

# 函數：參數傳遞、變數生命週期

資訊之芽語法班 2015 suhorng

# 參數傳遞怎麼 work ?

- 大家來找碴

```
void swap(int a, int b) {  
    int c = a;  
    a = b;  
    b = c;  
    // 執行完後 a 跟 b 的內容互換  
}  
  
int main() {  
    int x = 9, y = 23;  
    swap(x, y);  
    // 成功?  
}
```

## 參數傳遞怎麼 work ?

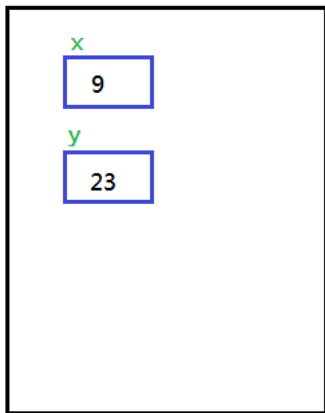
- 打比方. **x** 跟 **y** 從頭到尾沒動到

```
void swap(int a, int b) {  
    int c = a;  
    a = b;  
    b = c;  
    // 執行完後 a 跟 b 的內容互換  
}  
  
int main() {  
    int x = 9, y = 23;  
  
    // swap(x, y)  
    int a = x, b = y;  
    int c = a;  
    a = b;  
    b = c;  
  
}
```

