

A L G O R I T H M

P R A C T I C E

深度学习算法与实践

Steven Tang



基础

进阶

Python语法元素分析

程序的格式框架

Python

Python语言采用严格的“缩进”来表明程序的格式框架。缩进指每一行代码开始前的空白区域，用来表示代码之间的包含和层次关系。

1个缩进 (Tab) = 4个空格 (Space)

- 用以在Python中标明代码的层次关系
- 缩进是Python语言中表明程序框架的唯一手段

程序的格式框架

单层缩进

```
flag = False
name = 'luren'
if name == 'python':
    # 判断变量是否为 python
    flag = True
    # 条件成立时设置标志为真
    print('welcome!')
    # 并输出欢迎信息
else:
    print(name)
    # 条件不成立时输出变量名称
```

多层缩进

```
if True:
    if True:
        print( "welcome" )
```

缩进快捷键**Tab**

反缩进快捷键**Shift+Tab**

注释 (一个程序员的自我修养1)

注释：程序员在代码中加入的说明信息，不被计算机执行

注释的两种方法：

- 单行注释以#开头

```
#Here are the comments
```

- 注释快捷键Ctrl+1

- 多行注释以'''开头和结尾

```
'''
```

```
This is a multiline comment  
used in Python
```

```
'''
```

可移植性
可维护性

Python中文编码

-*- coding: utf-8 -*-

常见的编码格式： GBK

常见的编码格式： utf-8

```
1# -*- coding: gbk -*-  
2s="你好"  
3print(s)
```

标识符命名(一个程序员的自我修养2)

系统性

$$\underline{\underline{\pi = 3.1415926}}$$

马蜂窝命名

- 常量: 程序中值不发生改变的元素
- 变量: 程序中值发生改变或者可以发生改变的元素

Python语言允许采用大写字母、小写字母、数字、下划线(_) 和 汉字 等字符及其组合给变量命名，但名字的首字符不能是数字，中间不能出现空格，长度没有限制

duyile Name!

注意: 标识符对大小写敏感, python和Python是两个不同的名字

保留字

17 保留字

Import

Spyder

- ✓ 保留字，也称为关键字，指被编程语言内部定义并保留使用的标识符。
- ✓ 程序员编写程序不能定义与保留字相同的标识符(变量)。
- ✓ 每种程序设计语言都有一套保留字，保留字一般用来构成程序整体框架、表达关键值和具有结构性的复杂语义等。
- ✓ 掌握一门编程语言首先要熟记其所对应的保留字。

Python保留字

- ✓ Python 3.x保留字列表 (33个), 保留字与内置函数不同。

and	elif	<u>import</u>	raise
as	else	in	<u>return</u>
assert	except	is	try
break	finally	lambda	while
class	for	nonlocal	with
continue	from	not	yield
<u>def</u>	global	or	<u>True</u>
del	if	pass	<u>False</u>
			<u>None</u>

内建函数type

type()

```
In [6]: type(3)
```

```
Out[6]: int
```

```
In [7]: type(3.6)
```

```
Out[7]: float
```

```
In [8]: type(True)
```

```
Out[8]: bool
```

```
In [9]: type(None)
```

```
Out[9]: NoneType
```

```
In [10]:
```

保留字 True 和 False

True False 都是布尔值。

True和1、False和0除了数据类型以外都是等价的。

保留字None

None是一个特殊的常量

None和False不同

None不是0

None不是空字符串

None有自己的数据类型NoneType

你可以将None复制给任何变量，但是你不能创建其他NoneType对象

[None, 3]

[None, 3]

5x3

[None, 3]

内建函数help

help可以用来查看任意数据的信息。

```
In [11]: help(None)
Help on NoneType object:

class NoneType(object)
|   Methods defined here:
|
|   __bool__(self, /)
|       self != 0
|
|   __repr__(self, /)
|       Return repr(self).
|
|   -----
|   Static methods defined here:
|
|   __new__(*args, **kwargs) from builtins.type
|       Create and return a new object.  See help(type) for accurate signature.
```

Python 基本数据类型



数字类型

Python 不需要声明数据类型

a = None

a

long
long
double

```
# -*- coding: gbk -*-  
a=3  
b=2.3  
c=3+4j
```

int(a);

3 2
|
1

```
#include "stdafx.h"  
int main()  
{  
  
    int a = 3;  
    float b = 2.3;  
  
    return 0;  
}
```


