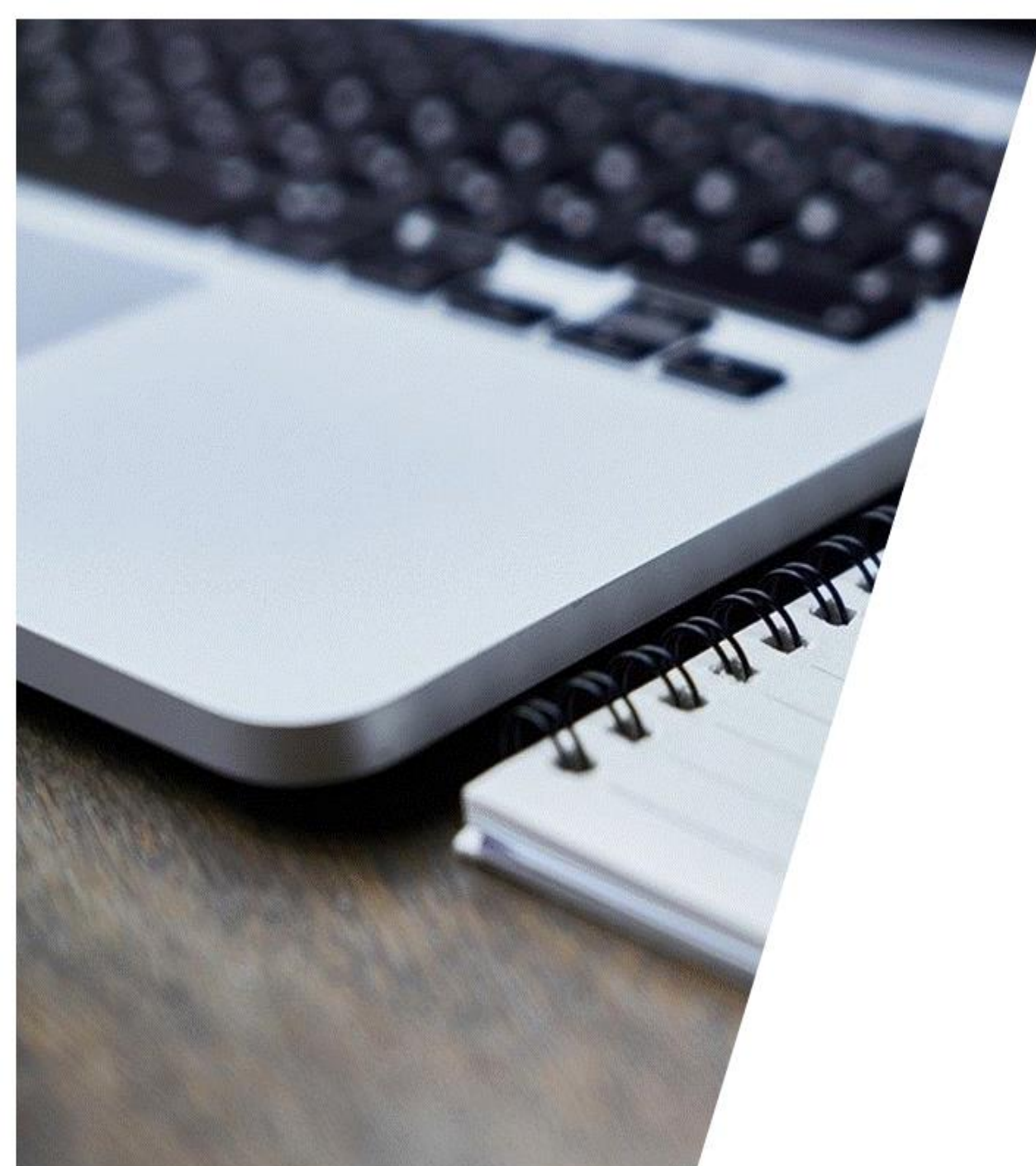




C 語言

重複結構-for



在C 語言有三種重複結構

```
初始值;  
while( 條件式 ){  
    敘述式;  
    條件的改變;  
}
```

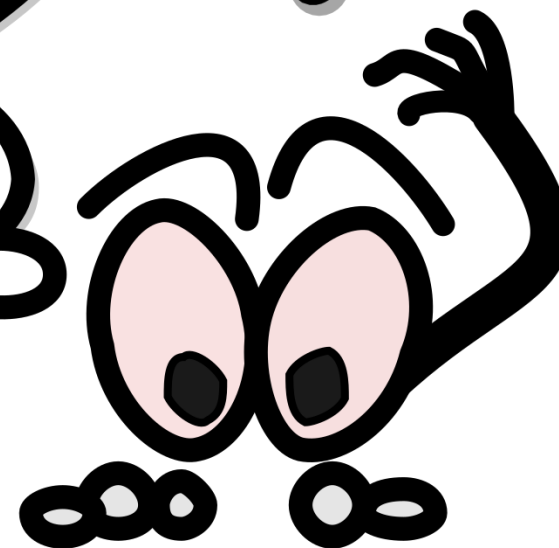
```
初始值;  
do{  
    敘述式;  
    條件的改變;  
}while( 判斷式 );
```

```
for( 初始值; 條件式; 條件變更 ){  
    敘述式;  
}
```

