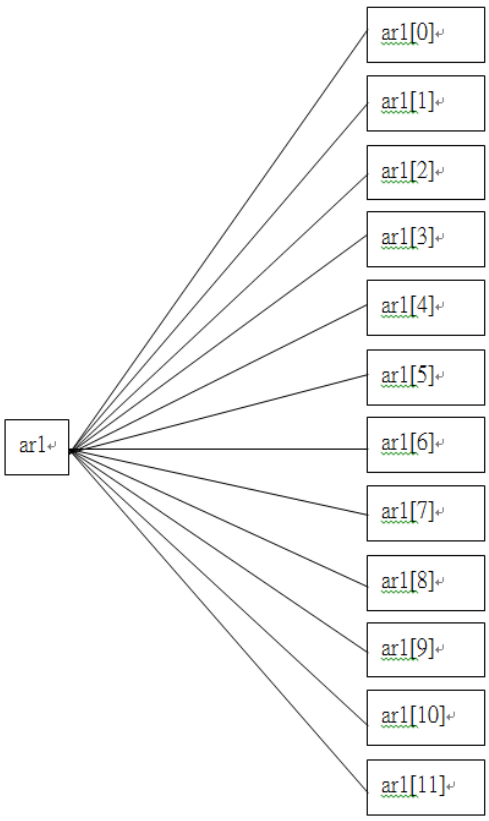
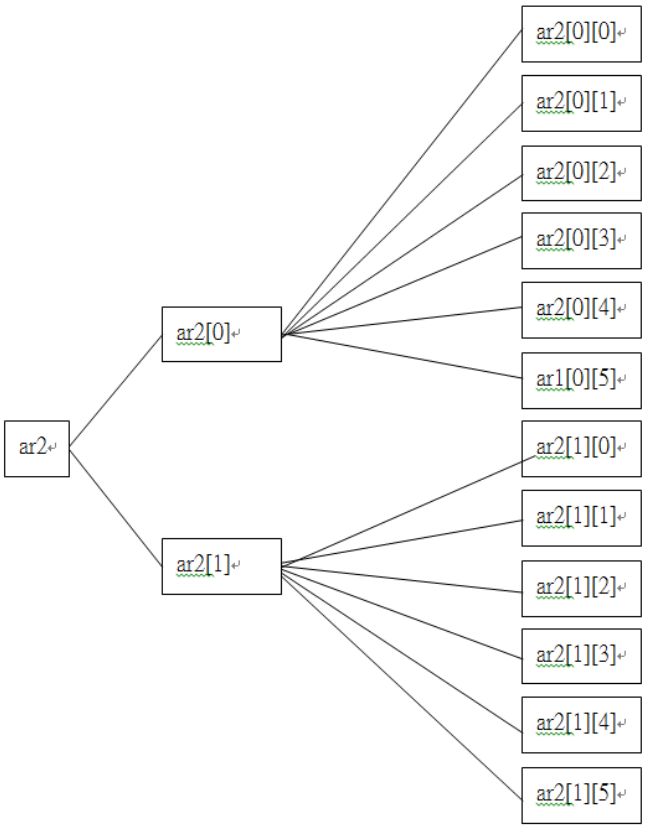
陣列結構

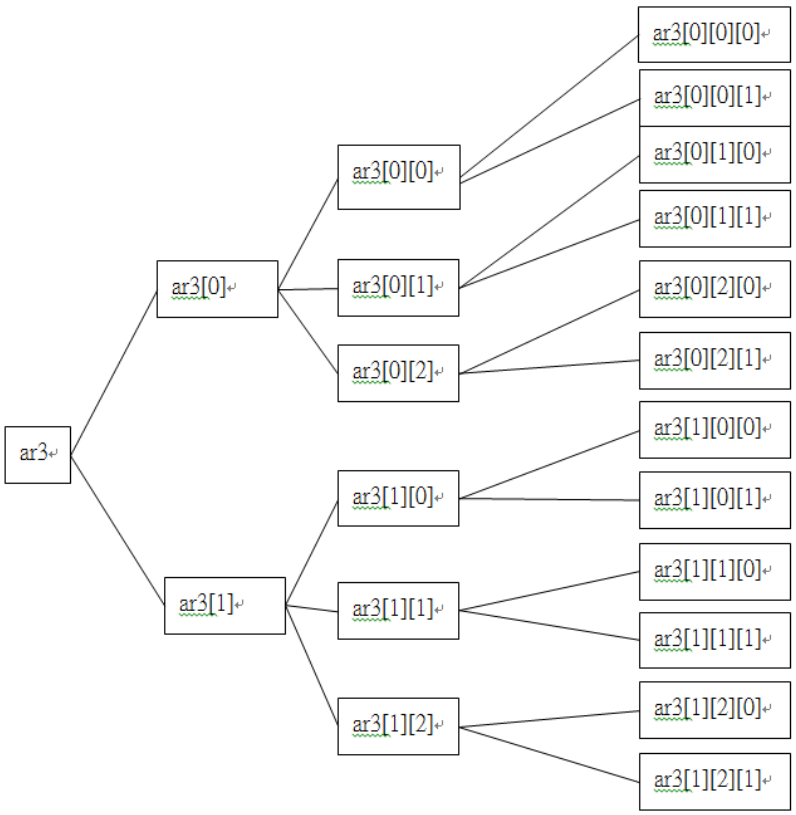
一維陣列➔int[ ] ar1=new int[12]