数字类型

Python语言包括三种数字类型

- 整数类型
- 浮点数类型
- 复数类型

整数类型

- 与数学中的整数概念一致，没有取值范围限制
- `pow(x, y)`函数：计算 x^y
 - 程序1：`pow(2,10)` , `pow(2,15)`
 - 程序2：`pow(2, 1000)`
 - 程序3：`pow(2, pow(2,15))`

$$\sqrt{-1} = j$$

整数类型

■ 示例

- 1010, 99, -217
- 0x7a, -0x29 (0x, 0X开头表示16进制数)
- 0b010, -0B101 (0b, 0B开头表示2进制数)
- 0o123, -00456 (0o, 0O开头表示8进制数)

0x

0b

0o

0X

0B

0O

→ Binary
Oct

内建函数id

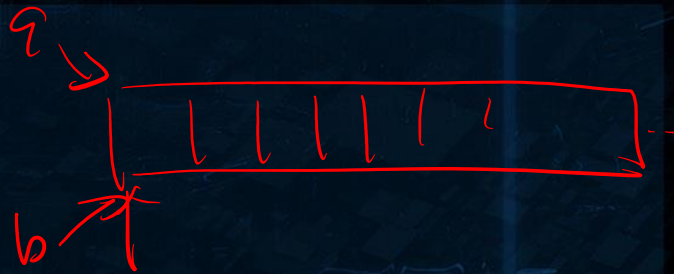
用内建函数 id() 可以查看每个对象的内存地址，即身份。

```
In [39]: id(2)
Out[39]: 140708879114704

In [40]: a=3
In [41]: b=3

In [42]: id(a)
Out[42]: 140708879114704

In [43]: id(b)
Out[43]: 140708879114704
```



浮点数类型

- 带有小数点及小数的数字
- Python语言中浮点数的数值范围存在限制，小数精度也存在限制。这种限制与在不同计算机系统有关

64 . 32

浮点数类型

◆ 0.0, -67., -2.17

◆ 9e4, 4.1e-2, 9.3E4

◆ 科学计数法使用字母“e”或“E”作为幂的符号，以10为基数。科学计数法含义如下：

$$\langle x \rangle e \langle y \rangle = x * 10^y$$

4. = 4.0

4.1×10^{-2}

0.041

复数类型

- ◆ $z = 1.24e-5 + 39j$ (实部和虚部是什么?)
- ◆ 对于复数 z , 可以用 $z.real$ 获得实数部分, $z.imag$ 获得虚数部分
- ◆ $z.real = 1.24e-05$ $z.imag = 39$

1.24×10^{-5} 39.0

数字类型的操作

内置的数值运算操作符

- 三种类型存在一种逐渐“扩展”的关系：

整数 ($a=2$) \rightarrow 浮点数 ($a=2.5$) \rightarrow 复数 ($a=2.5+4j$)

(整数是浮点数特例，浮点数是复数特例)

- 不同数字类型之间可以进行混合运算，运算后生成结果为最宽类型
- $123 + 4.0 = 127.0$ (整数 + 浮点数 = 浮点数)

内置的数值运算操作符

操作符	描述
$x + y$	x与y之和
$x - y$	x与y之差
$x * y$	x与y之积
x / y	x与y之商
$x // y$	x与y之整数商，即：不大于x与y之商的最大整数
$x \% y$	x与y之商的余数，也称为模运算
$-x$	x的负值，即： $x*(-1)$
$+x$	x本身
$x**y$	x的y次幂，即： x^y

内置的数值运算函数

Python解释器提供了一些内置函数，在这些内置函数之中，有6个函数与数值运算相关

函数	描述
<code>abs(x)</code>	x的绝对值
<code>divmod(x, y)</code>	$(x/y, x\%y)$ ，输出为元组形式（也称为元组类型）
<code>pow(x, y[, z])</code>	$(x**y)\%z$ ， <code>[..]</code> 表示该参数可以省略，即： <code>pow(x,y)</code> ，它与 <code>x**y</code> 相同
<code>round(x[, ndigits])</code>	对x四舍五入，保留ndigits位小数。 <code>round(x)</code> 返回四舍五入的整数值
<code>max(x₁, x₂, ..., x_n)</code>	x_1, x_2, \dots, x_n 的最大值，n没有限定
<code>min(x₁, x₂, ..., x_n)</code>	x_1, x_2, \dots, x_n 的最小值，n没有限定

