

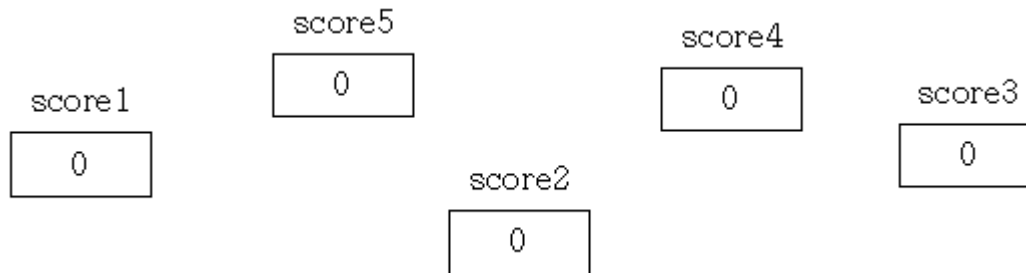
# JAVA 프로그래밍

## 배열(array)

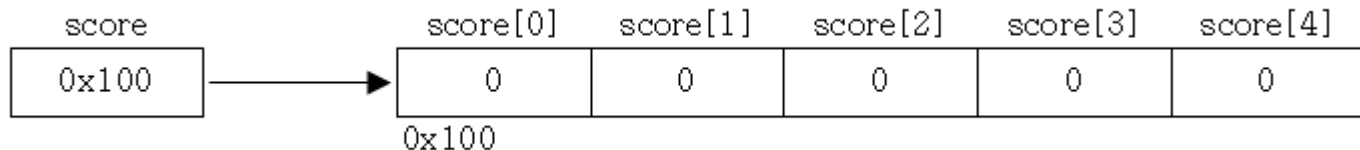
# 배열(array)

- 같은 타입의 여러 변수를 하나의 묶음으로 다루는 것
- 많은 양의 값(데이터)을 다룰 때 유용하다.
- 배열의 각 요소는 서로 연속적이다.

```
int score1=0, score2=0, score3=0, score4=0, score5=0 ;
```



```
int[] score = new int[5]; // 5개의 int 값을 저장할 수 있는 배열을 생성한다.
```



# 배열의 선언과 생성

- ① 타입 또는 변수이름 뒤에 대괄호[]를 붙여서 배열을 선언한다.

선언방법	선언 예
타입[] 변수이름;	<code>int[] score;</code> <code>String[] name;</code>
타입 변수이름[];	<code>int score[];</code> <code>String name[];</code>

- ② 배열을 선언한다고 값을 저장할 공간이 생성되는 것이 아니라, 배열을 다루는데 사용할 참조변수가 생성된다.

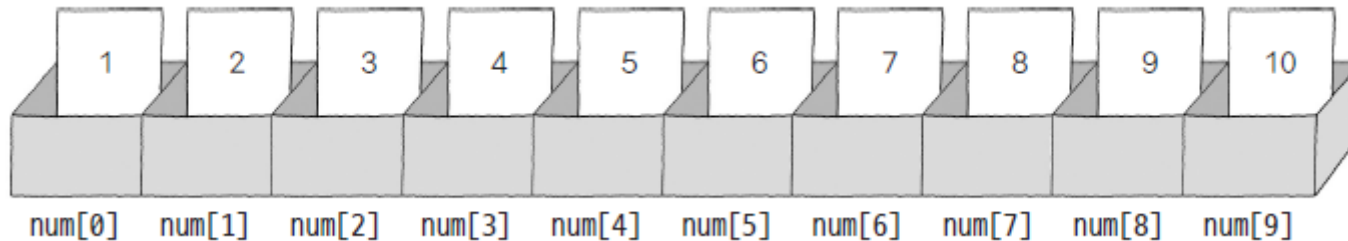
```
int[] score;           // 배열을 선언한다. (생성된 배열을 다루는데 사용될 참조변수 선언)
score = new int[5];    // 배열을 생성한다. (5개의 int값을 저장할 수 있는 공간생성)
```

위의 두 문장은 `int[] score = new int[5];`와 같이 한 문장으로 줄여 쓸 수 있다.

