上一次课，我们学习来基本的变量和常量的知识。今天，主要学习的是运算符号与可选类型。

1. 运算符

＋－＊／，＋＋，＋＝

/运算符

//+ - \* /

//单目运算符 一元运算符

var isMan:Bool=true

//取反

print(!isMan)

//＋＝，二目运算符，二元运算符

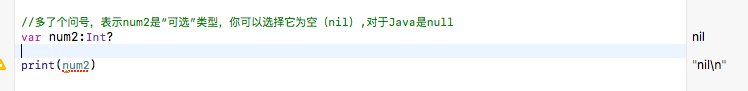
print(a+=a)// a=a+a

//三目运算符号，必须配合三个变量同时使用 (express)? value1:value2

var b:Int=(4<3) ? a : 0

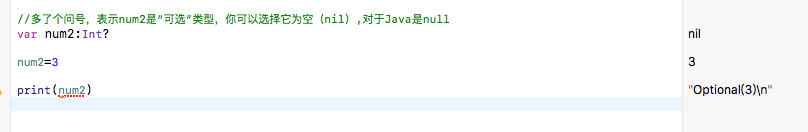
1. 可选类型（Swift）

C＃－>可空类型

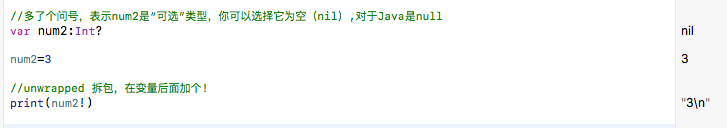


不赋值，它是nil

如果给num2赋值，则打印输出的就是Optional（3），提示还是一个可选类型，值是3.



通过拆包操作，让值不空，不可选

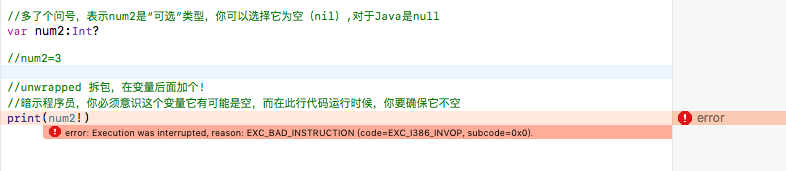


//unwrapped 拆包，在变量后面加个!

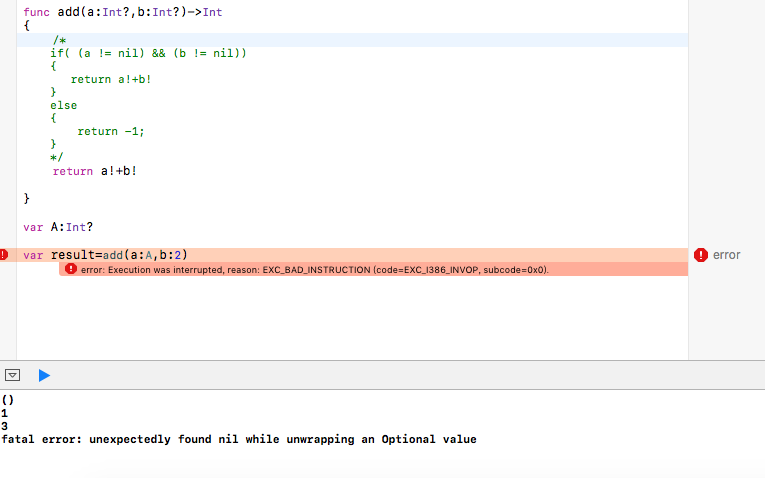
//暗示程序员，你必须意识这个变量它有可能是空，而在此行代码运行时候，你要确保它不空

print(num2!)

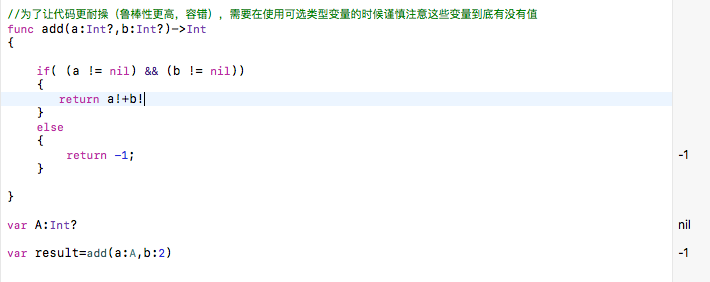
如果可选类型的变量，你不赋值就使用，会发生异常



在函数参数中使用可选类型的参数，如果外部没有给参数赋值，会直接引发异常



提高代码的鲁棒性（Robust），程序员需要考虑可选类型变量为空的情况。下面的代码，在使用可选类型变量时，需要先判断是否为nil，然后才计算。



1. 元组

//带有属性名的元组

var item=(name:"tom",age:12)

print(item.name+":"+String(item.age))

var item2=("jade",30)

//输出第一个属性

print(item2.0)

//函数可以返回元组，作用——一次带回多个返回值

func myfun()->(String,Int)

{

return (name:"pan",age:38)

}

let resutl2=myfun()