生態模擬: 以C語言為例

Class 03 (2018/03/22)

Variables

3.1 Variables
3.2 Identifier
3.3 Data Type
3.4 Declaration
3.5 Assignment, Initialization, Re-assignment
3.6 Standard Input

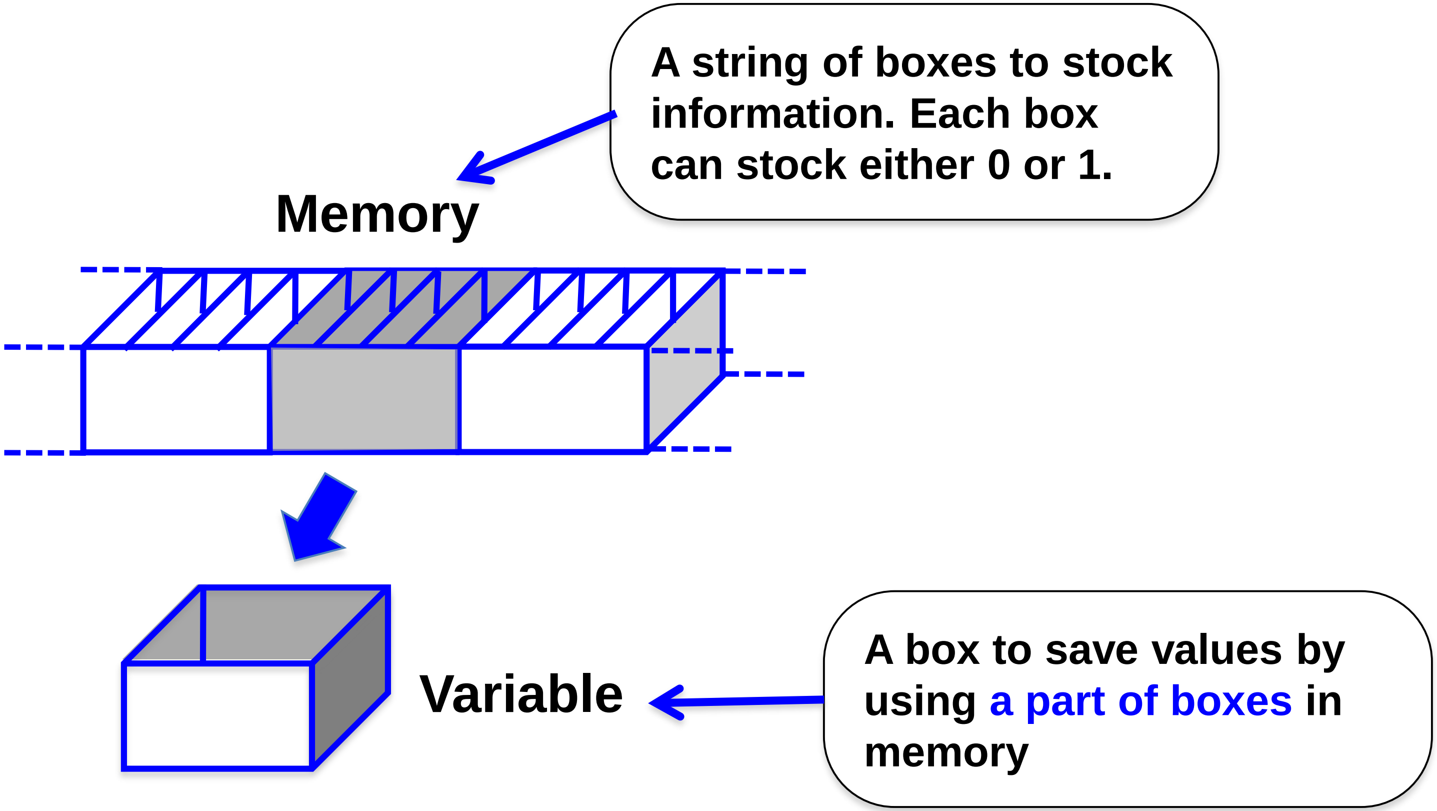
Takeshi Miki

三木　健（海洋研究所）

3.1 Variables (變數)

What is a variable?