# 參數傳遞怎麼 work ?

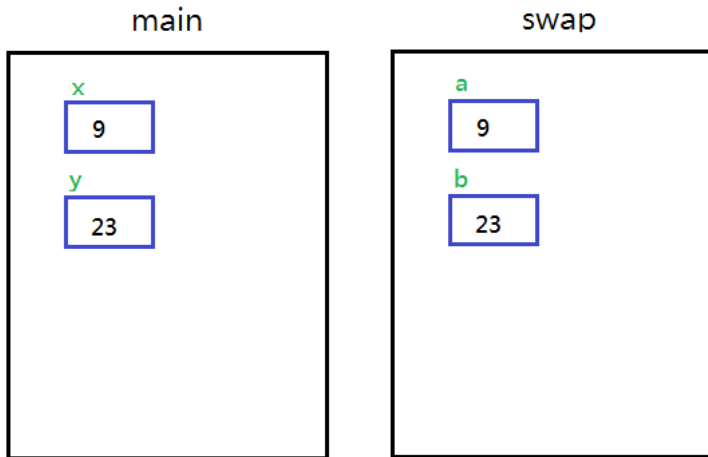
```
swap(x, y);
```

main



## 參數傳遞怎麼 work ?

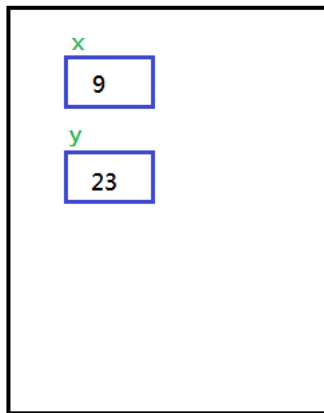
```
void swap(int a, int b) {
```



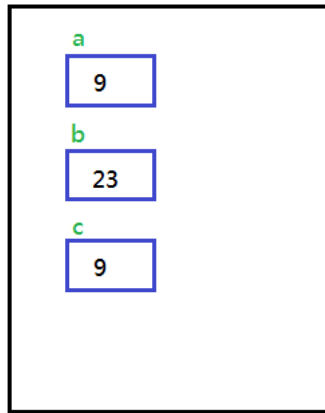
# 參數傳遞怎麼 work ?

```
int c = a;
```

main



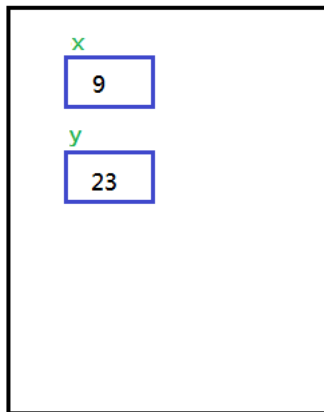
swap



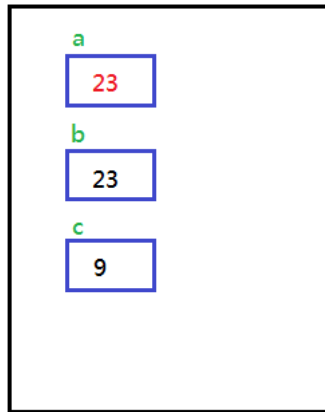
# 參數傳遞怎麼 work ?

```
a = b;
```

main



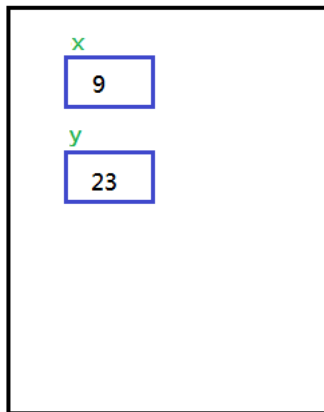
swap



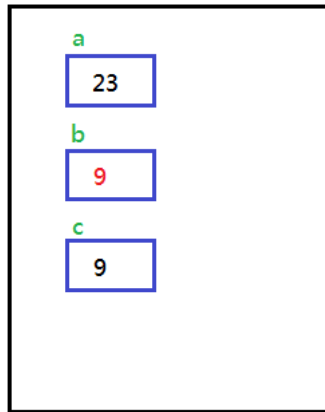
# 參數傳遞怎麼 work ?

```
b = c;
```

main



swap

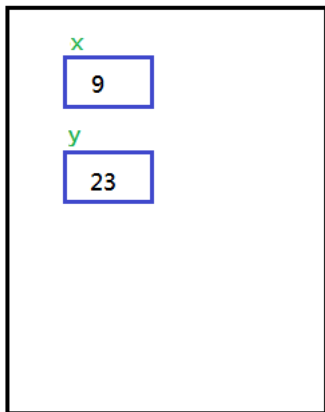




# 參數傳遞怎麼 work ?

```
}
```

main



## 參數傳遞怎麼 work ?

- 特殊情況：陣列

```
void setCdr(int a[2], int v) {  
    a[1] = v;  
}  
  
int main() {  
    int cons[2] = {1, 2};  
  
    // cons 是 {1, 2}  
    setCdr(cons, -1);  
    // cons 是 {1, -1}  
  
}
```

## 參數傳遞怎麼 work ? (錯誤)

- 可以打比方嗎？

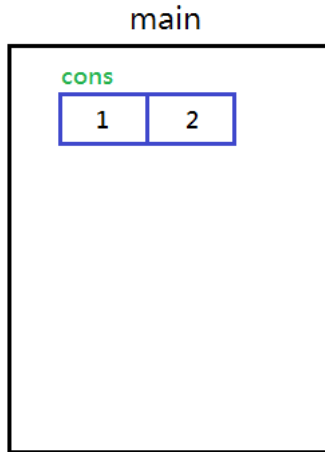
```
void setCdr(int a[2], int v) {  
    a[1] = v;  
}  
  
int main() {  
    int cons[2] = {1,2};  
  
    // cons 是 {1, 2}  
    // setCdr(cons, -1)  
    int a[2] = cons, v = -1;  
    a[1] = v;  
}
```

