

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

? ?? ??

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

`split()` 方法 接受 `(String regex)` 参数，该参数指定用于分割字符串的 `正则表达式`。

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? ??)

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

1.3 使用转义字符

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

2. 使用 `split()` 方法

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

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

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

3. 分隔符 转义 转义 转义

“ 分隔符 转义 转义 转义(\\) 转义 ! ”

分隔符	转义 转义 转义	split 转义 转义
.	转义 转义	"\\."
,	,	OR (转义)
*	转义	"*"
() []	转义 转义	"\\(", "\\)", "\\[", "\\]"

3.1 ???(.) ??

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

3.2 ???(|) ??

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

4. 数字/字母 数字 数字

4.1 数字 数字?? 数字??

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

4.2 数字?? 数字?? 数字??

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

5. 数字 字母 数字

5.1 数字(,) 数字 数字??(;) 数字??

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

5.2 AND/OR 数字?? 数字?? 数字??

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

6. 数字 数字 数字 数字 (数字??)

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

7. 문자열 배열

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

? ????: ?????? ????

문자열	의미	정규식
<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 하나 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 또는 b 또는 c	<code>"a[bc]d"</code> → "abd", "acd"
<code>[^abc]</code>	a, b, c가 아닌 문자	<code>"[^0-9]"</code> → 0~9가 아닌 문자
<code>+</code>	1개 이상 반복	<code>"\\s+"</code>
<code>*</code>	0개 이상 반복	<code>"a*"</code>
<code>,</code>	OR	OR

? ??

- `split()`은 공백을 기준으로 문자열을 배열로 나눕니다.
- 문자열을 `StringTokenizer`, `split()` 또는 `split()`으로 나눌 수 있습니다.
- 정규식을 사용하여 문자열을 나눌 수 있습니다(예: `split("\\s+").length`).

