

문자열

문장을 뜻하고 기본적으로 제공하는 클래스 중 하나

데이터 타입의 변수는 큰 따옴표로 나타낸다

문자열을 나타내는 자료형 string

문자열을 기교

문자열 메소드(equals)

양 쪽에 있는 내용을 비교한 값을 Boolean type으로 반환

```
a.equals(b)
```

값을 비교해서 같으면 true 다르면 false리턴

```
String a = "Java";
```

```
String b = new String("java");
```

```
System.out.println(a==b);
```

```
System.out.println(a.equals(b));
```

String , new string비교

문자열 메소드(equalsIgnoreCase)

대소문자를 무시하고 비교

```
a.equalsIgnoreCase(b)
```

문자열 메소드(compareTo)

두개의 값을 int(정수)값으로 변환

숫자의 비교 같은 경우는 단순히 크다(1) 같다(0), 작다(-1)의 관한 결과값을 리턴해 주는 반면

문자열의 비교 같은 경우는 같다(0), 그 외 양수/음수값 같이 결과를 반환해준다

문자열 메소드(compare TolgnoreCase)

대소문자를 무시하고 비교

문자열 메소드(charA+)

문자열에서 하나의 문자를 Index로 지정하고, 기정한 값을 문자형(char)으로 리턴

[charA+(index->인덱스 번호가 들어감)]

문자열 메소드(to____Case)

모두 대문자 or 소문자로 변환

[toUpperCase():대문자, toLowerCase():소문자]

문자열 메소드(substin)

문자열의 일부 변환

[substring(index시작,index끝)]

문자열 메소드(trim/contains)

앞뒤 공백 제거 후 반환=>[trim()]

문자열 포함 유무 조사=>[contains()]

문자열 메소드(startWith)

시작하는 문자열, 끝나는 문자열이 맞는지 조사

[startsWith():첫 문자, endsWith():끝 문자]

문자열 메소드(isEmpty)

문자열의 길이가 0이면 true 반환

```
String str1 = "hi";
```

```
String str2 = " Java";
```

```
String str3 ="Hi, Java";
```

```
String str4 ="Java";
```

```
System.out.println(str4.isEmpty()); // false
```

```
Str4="";
```

```
System.out.println(str4.isEmpty()); // true
```

배열

타입이 같은 데이터를 다룰 때 모든 데이터의 변수를 지정한다면 개수가 너무 많아지는 점을 배열을 통해 동일한 데이터 타입으로 집합을 쉽게 처리할 수 있음

각각 변수로 하기 싫어서 변수로 애용함

배열(선언과 생성)

[데이터타입] 변수or배열이름

```
Int[] arr1=new int[5]
```

```
Int arr2[]={10,100,90,50,40};
```

```
// 배열 요소에 접근하는 방법 1
```

```
int a = arr2[2]
```

```
system.out.printl(a);
```

//배열 요소에 접근하는 방법 2

```
for(int i=0; i<arr1.length; i++){  
    arr1[i] = sc.nextInt();  
    system.out.println(str1[i] + "");  
}
```

1차원 배열

조건식이 true일 때만 실행문 수행할 때 사용한다

```
Int[] arr1=new int[5]
```

```
Int arr2[]={10,100,90,50,40};
```

2차원배열

```
Int[][] str3= new int[3][5](3행 5열)
```

과제

백준 2845, 10039

String b = "JAVA" 와 String c = new String("JAVA")의 차이점

다차원 배열

Ex)

```
int [][][] = MultiArr = new int[2][5][2];
```

```
int [][][] MultiArr={  
    {{1,2,3,4,5},{6,7,8,9,10}},  
    {{11,12,13,14,15},{16,17,18,19,20}}
```

```
};
```

```
system.out.println(MultiArr[0][0][0]); // 1
```

```
system.out.println(MultiArr[0][1][1]); // 7
```

```
system.out.println(MultiArr[1][1][0]); //16
```