# 배열 사용하기

- [ ] : 인덱스 혹은 첨자 연산자  
배열의 위치를 지정하여 자료를 가져옴  
모든 배열의 순서는 0 부터 시작함  
n 개 배열은 0 부터 n-1 위치까지 자료가 존재

- 예) `int[ ] num = new int[ ]{1,2,3,4,5,6,7,8,9,10};`



# 배열의 초기화

<ArrayPractice1.java>

- 생성된 배열에 처음으로 값을 저장하는 것

```
int[] score = new int[5]; // 크기가 5인 int형 배열을 생성한다.  
score[0] = 100;           // 각 요소에 직접 값을 저장한다.  
score[1] = 90;  
score[2] = 80;  
score[3] = 70;  
score[4] = 60;
```

```
int[] score = { 100, 90, 80, 70, 60}; // 1번  
int[] score = new int[]{ 100, 90, 80, 70, 60}; // 2번
```

```
int[] score;  
score = { 100, 90, 80, 70, 60}; // 에러 발생!!!
```

```
int[] score;  
score = new int[]{ 100, 90, 80, 70, 60}; // OK
```

# 배열의 활용

- 배열에 값을 저장하고 읽어오기

```
score[3] = 100;          // 배열 score의 4번째 요소에 100을 저장한다.  
int value = score[3];    // 배열 score의 4번째 요소에 저장된 값을 읽어서 value에 저장.
```

<ArrayPractice2.java>

- ‘배열이름.length’는 배열의 크기를 알려준다.

```
int[] score = { 100, 90, 80, 70, 60, 50 };
```

```
for(int i=0; i < 6; i++) {  
    System.out.println(score[i]);  
}
```



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

# 문자 배열 만들기

- ✓ 아스키 값에 따른 문자 (A- Z)와 아스키 값이 차례로 출력

```
public class CharArray {  
    public static void main(String[ ] args) {  
        char[ ] alphabets = new char[26];  
        char ch = 'A';  
  
        for(int i = 0; i < alphabets.length; i++, ch++) {  
            alphabets[i] = ch; //아스키 값으로 각 요소에 저장  
        }  
  
        for(int i = 0; i < alphabets.length; i++) {  
            System.out.println(alphabets[i] + "," + (int)alphabets[i]);  
        }  
    }  
}
```

# 향상된 for문(Enhanced for)

- 배열을 사용하여 반복시킬 때 조금 더 편리한 방법으로 사용할 enhanced for문

```
for( 자료형 변수 : 배열 ) {    // 배열의 값이 있으면 자료형 변수로 값을 전달해 주고
    수행문 ;                  // 수행문 반복 실행
}
```

// 초기값, 종료 조건, 증감식을 숨기고 배열과 변수를 이용하여 실행문을 반복시키는 방법

```
public class ForEach {
    public static void main(String[] args) {
        int[] score = { 78, 70, 65, 98, 58 };
        int sum = 0;
        for (int i : score) {
            sum += i;           // sum = sum + i;
        }
        System.out.println("점수 합계 : " + sum);    // 점수 합계 : 369
    }
}
```



# Quiz 1차원 배열

<ArrayExam1.java> 3개의 정수 값을 저장하는 배열 변수를 선언하고 초기화한 후, 배열의 요소들의 합계와 평균을 아래와 같이 출력하시오.

[변수 선언]

정수형 배열 변수 array 선언과 초기화;

정수형 변수 sum 선언과 초기화;

실수형 변수 avg 선언과 초기화;

[출력 결과]

합계: 266

평균: 88.67

# Quiz 1차원 배열

<ArrayExam2.java> 정수 5개를 저장하는 배열 변수를 선언하고 초기화한 후 합계, 평균을 출력하고, 각 배열의 요소들은 for문을 사용하여 아래와 같이 출력하시오.

