



链滴

erlang 基本数据类型

作者: [lxer](#)

原文链接: <https://ld246.com/article/1614861187765>

来源网站: [链滴](#)

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

数值

- erlang数值分为 **整数**和**浮点数**。数值类型不用担心溢出问题，因为erlang内部采用了字符串加减。
- Erlang支持ASCII或者Unicode转换成整数值，同时支持整数不同进制的表示。
- 可以表示二进制，十六进制等

```
2#1010.  
> 10  
16#aa.  
17
```

原子类型

- 必须以小写字母开头，如果要大写、下划线或者其他的特殊字符开头，必须加上单引号
- 原子在Erlang里面是存在一张表上面的，原子的个数有上限，大概是在一百万个左右

```
normal.  
> normal  
'_a'.  
> 'a'
```

布尔

- true 和 false
- erlang 布尔类型并不能称作布尔类型，一般和布尔操作符一起使用

位字符串

- 在大多数情况下，二进制型里的位数都会是8的整数倍，因此对应一个字节串。如果位数不是8的整数倍，就称这段数据为位串（bitstring）。所以当我们说位串时，是在强调数据里的位数不是8的整数。位语法是一种表示法，用于从二进制数据里提取或加入单独的位或者位串

```
<<255,1>>  
> <<255,1>>  
<<256,1>>  
> <<0,1>>  
<<257, 1>>  
> <<1,1>>
```

二进制数据

- 一种数据类型，用来实现原始数据的高速存储。节省内存，输入输出更加高效。书写打印时，二进制数据以一个整数或者字符序列的形式出现，两端分别用尖括号括起来。其中的整数，每一个都要在0-255之间，如果二进制数据是可以打印的字符串，shell将显示字符串形式，否则会显示一串整数。
- binary是二进制类型，bitstring是位串类型；binary和bitstring都是二进制类型的数据

- 定义原型： 数值:大小/类型

```
<<11:16/signed-big-integer>>.
```

```
> <<0,225>>
```

```
%% 值为11,占16bit, 无符号整型, 使用大端字节序存储
```

元组

- 类似于Object, 不同的是, 它的长度是固定的。

```
A = {1, a, "b"}.
```

```
%% 取相应元素, 比如只取出1:
```

```
{T, _, _} = A.
```

```
%% T就是1
```

映射

- Record类似于Struct,Record表达式会在编译时转为tuple表达式

- 定义:

```
%% shell中的定义
```

```
rd(stu, {name, age})
```

```
%% 文件中定义
```

```
-record(stu, {name, age})
```

```
%% 使用
```

```
#stu{name = "lx",age = 23}
```

列表

- 列表可以是不同类型的元素
- 列表可以用++/--连接
- 第一个元素是列表的头
- 定义: [], [Head|Other]

```
List = [1, lx, "lx"].
```

```
%获取头元素
```

```
[Head|Other] = List.
```

```
% Head 为头部第一个元素, Other为剩余其他元素, 如果不想使用 可以用 _
```