二維陣列➔int[ ][ ] ar2=new int[2][6]



三維陣列➔int[ ][ ][] ar3=new int[2][3][2]



java 的陣列

1. 陣列是一堆同型態的變數，記憶體中是連續的空間，可使用迴圈
2. 使用動態記憶體配置的方式宣告，維度可用變數➔int[ ] ar1=new int[5]
3. 陣列的長度是 “常數值” 不可修改➔ ~~ar1.length=10~~
4. 陣列分類

   <1>.定義陣列➔空的陣列，有初始值的陣列

   <2>.陣列的維度 ➔一維陣列，二維陣列.....

   <3>.基本型態陣列（每一維裡面都放 “值”），物件陣列（每一維裡面都放 “址”）

   <4>.二維以上➔規則陣列，不規則陣列

```
public static void 一維基本型態陣列宣告() {
    //空
    int[] ar1 = new int[10];
    int ar2[] = new int[10];
    //有初始值
    int[] ar3 = {10, 20, 30, 40, 50};
    int[] ar4 = new int[]{10, 20, 30, 40, 50};

    int[] ar41;
    ar41 = new int[5];

    int[] ar42;
    ar42 = new int[]{10, 20, 30, 40, 50};
}
```

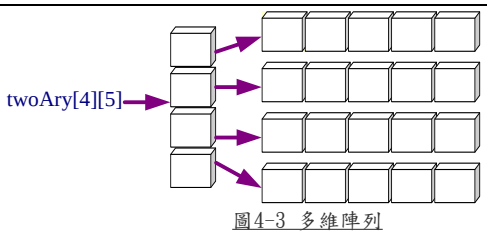


圖4-3 多維陣列

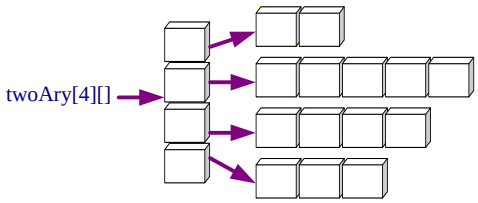


圖4-4 非矩形的多維陣列

```
public static void 二維基本型態陣列宣告() {
    //規則陣列-空
    int[][] ar5 = new int[2][3];
    int ar6[][] = new int[2][3];
    int[] ar7[] = new int[2][3];
    //規則陣列-有初始值
    int[][] ar8 = {{10, 20, 30}, {40, 50, 60}};
    int[][] ar9 = new int[][]{{10, 20, 30}, {40, 50, 60}};

    int[][] ar91;
    ar91 = new int[2][3];
    int[][] ar92;
    ar92 = new int[][]{{10, 20, 30}, {40, 50, 60}};
}
```

