

函數語法

hydai@0411

資訊之芽語法班 2015

函數宣告

宣告？

- 變數要使用以前要宣告
- 當然函數也要！

宣告 Con't

- `returnType` functionName (`arguments`);
- ex:
 - `int sum(int a, int b);`
 - `double add(double a, double b);`

return type

- 代表回傳的型態
 - int, double, char...
- 特例：不回傳東西時
 - void
- 如果不是“void”的話，務必要回傳東西

arguments

- 代表要吃進來的參數
- 可以有很多個，使用逗號(,)分開
- 如同宣告變數一樣：
 - `type var`

Example: sum

```
int sum (int a, int b);
```

- 這樣宣告了一個叫做 `sum` 的函數
- `sum` 的參數需要傳進來兩個整數，分別是 `a`, `b`
- 最後會回傳一個整數


```
int sum (int a, int b);
```

- 這樣宣告了一個叫做 `sum` 的函數
- `sum` 的參數需要傳進來兩個整數，分別是 `a`, `b`
- 最後會回傳一個整數

```
int sum (int a, int b);
```

- 這樣宣告了一個叫做 `sum` 的函數
- `sum` 的參數需要傳進來兩個整數，分別是 `a`, `b`
- 最後會回傳一個整數

函數實作

實作？

- 我們剛才已經定義好函數的模樣（宣告）
- 但是沒有定義說裡頭到底做了什麼事情
- 現在就來看一下怎麼把功能完成吧！

實作 con't

- `returnType` functionName (`arguments`) {
- `// Do something`
- `return XX;`
- }

```
int sum (int a, int b);
```

```
1  int sum (int a, int b) {  
2      return a+b;  
3  }
```

```
int sum (int a, int b);
```

```
1  int sum (int a, int b) {  
2      int c = a+b;  
3      return c;  
4  }
```

咦？怎麼好像看過類似的？

- 沒有錯啦 ○ A ○ ~
- 寫在函數裡面的東西跟在 `main()` 寫的東西沒兩樣
- 還記得嗎？我們曾經提過 `main()` 就是一個函數喔！

函數怎麼運作的？

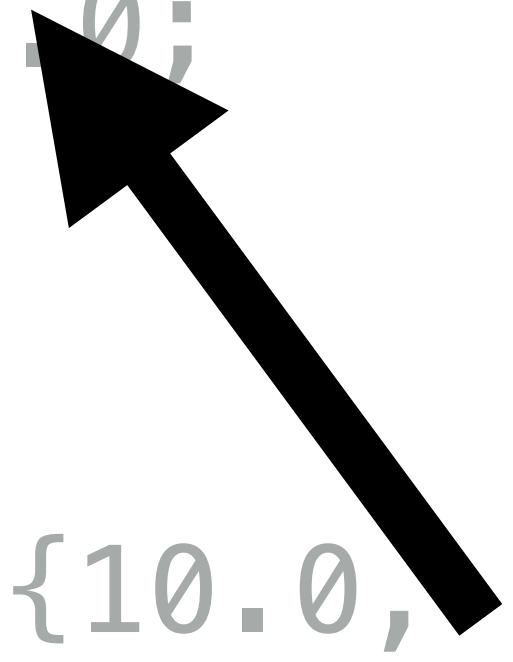
```
1  double average (double a, double b, double c) {
2      return (a+b+c)/3.0;
3  }
4  int main() {
5      ...
6      double arr[3] = {10.0, 20.0, 30.0};
7      double avg = average(arr[0], arr[1], arr[2]);
8      std::cout << avg << std::endl; // Output: 20
9      ...
10 }
```

呼叫函數

```
1 double average (double a, double b, double c) {  
2     return (a+b+c)/3.0;  
3 }  
4 int main() {  
5     ...  
6     double arr[3] = {10.0, 20.0, 30.0};  
7     double avg = average(arr[0], arr[1], arr[2]);  
8     std::cout << avg << std::endl; // Output: 20  
9     ...  
10 }
```

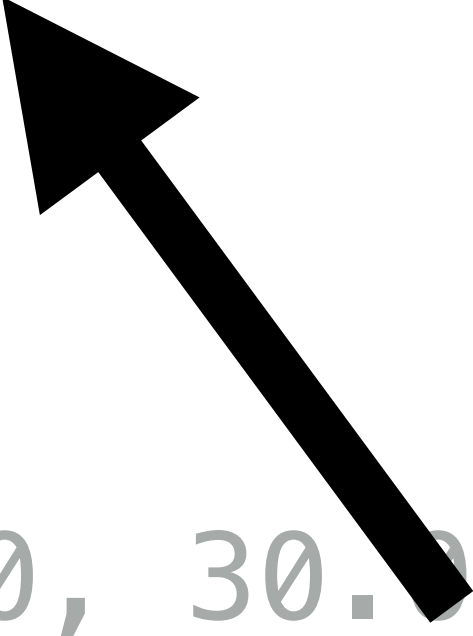
傳遞變數

```
1 double average (double a, double b, double c) {  
2     return (a+b+c)/3.0;  
3 }  
4 int main() {  
5     ...  
6     double arr[3] = {10.0, 20.0, 30.0};  
7     double avg = average(arr[0], arr[1], arr[2]);  
8     std::cout << avg << std::endl; // Output: 20  
9     ...  
10 }
```



