



請以純粹黑白列印

# Function Pointer

指向 function 的指標

井民全製作

# Introduction

- Function 與 variable 一樣都有位址
  - Function 位址 → function 的機器碼開始位置
  - Variable 位址 → variable 記憶體存放位置



我們可以把 function 位址傳給另一個 function.

Function 切換

# 簡單的問題

- 如果我要寫一個，能計算任意 function 的執行時間，怎麼辦？

Step 1:

先取得待測 function 的位址

Step 2:

宣告一個 **function pointer**  
指向該待測 function

Step 3:

利用 function pointer  
呼叫 **function**

# 如何取得 function 位址

- 就是 function name 去掉括弧

Think() → Think 即為 function address

解析

Process (**Think**); // 傳遞 Think address 給 process

比較 Think()

Thought ( Think() );

傳遞 Think 的 return value

# 宣告一個 function pointer

- Function pointer 的宣告必須包含
  - Return type & argument list

```
double pam (int); // prototype
```

Function Pointer

```
double (*pf) (int);
```

取得 **function address**:

```
double pam(int); → Function prototype
```

```
double (*pf) (int); → 宣告指標
```

```
pf=pam;
```

取得位址

```
double y = (*pf) (5)
```

呼叫範例

# 完整範例

```
#include <iostream>
using namespace std;      } for cout

double pam (int); // function prototype
int main( ) {
    // Step 1: 宣告一個 function pointer
    double (*pf) (int);

    pf=pam;

    // Step 2: 利用 function pointer 呼叫的範例
    int ans=(*pf)(123);

    cout << "ans="<<ans;
    return 0;
}
// pam function 定義
double pam (int x){
    return x*2.0;
}
```