

文章  
[Jingwei Wang](#) · 八月 25, 2021 阅读大约需分钟

## Object Script 基础知识(二)

# Object Script(二)

- [特殊符号](#)

ObjectScript没有定义任何关键字：你可以自由地使用任何字作为标识符(如变量名)。为了达到这个目的，ObjectScript使用了一组内置的命令以及特殊的字符(比如函数名的"\$"前缀)，以便标识符与其他语言元素区分开来。

示例	含义
<a href="#">abc^def</a>	调用 routine中的一个标签：  def是一个routine，abc是该routine中的一个标签。
<a href="#">i%abcdef</a>	i%<PropertyName>  用于从实例变量自身的 Get或Set访问器方法中引用该变量，或者绕过其Get或Set方法
<a href="#">abc-&gt;def</a>	这种语法只在 InterSystems的SQL语句中可以使用。它指定了一个隐含的左外连接。abc是你正在查询的类中的一个对象值字段，def是子类中的一个字段。
<a href="#">abc?def</a>	一个问号是模式匹配操作符，测试变量 abc中的值是否与def中指定的模式相匹配
<a href="#">abc[def</a>	左括号( [ ]是二进制包含运算符。这个表达式检测字符串"abc "是否包含字符串 "def"
<a href="#">abc]def</a>	右括号( ] )是二进制跟随运算符。这个表达式检测字符串"abc "是否在字符串 "def "之后，以ASCII的顺序排列。
<a href="#">abc]]def</a>	两个右括号( ]] )是二进制排序运算符。这个表达式检测字符串 "abc "是否排在字符串 "def "之后，以数字指标排序。
<a href="#">..</a>	相对点语法，两个句号组成一个类成员内部对另一个类成员的引用的开始。或者用于访问当前对象的一个属性调用一个方法

<a href="#">##Class()</a>	用于调用一个类的方法，将一个对象引用转换为另一个类来调用一个方法，或者用于访问一个类参数的值。
<a href="#">##super</a>	用于从子类方法中调用父类方法
<a href="#">\$abc</a>	系统变量或自定义变量
<a href="#">\$abc()</a>	系统函数或者自定义函数，括号中可用逗号分隔参数
<a href="#">\$abc.def.ghi(xxx)</a>	<p><a href="#">\$abc</a>是\$SYSTEM，<a href="#">def</a>是%SYSTEM包中的类的名称，<a href="#">ghi</a>是该类中一个方法的名称，<a href="#">xxx</a>是该方法的一个可选的逗号分隔的参数列表</p> <p><a href="#">\$SYSTEM</a>是 %SYSTEM包的别名，用来提供对该包中类的方法的访问</p>
<a href="#">\$\$abc</a>	<a href="#">abc</a> 是一个宏，美元符号不是其名称的一部分
<a href="#">\$\$abc()</a>	是一个定义在包含此引用的 routine或函数中的子程序。这种语法调用子程序 <a href="#">abc</a> 并获得其返回值
<a href="#">%abc</a>	InterSystems IRIS系统类中的大多数包、类和方法都以百分号开头。也有可能是InterSystems IRIS设置的一个局部变量，Routine或者嵌入SQL变量
<a href="#">%%abc</a>	有关于 InterSystems SQL，是伪字段的關鍵字
<a href="#">^abc</a>	一个函数或者一个执行程序 (Routine)，也可表示为全局参数
<a href="#">^\$abc</a>	结构化的系统变量
<a href="#">^ abc</a>	进程私有的全局参数，是一个用于临时存储大数据值的机制
<a href="#">^[XXX]%abc</a>	对另一个名字空间中的全局或例程的引用

- 常用系统变量 及 函数 :

变量 /函数/类	含义
<a href="#">\$SYSTEM</a>	提供系统信息
<a href="#">\$HOROLOG</a>	当前进程的日期和时间
<a href="#">\$zv</a>	系统版本信息
<a href="#">\$ZDH()</a>	用于检验日期格式,并将可读的日期格式转换成 InterSystems IRIS 数据库可读格式
<a href="#">\$zd()</a>	<p>用于检验日期格式,并将 InterSystems IRIS数据库日期格式转换成可读的日期格式。</p> <p>示例:</p> <p>字符串类型与日期类型相互转换 :\$ZDH/\$ZD</p> <pre>w \$ZDH("2017-03-13",3)</pre> <pre>w \$zd(0)</pre>
<a href="#">\$Random(range)</a>	返回符合范围内的随机正整数
<a href="#">\$zlog()</a>	Log指数计算
<a href="#">\$zsqr()</a>	平方根计算
<a href="#">\$Case</a>	根据第一个参数,可以返回相关的值,类似于 Switch case。或者返回一个procedure或者一个routine

示例:

```
s x =2
```

```
w $CASE(x,1:"single",2:"double",3:"triple",:"input error")
```

- [数组](#)

**本地数组: 生命周期在当前进程**

```
SET a(1) = "A local variable array"
```

```
SET a(1,1,1) = "Another local variable array"
```

**全局数组: 持久化数组, 直接存于磁盘**

```
SET ^a(1) = "A process-private global array"
```

```
SET ^a(1) = "A global array"
```

[#ObjectScript](#) [#Caché](#) [#Ensemble](#) [#InterSystems IRIS](#)

源 URL: <https://cn.community.intersystems.com/post/object-script-%E5%9F%BA%E7%A1%80%E7%9F%A5%E8%AF%86%E4%BA%8C>