[변수 선언]

정수형 배열 변수 array 선언과 초기화 { 83,65,78,96,54 };

[출력 결과]

array[0]의 값: 83

array[1]의 값: 65

array[2]의 값: 78

array[3]의 값: 96

array[4]의 값: 54

총점: 376

평균: 75.20

# Quiz 1차원 배열

〈ArrayExam3.java〉 국어, 영어, 수학, 총점, 평균을 배열로 선언하고 아래 출력 결과와 같이 구현하세요. (총점과 평균을 계산하는 for문과 배열 요소를 출력하는 for문 각각 사용)

[변수 선언]

```
String[ ] subject = { "국어", "영어", "수학", "총점", "평균" };
```

```
int[ ] jum = { 75, 89, 96, 0, 0 };
```

[출력 결과]

국어	영어	수학	총점	평균
----	----	----	----	----

75	89	96	260	86
----	----	----	-----	----

# Quiz 1차원 배열

<MaxMin.java>

정수 5개를 저장할 배열을 생성하고 정수를 입력 받은 후, 아래의 실행 결과와 같이 최대값과 최소값을 출력하세요.

[실행 결과]

5개의 정수를 입력하세요 . . .

13 7 -3 22 34

최대값 : 34

최소값 : -3

# Quiz 1차원 배열

〈RandomLotto.java〉

Math.random()메서드를 이용하여 다음과 같이 1~45 사이의 로또 번호를 생성한 결과를 출력해 보세요.

[실행 결과]

당첨 번호는 다음과 같습니다.

-----

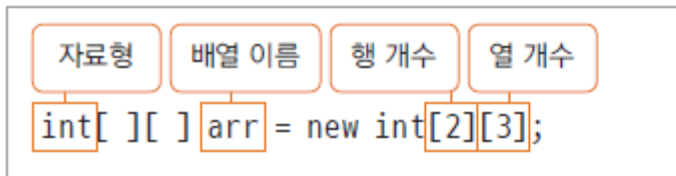
2, 26, 37, 13, 26, 43

-----

당첨을 축하드립니다! \*^^\*

# 다차원 배열

- 2차원 이상의 배열
- 지도, 게임, 등 평면이나 공간을 구현 할 때 많이 사용됨
- 이차원 배열의 선언과 구조



arr[0][0]	arr[0][1]	arr[0][2]
arr[1][0]	arr[1][1]	arr[1][2]

- 선언과 초기화

```
int[ ][ ] arr = {{1, 2, 3}, {4, 5, 6}};
```



arr[0][0]	arr[0][1]	arr[0][2]
1	2	3
arr[1][0]	arr[1][1]	arr[1][2]
4	5	6

# 다차원 배열의 선언과 생성

- ‘[ ]’의 개수가 차원의 수를 의미

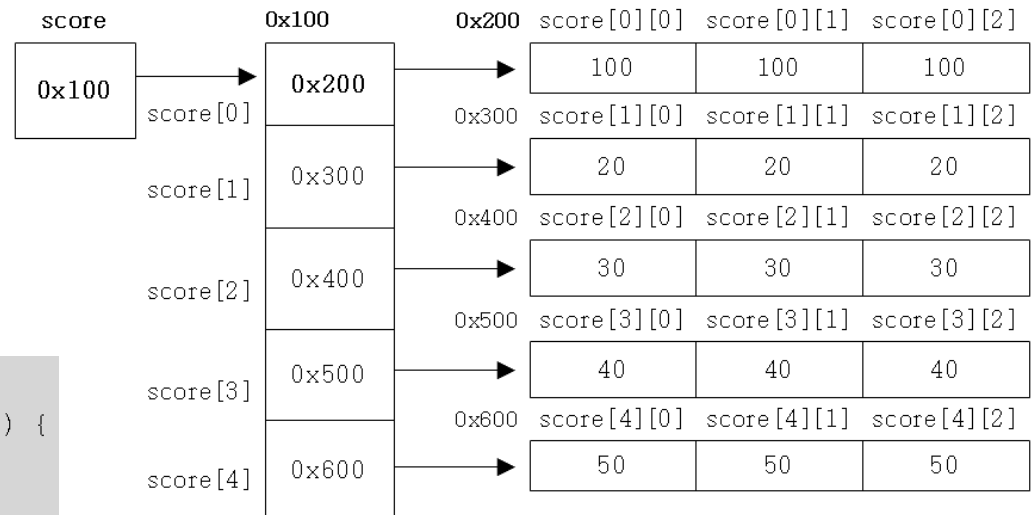
선언방법	선언예
타입 [ ] [ ] 변수이름;	int [ ] [ ] score;
타입 변수이름 [ ] [ ];	int score [ ] [ ];
타입 [ ] 변수이름 [ ];	int [ ] score [ ];

```
int [ ] [ ] score = {  
    { 100, 100, 100 },  
    { 20, 20, 20 },  
    { 30, 30, 30 },  
    { 40, 40, 40 },  
    { 50, 50, 50 },  
};
```

```
int [ ] [ ] score = new int [ 5 ] [ 3 ];    // 5행 3열의 2차원 배열을 생성한다.
```

