

xlang 中的基本类型操作

文档库地址 <https://xlang.link/documents/index.html>

xlang 中有以下基本类型:

名称	说明	类别
void	无类型	无
byte	无符号 8 位整数(0~255)	数值, 简单类型
char	无符号 16 位整数(0~65535)	数值, 简单类型
short	有符号 16 位整数(-32768~+32767)	数值, 简单类型
int	有符号 32 位整数(-2,147,483,648~2,147,483,647)	数值, 简单类型
long	有符号 64 位整数 (-9,223,372,036,854,775,808~9,223,372,036,854,775,807)	数值, 简单类型
bool	布尔类型(true,false)	真假, 简单类型
String	字符串类(multibyte characters string)	字符串, 简单类型
[](Array)	数组对象(POD 类型的数组为连续的 POD 类型数组(string 除外),其他类型的数组为指针引用)	对象
Object	对象基础类型	对象

简单类型: 所有相关操作都是以值传递的方式进行。

对象: 所有相关操作都是以对象引用传递的方式进行。

例如:

简单类型:

```
String a = "123";
String b = a;
b = "456";
// 此时 b 的值为"456", 但 a 的值仍为"123"
```

对象类型:

```
int [] a = {1, 2, 3};
int [] b = a;
b[0] = 0;
// 此时 b[0] 的值为 0, a[0]的值也为 0
```

类别中包含数值的所有类型之间可以互相转换，但当类型表示范围不同时，编译器将给予警告，超出表示范围的数据将被截断，无法表示的精度将丢失。

字符串为特殊的简单类型，

类别中含有简单类型的没有 GC 开销，类别为对象的存在 GC 开销。

所有非简单类型都有一个虚方法：`finalize`。

原型：

```
void finalize(){  
  
}
```

该方法用于在对象被回收销毁时执行以通知用户做最后的清理工作。

注意：该方法仅被执行一次，若对象在 `finalize` 中被复活，则下次生命周期结束时将直接销毁，`finalize` 将不会被再次执行。

简单类型可以通过 `String` 重载的操作符 `+` 来进行字符串操作；

```
String s = "my age is " + 18;
```

s 则等于 "my age is 18";

通过 `String` 转到简单类型：

```
String age = "18";
```

```
int nage = age.parseInt(); // 从字符串解析整数，其他数值类型类似
```

```
bool is_18_years_old = age.equals("18"); //比较字符串是否相等  
//可以使用 equalsIgnoreCase 忽略大小写
```