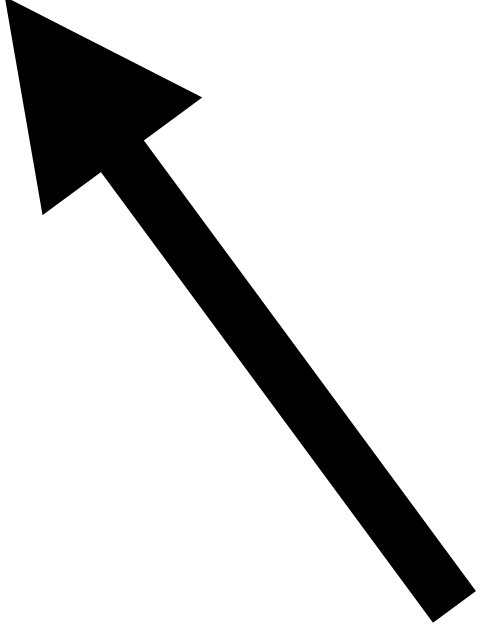
傳遞變數

```
1 double average (double a, double b, double c) {  
2     return (a+b+c)/3.0;  
3 }  
4 int main() {  
5     ...  
6     double arr[3] = {10.0, 20.0, 30.0};  
7     double avg = average(arr[0], arr[1], arr[2]);  
8     std::cout << avg << std::endl; // Output: 20  
9     ...  
10 }
```



傳遞變數

```
1 double average (double a, double b, double c) {  
2     return (a+b+c)/3.0;  
3 }  
4 int main() {  
5     ...  
6     double arr[3] = {10.0, 20.0, 30.0};  
7     double avg = average(arr[0], arr[1], arr[2]);  
8     std::cout << avg << std::endl; // Output: 20  
9     ...  
10 }
```

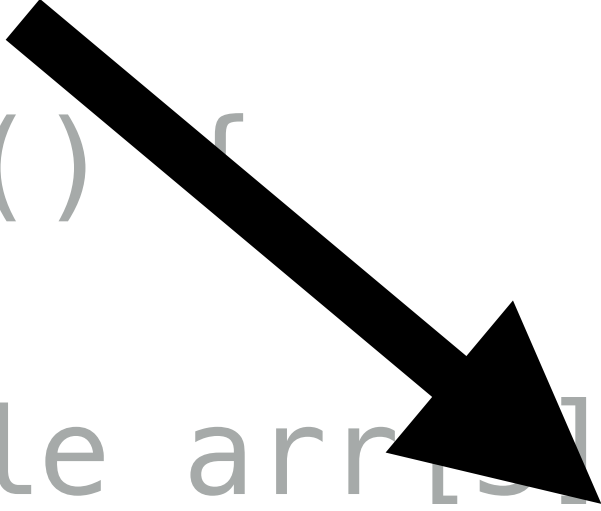


執行 average 裡頭的運算

```
1 double average (double a, double b, double c) {  
2     return (a+b+c)/3.0;  
3 }  
4 int main() {  
5     ...  
6     double arr[3] = {10.0, 20.0, 30.0};  
7     double avg = average(arr[0], arr[1], arr[2]);  
8     std::cout << avg << std::endl; // Output: 20  
9     ...  
10 }
```

回傳資料

```
1 double average (double a, double b, double c) {  
2     return (a+b+c)/3.0;  
3 }  
4 int main() {  
5     ...  
6     double arr[3] = {10.0, 20.0, 30.0};  
7     double avg = average(arr[0], arr[1], arr[2]);  
8     std::cout << avg << std::endl; // Output: 20  
9     ...  
10 }
```



換個方式看一遍

```
1  double average (double a, double b, double c) {
2      return (a+b+c)/3.0;
3  }
4  int main() {
5      ...
6      double arr[3] = {10.0, 20.0, 30.0};
7      double avg = average(arr[0], arr[1], arr[2]);
8      std::cout << avg << std::endl; // Output: 20
9      ...
10 }
```

```
1  double average (double a, double b, double c) {
2      return (a+b+c)/3.0;
3  }
4  int main() {
5      ...
6      double arr[3] = {10.0, 20.0, 30.0};
7      double avg = average(10.0, 20.0, 30.0);
8      std::cout << avg << std::endl; // Output: 20
9      ...
10 }
```

```
1  double average (a = 10.0, b = 20.0, c = 30.0) {
2      return (a+b+c)/3.0;
3  }
4  int main() {
5      ...
6      double arr[3] = {10.0, 20.0, 30.0};
7      double avg = average(10.0, 20.0, 30.0);
8      std::cout << avg << std::endl; // Output: 20
9      ...
10 }
```

```
1  double average (a = 10.0, b = 20.0, c = 30.0) {
2      return (10.0+20.0+30.0)/3.0;
3  }
4  int main() {
5      ...
6      double arr[3] = {10.0, 20.0, 30.0};
7      double avg = average(10.0, 20.0, 30.0);
8      std::cout << avg << std::endl; // Output: 20
9      ...
10 }
```

```
1  double average (a = 10.0, b = 20.0, c = 30.0) {
2      return 60.0/3.0;
3  }
4  int main() {
5      ...
6      double arr[3] = {10.0, 20.0, 30.0};
7      double avg = average(10.0, 20.0, 30.0);
8      std::cout << avg << std::endl; // Output: 20
9      ...
10 }
```

```
1  double average (a = 10.0, b = 20.0, c = 30.0) {
2      return 20.0;
3  }
4  int main() {
5      ...
6      double arr[3] = {10.0, 20.0, 30.0};
7      double avg = average(10.0, 20.0, 30.0);
8      std::cout << avg << std::endl; // Output: 20
9      ...
10 }
```

```
1  double average (a = 10.0, b = 20.0, c = 30.0) {
2      return 20.0;
3  }
4  int main() {
5      ...
6      double arr[3] = {10.0, 20.0, 30.0};
7      double avg =20.0;
8      std::cout << avg << std::endl; // Output: 20
9      ...
10 }
```