

- 输入输出

```
1 print("Target variable: %s" % name_y)
2 # 数组以字符串形式输出
3 print("Predictors:      "+str(names_x))
4 # 输出多个数据用 () 包起来
5 print("num samples = %d, num features = %d" % (nsamp, nfeatures))

1 #两种补0方法
2 s = n.zfill(5)
3 n = 123
4 s = "%05d" % n
5 assert s == "00123"
6
```

array输出方法:

1. 直接写**array**的名字, 运行
2. 如果要和字符串拼接的话, 必须要用**str()** 或者**repr()** 转换才能输出
print("x = " + str(x))
3. 使用**np**函数
print('x = '+np.array_str(x, precision = 3))

Python两种输出值的方式: 表达式语句和 `print()` 函数。

第三种方式是使用文件对象的 `write()` 方法, 标准输出文件可以用 `sys.stdout` 引用。

如果你希望输出的形式更加多样, 可以使用 `str.format()` 函数来格式化输出值。

如果你希望将输出的值转成字符串, 可以使用 **`repr()` 或 `str()`** 函数来实现。

- **`str()`**: 函数返回一个用户易读的表达式。
- **`repr()`**: 产生一个解释器易读的表达式。

`str.format()` 的基本使用如下:

```
print("xlast= {0:5.3f}".format(xlast))
```

表示: 下标0, 最少5位 (不足则前面补空格), 保留小数点后三位

```
>>> print('{}网址: "{}!"'.format('菜鸟教程', 'www.runoob.com'))
菜鸟教程网址: "www.runoob.com!"
```

括号及其里面的字符 (称作格式化字段) 将会被 `format()` 中的参数替换。

在括号中的数字用于指向传入对象在 `format()` 中的位置, 如下所示:

```
>>> print('{0} 和 {1}'.format('Google', 'Runoob'))
Google 和 Runoob
>>> print('{1} 和 {0}'.format('Google', 'Runoob'))
```

如果在 `format()` 中使用了关键字参数, 那么它们的值会指向使用该名字的参数。

```
>>> print('{name}网址: {site}'.format(name='菜鸟教程', site='www.runoob.com'))  
菜鸟教程网址: www.runoob.com
```

Python提供了 `input()` 置函数从标准输入读入一行文本, 默认的标准输入是键盘。

`input` 可以接收一个Python表达式作为输入, 并将运算结果返回。

```
#!/usr/bin/python3  
  
str = input("请输入: ");  
print ("你输入的内容是: ", str)
```

这会产生如下的对应着输入的结果:

```
请输入: 菜鸟教程  
你输入的内容是:  菜鸟教程
```