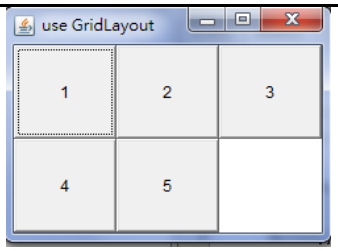
```
public static void 二維不規則陣列宣告() {
    //不規則陣列-空
    int[][] ar10 = new int[4][];
    ar10[0] = new int[2];
    ar10[1] = new int[5];
    ar10[2] = new int[4];
    ar10[3] = new int[3];
    //不規則陣列-有初始值
    int[][] ar11 = {{10, 20}, {30, 40, 50, 60, 70}, {80, 90, 100, 110}, {120, 130, 140}};
}
```

```java
public class ArrayDemo1 {

    public int x ;

    public void xyz() {
    }
}
```

```java
public static void 物件陣列宣告1() {

    ArrayDemo1 a1, a2, a3, a4, a5;

    a1 = new ArrayDemo1();
    a2 = new ArrayDemo1();
    a3 = new ArrayDemo1();
    a4 = new ArrayDemo1();
    a5 = new ArrayDemo1();

    a1.x = 10;
    a1.xyz();
}
```

```java
public static void 物件陣列宣告2() {

    ArrayDemo1[] a = new ArrayDemo1[5]; //每一個都是 null
    a[0] = new ArrayDemo1();
    a[1] = new ArrayDemo1();
    a[2] = new ArrayDemo1();
    a[3] = new ArrayDemo1();
    a[4] = new ArrayDemo1();

    a[0].x = 10;
    a[0].xyz();

    //或-------
    ArrayDemo1[] b = new ArrayDemo1[5]; //每一個都是 null
    for (int i = 0; i < b.length; i++) {
        b[i] = new ArrayDemo1();
    }

    b[0].x = 10;
    b[0].xyz();
    //或--------
    ArrayDemo1[] c = {new ArrayDemo1(), new ArrayDemo1(),
        new ArrayDemo1(), new ArrayDemo1(), new ArrayDemo1()};

    c[0].x = 10;
    c[0].xyz();
}
```

```java
public class Frame099 extends JFrame {

    public JButton B1, B2, B3, B4, B5;
    public Container ContentPane; //內容桌布

    public Frame099() {

        ContentPane = this.getContentPane(); //取得內容桌布
        GridLayout layout = new GridLayout(2, 3);//取得版面配置
        ContentPane.setLayout(layout);//內容桌布設定版面配置

        B1 = new JButton("1");
        ContentPane.add(B1);//加入內容桌布
        B2 = new JButton("2");
        ContentPane.add(B2);
        B3 = new JButton("3");
        ContentPane.add(B3);
        B4 = new JButton("4");
        ContentPane.add(B4);
        B5 = new JButton("5");
        ContentPane.add(B5);

        this.setBounds(200, 100, 250, 180);
        this.setTitle("控制項陣列");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame099();
    }
}
```




```java
public static void 字串陣列宣告() {
    //空的
    String[] ar1 = new String[5];
    //給值
    ar1[0] = "aaa";   //ar1[0]=new String("aaa");
    ar1[1] = "bbb";
    ar1[2] = "ccc";
    ar1[3] = "ddd";
    ar1[4] = "eee";
    //有初始值
    String[] ar2 = {"aaa", "bbb", "ccc", "ddd", "eee"};
}
```

```java
public static void 一維空陣列() {
    int[] ar1 = new int[5];
    StringBuilder str = new StringBuilder("");
    Scanner s = new Scanner(System.in);
    int sum = 0;

    for (int i = 0; i < ar1.length; i++) {
        System.out.printf("ar1[%d]=", i);
        ar1[i] = s.nextInt();
        str.append(ar1[i]).append(" ");
        sum = sum + ar1[i];
    }
    System.out.println(sum);
    System.out.println(str);
}
```

```java
public static void 一維有初始值陣列() {
    int[] ar1 = new int[5];
    StringBuilder str = new StringBuilder("");
    Scanner s = new Scanner(System.in);
    int sum = 0;

    for (int i = 0; i < ar1.length; i++) {
        str.append(ar1[i]).append(" ");
        sum = sum + ar1[i];
    }
    System.out.println(sum);
    System.out.println(str);
}
```

```
sum=150
10 20 30 40 50
```

```java
public static void 二維空陣列() {
    int[][] ar1 = new int[2][3];
    StringBuilder str = new StringBuilder("");
    Scanner s = new Scanner(System.in);
    int sum = 0;

    for (int i = 0; i < ar1.length; i++) {
        for (int j = 0; j < ar1[i].length; j++) {
            System.out.printf("ar1[%d][%d]=", i, j);
            ar1[i][j] = s.nextInt();
            str.append(ar1[i][j]).append(" ");
            sum = sum + ar1[i][j];
        }
        if (i < ar1.length - 1) {
            str.append("\n");
        }
    }
    System.out.println(sum);
    System.out.println(str);
}
```

```java
public static void 二維有初始值陣列() {
    int[][] ar1 = new int[2][3];
    StringBuilder str = new StringBuilder("");
    Scanner s = new Scanner(System.in);
    int sum = 0;

    for (int i = 0; i < ar1.length; i++) {