类型的转换

数值运算操作符可以隐式地转换输出结果的数字类型

例如，两个整数采用运算符“/”的除法将可能输出浮点数结果。此外，通过内置的数字类型转换函数可以显式地在数字类型之间进行转换

函数	描述
<code>int(x)</code>	将x转换为整数，x可以是浮点数或字符串
<code>float(x)</code>	将x转换为浮点数，x可以是整数或字符串
<code>complex(re, im)</code>	生成一个复数，实部为re，虚部为im，re可以是整数、浮点数或字符串，im可以是整数或浮点数但不能为字符串
<code>bool(x)</code>	将x转换为bool值（True或者False），x可以是数字类型也可以其他类型

数字类型的转换

- ◆ 三种类型可以相互转换

函数: `int()`, `float()`, `complex()`

- ◆ 示例:

- ◆ `int(4.5) = 4` (直接去掉小数部分)
- ◆ `int('15',16)=21` (将15转成16进制输出, base从6到36)
- ◆ `float(4) = 4.0` (增加小数部分)
- ◆ `complex(4) = 4 + 0j`

bool类型

```
In [26]: bool(true)
Traceback (most recent call last):

  File "<ipython-input-26-0e7e8772b38a>", line 1, in <module>
    bool(true)
NameError: name 'true' is not defined

In [27]:

In [27]: bool(True)
Out[27]: True

In [28]: bool(False)
Out[28]: False

In [29]: bool(1)
Out[29]: True

In [30]: bool(0)
Out[30]: False

In [31]: bool(None)
Out[31]: False

In [32]: bool("hello world!")
Out[32]: True
```

数字类型的转换

- 示例: `complex(4.5) = 4.5 + 0j`

```
>>> float(4.5+0j)
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    float(4.5+0j)
TypeError: can't convert complex to float
>>>
```

同端对齐
↓
1024 × 968 × 3

字符串

"s" 自然语言 结束

- Python语言中，字符串是用两个双引号 "" 或者单引号 ' ' 括起来的一个或多个字符。
- Python字符串的两种序号体系

"abc" = ['a', 'b', 'c']

6 7 = 6 + 1
-5 ⑤

Python

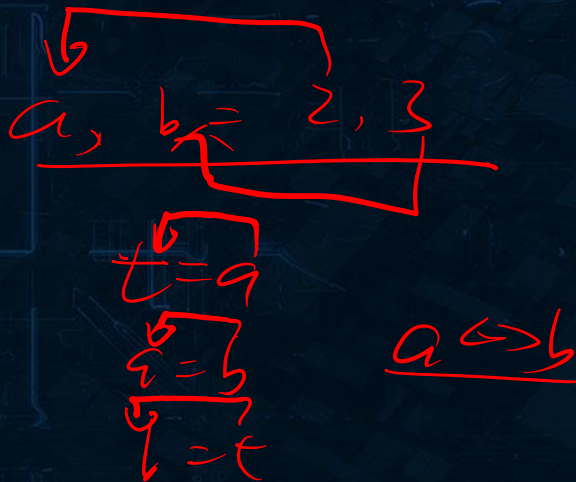
反向递减序号										
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
H	e	l	l	o		W	o	r	l	d
0	1	2	3	4	5	6	7	8	9	10
正向递增序号										

赋值语句

■ Python语言中, = 表示“赋值”, 即将等号右侧的值计算后将结果值赋给左侧变量, 包含等号 (=) 的语句称为“赋值语句”

■ 同步赋值语句: 同时给多个变量赋值

<变量1>, ..., <变量N> = <表达式1>, ..., <表达式N>



input()函数

变量

SiChu

- 获得用户输入之前，input()函数可以包含一些提示性文字

2.7

<变量> = input(<提示性文字>)

- input函数会把输入的数据传给左边的变量来保存，2.7版本中该变量是数字类型，3.x版本当中是字符串类型。

```
In [16]: input("请输入: ")
```

```
请输入: python
```

```
Out[16]: 'python'
```

```
In [17]: input("请输入: ")
```

```
请输入: 123.321
```

```
Out[17]: '123.321'
```

eval() 函数

eval

evaluated)