	국어	영어	수학
1	100	100	100
2	20	20	20
3	30	30	30
4	40	40	40
5	50	50	50

```
for (int i=0; i < score.length; i++) {  
    for (int j=0; j < score[i].length; j++) {  
        score[i][j] = 10;  
    }  
}
```



## 2차원 배열

```
public class TwoDemensionArray {  
    public static void main(String[] args) {  
        int[][] array1 = { {1,2,3}, {4,5,6}, {7,8,9}, {10,11,12} };  
        int[][] array2 = new int[2][3];  
        array2[0][0]=1;  
        array2[0][1]=2;  
        array2[0][2]=3;  
        array2[1][0]=4;  
        array2[1][1]=5;  
        array2[1][2]=6;  
  
        System.out.println(array1[0][0] + " " + array1[0][1] + " " + array1[0][2]); // 0행 출력  
        System.out.println(array1[1][0] + " " + array1[1][1] + " " + array1[1][2]); // 1행 출력  
        System.out.println(array1[2][0] + " " + array1[2][1] + " " + array1[2][2]); // 2행 출력  
        System.out.println(array1[3][0] + " " + array1[3][1] + " " + array1[3][2]); // 3행 출력  
        System.out.println("-----");  
        System.out.println(array2[0][0] + " " + array2[0][1] + " " + array2[0][2]); // 0행 출력  
        System.out.println(array2[1][0] + " " + array2[1][1] + " " + array2[1][2]); // 1행 출력  
    }  
}
```



# 2차원 배열 사용하기

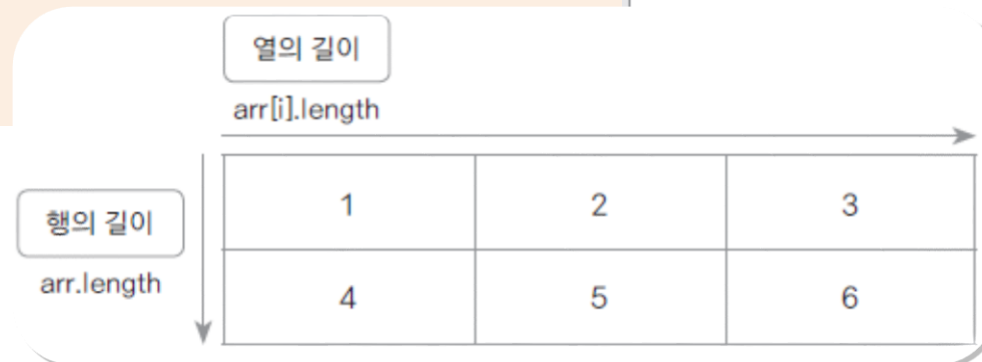
- ✓ 전체 모든 요소를 출력하려면 중첩된 for 문을 사용해야 함

```
public class TwoDimension {  
    public static void main(String[ ] args) {  
        int[ ][ ] arr = {{1, 2, 3}, {4, 5, 6}};  
  
        for(int i = 0; i < arr.length; i++) {  
            for(int j = 0; j < arr[i].length; j++) {  
                System.out.println(arr[i][j]);  
            }  
            System.out.println( ); //행 출력 끝난 후 한 줄 띄움  
        }  
    }  
}
```

이차원 배열 선언과 동시에 초기화



```
Problems  Javadoc  Declaration  Console  ?  
<terminated> TowDimension [Java Application] C:\Program Files  
1  
2  
3  
4  
5  
6
```



# Quiz 2차원 배열

〈TwoDemensionArrayExam1.java〉 6개의 값을 저장하는 2차원 배열은 문자 (char)형으로 초기화 시키고, 각 요소들을 모두 출력하시오. 6개의 값들을 출력하는 문장은 for문으로 작성하시오.