A **variable** can save characters/numerical values **using memory**.



A string of boxes to stock information. Each box can stock either 0 or 1.

Memory

Variable

A box to save values by using a part of boxes in memory

3.2 Identifier: Specify a name of variable

If you'd like to use a variable in source codes, you need to specify two things:

(1)To determine "name" of the variable,

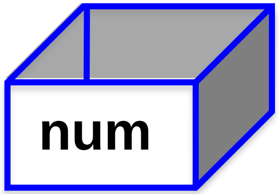
(2)To specify *data type* of the variable.

A combination of characters that can be used as a name of variable is called ***identifier***. For example, a name '**num**' is an identifier.

重要

There are rules for identifiers:

- Only alphabet, number, and _ can be used.
- Some keywords cannot be used, e.g. return.
- Cannot start with number
- Upper and lower letters are distinguished
- Not so long (e.g. < 31 characters)



例

a abc ab_c F1 ← **OK!**

12a return is-a ← **NG..**

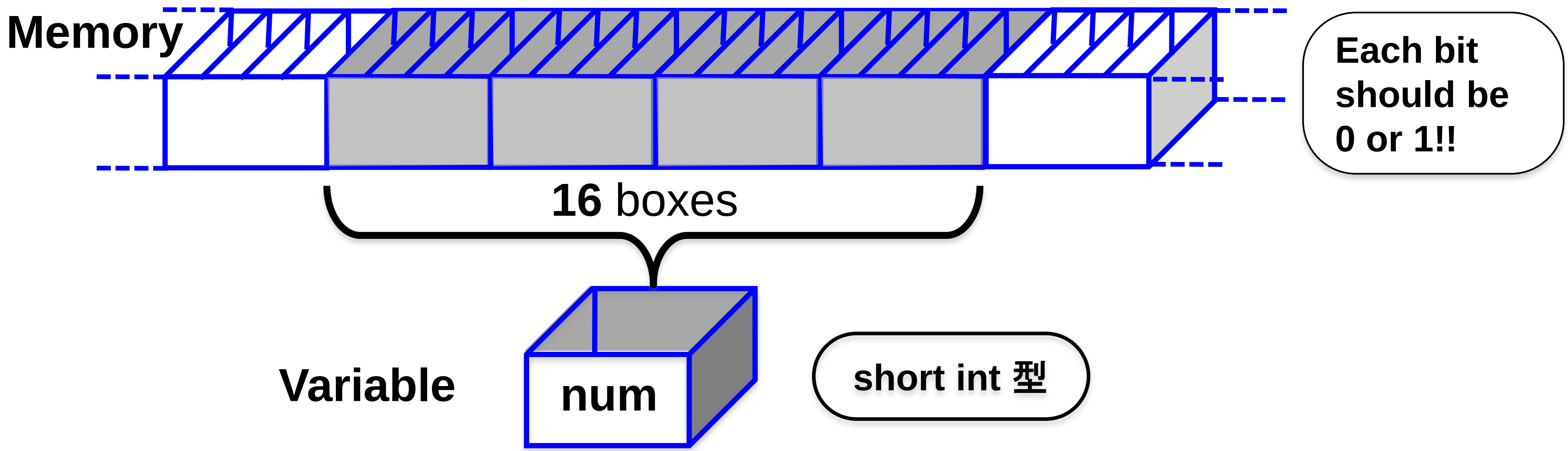
3.3 Data type (資料型)

As the second step, you need to **specify *data type* of the variable.**

This is because the program need to specify ***how many boxes in memory is allocated to the variable***.

e.g.) A data type ‘***short int***’ use **16** boxes (= **16 bits** = **2 bytes**) in the memory, which can represent **2^{16} = 65, 566** states.
This is used for integers from – 32768 (- 2^{15}) to 32767 (2^{15}-1).

e.g.) 00000000 00000101 = +5, 10000000 00000001 = -32767
[**二進位: binary numeral system**]



Memory

Each bit should be 0 or 1!!