- eval(<字符串>)函数是Python语言中一个十分重要的函数，它能够以Python表达式的方式解析并执行字符串，将返回结果输出

```
In [18]: a=3
In [19]: b=2
In [20]: eval("a+b")
Out[20]: 5
```

eval("a+b")

a+b

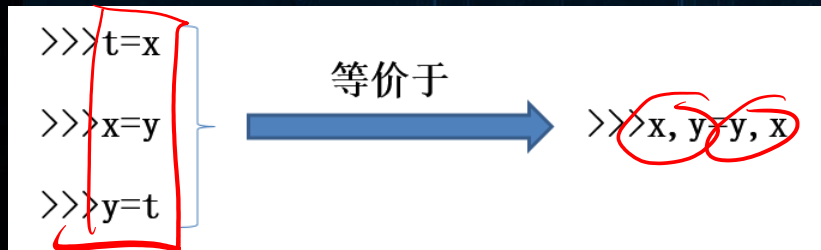
赋值语句

例：将变量x和y交换

■采用单个赋值，需要3行语句：

即通过一个临时变量t缓存x的原始值，然后将y值赋给x，再将x的原始值通过t赋值给y。

■采用同步赋值语句，仅需要一行代码：



输出函数

- ◆ print()函数用来输出字符信息，或以字符形式输出变量。
- ◆ print()函数可以输出各种类型变量的值。
- ◆ print()函数通过%来选择要输出的变量。

不带回车:

```
In [5]: print("没有回车",end='');print("没有回车",end='')  
没有回车没有回车
```

```
In [6]: print("有回车");print("有回车")  
有回车  
有回车
```

```
In [7]:
```

格式化输出:

```
In [8]: s='Duyi jiaoyu'  
...: x=len(s)  
...: print('The length of %s is %d' %(s,x))  
The length of Duyi jiaoyu is 11
```

```
In [9]:
```

String
decimal

%f

%x

%b

拼接输出:

```
In [9]: a=10
```

```
In [10]: print("Number is ",a)  
Number is 10
```

```
In [11]:
```

格式符

%% 百分号标记 #就是输出一个%

%c 字符及其ASCII码

%s 字符串

%d 有符号整数(十进制)

%u 无符号整数(十进制)

%o 无符号整数(八进制)

%x 无符号整数(十六进制)

%X 无符号整数(十六进制大写字符)

%e 浮点数字(科学计数法)

%E 浮点数字(科学计数法, 用E代替e)

%f 浮点数字(用小数点符号)

%g 浮点数字(根据值的大小采用%e或%f)

%G 浮点数字(类似于%g)

%p 指针(用十六进制打印值的内存地址)

%n 存储输出字符的数量放进参数列表的下一个变量中

format()方法的基本使用

字符串format()方法的基本使用格式是：

<模板字符串>.format(<逗号分隔的参数>)

"{1} {0} {1}".format("hello", "world") 输出结果为 "world hello world"。

8 5 8 8 8

字符串类型及其操作

字符串类型

- 字符串是用双引号"或者单引号"括起来的一个或多个字符。
- 字符串可以保存在变量中，也可以单独存在。

字符串类型

■ Python语言转义符:

■ 输出带有引号的字符串，可以使用转义符

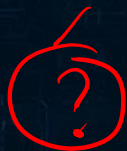
■ 使用 `\\` 输出带有转移符的字符串

```
>>> print("\\"大家好\\")
"大家好"
>>>
```

正反斜杠总结

正斜杠 / : 除法运算、相对路径

反斜杠 \ : 转义字符、绝对路径



字符串类型

- 字符串是一个字符序列：字符串最左端位置标记为0，依次增加。字符串中的编号叫做“索引”

H	E	L	L	O		W	O	R	L	D	!
0	1	2	3	4	5	6	7	8	9	10	11

字符串类型

- 单个索引辅助访问字符串中的特定位置

格式为<string>[<索引>]

() \leftarrow 定义

[] \leftarrow 取值

{ } \leftarrow 集合

```
In [13]: string "Hello world!"  
In [14]: print(string[3])  
l  
In [15]: print(string[-3])  
l  
In [16]:
```

字符串类型

- Python中字符串索引从0开始，一个长度为L的字符串最后一个字符的位置是L-1
- Python同时允许使用负数从字符串右边末尾向左边进行反向索引，最右侧索引值是-1

