

StringTokenizer 클래스

StringTokenizer 클래스

❖ 문자열 분리 방법

- String의 split() 메소드 이용
- java.util.StringTokenizer 클래스 이용

❖ String의 split()

- 정규표현식을 구분자로 해서 부분 문자열 분리
- 배열에 저장하고 리턴

홍길동&이수홍,박연수,김자바-최명호

```
String[] names = text.split("&|,|-");
```

StringTokenizer 클래스

❖ StringSplitExample.java

```
public class StringSplitExample {  
    public static void main(String[] args) {  
        String text = "홍길동&이수홍,,박연수,김자바--최명호";  
        String[] names = text.split("&|,|-");  
  
        for (String name : names) {  
            System.out.println(name);  
        }  
    }  
}
```

StringTokenizer 클래스

❖ StringTokenizer 클래스

- Iterator 패턴으로 분리된 문자열 처리

```
String text = "홍길동/이수홍/박연수";  
StringTokenizer st = new StringTokenizer(text, "/");
```

```
while( st.hasMoreTokens() ) {  
    String token = st.nextToken();  
    System.out.println(token);  
}
```

메소드		설명
int	countTokens()	꺼내지 않고 남아있는 토큰의 수
boolean	hasMoreTokens()	남아 있는 토큰이 있는지 여부
String	nextToken()	토큰을 하나씩 꺼내옴

StringTokenizer 클래스

❖ StringTokenizerExample.java

```
import java.util.StringTokenizer;

public class StringTokenizerExample {
    public static void main(String[] args) {
        String text = "홍길동/이수홍/박연수";

        // how1: 전체 토큰 수를 얻어 for문으로 루핑
        StringTokenizer st = new StringTokenizer(text, "/");
        int countTokens = st.countTokens();
        for (int i = 0; i < countTokens; i++) {
            String token = st.nextToken();
            System.out.println(token);
        }
        System.out.println();

        // how2: 남아 있는 토큰을 확인하고 while문으로 루핑
        st = new StringTokenizer(text, "/");
        while (st.hasMoreTokens()) {
            String token = st.nextToken();
            System.out.println(token);
        }
    }
}
```