        for (int j = 0; j < ar1[i].length; j++) {
            str.append(ar1[i][j]).append(" ");
            sum = sum + ar1[i][j];
        }
        if (i < ar1.length - 1) {
            str.append("\n");
        }
    }
    System.out.println(sum);
    System.out.println(str);
}
```

```
sum=210
10 20 30
40 50 60
```

```
public static void 函數產生陣列1() {
    String[] ar1;
    String str = "aa,bb,cc";
    ar1 = str.split(",");

    for (int i = 0; i < ar1.length; i++) {
        System.out.println(ar1[i]);
    }
}
```

```
aa
bb
cc
```

```
public static void 函數產生陣列2() {
    char[] ar1;
    String str = "abcde";

    ar1 = str.toCharArray();
    for (int i = 0; i < ar1.length; i++) {
        System.out.println(ar1[i]);
    }
}
```

```
a
b
c
d
e
```

```
public static void 函數產生陣列3() {
    String[] ar1;

    try {
        FileReader fr = new FileReader("條碼.txt");
        BufferedReader br = new BufferedReader(fr);
        String data;
        while ((data = br.readLine()) != null) {
            ar1 = data.split(",");
            for(int i=0;i<ar1.length;i++){
                System.out.println(ar1[i]);
            }
            System.out.println("======================");
        }
    } catch (IOException e) {
        System.out.println(e);
    }
}
```

```
AA-982
2011029F
4712124010141
===============
AA-982
2011029H
4712124010142
===============
AA-982
2011029A
4712124010173
```

```java
public static void 快捷迴圈() {
    int[] ar11 = {1, 3, 5, 7, 9, 11};
    double[] ar22 = new double[5];
    boolean[] ar33 = new boolean[5];
    char[] ar44 = new char[5];
    ArrayDemo1[] ar55 = new ArrayDemo1[5];
    String[] ar66 = new String[5];

    int[][] ar77 = {{1, 3, 5}, {7, 9, 11}};

    for (int x : ar11) {
        System.out.print(x + " ");
    }
    System.out.println("\n============================");
    for (double x : ar22) {
        System.out.print(x + " ");
    }
    System.out.println("\n============================");
    for (boolean x : ar33) {
        System.out.print(x + " ");
    }
    System.out.println("\n============================");
    for (char x : ar44) {
        System.out.print(x + " ");  //\u0000
    }
    System.out.println("\n============================");
    for (ArrayDemo1 x : ar55) {
        System.out.print(x + " ");  //null
    }
    System.out.println("\n============================");
    for (String x : ar66) {
        System.out.print(x + " ");  //null
    }
    System.out.println("\n============================");
}
```

```
1 3 5 7 9 11
============================
0.0 0.0 0.0 0.0 0.0
============================
false false false false false
============================

============================
null null null null null
============================
null null null null null
============================
```

```java
public static void 陣列的比較() {

    int[] ar1 = {10, 20, 30};
    int[][] ar2 = {{10, 20, 30, 40, 50}, {60, 70, 80, 90, 100}};
    int[][][] ar3 = new int[2][3][4];  //全都是 0

    //System.out.println(ar1==ar2);  //錯誤
    //System.out.println(ar2[0]==ar3[0]); //錯誤
    System.out.println(ar1 == ar2[0]);  //正確 但   址 比 址 false

    ar1 = ar2[0];  //不是 5 個 塞給 3 個 只是 位址取代

    for (int i : ar1) {
        System.out.print(i + " ");
    }
    System.out.println();

    //遇到值比較的時候要小心
    System.out.println(ar1[0] == ar2[1][2]); // 值 false
    System.out.println(ar2 == ar3[0]);  //址  false
    System.out.println(ar2[0][2] == ar3[0][0][0]);  //  值 false
}
```

```java
public static void 泡沫排序法() {
    int[] num = {5, 8, 7, 6, 15, 2, 47, 16, 99, 14};
    int tmp;

    for (int i = 0; i < num.length - 1; i++) {
        for (int j = i + 1; j < num.length; j++) {
            if (num[i] > num[j]) {
                tmp = num[i];
                num[i] = num[j];
                num[j] = tmp;
            }
        }
    }
    System.out.print("由小到大排: ");;
    for (int i = 0; i < num.length; i++) {
        System.out.print(num[i] + " ");
    }
    System.out.println();
}
```

