# 字符串

Java Reference

## String methods

一个 String 字符串是文本。常用方法：.length()、.charAt(i)、.substring(a, b)、.indexOf(x)、.toUpperCase()、.equals(...)。字符串是不可变不可变的：其内容无法更改。方法可能返回数字、布尔值或字符串；返回的 String 对象可能会重用现有对象。

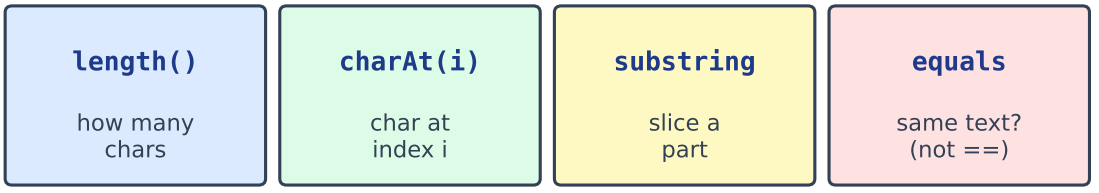
```
public class Main {
    public static void main(String[] args) {
        String s = "Python";
        System.out.println(s.length());          // 6
        System.out.println(s.charAt(0));         // P
        System.out.println(s.substring(0, 3));   // Pyt
        System.out.println(s.toUpperCase());     // PYTHON
    }
}
```

天天都会用到的方法:

| 方法 | 含义 | 示例 → 结果 |
|---|---|---|
| .length() | 有多少个字符 | "Hi".length() → 2 |
| .charAt(i) | 取一个字符 | "Hi".charAt(0) → H |
| .substring(a, b) | 取一段, 到 b 之前为止 | "Python".substring(0, 3) → Pyt |
| .substring(a) | 从 a 到末尾 | "Python".substring(3) → hon |
| .indexOf(x) | 第一次出现的位置,没有则 -1 | "banana".indexOf("na") → 2 |
| .equals(s) | 文本相同? | "hi".equals("hi") → true |
| .compareTo(s) | 顺序:负数 / 0 / 正数 | "apple".compareTo("banana") → 负数 |

`compareTo` 按字典序比较 UTF-16 代码单元，区分大小写（例如，`"Z"` 排在`"a"` 之前）—AP 考试用它来排序题目：

```
public class Main {
    public static void main(String[] args) {
        String a = "apple", b = "banana";
        System.out.println(a.compareTo(b) < 0);     // true (apple comes
↪   first)
        System.out.println(a.compareTo("apple"));  // 0 (equal)
    }
}
```




Exam tip: Strings are objects — use equals for text, not ==; indices start at 0

*Key String methods: length, charAt, substring, equals*

## Building & traversing strings

用 + 连接字符串 (拼接,concatenation)。用循环和 `.charAt(i)` 访问每个字符。

```
public class Main {
    public static void main(String[] args) {
        String word = "banana";
        int count = 0;
        for (int i = 0; i < word.length(); i++) {
            if (word.charAt(i) == 'a') count++;
        }
        System.out.println(count);   // 3
    }
}
```

在循环里构建长字符串时,StringBuilder 快得多: 先逐段 append, 最后调用一次 .toString()。

```
public class Main {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder();
        for (int i = 1; i <= 5; i++) {
            sb.append(i).append(" ");
        }
        System.out.println(sb.toString().trim());   // 1 2 3 4 5
    }
}
```

### 常见错误

- 字符串是不可变的: s.toUpperCase() 返回大写结果而不改变 s, 因此请存储该结果。
- 取字符用 s.charAt(i); 长度是 s.length()(方法, 带 ())。
- 在大循环里用 += 拼字符串很慢; 要用 StringBuilder。