16 boxes

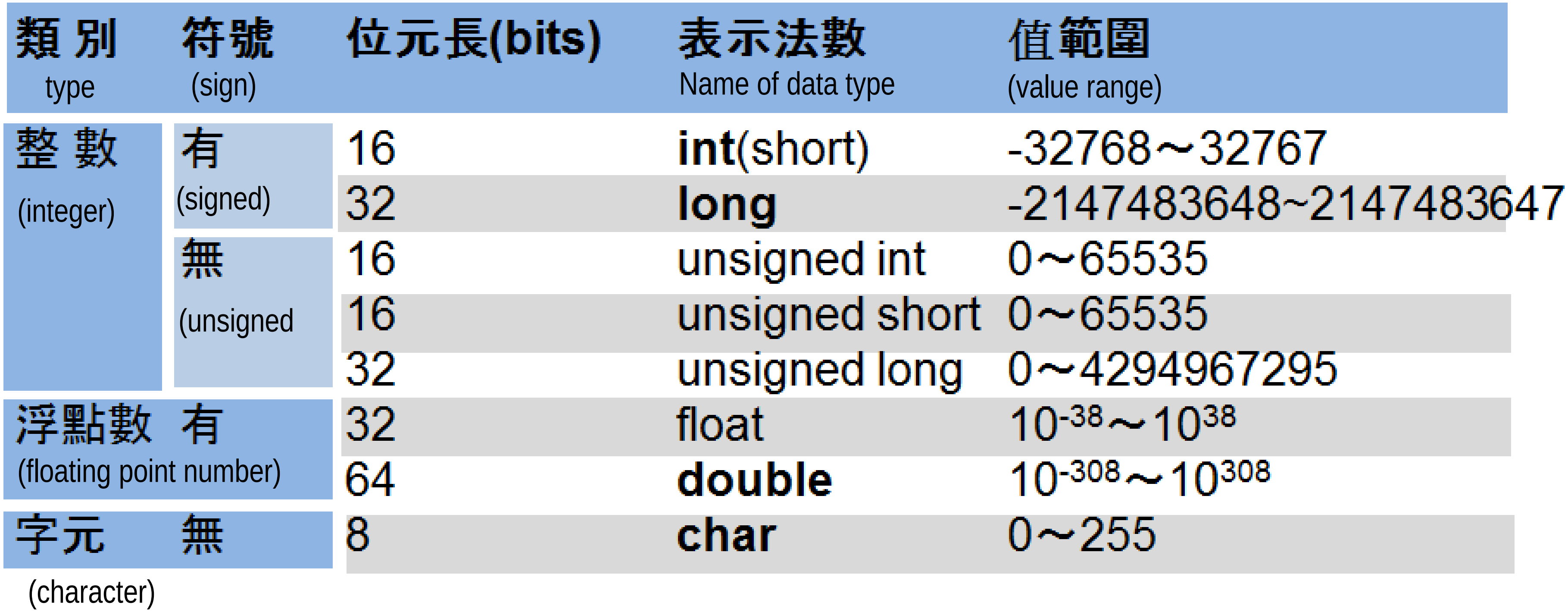
Variable

num

short int 型

3.3 Data type (資料型):變數的資料型態種類

The list of data types that are frequently used.



類 別 type	符號 (sign)	位元長(bits)	表示法數 Name of data type	值範圍 (value range)
整 數 (integer)	有 (signed)	16	**int**(short)	-32768～32767
		32	**long**	-2147483648~2147483647
	無 (unsigned	16	unsigned int	0～65535
		16	unsigned short	0～65535
		32	unsigned long	0～4294967295
浮點數 (floating point number)	有	32	float	10^{-38}～10^{38}
		64	**double**	10^{-308}～10^{308}
字元 (character)	無	8	**char**	0～255

3.4 Declaration (變數的宣告方法)

Once you decide the name and type of a variable, you need to declare it for using in the program.

構文(Syntax):

```
data_type identifier;
```

e.g.)