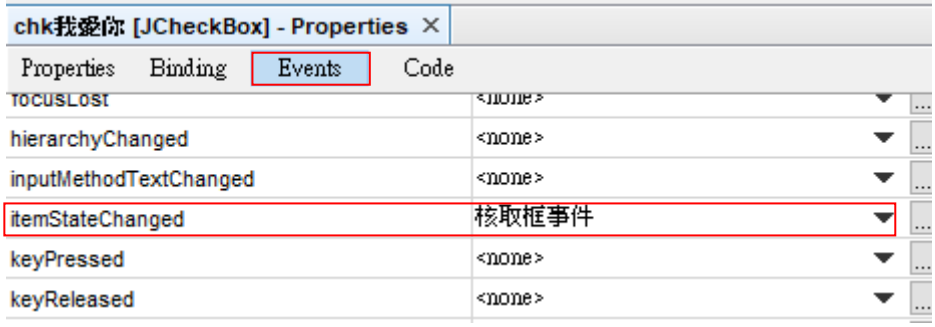
```
由小到大排: 2 5 6 7 8 14 15 16 47 99
```

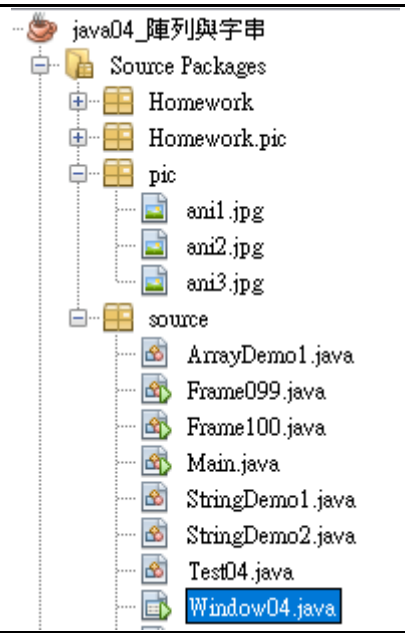
```java
public static void 找陣列中的字串() {
    Scanner s = new Scanner(System.in);
    String input;
    System.out.print("庫房 : ");
    input = s.next();

    if (input.equals("23") || input.equals("25") || input.equals("26")
            || input.equals("43")) {
        System.out.println("資料輸入正確");
    } else {
        System.out.println("資料輸入錯誤請重新輸入");
    }
}
```

```java
public static void 找陣列中的字串_陣列() {
    Scanner s = new Scanner(System.in);
    String input;

    String[] ar1 = {"23", "25", "26", "43"};
    boolean ok = false;

    System.out.print("庫房 : ");
    input = s.next();

    for (int i = 0; i < ar1.length; i++) {
        if (input.equals(ar1[i])) {
            ok = true;
            break;
        }
    }
    //跳出迴圈後
    if (ok) {
        System.out.println("資料輸入正確");
    } else {
        System.out.println("資料輸入錯誤請重新輸入");
    }
}
```

```java
public static void 找陣列中的字串_副程式() {
    Scanner s = new Scanner(System.in);
    String input;

    System.out.print("庫房 : ");
    input = s.next();

    if (found_陣列(input)) {
        System.out.println("資料輸入正確");
    } else {
        System.out.println("資料輸入錯誤請重新輸入");
    }
}

public static boolean found_陣列(String input) {

    String[] ar1 = {"23", "25", "26", "43"};

    for (int i = 0; i < ar1.length; i++) {
        if (input.equals(ar1[i])) {
            return true;
        }
    }
    return false;
}
```


```
private void 核取框事件(java.awt.event.ItemEvent evt) {

    StringBuilder str = new StringBuilder("");
    int[] ar1 = {10, 20, 30};
    int sum = 0;
    JCheckBox[] chk = {chk我愛你, chk我恨你, chk我喜歡你};

    //        if (chk我愛你.isSelected()) {
    //            str.append(chk我愛你.getText()).append(" ");
    //            sum=sum+ar1[0];
    //        }
    //        if (chk我恨你.isSelected()) {
    //            str.append(chk我恨你.getText()).append(" ");
    //             sum=sum+ar1[1];
    //        }
    //        if (chk我喜歡你.isSelected()) {
    //            str.append(chk我喜歡你.getText()).append(" ");
    //            sum=sum+ar1[2];
    //        }

    for (int i = 0; i < chk.length; i++) {

        if (chk[i].isSelected()) {
            str.append(chk[i].getText()).append(" ");
            sum = sum + ar1[i];
        }
    }
    lbl答案.setText(str.toString() + "==> 共 " + sum + " 元 ");
}
```






```
private void 選項鈕選圖片(java.awt.event.ActionEvent evt) {
    if (rdo企鵝.isSelected()) {
        lbl圖片.setIcon(new javax.swing.ImageIcon(getClass().getResource("/pic/ani1.jpg"))); // NOI18N
    } else if (rdo小貓.isSelected()) {
        lbl圖片.setIcon(new javax.swing.ImageIcon(getClass().getResource("/pic/ani2.jpg"))); // NOI18N
    } else {
        lbl圖片.setIcon(new javax.swing.ImageIcon(getClass().getResource("/pic/ani3.jpg"))); // NOI18N
    }
}
```


```
public class Window04 extends javax.swing.JFrame {

    public ImageIcon[] ic = new ImageIcon[3];
    public JCheckBox[] chk;
    public JLabel[] lbl;

    public Window04() {
        initComponents();

        ic[0] = new ImageIcon(getClass().getResource("/pic/ani1.jpg"));
        ic[1] = new ImageIcon(getClass().getResource("/pic/ani2.jpg"));
        ic[2] = new ImageIcon(getClass().getResource("/pic/ani3.jpg"));

        chk = new JCheckBox[]{chk企鵝, chk小貓, chk小狗};
        lbl = new JLabel[]{lbl企鵝, lbl小貓, lbl小狗};

        for(int i=0;i<lbl.length;i++){
            lbl[i].setIcon(ic[i]);
           lbl[i].setVisible(false);
        }
    }
```

```
private void 核取框選圖片(java.awt.event.ActionEvent evt) {
    for (int i = 0; i < chk.length; i++) {
        if (chk[i].isSelected()) {
            lbl[i].setVisible(true);
        }else{
             lbl[i].setVisible(false);
        }
    }
}
```

