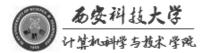


## 3.3 输入和输出

# 神经网络&深度学习 Google ENSORFLOW

## Python语法初步:

- 使用 "#" 作为单行注释符。
- Python变量在使用之前不需要声明。
- 语句块通过代码的缩进来表示。
- 标识符是大小写敏感的。
- print()函数在输出中自动包含换行。





#### ■ 输入函数:

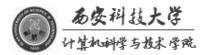
input (提示信息)

接收用户的输入,并以字符串类型返回

在屏幕上输出提示信息,并等待键盘输入,接收到的输入将被存储在字符串string中

提示信息

string = input("Please input some words:")



#### □ 接收用户输入

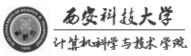
■ 命令行

```
>>> string = input("Please input some words:")
Please input some words:
>>> string = input("Please input some words:")
Please input some words:Hello!

提示信息
用户输入
```

**■** Jupyter Notebook

```
In [*]: string = input ("Please input some words:")
Please input some words:
In [*]: string = input ("Please input some words:")
Please input some words Hello!
```



提示信息 用户输入

Python 语言基础

```
# 判断变量num是否是正数
num=0

if num > 0:
    print ("num是正数")

else:
    print ("num可能是0")
    print ("num也可能是负数")

print ("num也可能是负数")

* num=input("请输入一个整数: ")

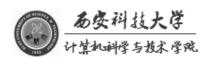
* 类型错误 原因: input函数的返回值是字符串类型
TypeError: unorderable types: str() > int()

* 类型转换函数

* num=input("请输入一个整数: "))
```

```
#判断变量num是否是正数
num=int(input("请输入一个整数: "))
print ("您输入的整数是: %d" %(num))
if num > ∅:
    print ("num是正数")
else:
    print ("num可能是0")
    print ("num也可能是负数")
```

接收用户输入





#### □ 接收用户输入

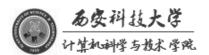
```
#判断变量num是否是正数
num=int(input("请输入一个整数: "))
print ("您输入的整数是: %d" %(num))
if num > 0:
    print ("num是正数")

else:
    print ("num可能是0")
    print ("num也可能是负数")
```

请输入一个整数: 2 num是正数

请输入一个整数: 0 num可能是0 num也可能是负数

请输入一个整数: -2 num可能是∅ num也可能是负数



### ■輸出函数

常量、变量、表达式

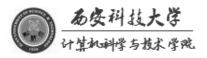
print (输出内容)

- □ 輸出字符串常量
- >>>print("Hello, Sir")
  Hello, Sir
- □ 输出数学表达式

>>>print(1+2)

□ 輸出变量

>>>x = 12 >>>print(x)



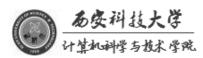
#### □ 格式化参数的使用

```
print ("您输入的整数是: %d" %(num))
```

格式化参数

变量

```
#判断变量num是否是正数
num=int(input("请输入一个整数: "))
print ("您输入的整数是: %d" %(num))
if num > 0:
    print ("num是正数")
else:
    print ("num可能是0")
    print ("num也可能是负数")
```



Python 语言基础

#### 运行结果

□ 输入正数: 2

□ 输入0

□ 输入负数: -2

请输入一个整数: 2

print ("您输入的整数是: %d" %(num))

您输入的整数是: 2

num是正数

请输入一个整数: 0

您输入的整数是: 0

num可能是0

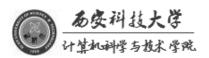
num也可能是负数

请输入一个整数: -2

您输入的整数是: -2

num可能是0

num也可能是负数

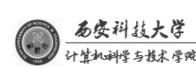


#### 3.3 输入和输出



#### ■ 常用的格式化参数

符号	描述	符号	描述
%с	格式化字符及其ASCII码	%x	格式化无符号十六进制数
%s	格式化字符串	%X	格式化无符号十六进制数 (大写)
%d	格式化整数	%f	格式化浮点数字,可指定小数点后的精度
%u	格式化无符号整型	%e	用科学计数法格式化浮点数
%o	格式化无符号八进制数	%p	用十六进制数格式化变量的地址



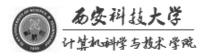
■ 字符串格式化参数

```
>>>print("His name is %s" %("Mike"))
His name is Mike
```

■ 使用多个格式化参数

```
print("...%s...%s...%s" %(string1,string2,...,stringn))
```

```
>>>yourname="Mike"
>>>myname="Devid"
>>>print("Hello,%s! I am %s." %(yourname,myname))
Hello,Mike! I am Devid.
```



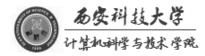
Python 语言基础

#### 3.3 输入和输出



#### □ 转义字符

转义字符	描述	转义字符	描述
\r	回车	\(行尾)	续行符
\n	换行	//	反斜杠符号
\t	横向制表符	\'	单引号
\v	纵向制表符	\"	双引号
\f	换页	\000	空
\a	响铃	\оууу	八进制数yyy代表的字符
\p	退格 (Backspace)	\xyy	十进制数yy代表的字符



■ 输出换行符

>>>print("纸上得来终觉浅, \newline \

转义字符

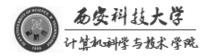
■ 输出引号

>>>print("使用转义字符输出一个双引号: \"")输出一个双引号: "

转义字符

■ 输出 \

>>>print("使用转义字符输出一个反斜杠: \\") 使用转义字符输出一个反斜杠: \

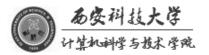




#### 失效转义字符

```
>>>print("C:\MyProgram\rencent\num\test\score")
>>>print("C:\MyProgram\rencent\num\test\score")
ecentrogram
um est\score

>>>print(r'C:\MyProgram\rencent\num\test\score")
C:\MyProgram\rencent\num\test\score
>>>print(R'C:\MyProgram\rencent\num\test\score")
C:\MyProgram\rencent\num\test\score
>>>print(R'C:\MyProgram\rencent\num\test\score")
C:\MyProgram\rencent\num\test\score
```



□ end参数

- print (输出内容, end)
- 表示输出信息结束之后,附加的字符串
- 默认值是换行\n。
- 设置这个参数,可以改变输出效果

设置end参数为空串,连续输出

```
不设置end参数,默认为\n,换行
```

```
>>>print( "Python")
>>>print( "3.5")
Python
3.5
```

```
>>>print( "Python", end="")
>>>print( "3.5")
Python3.5
```

```
>>>print( "Python", end=" ")
>>>print( "3.5")
Python 3.5

空格

设置end参数为空格
```

