

Operator

王淇 (LittleCube)

台南一中資訊社 TFCIS

2021.9.22

Outline

- 1 算術運算子
- 2 比較運算子
- 3 邏輯運算子
- 4 賦值運算子
- 5 增減運算子
- 6 其他

算術運算子

- $+a$
- $-a$
- $a + b$
- $a - b$
- $a * b$
- a / b
- $a \% b$

算術運算子

- $+a$

- $-a$

- $a + b$

- $a - b$

- $a * b$

- a / b

- $a \% b$

- 除以 0 或求餘 0 是 Undefined Behavior

- 位元運算

算術運算子

- 位元運算
- 電腦怎麼儲存數字？

- 以 32 位元 (`int`) 為例：

算術運算子

- 以 32 位元 (`int`) 為例：

- $48763 = \underbrace{0}_{\textit{Sign}} \underbrace{00000000000000001011111001111011}_{\textit{Value}}$

算術運算子

- 以 32 位元 (`int`) 為例：

- $48763 = \underbrace{0}_{\text{Sign}} \underbrace{00000000000000001011111001111011}_{\text{Value}}$

- $0 = \underbrace{0}_{\text{Sign}} \underbrace{00000000000000000000000000000000}_{\text{Value}}$

算術運算子

■ 以 32 位元 (`int`) 為例：

■ 48763 = $\underbrace{0}_{\text{Sign}} \underbrace{00000000000000001011111001111011}_{\text{Value}}$

■ 0 = $\underbrace{0}_{\text{Sign}} \underbrace{00000000000000000000000000000000}_{\text{Value}}$

■ -1 = $\underbrace{1}_{\text{Sign}} \underbrace{11111111111111111111111111111111}_{\text{Value}}$

算術運算子

- 以 32 位元 (`int`) 為例：

- $48763 = \underbrace{0}_{\text{Sign}} \underbrace{00000000000000001011111001111011}_{\text{Value}}$

- $0 = \underbrace{0}_{\text{Sign}} \underbrace{00000000000000000000000000000000}_{\text{Value}}$

- $-1 = \underbrace{1}_{\text{Sign}} \underbrace{11111111111111111111111111111111}_{\text{Value}}$

- $-48763 = \underbrace{1}_{\text{Sign}} \underbrace{111111111111111110100000110000101}_{\text{Value}}$

算術運算子

- $\sim a$
位元 NOT 運算 (反轉所有位元)
- $a \& b$
位元 AND 運算 (取出兩個數字皆有的位元)
- $a | b$
位元 OR 運算 (取出兩個數字其中一個有的位元)
- $a \wedge b$
位元 XOR 運算 (取出兩個數字只有其中一個有的位元)

算術運算子

■ `a << b`

位元左移運算 (相當於乘上 2^b)

對於不帶號 (`unsigned`) 的版本，超出去的會刪掉；

對於帶號 (`signed`) 的版本，超出去的會導致 `Undefined Behavior`，負數也會

■ `a >> b`

位元右移運算 (相當於除掉 2^b)

對於不帶號 (`unsigned`) 跟正數的版本，超出去的會刪掉；

對於帶號 (`signed`) 的版本，負號通常不會一起右移

比較運算子

- `a == b`
判斷相等
- `a != b`
判斷不相等
- `a < b`
判斷小於
- `a > b`
判斷大於
- `a <= b`
判斷小於等於
- `a >= b`
判斷大於等於

邏輯運算子

- `!a`
NOT，把 `true` 跟 `false` 互換
- `a && b`
AND，兩個都是 `true` 才是 `true`，否則就是 `false`
- `a || b`
OR，其中一個是 `true` 就是 `true`，否則就是 `false`
- `a ^ b`
XOR，這裡的 XOR 其實是對算出來的結果 (`bool`) 做 XOR，並不是邏輯運算子內的東西

賦值運算子

- $a = b$

單純賦值，把 a 設成跟 b 一樣
少數由右往左的運算元

賦值運算子

- `a = b`

單純賦值，把 `a` 設成跟 `b` 一樣
少數由右往左的運算元

- `a += b`

- `a -= b`

- `a *= b`

- `a /= b`

- `a %= b`

- `a &= b`

- `a |= b`

- `a ^= b`

- `a <<= b`

- `a >>= b`

增減運算子

- **++a**
先加 1，再執行剩下的
- **a++**
先執行剩下的，再加 1

增減運算子

- `++a`

先加 1，再執行剩下的

- `a++`

先執行剩下的，再加 1

- `--a`

- `a--`

- 成員運算子
- `a ? b : c`
 - 如果 `a` 是 `true`，答案就是 `b`
 - 如果 `a` 是 `false`，答案就是 `c`