# 修飾子

sprout 2019

感謝 2017 子期的投影片

# 來自第一周的連結

https://en.cppreference.com/w/cpp/language/types

這些長長短短是什麼啊?
還有謎樣的unsigned...

| Type specifier | Equivalent type | Width in bits by data model | | | | |
|---|---|---|---|---|---|---|
| | | C++ standard | LP32 | ILP32 | LLP64 | LP64 |
| short | short int | at least 16 | 16 | 16 | 16 | 16 |
| short int | | | | | | |
| signed short | | | | | | |
| signed short int | | | | | | |
| unsigned short | unsigned short int | | | | | |
| unsigned short int | | | | | | |
| int | int | at least 16 | 16 | 32 | 32 | 32 |
| signed | | | | | | |
| signed int | | | | | | |
| unsigned | unsigned int | | | | | |
| unsigned int | | | | | | |
| long | long int | at least 32 | 32 | 32 | 32 | 64 |
| long int | | | | | | |
| signed long | | | | | | |
| signed long int | | | | | | |
| unsigned long | unsigned long int | | | | | |
| unsigned long int | | | | | | |
| long long | long long int (C++11) | at least 64 | 64 | 64 | 64 | 64 |
| long long int | | | | | | |
| signed long long | | | | | | |
| signed long long int | | | | | | |
| unsigned long long | unsigned long long int (C++11) | | | | | |
| unsigned long long int | | | | | | |

# 修飾子

- “宣告” 額外性質
- 常用修飾子:
  - short、long、long long
  - unsigned
  - const
  - static

# 跟記憶體大小有關的修飾子

- 只有定義:

  sizeof(short)≦ sizeof(int) ≦ sizeof(long)≦ sizeof(long long)

- 沒特別寫出來的時候default後面接int

  short a;＝short int a;

```
#include<iostream>

int main(){
    std::cout<<"size of int = "<<sizeof(int)<<"\n";
    std::cout<<"size of long int = "<<sizeof(l
    std::cout<<"size of long long int = "<<siz
    std::cout<<"size of short int = "<<sizeof(
}
```

```
size of int = 4
size of long int = 4
size of long long int = 8
size of short int = 2
```

# limits.h

```
#define MB_LEN_MAX 5
#define SHRT_MIN (-32768)
#define SHRT_MAX 32767
#define USHRT_MAX 0xffffU
#define INT_MIN (-2147483647 - 1)
#define INT_MAX 2147483647
#define UINT_MAX 0xffffffffU
#define LONG_MIN (-2147483647L - 1)
#define LONG_MAX 2147483647L
#define ULONG_MAX 0xffffffffUL
#define LLONG_MAX 9223372036854775807ll
#define LLONG_MIN (-9223372036854775807ll - 1)
#define ULLONG_MAX 0xffffffffffffffffull
```

short 的上下界

int 的上下界

long 的上下界

# limits.h

```
#define MB_LEN_MAX 5
#define SHRT_MIN (-32768)
#define SHRT_MAX 32767
#define USHRT_MAX 0xffffU
#define INT_MIN (-2147483647 - 1)
#define INT_MAX 2147483647
#define UINT_MAX 0xffffffffU
#define LONG_MIN (-2147483647L - 1)
#define LONG_MAX 2147483647L
#define ULONG_MAX 0xffffffffUL
#define LLONG_MAX 9223372036854775807ll
#define LLONG_MIN (-9223372036854775807ll - 1)
#define ULLONG_MAX 0xffffffffffffffffull
```

short 的上下界

int 的上下界

long 的上下界

實際使用的時候可能會因為不同硬體、作業系統、編譯器而有不同的結果

# unsigned

- signed : 有正、負，用MSB分辨
- unsigned : 只有正數→上限變兩倍
  - -( 2 ^ 31 ) ~ ( 2 ^ 31 ) - 1  0 ~ ( 2 ^ 32 ) - 1

# 2的補數 (2's complement)

- 電腦紀錄負數的方法
- MSB==0表示正數， MSB==1表示負數
- 先寫出正數，01互換以後+=1
- 例:
  - -5

    5 = 00001001 → 11110110 → 11110111
  - -1

    1 = 00000001 → 11111110 → 11111111

# Overflow (溢位)

- 大小超過上下界
- Undefined behavior

```cpp
#include<iostream>
#include<limits.h>
int main(){
    int a=INT_MAX;
    std::cout<<"a = "<<a<<'\n';
    a=a+1;
    std::cout<<"a+1 = "<<a<<'\n';
    return 0;
}
```

```
a = 2147483647
a+1 = -2147483648
```

# Const

- 避免不小心被更動
- 可用於參數的傳遞

```cpp
#include<iostream>
int main(){
    const double pi=3.1415926535;
    double r;
    std::cin>>r;
    pi=3.14;
    std::cout<<r*r*pi<<'\n';
    return 0;
}
```

```
d:\資芽\teach\2019>g++ -std=c++14 cycle_area.cpp -o cycle_area.exe
cycle_area.cpp: In function 'int main()':
cycle_area.cpp:6:5: error: assignment of read-only variable 'pi'
  pi=3.14;
     ^~~~
```

# Static

- 固定在記憶體中的變數
- 只宣告 / 初始化一次，程式執行期間都會存在
  - 不隨函式結束而消失 → 統計
- 全域靜態變數 / 函式→只能在檔案內被調用
  - 宣告一個 static 變數 a 在標頭檔裡，引用該標頭檔的程式亦無法存取變數 a

# 例子

```
#include<iostream>
int count(void);
int main(){
    for(int i=0;i<5;i++){
        std::cout<<count()<<' ';
    }
    std::cout<<'\n';
    return 0;
}
int count(void){
    int i=0;
    i++;
    return i;
}
```

輸出: 1 1 1 1 1

# 例子

```
#include<iostream>
int count(void);
int main(){
    for(int i=0;i<5;i++){
        std::cout<<count()<<' ';
    }
    std::cout<<'\n';
    return 0;
}
int count(void){
    static int i=0;
    i++;
    return i;
}
```

輸出: 1 2 3 4 5

# 可以用多個修飾子

```
static const unsigned long long int solong=0;
```

↑ 這是合法的宣告