- 沒有 `int a[2] = cons` 這種寫法：這是什麼意思？

# 參數傳遞怎麼 work ?

- 更改前

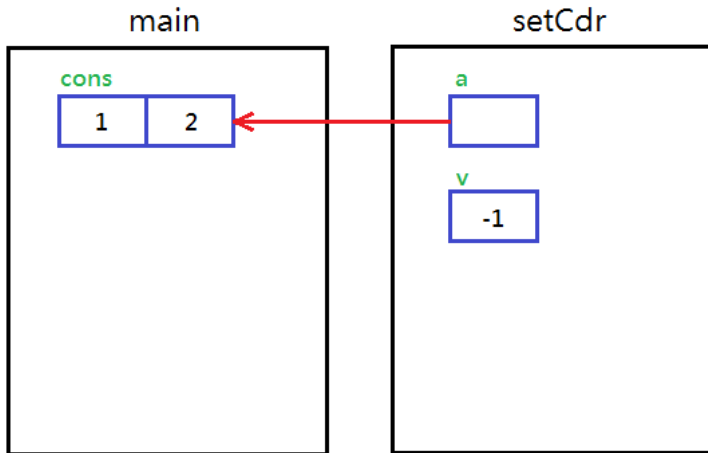
```
setCdr(cons, -1);
```



## 參數傳遞怎麼 work ?

- 更改前

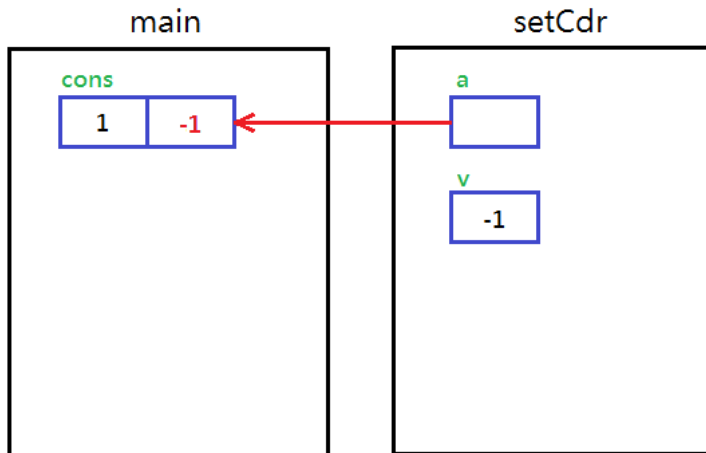
```
void setCdr(int a[2], int v) {
```



# 參數傳遞怎麼 work ?

- 更改後

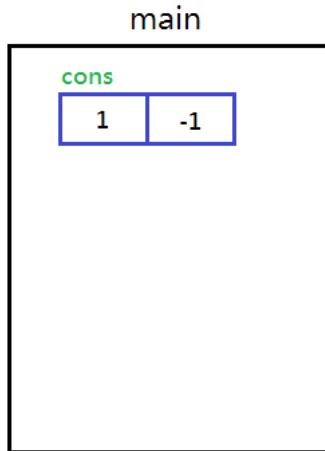
```
a[1] = v;
```



# 參數傳遞怎麼 work ?

- 更改後； `a` 是個指標，裡面存著 `cons` 陣列的起始位址

```
}
```

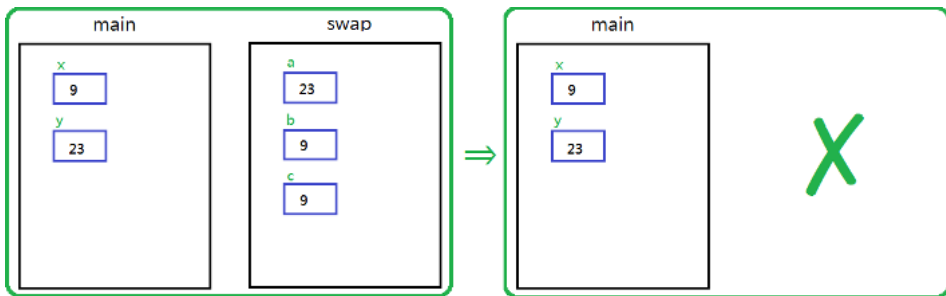


## 區域變數、參數的生命週期

- 執行時，參數、區域變數生命週期只在函數內

```
void swap(int a, int b) {  
    int c = a;  
    a = b;  
    b = c;  
}
```

- 剛才圖中的 `a`, `b`, `c`





# 函數名稱的可見性

- 宣告後函數才可見

```
// test 不可見
```

```
void test();
```

```
// test 可見
```

# 函數名稱的可見性

- 直接實作亦可

```
// test 不可見
```

```
void test() {  
}
```

```
// test 可見
```

# 函數名稱的可見性

- 範例

```
void test();

int main() {
    test();
}

void test() {
    std::cout << "in test()\n";
}
```