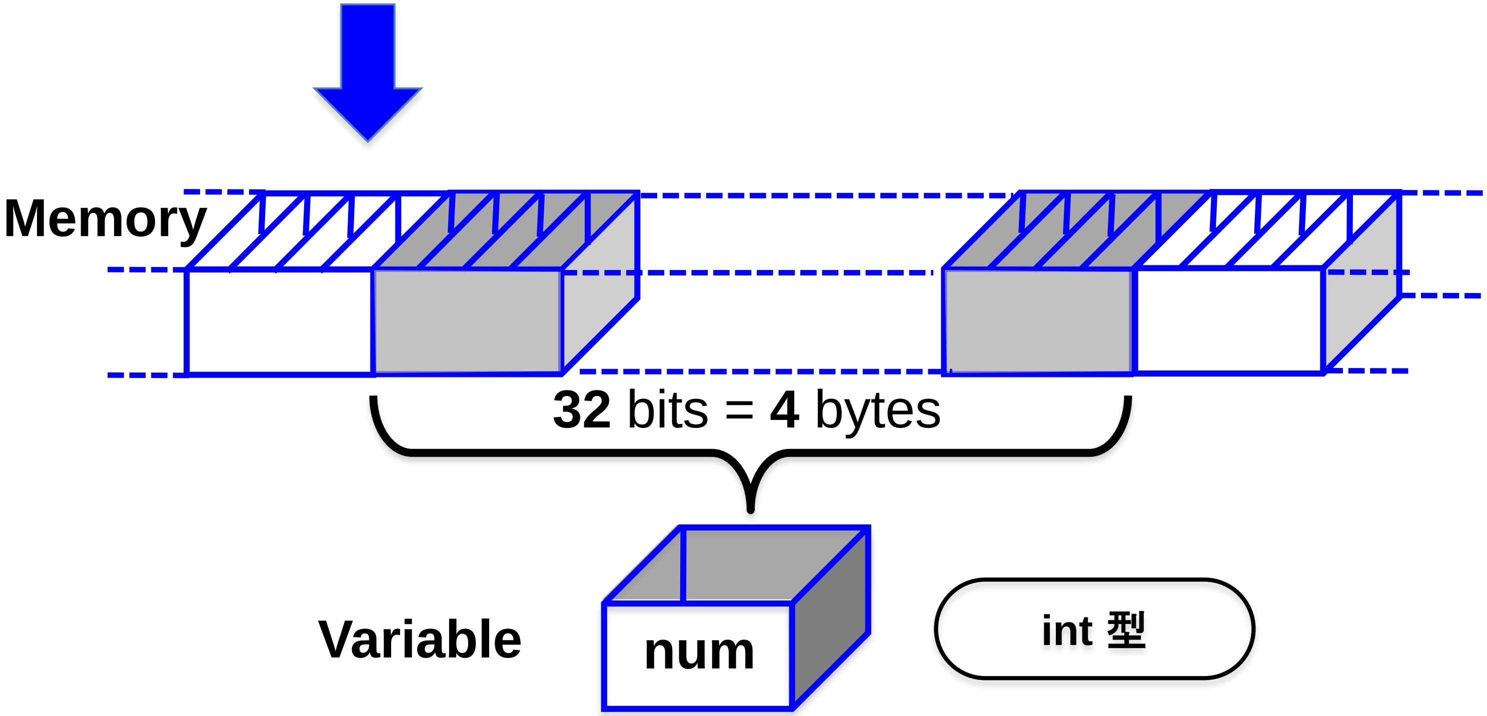
```
int num;
char c;
double db1, db2;
```

You can declare multiple variables with the same type in a single statement.

3.4 Declaration (變數的宣告方法)

By the declaration of the variable, the memory space is allocated to the variable.

```
int num;
```



3.5 Assignment, Initialization, Re-assignment

How to use variables?

