

## 1. 문자열

```
package week6_1;
import java.util.Scanner;
public class ex {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        String s1 = "a";
        String s2 = "a";
        String s3 = new String("a");
        String s4 = new String("a");

        System.out.println(s1==s2);
        System.out.println(s1.equals(s3));    true
        System.out.println(s3==s4);          true
                                              false
    }
}
```

- ➔ s1과 s2는 주소값이 같지만 s3과 s4는 새로운 String 으로 만들었기 때문에 주소값이 다름
- ➔ `String s1 = "a";` 에서 s1을 a로 선언하고 정의, 같은 방식으로 s2를 a로 선언하고 정의했기 때문에 주소값이 같다 , `System.out.println(s1==s2);` 는 True
- ➔ `String s3 = new String("a");` 에서 s3를 새로운 주소값 a로 선언 후 정의 같은 방식으로 s4를 a로 선언 후 정의 하지만 `System.out.println(s1.equals(s3));` 에서 `.equals()` 의 경우 값만 같으면 True이다.
- ➔ `System.out.println(s3==s4);` 는 False, s3과 s4의 값이 새로운 String으로 정의되었기 때문에 주소값이 다르다, 때문에 False , 만약 `.equals()` 를 사용했다면 True이다.

```

package week6_1;

public class ex2 {

    public static void main(String[] args) {
        String s1 = "abc";
        String s2 = s1.toUpperCase();
        System.out.println(s2);
    }
}

```

- ➔ S1이 abc로 선언 후 정의되었다.
- ➔ S2는 s1에 `toUpperCase();` 을 사용해 선언 후 정의했다.
- ➔ S2를 출력하면 `toUpperCase();` 로 인해 ABC로 출력되었음을 알 수 있다.

## 2. 배열

```

Scanner sc = new Scanner(System.in);
// 배열 크기는 3으로 지정함
//1
int [] num = {10,11,12};
//2
int [] num1 = new int [] {10,11,12};
//3
int [] num3;
num3 = new int [] {10,11,12};

```

- ➔ 배열은 자료형 타입 바로 옆에 `[]` 기호를 사용해 표현한다. 즉 배열이랑 자료형의 종류가 아닌 자료형의 집합을 의미한다.
- ➔ 배열을 표현하는 방법엔 3가지가 있다.
  1. 데이터타입 `[]` 배열이름 = {집합};
  2. 데이터타입 `[]` = new int [배열의 크기] {집합};
  3. 데이터 타입 `[]` 배열이름 ;
 

배열이름 = new int [배열의 크기] {집합};

```
//데이터 타입 [] 배열 이름 = new int [배열의 크기];
int n = sc.nextInt();
int [] arr = new int [n];

//배열에서 인덱스      0  1  2  3  4
int [] abc = new int [] {1, 2, 3, 4, 5};
System.out.println(abc[0]);
System.out.println(abc[1]);
System.out.println(abc[4]);
```

→ 배열에서의 인덱스 번호는 순서대로 0부터 시작한다.

```
System.out.println(abc[0]); 1
System.out.println(abc[1]); 2
System.out.println(abc[4]); 5
```

→ 보여지는 것과 같이 0번 인덱스는 1 , 1번 인덱스는 2 , 4번 인덱스는 5이다.

```
Scanner sc = new Scanner(System.in);
int sum = 0;

int [] num = new int [3];

for(int i = 0; i < num.length; i++) {
    num[i] = sc.nextInt();
}

for(int i = 0; i < num.length; i++) {
    sum += num[i];
}

System.out.println(sum/num.length);
```

→ `int [] num = new int [3];` 에서 크기가 3인 배열을 만들었다.

→ `for(int i = 0; i < num.length; i++)`  
`sum += num[i];` 에서 num의 크기를 length를 통해 값을 확인하여 for문을 통해 반복 입력 받는다. 그리고 sum에 입력받은 값을 저장한다.

→ `System.out.println(sum/num.length);` 에서 똑같이 length를 통해 num의 값을 확인한 뒤 sum의 값을 나누어 평균을 계산해 출력한다.