[변수 선언]

6개의 값을 저장하는 문자형 배열 변수 선언;

[출력 결과]

ABC

DEF

# Quiz 2차원 배열

<TwoDemensionArrayExam2.java> 5개 값을 저장하는 2차원 배열에 문자열 (String)로 초기화 시키고 각각의 값들을 모두 출력하시오. 5개의 값들을 출력하는 문장은 for문으로 작성하시오.

[변수 선언]

```
String[][] array = { {"BASIC"}, {"C"}, {"C++"}, {"C#"}, {"JAVA"} };
```

[출력 결과]

BASIC

C

C++

C#

JAVA

# Quiz 2차원 배열

<TwoDemensionArrayExam3.java> 2차원 배열을 사용하여 구구단을 아래와 같이 출력하시오. 출력은 이중 for문을 사용하시오.

[변수 선언]

```
int[][] array = new int[8][9];
```

[출력 결과]

```
2단 : 02 04 06 08 10 12 14 16 18
3단 : 03 06 09 12 15 18 21 24 27
4단 : 04 08 12 16 20 24 28 32 36
5단 : 05 10 15 20 25 30 35 40 45
6단 : 06 12 18 24 30 36 42 48 54
7단 : 07 14 21 28 35 42 49 56 63
8단 : 08 16 24 32 40 48 56 64 72
9단 : 09 18 27 36 45 54 63 72 81
```

# Quiz 2차원 배열

<TwoDemensionArrayExam4.java> 행의 길이를 입력 받은 후, 그 숫자를 사용한 이차원 배열을 선언하고 반복문을 이용해 배열 요소들을 아래 실행결과와 같이 출력 하세요.

예)

3 입력 → 3행 3열의 이차원 배열을 선언하여 배열 값 출력

5 입력 → 5행 5열의 이차원 배열을 선언하여 배열 값 출력

[실행 결과]

행의 길이 입력 : 4

001	002	003	004
005	006	007	008
009	010	011	012
013	014	015	016

# Quiz 2차원 배열

<ForEachTest.java> 다음 이차원 배열을 Enhanced for 구문으로 실행 결과와 같이 출력해 보세요.

```
int[][] sales = { { 110, 100, 110, 120 },    // 지지난해  
                  { 120, 110, 100, 120 },    // 지난해  
                  { 130, 140, 120, 150 }     // 올해  
                };
```

[실행 결과]

지난 3년간 분기별 매출은 아래와 같고,

110, 100, 110, 120,

120, 110, 100, 120,

130, 140, 120, 150,

지난 3년간 매출 총액은 1430이며,

연평균 476.67입니다.

# 배열의 복사

- for문을 이용한 배열의 복사

```
int[] number = {1,2,3,4,5};  
int[] newNumber = new int[10];  
  
for(int i=0; i<number.length;i++) {  
    newNumber[i] = number[i]; // 배열 number의 값을 newNumber에 저장한다.  
}
```

number

1	2	3	4	5
---	---	---	---	---

newNumber

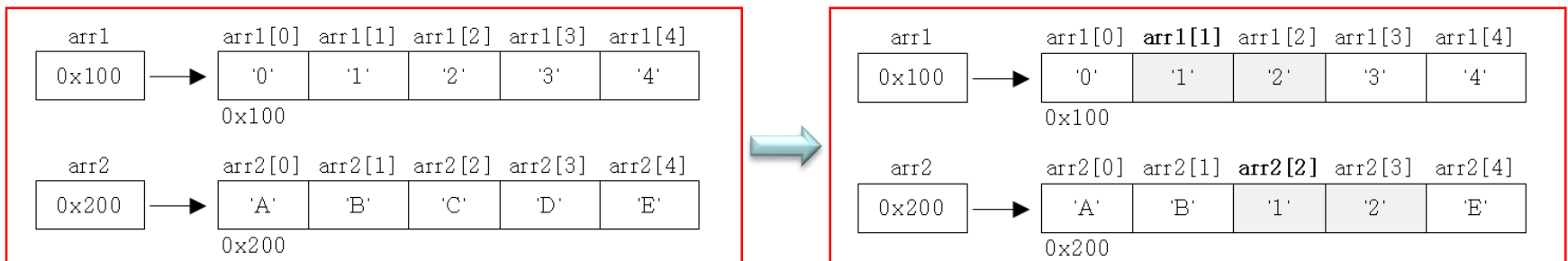
0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

