



链滴

003 选择

作者: [pzs233](#)

原文链接: <https://ld246.com/article/1537246277709>

来源网站: [链滴](#)

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

3.2 boolean 数据类型

boolean 数据类型声明一个具有值 true 或者 false 的变量。

关系操作符：

- <
- <=
- >
- >=
- ==
- !=

3.3 if 语句

语法：

```
if (布尔表达式){  
    语句 (组) ;  
}
```

3.4 if-else 语句

语法：

```
if (布尔表达式){  
    为真时执行的语句 (组) ;  
} else {  
    为假时执行的语句 (组) ;  
}
```

3.5 嵌套的 if 语句和多分支 if-else 语句

3.6 常见错误和陷阱

在同一个块中，else 总是和离它最近的 if 子句匹配。

两浮点值只能比较它们是否足够接近，通过测试两个数的差距小于某个阈值，通常 float 类型的值用 10^{-7} 当阈值，double 类型的值用 10^{-14} 。

eg:

```
final double EPSILON = 1E-14;  
double x = 1.0 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1;  
if (Math.abs(x - 0.5) < EPSILON){  
    System.out.println(x + " is approximately 0.5");  
}
```

3.7 产生随机数

`Math.random()`，调用这个方法会返回一个双精度的随机值 d ，且满足 $0.0 \leq d < 1.0$ 。

通过组合就能产生很多类型的随机数。

eg: 产生 0 ~ 9 之间的随机整数 `(int)(Math.random() * 10)`

`System.exit(status)` 是在 `System` 类中定义的，调用这个方法可以终止程序。参数 `status` 为 0 表明程序正常结束，非 0 表示非正常结束。

3.10 逻辑操作符

布尔操作符

操作符	名称	说明
!	非	逻辑非
&&	与	逻辑与
	或	逻辑或
^	异或	逻辑异或

`!(condition1 && condition2)` 和 `!(condition1 || condition2)` 是等价的；

`!(condition1 || condition2)` 和 `!(condition1 && condition2)` 是等价的。

3.13 switch 语句

语法：

```
switch (switch 表达式) {
    case value1 : 语句 (组) 1;
                break; // 可选, break 语句会立即终止 switch 语句
    case value2 : 语句 (组) 2;
                break;
    case value3 : 语句 (组) 3;
                break;
    ...
    case valueN: 语句 (组) ;
                break;
    default : 默认以上无匹配时执行的语句 (组) ; // 可选
}
```

3.14 条件表达式

语法：`boolean-expression ? expression1 : expression2;`

Java 中唯一的三元操作符。

3.15 操作符的优先级和结合规则

操作符的优先级:

1. `var++`、`var--`
2. `+`、`-`、`++var`、`--var`
3. `(type)` (类型转换符)
4. `!` (非)
5. `*`、`/`、`%`
6. `+`、`-`
7. `<`、`<=`、`>`、`>=`
8. `==`、`!=`
9. `^` (异或)
10. `&&`
11. `||`
12. `=`、`+=`、`-=`、`/=`、`%=`

关键术语

- Boolean expression
- Boolean data type
- conditional operator
- dangling-else ambiguity
- debugging
- fall-through behavior
- flowchart
- lazy operator
- operator associativity
- operator precedence
- selection statement
- short-circuit evaluation

End