

資料型別

整數 `int`

- 宣告變數

- `int birds;`

- `int cats, dogs;`

- 宣告變數順便初始化

- `int birds = 10;`

- `int cats = 3, dogs = 2;`

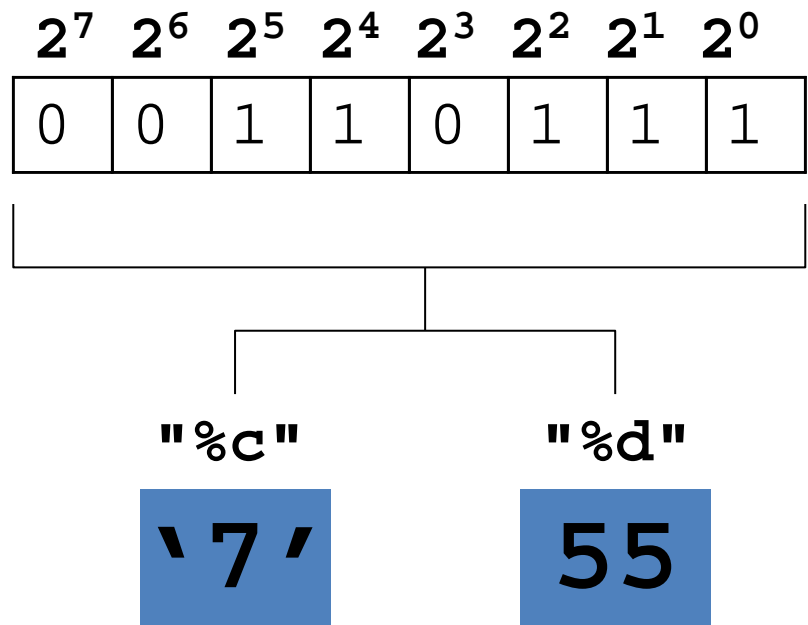
- `int cats, dogs = 2;`



找到一塊儲存空間，標示成 `birds`，
把 10 存放在裡面。

整數和浮點數

- 整數會完全用二進位編碼來儲存整個數值
- 浮點數則會拆成幾個部份，包括正負號、底數、指數三個組成元素。
- 32bit 整數範圍: -2^{31} 到 $2^{31}-1$



使用浮點數型別 `float`, `double`

```
#include <stdio.h>
int main(void)
{
    float x = 12345.0;
    double y = 2.34e12;

    printf("%f or %e\n", x, x);
    printf("%f or %e\n", y, y);

    return 0;
}
```

輸出:

```
12345.000000 or 1.234500e+004
```

```
2340000000000.000000 or 2.340000e+012
```

使用字元 char 型別

- char 型別會用到 8-bit 的記憶體空間，靠 8 bits 所儲存的數值來表達字元，譬如 'A', 'e', '7' 等等單一的英文字母或數字。
- ASCII code 編碼，可以把 0 到 127 的數值對應到不同的字元。
- **%c**
 - **scanf()** 裡用 **%c** 表示輸入的資料會被當成字元，譬如輸入的是 7，就會把 7 當成字元而不是整數 7。同樣地，**printf()** 裡用 **%c**，也是要把傳入參數的數值所對應的字元顯示出來。

使用 `sizeof` 查詢型別的佔用的位元組數

```
#include <stdio.h>
int main(void)
{
    float x;
    printf("int: %u bytes, char: %u bytes,\n",
sizeof(int), sizeof(char));
    printf("long: %u bytes, double: %u bytes.\n",
sizeof(long), sizeof(double));
    printf("x: %u bytes\n", sizeof(x));
    return 0;
}
```

執行結果和作業系統有關

型別轉換

- 當 `expression` 或 `statement` 中所出現的變數或常數之間型別不同時，C 會將變數或常數轉成相同型別。
 - 在 `expression` 中的 `char` 會轉換成 `int`，譬如我們前面看到的：`i <= 'z'`。
 - 當兩種型別混用時，位階較低的型別會轉成位階較高的；位階高低順序如下：
 - `double, float, unsigned long, long, unsigned int, int`
 - 在等式的 `statement` 中，型別會被轉換成等號左邊的變數的型別。
 - `X_double = Y_int + Z_int;`

強制型別轉換 cast

- 在數值或變數前面加上 (型別)

```
x = (int) 2.3 + (int) 2.8;
```



```
x = (int) (2.3 + 2.8 );
```

