



链滴

# 四大函数式接口 - 重点必须掌握 (Juc-07)

作者: [yscxy](#)

原文链接: <https://ld246.com/article/1606116556180>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 概述

新时代的程序员必须要掌握的：lambda表达式、链式编程、函数式接口、Stream流式计算

## Function函数式接口

```
package net.yscopy.function;

import java.util.function.Function;

/**
 * @Author WangFuKun
 * @create 2020/11/23 14:53
 */
/*
 *Function 函数式接口，有一个输入参数，有一个输出
 * 只要是函数式接口都可以用Lambda表达式简化
 */
public class Demo01 {
    public static void main(String[] args) {
        Function<String, String> function = new Function<String, String>() {
            @Override
            public String apply(String s) {
                return s;
            }
        };
        //Lambda表达式简化
        Function<String, String> function1 = (str) -> {
            return str;
        };
        System.out.println(function1.apply("abc"));
    }
}
```

也可以更简单这样写

```
Function<String, String> function1 = str -> {
    return str;
};
```

## Predicate

有一个参数返回值只能是布尔值

```
package net.yscopy.function;

import java.util.function.Predicate;

/**
 * @Author WangFuKun
 * @create 2020/11/23 15:02
 */
```

```

/*
 * 断定式接口，也就是有一个输入参数，返回值只能是布尔值
 */
public class Demo02 {
    public static void main(String[] args) {
        // Predicate<String> predicate = new Predicate<String>() {
        //     @Override
        //     public boolean test(String str) {
        //         return str.isEmpty();
        //     }
        // };
        // System.out.println(predicate.test("d"));
        Predicate<String> predicate = str -> {
            return str.isEmpty();
        };
    }
}

```

## Consumer消费型接口

消费型接口，只有输入，没有返回值

```

package net.yscopy.function;

import java.util.function.Consumer;

/**
 * @Author WangFuKun
 * @create 2020/11/23 15:14
 */
/*
 * 消费型接口，只有输入，没有返回值
 */
public class Demo03 {
    public static void main(String[] args) {
        // Consumer<String> consumer = new Consumer<String>() {
        //     @Override
        //     public void accept(String str) {
        //         System.out.println("消费了->" + str);
        //     }
        // };
        // consumer.accept("strsssssss");
        Consumer<String> consumer = str->{
            System.out.println(str);
        };
        consumer.accept("test");
    }
}

```

## Supplier供给型接口

供给型接口，没有参数，只有返回值

```
package net.yscopy.function;

import java.util.function.Supplier;

/**
 * @Author WangFuKun
 * @create 2020/11/23 15:21
 */
/*
 * 供给类型接口，没有参数，只有返回值
 */
public class Demo04 {
    public static void main(String[] args) {
        // Supplier<String> stringSupplier = new Supplier<String>() {
        //     @Override
        //     public String get() {
        //         return "abs";
        //     }
        // };
        // System.out.println(stringSupplier.get());
        Supplier<String> stringSupplier = () -> {
            return "abs";
        };
        System.out.println(stringSupplier.get());
    }
}
```