java 的字串➔不可變 String

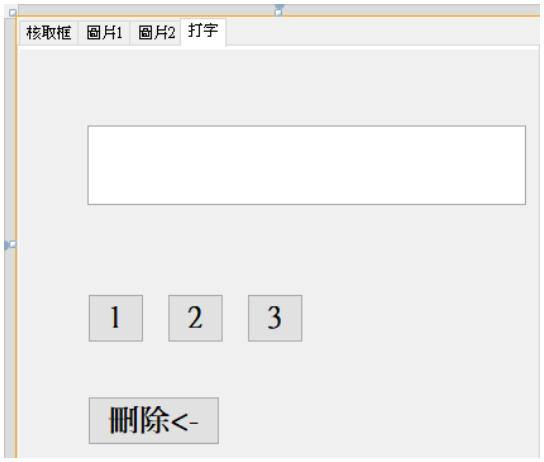
1. 單純的字元是屬於 char 原始資料型態
2. 字串主要以 String 類別來處理➔以陣列的方式存放字串
3. String 類別有一個 length()方法會傳回該字串長度的 int 值

```
public static void 字串長度() {
    String str1 = "我是 fish!";
    StringBuffer str2 = new StringBuffer("我是 fish!");
    StringBuilder str3 = new StringBuilder("我是 fish!");
    //無論中、英文或空白，每個都是 1 個字元
    System.out.println("str1 字串長度 = " + str1.length());
    System.out.println("str2 字串長度 = " + str2.length());
    System.out.println("str3 字串長度 = " + str3.length());
}
```

```
str1 字串長度 = 8
str2 字串長度 = 8
str3 字串長度 = 8
```

```
public static void 取出子字串() {
    String str1 = "abcdef";
    String stra = str1.substring(2);
    String strb = str1.substring(2, 5);
    System.out.println("stra 字串 = " + stra);
    System.out.println("strb 字串 = " + strb);
}
```

```
stra 字串 = cdef
strb 字串 = cde
```




```
private void 數字(java.awt.event.ActionEvent evt) {
    JButton btn = (JButton) evt.getSource();

    if (txtInput.getText().equals("0")) {
        txtInput.setText("");
    }

    txtInput.setText(txtInput.getText() + btn.getText());
}

private void 刪除(java.awt.event.ActionEvent evt) {
    int len;
    len = txtInput.getText().length();

    if (len != 0) {
        String str = txtInput.getText().substring(0, len - 1);
        txtInput.setText(str);
        if (txtInput.getText().length() == 0) {
            txtInput.setText("0");
        }
    }
}
```

4. equals 的覆寫

```
public class Object {

    private static native void registerNatives();
    [...]

    /**...*/
    public final native Class<?> getClass();

    /**...*/
    public native int hashCode();

    /**...*/
    public boolean equals(Object obj) {
        return (this == obj);
    }

    /**...*/
    protected native Object clone() throws CloneNotSupportedException;

    /**...*/
    public String toString() {
        return getClass().getName() + "@" + Integer.toHexString(hashCode());
    }
```

<1>.自訂的類別，除非覆寫 Object 類別的 equals 方法，否則無法比較兩物件，因 Object 類別的 equals 方法也是用 == 在比較

(1).String ➔有覆寫

(2).Byte , Short , Integer , Long ,Float ,Double , Character , Boolean 八個包裝類別➔有覆寫

※ StringBuffer , StringBuilder 與 自訂類別... ➔沒有覆寫

5. 判斷兩個 String 字串內容是否相同➔使用 equals()方法(比較參考型態)，不能用= =(比較基本型態)

有 new 與沒有 new 的差別➔沒有 new 時會先進字串池(String Pool) 搜尋，沒有搜尋到時會 new，順便將字串移進 String Pool

```
public static void String產生方式() {
    //每次必要空間
    String a1 = new String("fish");
    String a2 = new String("fish");
    String a3 = new String("fish");

    //先搜尋 String Pool
    String b1 = "fish";
    String b2 = "fish";
    String b3 = "fish";

    System.out.println("(a1==a2) ==> " + (a1 == a2));  //false
    System.out.println("(a1.equal(a2) ==> " + (a1.equals(a2))); //true

    System.out.println("(b1==b2) ==> " + (b1 == b2)); //true  虛
    System.out.println("(b1.equal(b2) ==> " + (b1.equals(b2))); //true
}
```

```java
//自訂類別==>沒有覆寫 equals
public class StringDemo1 {

    public int x = 10;

    public void xyz() {

    }
}
```

```java
//自訂類別==>有覆寫 equals
public class StringDemo2 {

    public int x = 10;

    public void xyz() {
    }

    public boolean equals(Object obj) {
        if ((obj != null && obj instanceof StringDemo2)) {
            if ((x == ((StringDemo2) obj).x)) {
                return true;
            }
        }
        return false;
    }
}
```

```java
//自訂類別因沒覆寫 Object 類別的 equals 方法 , 因此全部都是 用 == 在比
public static void equals用法11() {
    StringDemo1 a1 = new StringDemo1();
    StringDemo1 a2 = new StringDemo1();

    System.out.println("(a1==a2) ==> " + (a1 == a2));  //false
    System.out.println("(a1.equal(a2) ==> " + (a1.equals(a2)));  //false
}

public static void equals用法12() {
    StringDemo2 a1 = new StringDemo2();
    StringDemo2 a2 = new StringDemo2();

    System.out.println("(a1==a2) ==> " + (a1 == a2));  //false
    System.out.println("(a1.equal(a2) ==> " + (a1.equals(a2)));  //true
}
```

```java
//String有覆寫
    public static void equals用法21() {
        String a1 = new String("fish");
        String a2 = new String("fish");

        System.out.println("(a1==a2) ==> " + (a1 == a2));  //false
        System.out.println("(a1.equal(a2) ==> " + (a1.equals(a2)));  //true
    }
```

```java
//StringBuffer沒有覆寫
public static void equals用法31() {
    StringBuffer a1 = new StringBuffer("fish");
    StringBuffer a2 = new StringBuffer("fish");

    System.out.println("(a1==a2) ==> " + (a1 == a2));  //false
    System.out.println("(a1.equal(a2) ==> " + (a1.equals(a2)));  //false
}

//StringBuilder沒有覆寫
public static void equals用法32() {
    StringBuilder a1 = new StringBuilder("fish");
    StringBuilder a2 = new StringBuilder("fish");

    System.out.println("(a1==a2) ==> " + (a1 == a2));  //false
    System.out.println("(a1.equal(a2) ==> " + (a1.equals(a2)));  //false
}
```

```java
//八個包裝類別有覆寫
    public static void equals用法41() {
        Integer a1 = new Integer(4);
        Integer a2 = new Integer(4);

        System.out.println("(a1==a2) ==> " + (a1 == a2));  //false
        System.out.println("(a1.equal(a2) ==> " + (a1.equals(a2)));  //true
    }

    public static void equals用法42() {
        Long a1 = new Long(4);
        Long a2 = new Long(4);

        System.out.println("(a1==a2) ==> " + (a1 == a2));  //false
        System.out.println("(a1.equal(a2) ==> " + (a1.equals(a2)));  //true
    }
```

```java
public static void 兩物件比較() {

    StringDemo1 a = new StringDemo1();
    StringDemo1 b = new StringDemo1();
    StringDemo2 c = new StringDemo2();

    System.out.println("(a==b) ==> " + (a == b));  //false
    //system.out.println("(a==c) ==> " + (a == c));

    System.out.println("(a.equals(b)) ==> " + a.equals(b));  //同型態 false
    System.out.println("(a.equals(c)) ==> " + a.equals(c));  //false
}
```

