

Java: String.split() ??? ?? ??

? ?? ??

```
String[] result = obj.split("regex");
```

split() 方法 接受 (String regex) 参数，返回包含子字符串的数组。默认情况下，正则表达式是 `\\s*`，即任何空白字符。

1. 字符串分割

1.1 基本用法

```
String s = "hello world";  
String[] arr = s.split(" ");
```

1.2 使用正则表达式 (1.1 的进阶)

```
String s = "hello world java";  
String[] arr = s.split("\\s+"); // \\s = 任意空白字符, + 1 或多个
```

1.3 使用正则表达式 (1.2 的进阶)

```
String s = "a\tb c";  
String[] arr = s.split("[ \\t]+");
```

2. 使用逗号分隔符

```
String s = "apple,banana,grape";
String[] arr = s.split(",");
```

2.1 ?? + ?? ?? (CSV ??)

```
String s = "apple, banana, grape";
String[] arr = s.split(",\\s*"); // ,   
```

3. [] [] [] []

“ [] [] [] [] () [] !

[]	[] [] [] [] [] []	split [] [] []
.	[] []	"\\."
,	,	OR ([])
*	[] []	"*"
() []	[] []	"\\(", "\\]"

3.1 ???(.) ??

```
String s = "www.example.com";
String[] arr = s.split("\\.");
```

3.2 ???(|) ??

```
String s = "red|green|blue";
String[] arr = s.split("\\|");
```

4. □□/□□ □□ □□

4.1 ??? ????? ???

```
String s = "abc123def456ghi";  
String[] arr = s.split("\\d+"); // □□(1□ □□) □□ □□
```

4.2 ?? ?? ???

```
String s = "123abc456def";  
String[] arr = s.split("[a-zA-Z]+"); // □□□ □□ □□
```

5. □□ □□□ □□

5.1 ??(,) ?? ?????(;) ?? ??

```
String s = "apple,banana;grape orange";  
String[] arr = s.split("[,; ]");
```

5.2 AND/OR ?? ??? ??

```
String s = "red and blue or green";  
String[] arr = s.split("\\s+(and|or)\\s+"); // □□ □□□ and/or □□
```

6. □□ □ □□ □□ (□□□)

```
String s = "Hi. I am John! Nice to meet you?";  
String[] arr = s.split("[.!?]\\s*"); // . ! ? □ □□□□ □□
```

7. 문자열 분할

```
String s = "a,,b,c";  
String[] arr = s.split(",", -1); // -1 공백을 1개 이상으로  
System.out.println(Arrays.toString(arr));  
// 출력: [a, , b, c]
```

정규 표현식: 정규 표현식

표현식	의미	예시
<code>\\s</code>	공백 문자 (공백, 탭, 줄바꿈)	<code>"\\s+"</code> → 공백 문자 하나 이상
<code>\\d</code>	숫자 (0~9)	<code>"\\d+"</code> → 숫자 1개 이상
<code>\\w</code>	문자, 밑줄, 숫자	<code>"\\w+"</code>
<code>.</code>	어떤 문자든	<code>"a.b"</code> → "acb", "a1b" 등
<code>[abc]</code>	a, b, c 중 하나	<code>"a[bc]d"</code> → "abd", "acd"
<code>[^abc]</code>	a, b, c가 아닌 문자	<code>"[^0-9]"</code> → 숫자가 아닌 문자
<code>+</code>	1개 이상 반복	<code>"\\s+"</code>
<code>*</code>	0개 이상 반복	<code>"a*"</code>
<code>`</code>	원시 문자	OR

정규 표현식

- `split()`은 공백을 기준으로 문자열을 분할한다. `split()`은 공백을 기준으로 문자열을 분할한다.
- `StringTokenizer`, `split()`은 공백을 기준으로 문자열을 분할한다.
- `split()`은 공백을 기준으로 문자열을 분할한다. `split()`은 공백을 기준으로 문자열을 분할한다.

