

function

5/3, hyundai

為什麼需要？

- 假設我們的程式需要很大量的同樣性質的操作
- 對陣列裡每個數都開根號之類的
- 如果可以用一行就做完不是很好嗎？

Before

```
int main() {  
  
    A, B, C;  
    A = A*2+3-A/2;  
    B = B*2+3-B/2;  
    C = C*2+3-C/2;  
  
    return 0;  
}
```

After

```
int calc(int x) {  
    return (x*2+3-x/2);  
}  
int main() {  
  
    int A, B, C;  
    A = calc(A);  
    B = calc(B);  
    C = calc(C);  
  
    return 0;  
}
```

分離後的好處

- 如果操作需要更動，只要改函式就可以適用全部
- 易讀的程式碼，不容易出現 bug
- 較清晰的程式結構

猜猜我是OJ上的哪題？

```
#include <stdio>
int main(int argc, char *argv[])
{
    int a1, a2, a3, b1, b2, b3;
    while (~scanf("%d%d%d%d%d%d", &a1, &a2, &a3, &b1, &b2, &b3)) {
        printf("%d ", a1*b1+a2*b2+a3*b3);
        printf("(%d,%d,%d)\n", (a2*b3-a3*b2), (a3*b1-a1*b3), (a1*b2-a2*b1));
    }
    return 0;
}
```

45 / Dot & Cross

看名字想意義

```
4  int a1, a2, a3, b1, b2, b3;  
5  scanf("%d%d%d%d%d%d", &a1, &a2, &a3, &b1, &b2, &b3);  
6  printDot(a1, a2, a3, b1, b2, b3);  
7  printCross(a1, a2, a3, b1, b2, b3);
```

printDot => 印出 dot 值

printCross => 印出 cross 值

函式架構

- 函式名稱
- 回傳值
- 參數
- 函式內容

```
bool isDigit(char ch) {  
    if (ch >= '0' && ch <= '9')  
        return true;  
    else  
        return false;  
}
```

函式名稱

和變數命名的規則一樣
用來讓我們呼叫他的

```
bool isDigit(char ch) {  
    if (ch >= '0' && ch <= '9')  
        return true;  
    else  
        return false;  
}
```


函式內容

如同我們在 `int main()` 裡面寫的東西

```
bool isDigit(char ch) {  
    if (ch >= '0' && ch <= '9')  
        return true;  
    else  
        return false;  
}
```

參數

通常一個函式會吃 input
決定要做什麼事情等等

```
bool isDigit(char ch) {  
    if (ch >= '0' && ch <= '9')  
        return true;  
    else  
        return false;  
}
```

回傳值

函式算完以後，可能有東西回傳
在函式的最前頭要說明回傳的形態（這裡是 bool）
然後用 return xxx; 來傳回值

```
bool isDigit(char ch) {  
    if (ch >= '0' && ch <= '9')  
        return true;  
    else  
        return false;  
}
```

參數如何傳遞

main()

c: 7



isDigit(c);

複製一份到函式

isDigit(char ch)

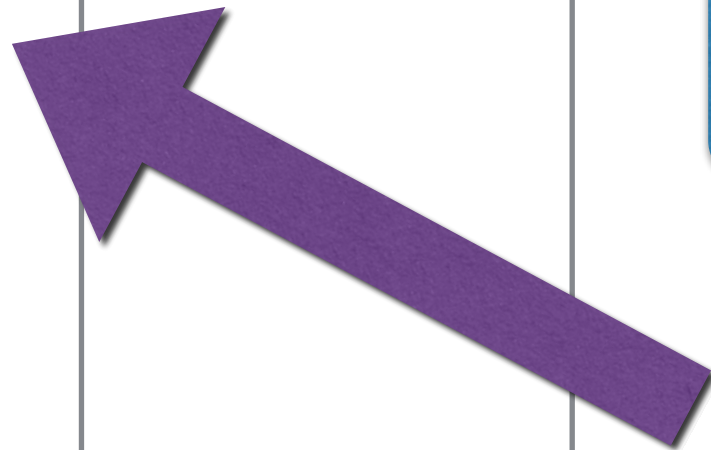
ch: 7

•
•
•

main()

c: 7

isDigit(c);



兩邊互不相關

isDigit(char ch)

ch: 7



ch = 'a';

main()

c: 7

isDigit(c);

兩邊互不相關

isDigit(char ch)

ch: 7

ch = 'a';

ch: a

main()

c: 7

isDigit(c);

兩邊互不相關

main()

c: 7

isDigit(c);



true

練習 (1)

- 使用函式實作 isAlpha
- 如果輸入的字是 a-z, A-Z 請回傳 bool/true
- 否則回傳 bool/false;

練習 (2)

- 利用函式 isDigit, isAlpha
- 如果輸入的是數字，請輸出 isDigit
- 如果輸入的是字母，請輸出 isAlpha
- 如果都不是上述兩者，請輸出 unknown

更複雜的參數

傳遞 array

- 給你一個陣列，要求找出最大值
- `int findMax(int array[], int size);`
- 因為在函式裡不知道原本陣列的大小
- 要記得傳入陣列的 `size`
- 等待講完指標後，會再提。