

Lambda表达式只是一个“语法糖”，由编译器推断并包装成常规的代码，可以使用更少的代码来实现同样的功能。

基本的Lambda例子

现在,我们已经知道什么是lambda表达式,让我们先从一些基本的例子开始。在本节中,我们将看到lambda表达式如何影响我们编码的方式。假设有一个玩家List,程序员可以使用 for 语句 ("for 循环")来遍历,在Java SE 8中可以转换为另一种形式:

```
String[] atp = {"Rafael Nadal", "Novak Djokovic",
    "Stanislas Wawrinka",
    "David Ferrer", "Roger Federer",
    "Andy Murray", "Tomas Berdych",
    "Juan Martin Del Potro"};
List<String> players = Arrays.asList(atp);

// 以前的循环方式
for (String player : players) {
    System.out.print(player + "; ");
}

// 使用 lambda 表达式以及函数操作(functional operation)
players.forEach((player) -> System.out.print(player + "; "));

// 在 Java 8 中使用双冒号操作符(double colon operator)
players.forEach(System.out::println);
```

下面是使用lambdas 来实现 Runnable接口 的示例:

```
// 1.1使用匿名内部类
new Thread(new Runnable() {
    @Override
    public void run() {
        System.out.println("Hello world !");
    }
}).start();

// 1.2使用 lambda expression
new Thread(() -> System.out.println("Hello world !")).start();

// 2.1使用匿名内部类
Runnable race1 = new Runnable() {
    @Override
    public void run() {
        System.out.println("Hello world !");
    }
};

// 2.2使用 lambda expression
Runnable race2 = () -> System.out.println("Hello world !");

// 直接调用 run 方法(没开新线程哦!)
race1.run();
race2.run();
```