



开始

基本运算符

在[上一章](#)中，我们看到 Elixir 提供了 `+`，`-`，`*`，`/`，`div/2`，`rem/2` 作为算术运算符，以及函数和整数除法和余数。

Elixir 还提供和操作列表：`++` `--`

```
iex> [1, 2, 3] ++ [4, 5, 6]
[1, 2, 3, 4, 5, 6]
iex> [1, 2, 3] -- [2]
[1, 3]
```

字符串连接是用：`<>`

```
iex> "foo" <> "bar"
"foobar"
```

Elixir 还提供了三个布尔运算符：`and`，`or`，`not`。这些运算符在某种意义上是严格的，因为它们期望计算结果为布尔值（或）的东西作为他们的第一个参数：`or` `and` `not` `true` `false`

```
iex> true and true
true
iex> false or is_atom(:example)
true
```

提供非布尔值将引发异常：

```
iex> 1 and true
** (BadBooleanError) expected a boolean on left-side of
```

新闻：[Elixir v1.15 发布](#)

搜索。。。

[接口文档](#)

开始

- [1. 介绍](#)
- [2. 基本类型](#)
- [3. 基本运算符](#)
- [4. 模式匹配](#)
- [5. 案例、cond 和 if](#)
- [6. 二进制文件、字符串和字符列表](#)
- [7. 关键字列表和地图](#)
- [8. 模块和功能](#)
- [9. 递归](#)
- [10. 枚举项和流](#)
- [11. 过程](#)
- [12. IO 和文件系统](#)
- [13. 别名、要求和导入](#)
- [14. 模块属性](#)
- [15. 结构体](#)
- [16. 协议](#)
- [17. 理解](#)
- [18. 印记](#)
- [19. 尝试、捕捉和救援](#)
- [20. 可选语法表](#)
- [21. Erlang 库](#)

```
"and", got: 1
```

`or` 并且是短路运营商。仅当左侧不足以确定结果时，它们才会执行右侧：`and`

```
iex> false and raise("This error will never be raised")
false
iex> true or raise("This error will never be raised")
true
```

除了这些布尔运算符，Elixir 还提供了，并且接受任何类型的参数。对于这些运算符，除 `and` 和 `or` 之外的所有值的计算结果均为

true: `||` `&&` `!` `false` `nil`

```
# or
iex> 1 || true
1
iex> false || 11
11

# and
iex> nil && 13
nil
iex> true && 17
17

# not
iex> !true
false
iex> !1
false
iex> !nil
true
```

根据经验，使用 `and` 和 `or` 当您需要布尔值时。如果任何参数是非布尔参数，请使用 `||` 和 `&&`。

Elixir 还提供，`==`，`!=`，`===`，`!==`，`<=`，`>=`，`<`，`>` 和作为比较运算符：

```
iex> 1 == 1
true
iex> 1 != 2
```

22. 调试

23. 类型规格和行为

24. 下一步去哪里

混合和一次性密码

1. 混音简介

2. 代理

3. GenServer

4. 主管和申请

5. 动态主管

6. 电子交易体系

7. 依赖项和伞形项目

8. 任务和 `gen_tcp`

9. 文档测试，模式和

10. 分布式任务和标签

11. 配置和发布

ELIXIR 中的元编程

1. 报价和取消报价

2. 宏

3. 域特定语言

```
true  
iex> 1 < 2  
true
```

和之间的区别在于后者在比较整数和浮点数时更严格： `==` `===`

```
iex> 1 == 1.0  
true  
iex> 1 === 1.0  
false
```

我们说这些运算符执行*结构比较*。有关更多信息，您可以阅读我们关于[结构与语义比较](#)的文档。

在下一章中，我们将讨论通过使用 `match` 运算符进行模式匹配。 `=`

[← 上一页](#) [返回页首](#) [下一→](#)

有什么不对吗？ [在 GitHub 上编辑此页面。](#)

© 2012–2023 长生不老药团队。

Elixir和Elixir标志是[The Elixir Team](#) 的注册商标。