6. Java 的字串有 3 種型態 ➔String , StringBuffer , StringBuilder(5.0)

String➔ 不可變 , StringBuffer 與 StringBuilder➔ 可變

```
public static void 字串的運算1() {
    String ss = "賴玉珊";
    ss.concat("我愛你");
    System.out.println(ss); //賴玉珊
    ////////////////////////////////////
    ss = ss.concat("我愛你");  //要重新指派
    System.out.println(ss); //賴玉珊我愛你
}

public static void 字串的運算2() {
    StringBuffer ss = new StringBuffer("賴玉珊");
    ss.append("我愛你");
    System.out.println(ss); //賴玉珊我愛你
}

public static void 字串的運算3() {  //5.0
    StringBuilder ss = new StringBuilder("賴玉珊");
    ss.append("我愛你");
    System.out.println(ss); //賴玉珊我愛你
}
```

```
public static void 字串的運算4() {
    String ss = "";
    for (int i = 1; i <= 10; i++) {
        ss = ss.concat(i + " ");
    }
    System.out.println(ss);
}

public static void 字串的運算5() {
    StringBuffer ss = new StringBuffer("");
    for (int i = 1; i <= 10; i++) {
        ss.append(i).append(" ");
    }
    System.out.println(ss);
}

public static void 字串的運算6() {
    StringBuilder ss = new StringBuilder("");
    for (int i = 1; i <= 10; i++) {
        ss.append(i).append(" ");
    }
    System.out.println(ss);
}
```

作業

```
public static void 總分與平均_陣列() {
    Scanner s = new Scanner(System.in);
    String[] name = {"甲同學", "乙同學", "丙同學"};
    double[][] grade = new double[3][3];
    double[] total = new double[3];
    double[] average = new double[3];

    for (int i = 0; i < grade.length; i++) {

        for (int j = 0; j < grade[i].length; j++) {

        }

    }
}
```

```
第1人
-------第1科=95
-------第2科=78
-------第3科=85
第2人
-------第1科=74
-------第2科=65
-------第3科=78
第3人
-------第1科=88
-------第2科=95
-------第3科=67
甲同學 總分= 258.0 平均= 86.0
乙同學 總分= 217.0 平均= 72.33333333333333
丙同學 總分= 250.0 平均= 83.33333333333333
```

```
public static void 統計字元() {
    String str;
    Scanner s = new Scanner(System.in);
    char[] var;
    int n1 = 0, n2 = 0, n3 = 0;


}
```

```
輸入字串 = ABCxyz123treYRT789
數字=6 個
小寫字母=6 個
大寫字母=6 個
```

```java
public static void 總分與平均_檔案() {
    String[] ar1;
    double[] total = new double[3];
    double[] average = new double[3];
    int i = 0;
    try {
        FileReader fr = new FileReader("分數.txt");
        BufferedReader br = new BufferedReader(fr);
        String data;
        while ((data = br.readLine()) != null) {
            ar1 = data.split(",");
            //
            //
            //
            i++;
        }
    } catch (IOException e) {
        System.out.println(e);
    }
}
```