(1) ***Assignment*** (代入) of a value to the variable

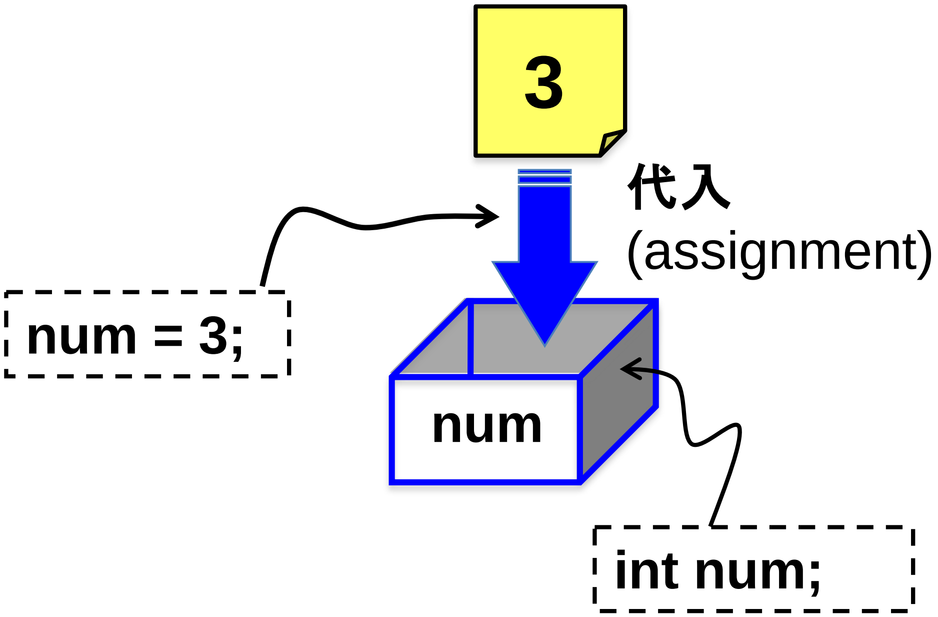
構文(Syntax):

```
name = expression;
變數名=式;
```

*We'll learn more about ‘expression’ in the next week.

This does not mean an equation or equality!

```
num = 3;
```



3.5 Assignment, Initialization, Re-assignment

How to use variables?

(2) ***Output*** of the value of the variable

Prepare a new sample file of C codes.

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

int main (void)
{
     int num;

      num = 3;

      printf("The value of the variable num is %d.\n", num);

      return 0;
}
```

宣告

代入

出力

3.5 Assignment, Initialization, Re-assignment

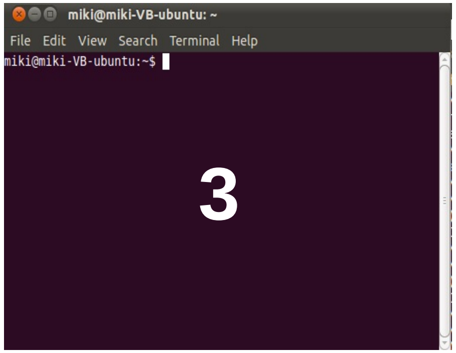
How to use variables?

(2) ***Output*** of the value of the variable

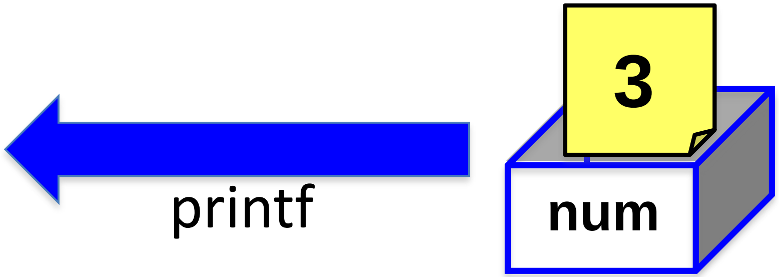
When it is executed, the output in the screen is:

```
The value of the variable num is 3.
```

```
printf(“The value of the variable num is %d.\n”,  num );
```



```
miki@miki-VB-ubuntu:~$ 

            3
```



3.5 Assignment, Initialization, Re-assignment

How to use variables?

(3) ***Initialization*** of the value of the variable

The following two statements can be modified;

```
int num;
num = 3;
```

The two statements can be replaced by a single statement (**initialization**).