- System.arraycopy()를 이용한 배열의 복사

```
System.arraycopy(arr1, 0, arr2, 0, arr1.length);
```

arr1[0]에서 arr2[0]으로 arr1.length개의 데이터를 복사

**System.arraycopy(arr1, 1, arr2, 2, 2);**



# 배열 복사하기

- 기존 배열과 같은 배열을 만들거나 배열이 꽉 찬 경우 더 큰 배열을 만들고 기존 배열 자료를 복사할 수 있음
- `System.arraycopy(src, srcPos, dest, destPos, length);`

매개변수	설명
src	복사할 배열 이름
srcPos	복사할 배열의 첫 번째 위치
dest	복사해서 붙여 넣을 대상 배열 이름
destPos	복사해서 대상 배열에 붙여 넣기를 시작할 첫 번째 위치
length	src에서 dest로 자료를 복사할 요소 개수



# 배열의 복사

```
public class ArrayCopyTest {  
    public static void main(String[] args) {  
        int[] num = { 1,2,3,4,5 };  
        int[] newNum = new int[10];  
  
        for(int i=0; i<num.length; i++) {  
            newNum[i] = num[i]; // 방법1: (대입 연산자 사용)  
            System.out.println("num["+i+"]의 값 : " + num[i]);  
            System.out.println("newNum["+i+"]의 값 : " + newNum[i]);  
        }  
  
        int[] arr1 = { 6,7,8,9,10 };  
        int[] arr2 = new int[5];  
  
        System.arraycopy(arr1, 0, arr2, 0, arr1.length); // 방법2: (System.arraycopy 사용)  
        for(int i=0; i<arr1.length; i++) {  
            System.out.println("arr1[" + i + "] : " + arr1[i]  
                               + " , arr2[" + i + "] : " + arr2[i] );  
        }  
    }  
}
```

# Quiz 배열의 복사

<ArrayCopyExam1.java> 문자형 배열 2개를 각각의 배열 객체를 사용하여 아래와 같이 출력하세요. 출력문2, 출력문3에서는 System.arraycopy() 메서드를 사용하여 복사한 뒤 출력하세요.

[변수 선언 및 초기화]

```
char[ ] abc = { 'A', 'B', 'C', 'D' };
```

```
char[ ] number = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };
```

[출력 결과]

===출력문1===

ABCD

0123456789

===출력문2===

ABCD456789

===출력문3===

ABCD45ABC9

# Quiz 배열의 복사

<ArrayCopyExam2.java> System.arraycopy() 메서드를 사용하여 5개의 정수 값을 갖는 배열 2개를 합친 후, 모든 배열의 값들을 출력하세요.

[변수 선언과 초기화]

```
int[] arr1 = { 1 , 2 , 3 , 4 , 5 };  
int[] arr2 = { 6 , 7 , 8 , 9 , 10 };
```

[출력 결과]

1

2

3

4

5

6

7

8

9

10

# 사용자 입력 받기 - 커맨드 라인

- 커맨드 라인에서 입력된 값들은 String 배열에 담겨 main() 메서드에 전달된다.

```
public static void main(String[] args)
{
    System.out.println("매개변수의 개수:"+args.length);
    for(int i=0;i< args.length;i++) {
        System.out.println("args[" + i + "] = \""+ args[i] + "\"");
    }
}
```

## [실행 결과]

```
매개변수의 개수:3
args[0] = "abc"
args[1] = "123"
args[2] = "Hello world"
```

# 사용자 입력 받기 - 커맨드 라인

- 커맨드 라인에서 입력된 값들은 String 배열에 담겨 main() 메서드에 전달된다.

<ArrayArgsTest.java>

// Run → Run Configurations → Arguments → 인수 값 입력 → Run

```
public static void main(String[] args) {  
    System.out.println("매개변수의 개수 : " + args.length);  
    for(int i=0 ; i<args.length ; i++)  
    {  
        System.out.println("args["+ i+ "] = ₩" + args[i] + "₩");  
    }  
}
```