切片操作

= 值

切片操作

- 可以通过两个索引值确定一个位置范围，返回这个范围的子串
格式： <string>[<开始>:<结束>]
- start和end都是整数型数值，这个子序列从索引start开始直到索引end结束，但不包括end位置。

```
In [16]: string[0:3]  
Out[16]: 'Hel'
```

```
In [17]: string[:3]  
Out[17]: 'Hel'
```

```
In [18]:
```


字符串类型

- 字符串之间可以通过+或*进行连接
- 加法操作(+)将两个字符串连接成为一个新的字符串
- 乘法操作(*)生成一个由其本身字符串重复连接而成的字符串

```
In [18]: "pine"+"apple"  
Out[18]: 'pineapple'  
  
In [19]: 4*"pine"  
Out[19]: 'pinepinepinepine'  
  
In [20]:
```

内置函数len()

length

len

字符串

- len()函数能否返回一个字符串的长度

```
In [3]: len("Python")  
Out[3]: 6
```

```
In [4]: len("好好学习，天天向上")  
Out[4]: 10
```

内置函数str

String

- 大多数数据类型都可以通过str()函数转换为字符串

```
In [8]: str(25)
Out[8]: '25'

In [9]: str(25.555)
Out[9]: '25.555'

In [10]: str(25e5)
Out[10]: '2500000.0'

In [11]: str(True)
Out[11]: 'True'

In [12]: str(None)
Out[12]: 'None'
```

字符串的操作

for 循环

可以通过 for 和 in 组成的循环来遍历字符串中每个字符

■ 格式如下：

for <var> in <string>:

操作

```
In [20]: string="Hello world!"
```

```
In [21]: for s in string:  
...:     print(s)  
...:
```

H

e

l

l

o

w

o

r

l

d

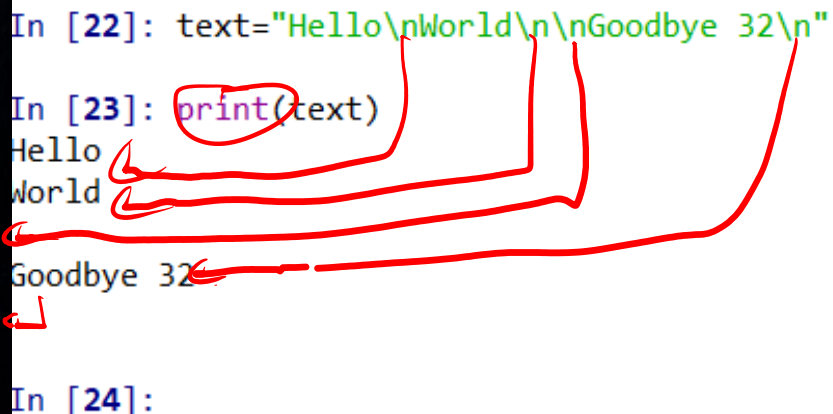
!

字符串的操作

用转义符可以在字符串中表达一些不可直接打印的信息

例如：用\n表示换行

```
In [22]: text="Hello\nWorld\n\nGoodbye 32\n"  
  
In [23]: print(text)  
Hello  
World  
  
Goodbye 32  
  
In [24]:
```



字符串 ljust, rjust

format

`ljust()` `rjust()`方法返回一个原字符串左(右) 对齐,并使用空格填充至指定长度的新字符串。如果指定的长度小于原字符串的长度则返回原字符串。

Left

```
In [50]: 'Hello world!'.ljust(20)
Out[50]: 'Hello world!'
```

```
In [51]: 'Hello world!'.rjust(20)
Out[51]: '      Hello world!'
```

```
In [52]: 'Hello world!'.ljust(20, '*')
Out[52]: 'Hello world!*****'
```

内置的字符串处理函数

操作	含义
+	连接
*	重复
<string>[]	索引
<string>[:]	剪切
len(<string>)	长度
<string>.upper()	字符串中字母大写
<string>.lower()	字符串中字母小写
<string>.strip()	去两边空格及去指定字符
<string>.split()	按指定字符分割字符串为数组
<string>.join()	连接两个字符串序列
<string>.find()	搜索指定字符串
<string>.replace()	字符串替换
for <var> in <string>	字符串迭代

小练习

查看变量类型的Python内置函数是_____。

查看变量内存地址的Python内置函数是_____。

表达式 `int('123', 16)` 的值为_____。

表达式 `'.'.join('abcdefg'.split('cd'))` 的值为_____。

表达式 `len('abcdefg'.ljust(3))` 的值为_____。