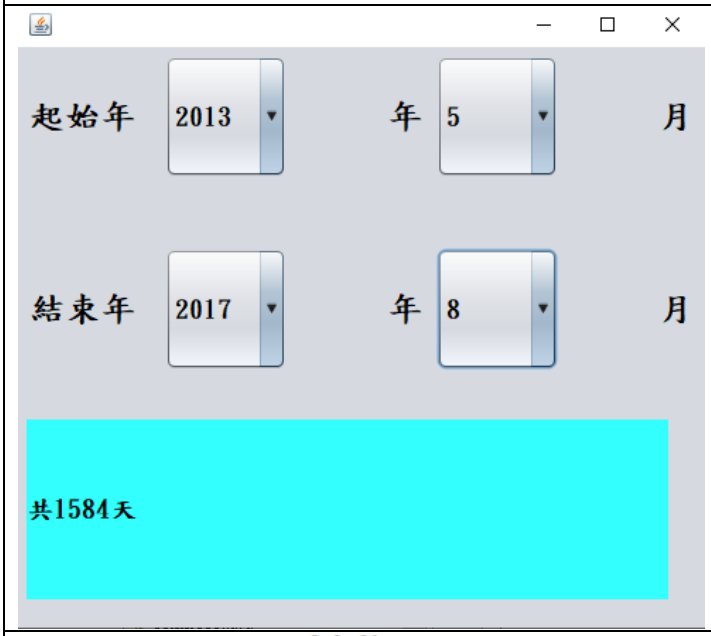
```
韓國瑜 總分= 238.0 平均= 79.33333333333333
柯文哲 總分= 178.0 平均= 59.333333333333336
蔡英文 總分= 240.0 平均= 80.0
```

```java
public static void 分數範圍() {

    int[] grade; //分數
    int[] bar = new int[11]; //長條圖 0-100 分共 11 個區間
    int number;
    Scanner s = new Scanner(System.in);


}
```

```
請輸入學生人數 (1-10 人) : 12
人數範圍為 1 ~ 10 ,請重新輸入 : 10
請輸入學生分數(0-100 分)
第 1 位 : 99
第 2 位 : 65
第 3 位 : 74
第 4 位 : 45
第 5 位 : 92
第 6 位 : 31
第 7 位 : 78
第 8 位 : 15
第 9 位 : 63
第 10 位 : 47
 === 分數分布直方圖===

100   :
90-99 : **
80-89 :
70-79 : **
60-69 : **
50-59 :
40-49 : **
30-39 : *
20-29 :
10-19 : *
0-9 :
```

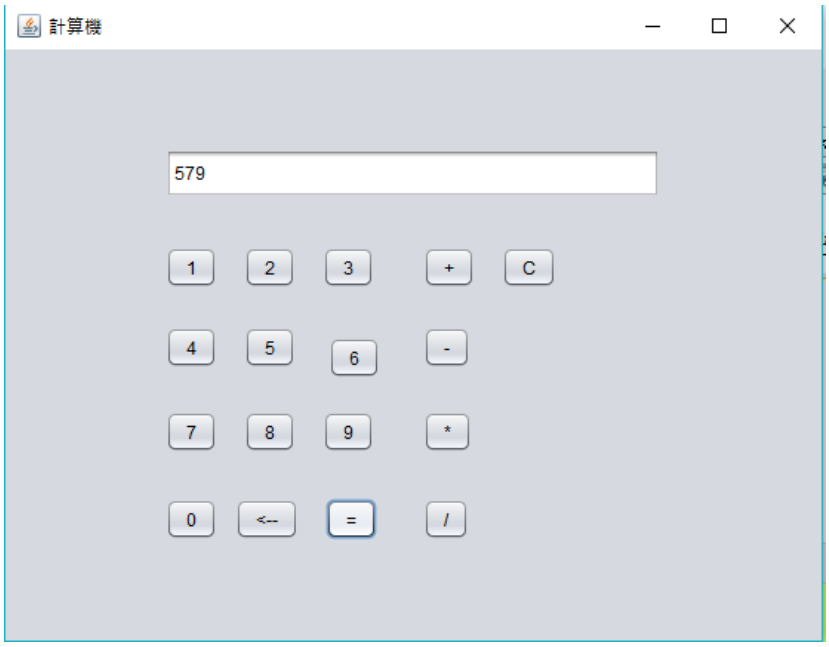
```java
public static void 計算天數() {
    int[] day1 = {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; //閏年
    int[] day2 = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; //平年
    Scanner s = new Scanner(System.in);
    int startyear;
    int startmonth;
    int finishyear;
    int finishmonth;
    int total = 0;
    System.out.print("起始年 = ");
    startyear = s.nextInt();
    System.out.print("起始月 = ");
    startmonth = s.nextInt();
    System.out.print("終止年 = ");
    finishyear = s.nextInt();
    System.out.print("終止月 = ");
    finishmonth = s.nextInt();
    //
    //
    //

    System.out.println("總天數=" + total + " 天");
}
```

```
起始年 = 2013
起始月 = 5
終止年 = 2017
終止月 = 8
總天數=1584 天
```




```java
public static void 威力彩() {

    boolean[] ok = new boolean[49 + 1]; //49個狀態
    int[] data = new int[6]; //6個數字


}
```


奶塔(30)
焗烤(150)
紅豆湯(20)
牛排(200)
蛋糕(50)
雞排(100)
品名 數量 價格
牛排 1 200
奶塔 1 30
紅豆湯 1 20
蛋糕 1 50
總金額=300元


計算機
579
1 2 3 + C
4 5 6 -
7 8 9 *
0 <-- = /