

# C++程式設計

---

陣列

# 陣列

陣列(Array): 一群相同資料型態的變數所組成的資料結構

變數: `int a;`

a 










`a = 0;`

`sum = sum + a;`

陣列: `int a[10];`

`a[0] = 0;`

`sum = sum + a[i];`

a[0]   
a[1]   
a[2]   
a[3]   
a[4]   
a[5]   
a[6]   
a[7]   
a[8]   
a[9] 

# 陣列宣告

---

資料型態 陣列名稱[陣列大小];

```
int a[1000];
```

 ✓

```
int n=500;  
int b[n];
```

 ✓

```
int i;  
int c[i];
```

 ✗

# 陣列初值設定

資料型態 陣列名稱[n] = {初值0, 初值1, ..., 初值n-1};

```
int day[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

```
int day[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; //根據初值個數決定陣列長度
```

	0	1	2	3	4	5	6	7	8	9	10	11
day	31	28	31	30	31	30	31	31	30	31	30	31

# 陣列初值設定

---

資料型態 陣列名稱[n] = {初值0, 初值1, ..., 初值n-1};

```
int data[5] = {1, 2, 3}; //不足的會補0
```

	0	1	2	3	4
data	1	2	3	0	0

```
int data[5] = {}; //初值皆為0
```

	0	1	2	3	4
data	0	0	0	0	0

# 陣列範例

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int score[] = {80,75,88,92,91};
7      float sum=0;
8
9      for(int i=0;i<5;i++)
10         sum += score[i];
11
12     cout << sum/5 << endl;
13 }
```

85.2

# 陣列範例-最大(小)值

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int score[]={80,75,88,92,91};
7      int min=score[0],max=score[0];
8      int length=sizeof(score)/sizeof(int); //計算陣列元素個數
9
10     cout << "陣列的元素: ";
11     for(int i=0;i<length;i++)
12     {
13         cout << score[i] << " ";
14         if(score[i]>max)
15             max=score[i];
16         if(score[i]<min)
17             min=score[i];
18     }
19     cout << endl;
20     cout << "最大值: " << max << endl;
21     cout << "最小值: " << min << endl;
22     return 0;
23 }
```

陣列的元素: 80 75 88 92 91  
最大值: 92  
最小值: 75

# 多維陣列

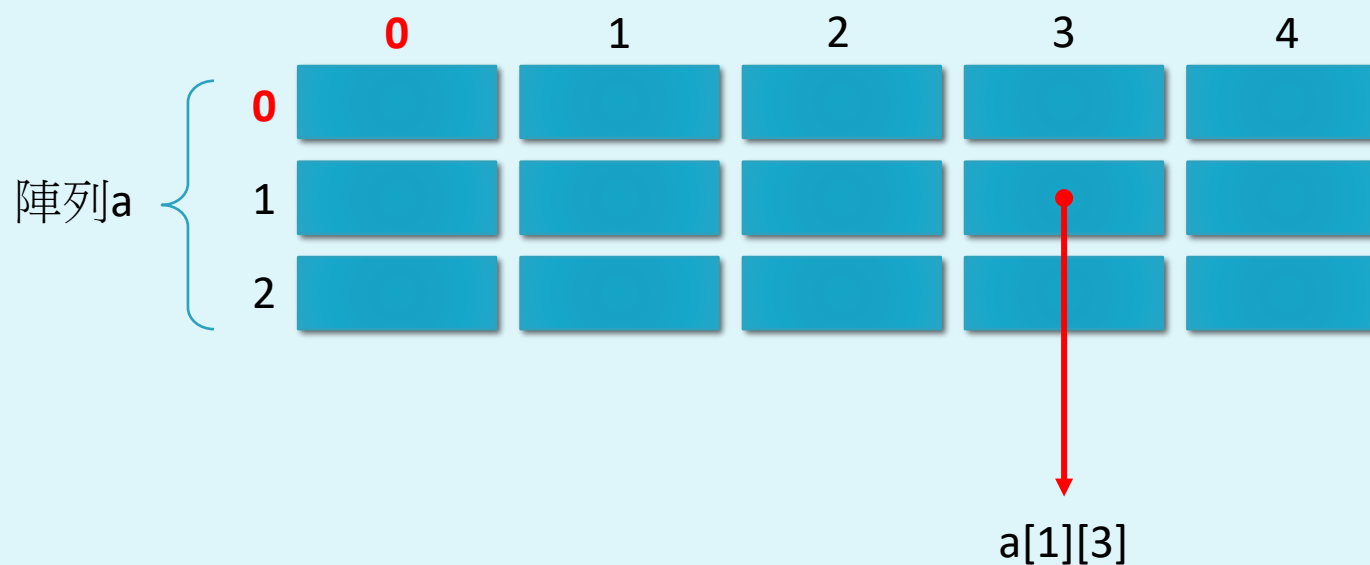
多維陣列: 即陣列的陣列, 以下為2維陣列範例

2維陣列:

```
int a[3][5];
```

```
a[2][2] = 0;
```

```
sum += a[i][j];
```





# 多維陣列初值設定

資料型態 陣列名稱[列數][行數] = { { 第0列初值 },  
{ 第1列初值 },  
{ ... },  
{ 第n列初值 } };

```
int a[2][4] = { {80, 75, 86, 90},  
               {78, 88, 94, 92} };
```

```
int a[][4] = { {80, 75, 86, 90},  
              {78, 88, 94, 92},  
              {86, 84, 83, 89} };
```

多維陣列的第一個註標值可以省略

		0	1	2	3
a	0	80	75	86	90
	1	78	88	94	92

		0	1	2	3
a	0	80	75	86	90
	1	78	88	94	92
	2	86	84	83	89

# 多維陣列

多維陣列的維度並無限制,但使用上應注意所佔記憶體空間大小

如: 以下陣列將消耗超過3GB的記憶體

```
char century[100][365][24][60][60];
```

多維陣列僅為提供程式設計者方便的抽象表示法,

其結果與註標值相乘的1維陣列相同

如: `int a[3][5];`

與 `int a[15];`

相同



# 多維陣列範例

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[3][5];
7
8      for(int i=0;i<3;i++)
9      {
10         for(int j=0;j<5;j++)
11         {
12             a[i][j]=(i+1)*(j+1);
13             cout << a[i][j] << '\t';
14         }
15         cout << endl;
16     }
17     return 0;
18 }
```

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[3*5];
7
8      for(int i=0;i<3;i++)
9      {
10         for(int j=0;j<5;j++)
11         {
12             a[i*5+j]=(i+1)*(j+1);
13             cout << a[i*5+j] << '\t';
14         }
15         cout << endl;
16     }
17     return 0;
18 }
```

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15