```
int num = 3;
```

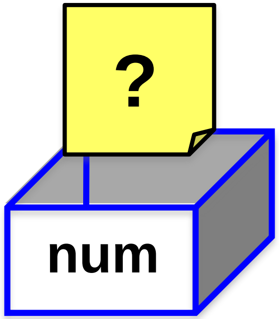
3.5 Assignment, Initialization, Re-assignment

How to use variables?

(3) ***Initialization*** of the value of the variable

Why is the initialization is important and useful?
--If you forget to assign the value to the variable, you cannot use it and leads to an error during compiling the source.

```
int num;
printf (“The value of the variable num is %d.\n”, num);
```



3.5 Assignment, Initialization, Re-assignment

How to use variables?

(4) ***Re-assimgment*** of the value of the variable

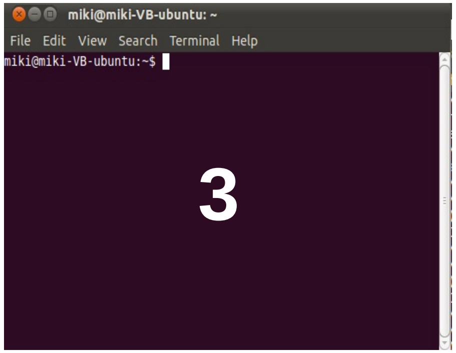
Prepare a new sample file of C codes.

```
#include <stdio.h>

int main (void)
{
        int num = 3;  初期化
        printf (“The initial value of the variable num is %d.\n”, num);  出力

        num = 5;      再代入

        printf (“The value of the variable num has been changed. \n”);
        printf (“The new value of the variable num is %d. \n”, num);

        return 0;
}
```

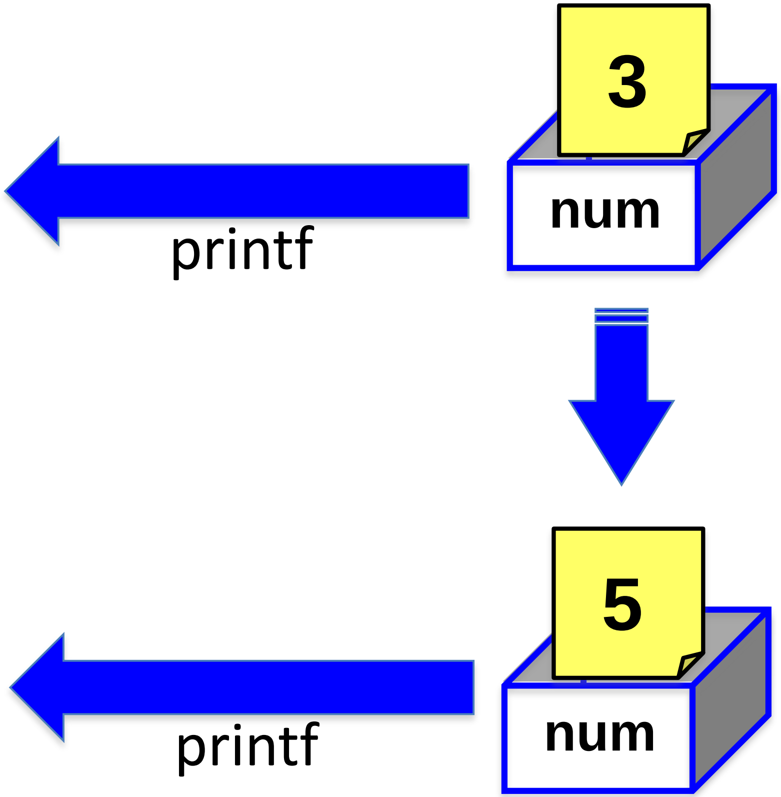
3.5 Assignment, Initialization, Re-assignment

How to use variables?

(4) ***Re-assimgment*** of the value of the variable

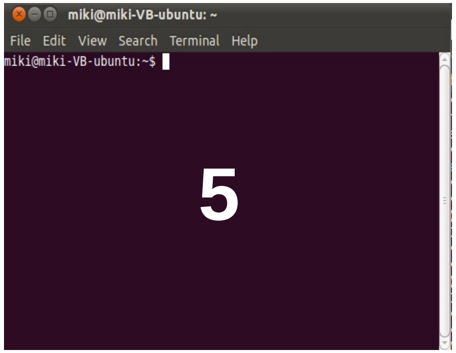


