/*
   o=82]5<<+(+3+1+&.(+  m  +-+1.)<)<|<|.6>4>-+(>    m-        &-1.9-2-)-|-|.28>-w-?-m.:>([28+
  */"read":"writ");for  (   a=k=u= 0;y[u];  u=2    +u){y[k++   ]=y[u];}if((a=fread(b,1,1024/*
 ,mY/R*Y"R*/,p/*U*/)/*          R*/ )>/*U{  */   2&& b/*Y*/[0]/*U*/=='P' &&4==/*"y*r/y)r\}
 */sscanf(b,d,&k,& A,&           i,  &r)&&        !   (k-6&&k -5)&&r==255){u=A;if(n>3){/*
 ]&<1<6<?<m.-+1>3> +:+ .1>3+++     .   -m-)      -;.u+=++.1<0< <; f<o<r<(.;<([m(=)/8*/
 u++;i++;}fprintf   (q,    d,k,           u      >>1,i>>1,r);u  = k-5?8:4;k=3;}else
   /*]>*/{(u)=/*{   p> >u  >t>-]s                >++(.yryr*/+(    n+14>17)?8/4:8*5/
      4;}for(r=i=0  ;  ;){u*=6;u+=                (n>3?1:0);if    (y[u]&01)fputc(/*
       <g-e<t.c>h.a r  -(-).)8+<1.                 >;+i.(<)<     <)+{+i.f>([180*/1*
       (r),q);if(y[u   ]&16)k=A;if                              (y[u]&2)k--;if(i/*
       ("^w^NAMORI; {   I*/==a/*"                               )*/){/**/i=a=(u)*11
        &255;if(1&&0>=     (a=                                 fread(b,1,1024,p))&&
         ")]i>(w)-;} {                                         /i-f-(-m--M1-0.)<{"
          [ 8]==59/* */                                       )break;i=0;}r=b[i++]
             ;u+=(/**>>                                      *..</<<<)<[[;]**/+8&*
             (y+u))?(10-              r?4:2):(y[u]          &4)?(k?2:4):2;u=y[u/*
              49;7i\(w)/;}             y}ru\=*ri[          ,mc]o;n}trientuu ren (
              */]-(int)'``;}             fclose(           p);k= +fclose( q);
               /*] <*.na/m*o{ri{                        d;^w^;}  }^_^}}
                 "   */   return  k-                 -1+   /*\'   '-`*/
                       (   -/*}/   */0x01         );       {;{    }}
                             ;           /*^w^*/         ;}
```

更多資訊可參考比賽官網：http://www.ioccc.org/