練習

1到10的和

??



```
#include <stdio.h>

int main() {
    int i = 1, sum = 0;
    do{
        sum = sum + i;
        i = i + 1;
    }while(i<=10);
    printf("%d\n", sum);
    return 0;
}
```

用do...while解

```
#include <stdio.h>

int main() {
    int i = 1, sum = 0;
    while(i<=10){
        sum = sum + i;
        i = i + 1;
    }
    printf("%d\n", sum);
    return 0;
}
```

用while解



有比while
更簡潔的寫法喔

for 子句執行過程

```
for( 控制變數初始值; 迴圈反覆的條件; 增量表達式){  
    語句1;  
    語句2;  
}
```



- 1 執行控制變量初始化語句，使控制變量獲得一個初值
- 2 判斷控制變量是否滿足條件表達式，若滿足條件則執行循環體
- 3 循環體
- 4 根據增量表達式，計算出變量

for：將初始值、條件式與條件改變寫到結構中

```
#include <stdio.h>
```

```
int main() {  
    int x;  
    for(x=1; x<=10; x=x+1){  
        printf("%d\n",x);  
    }  
    return 0;  
}
```



- 1 設定初值
使得x為1
- 2 是否滿足條件式
x是否小於等於10
- 3 循環體
印出x
- 4 計算變量
X每一次遞增1

將while的三個部分，精簡爲一行

第一個部分：
設定x的初始值
爲1，即 `a=1`;

第二個部分：
設定回圈條件，
即 `a<=10`;

第三個部分：
a
每次增加1，即
`a=a+1`;

```
for(x=1; x<=10; x=x+1){  
    printf("%d\n",x);  
}
```

改寫計算一個10人班級的平均分數計算

```
#include <stdio.h>
int main(){
    int counter=1;
    int grade;
    int total=0;
    float average;
    while(counter<=10){
        scanf( "%d" ,&grade);
        total=total+grade;
        counter=counter+1;
    }
    average=(float)total/10;
    printf( "%.2f\n" ,average);
    return 0;
}
```

- 用for改寫迴圈結構中的while

```
#include <stdio.h>
int main(){
    int counter;
    int grade;
    int total=0;
    float average;
    for (counter=1;counter<=10;counter=counter+1){
        scanf( "%d" ,&grade);
        total=total+grade;
    }
    average=(float)total/10;
    printf( "%.2f\n" ,average);
    return 0;
}
```

for 子句執行過程

```
for( 控制變數初始值(expression 1); 迴圈反覆的條件(expression 2); 增量表達式(expression 3)){  
    語句1;  
    語句2;  
}
```

- for敘述式裡的三個運算式都是可有可無的，但是敘述中的兩個；號是不可以省略的
 - 如果初始值已經設定好，expression1可以省略，也可以多個初始值，之間以，隔開
 - 如果省略了expression2的話，則C會認為控制條件永遠為真，因而建立一個無窮迴圈
 - 如果遞增的動作在迴圈主體中執行，或者不需要遞增動作，則expression3便可省略

for 敘述式的注意事項

- 初始值指定，迴圈繼續條件，和遞增的部分都可以包含算術運算式
- 遞增量可以是負的
- 如果迴圈繼續條件一開始為偽，則迴圈的本體部分將不會執行。程式會接著會由for敘述式之後的第一個敘述式開始執行
- 控制變數經常會由迴圈本體列印或使用，但並非一定要這麼做。我們也常只用控制變數來控制重複的次數，而在迴圈本體內並不會執行。

使用for 敘述式的例子

- 將控制變數從1遞增到100
`for(i=1;i<=100;i++)`
- 將控制變數從100遞減到1
`for(i=100;i>=1;i--)`
- 將控制變數從8到88每次遞增8
`for(i=8;i<=88;i+=8)`
- 將控制變數從20到2每次遞增-2
`for(i=20;i>=2;i-=2)`
- 將控制變數按照下列數列進行改變：2, 6, 10, 14, 18, 22, 26
`for(i=2;i<=26;i+=4)`
- 將控制變數按照下列數列進行改變：99, 88, 77, 66, 55, 44, 33, 22, 11, 0
`for(i=99;i>=0;i-=11)`

for是否比while好用呢?

- 簡潔多了
- 特別適合用在迴圈計次時，如重複100次，或重複n次時
- 並不是所有狀況都比while好用喔