```
miki@miki-VB-ubuntu:~$

                3
```



```
int num = 3;
```

```
num = 5;
```



```
miki@miki-VB-ubuntu:~$

                5
```

3.5 Assignment, Initialization, Re-assignment

How to use variables?

(5) ***Assignment*** of the value of the variable **from another variable**

Prepare a new sample file of C codes.

```
#include <stdio.h>

int main (void)
{
    int num1, num2;
    num1 = 3;
    printf (“The value of num1 is %d.\n”, num1);

    num2 = num1;

    printf (“The value of num2 is %d.\n”, num2);

    return 0;
}
```

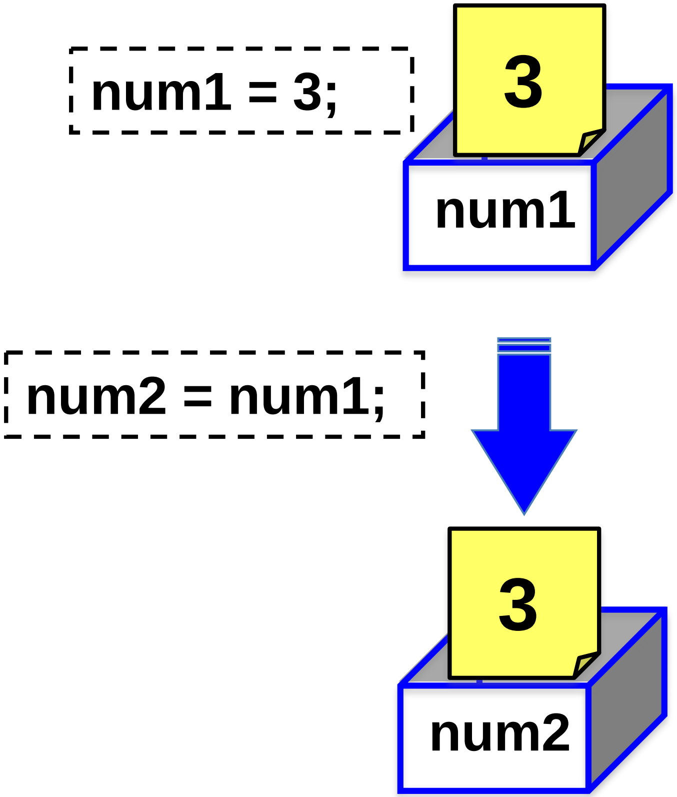
宣告

The assignment of the value of num1 to num2

3.5 Assignment, Initialization, Re-assignment

How to use variables?

(5) ***Assignment*** of the value of the variable **from another variable**

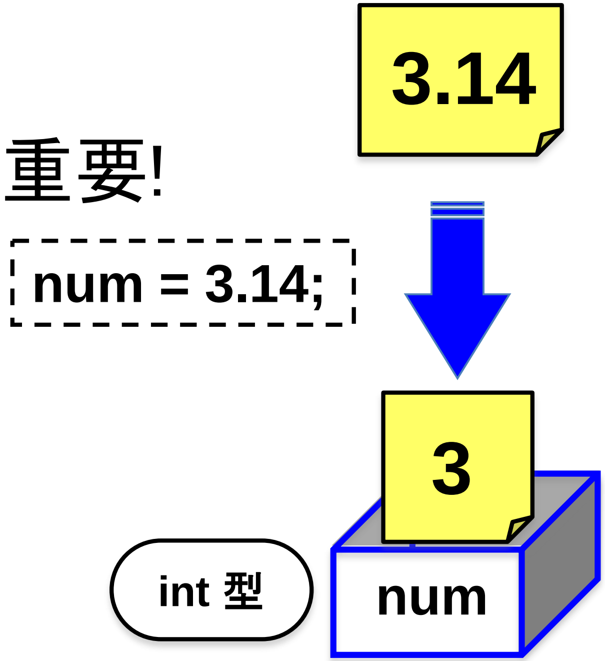


3.5 Assignment, Initialization, Re-assignment

How to use variables?

(6) ***Remark*** for the assignment of values

