



6. For 迴圈

2024資訊研究社語法班
Made with ❤️ by jheanlee

While很好用 但有點亂

如果你要寫從0數到9的程式，
你可能會這樣寫：

```
int i = 0;
while (i < 10) {
    cout << i << '\n';
    i++;
}
```

但是每次都要在外面放變數，迴圈後面還要做加減
不是很亂嗎

所以一個新的東西出現了：for迴圈

```
for (int i = 0; i < 10; i++) {
    cout << i << '\n';
}
```

for迴圈怎麼用

```
for (初始變數; 終止條件; 加減) {  
    code_true();  
}
```

有什麼好處?

程式比較整齊 比較好讀懂

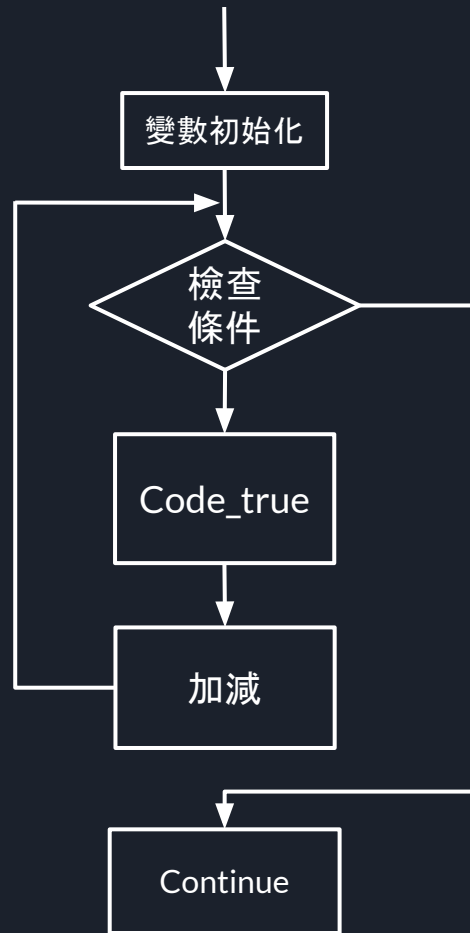
相較while比較不會忘記加減 (忘記程式就不會停了)

for用在哪裡?

在有固定終止條件、無過多變化的迴路中常見

那while是不是就沒用了?

while在動態、難以預測、無固定終止條件、複雜的迴路中常被使用



強制終止的break 直接跳過的continue

break 是強制終止一層迴圈的關鍵字
目前學過的迴圈: for, while
(未來有機會會講switch)


```
int i = 0;
while (i < 10000000) {

    if (i == 10) {
        break; // 碰到這裡會直接離開迴圈
    }

    cout << i << '\n';
    i++;
}
```

continue 是強制跳過一次迴路的關鍵字
目前學過的迴圈: for, while

```
0  for (int i = 0; i < 10; i++) {      0
1      if (i == 5) {                  1
2          continue;                  2
3      }                              3
4                                      4
5                                      5 5被跳過了
6                                      6
7      cout << i << '\n';              7
8  }                                  8
9                                      9
```



```
for (int i = 0; i < 3; i++) {  
    cout << "第" << i << "次 continue前面" << endl;  
    if (i == 1) {  
        continue; //如果 i == 1 就跳過  
    }  
    cout << "第" << i << "次 continue後面" << endl;  
}
```

第0次 continue前面
第0次 continue後面
第1次 continue前面
第2次 continue前面
第2次 continue後面

```
for (int i = 0; i < 3; i++) {  
    cout << "第" << i << "次 break前面" << endl;  
    if (i == 1) {  
        break; //如果 i == 1 就終止  
    }  
    cout << "第" << i << "次 break後面" << endl;  
}
```

第0次 break前面
第0次 break後面
第1次 break前面