```
#include <stdio.h>

int main (void)
{
        int num;

        num = 3.14;

        printf ("The value of num is %d.\n", num);

        return 0;
}
```



You cannot assign full information of the float to int type variable!!

3.5 Assignment, Initialization, Re-assignment

How to use variables?

(7) ***Remark*** for the declaration of variables;

重要!

```
#include <stdio.h>

int main (void)
{
        int num;


        num = num + 1;
        printf (“XYZ….\n”, num);

        return 0;
}
```

The declaration of variables should be the initial part of the block before other statements

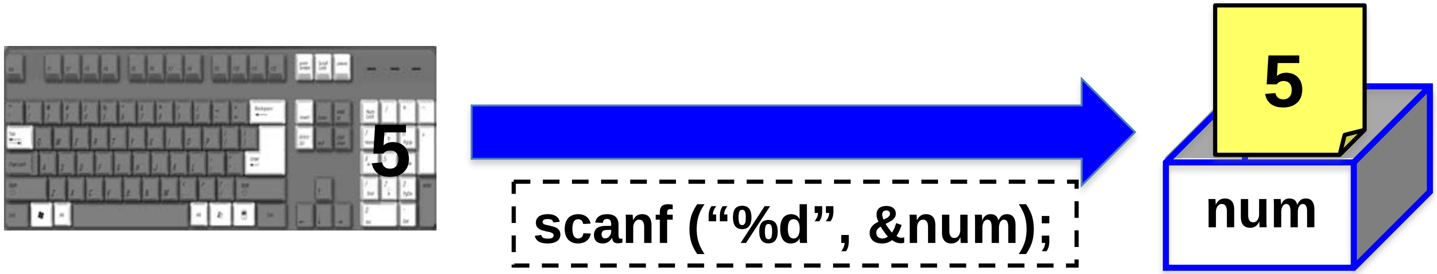
3.6 Standard Input

Input values from keyboard into variables

構文(Syntax):

```
#include <stdio.h>

int main (void)
{
    宣告;

    scanf (“conversion specification”, &variable);
    ....
}
```

You can assign the value input from the keyboard to this variable.



```
scanf (“%d”, &num);
```

3.6 Standard Input

Prepare a new sample file of C codes.

```
#include <stdio.h>

int main (void)
{
        int num;
        float f1;

        printf(“Please input an integer. \n”);
        scanf(“%d”, &num);

        printf(“Your input is %d.\n”, num);

        printf(“Please input a float. \n”);
        scanf(“%f”, &f1);

        printf(“Your input is %f.\n”, f1);

        return 0;
}
```

```
#include <stdio.h>

int main (void)
{
        int num;
        double f1;

        printf(“Please input an integer. \n”);
        scanf(“%d”, &num);

        printf(“Your input is %d.\n”, num);

        printf(“Please input a float. \n”);
        scanf(“%lf”, &f1);

        printf(“Your input is %lf.\n”, f1);

        return 0;
}
```

3.6 Standard Input

Input a **character** from the keyboard

```
#include <stdio.h>

int main (void)
{
        char ch;

        printf(“Please input a character. \n”);

        ch = getchar();

        printf(“Your input is %c.\n”, ch);

        return 0;
}
```

Homework this week

(1) Write a source code, which is intended to output/input the followings:

```
How old are you?                 (output)
23 ⏎                             (input)
You are 23 years old.            (output)
```

(2) Write a source code, which is intended to output/input the followings:

```
Please input your age and height (cm).            (output)
23 ⏎                                              (input)
172.3 ⏎                                           (input)
Your age is 23 and your height is 172.3 cm.       (output)
```