

在人工智能时代下，机器学习、深度学习、大数据分析处理、云计算等新技术的提出极大地促进了Python语言的发展，2017-2020年连续4年，IEEE Spectrum 发布的编程语言交互式排行榜上，Python独占榜首。Python以面向对象、容易学习、跨平台以及由此产生的低维护成本，深受Python爱好者喜欢并获得好评。

1 Python的特点

1.1 代码简单优雅

尽量写容易看明白的代码，尽量写少的代码

1.1.1 输出“Hello World!”

In [1]:

```
print("Hello World!")
```

Hello World!

1.1.1.1 C语言代码

```
#include<stdio.h>
int main()
{
    printf("Hello World!\n");
}
```

1.1.1.2 Java语言代码

```
public class HelloWorld{
    public static void main(String[] args){
        System.out.println("Hello World");
    }
}
```

比如，完成同一个任务，C语言要写1000行代码，Java只需要写100行，而Python可能只要20行。

1.1.2 计算1+2+...+100的和

In [2]:

```
print(sum(range(1, 101)))
```

5050

1.1.3 列出100~200之间被7整除的数。

In [3]:

```
print(' '.join([str(n) for n in range(100,201) if n%7==0]))
```

105 112 119 126 133 140 147 154 161 168 175 182 189 196

1.1.4 交换两个变量的值

In [4]:

```
a = 11
b = 22
print(f"a,b的值分别为: {a}, {b}")
a, b = b, a
print(f"交换变量的值后, a,b的值分别为: {a}, {b}")
```

a,b的值分别为: 11, 22

交换变量的值后, a,b的值分别为: 22, 11

1.1.5 打印九九乘法表

In [5]:

```
print('\n'.join(['\t'.join([f'{y} × {x}={x*y}' for y in range(1, x + 1)]) for x in range(1, 10)]))
```

```
1×1=1
1×2=2  2×2=4
1×3=3  2×3=6  3×3=9
1×4=4  2×4=8  3×4=12  4×4=16
1×5=5  2×5=10  3×5=15  4×5=20  5×5=25
1×6=6  2×6=12  3×6=18  4×6=24  5×6=30  6×6=36
1×7=7  2×7=14  3×7=21  4×7=28  5×7=35  6×7=42  7×7=49
1×8=8  2×8=16  3×8=24  4×8=32  5×8=40  6×8=48  7×8=56  8×8=64
1×9=9  2×9=18  3×9=27  4×9=36  5×9=45  6×9=54  7×9=63  8×9=72  9×9=81
```

1.2 动态类型语言

变量不需要声明数据类型。根据变量的赋值，Python解释器自动确定其数据类型。

要了解什么是动态语言，要首先了解类型检查。

类型检查是验证类型约束的过程，编译器或解释器通常在编译阶段或运行阶段做类型检查。

类型检查就是查看变量和它们的类型，然后判断表达式是否合理。例如，不能拿一个string类型变量除以浮点数变量。

类型检查发生在程序运行阶段(run time)的是动态类型语言(dynamically typed languages)。

常见的动态语言包括：Python, Javascript, PHP。

类型检查发生在编译阶段(compile time)的是静态类型语言 statically typed languages)。

常见的静态类型语言包括：C, C++, Java, C#, Scala。

1.3 强类型语言

每个变量指向的值均属于某个数据类型，即只支持该类型允许的运算操作。

强类型语言是指：不管是在编译阶段还是运行阶段，一旦某种类型绑定到变量后，此变量便会持有此类型，并且不能同其他类型在计算表达式时混合使用。

例如，在交互式工具IPython中输入如下两行代码：

```
In [1] a = 5
```

```
In [2] a = a + 's'
```

程序会抛出TypeError异常：unsupported operand type(s) for + 'int' and 'str'，不支持 int变量和 str变量相加。

常见的强类型语言有：Python, Java, C#, Scala。

与之对应的是弱类型语言，弱类型语言容易与其他类型混合计算。

弱类型语言代表 Javascript支持如下操作：

```
var data = 5
```

```
data = data + 'xiaoming'
```

string和int结合自动转化为string

常见的弱类型语言有：C, C++, PHP, Javascript。

In [6]:

```
a = 7
b = 11
c = "22"
print(a+b)
```

18

In [7]:

```
print(a+c)
```

```
-----
-
TypeError                                Traceback (most recent call last)
<ipython-input-7-29620a3af8d7> in <module>
----> 1 print(a+c)
```

TypeError: unsupported operand type(s) for +: 'int' and 'str'

1.4 第三方库丰富



1.5 应用领域广泛

1. WEB开发——最火的Python web框架Django, 支持异步高并发的Tornado框架, 短小精悍的flask, Sanic
2. 网络编程——支持高并发的Twisted网络框架, py3引入的asyncio使异步编程变的非常简单
3. 爬虫——爬虫领域, Python几乎是霸主地位, Scrapy\Requrest\BeautifuSoap\urllib等, 想爬啥就爬啥
4. 云计算——目前最火最知名的云计算框架就是OpenStack, Python现在的火, 很大一部分就是因为云计算
5. 人工智能、数据分析—— Python 是目前公认的人工智能和数据分析领域的必备语言
6. 自动化运维\测试——DevOps、SRE工程师标配语言, 腾讯蓝鲸运维平台、JumpServer、Saltstack、Ansible
7. 金融量化——我个人之前在金融行业CICC, 2010年的时候, 我们公司写的好多分析程序、高频交易软件就是用的Python, 到目前, Python是金融分析、量化交易领域里用的最多的语言
8. 科学运算—— 97年开始, NASA就在大量使用Python在进行各种复杂的科学运算, 随着NumPy, SciPy, Matplotlib, Enthought librarys等众多程序库的开发, 使的Python越来越适合于做科学计算、绘制高质量的2D和3D图像。
9. 游戏开发——知名的游戏 < 文明 >、< 大话西游 >、< Eve > 就是用Python写的

Python有哪些用途？

数据驱动的五大数据领域分层架构图



1.6 相关岗位工资高



2 Python的四大基本语法

主要从变量命名基本规则等方面总结Python的基本语法规则。

2.1 命名规则

Python的变量命名规则主要包括两条：

- (1) 允许包括英文、数字以及下划线(_)，不能以数字开头。
- (2) 名称区分大小写。

特别说明以下划线开头的变量是有特殊意义的：

类变量若以单下划线(_)开头, 代表不能直接被访问。

类变量若以双下划(__)开头, 表示为类的私有成员, 不能被导入和其他类变量访问。

以双下划开头和双下划线结尾的变量是Python里的专用标识, 有特殊的身分。

如Python自定义类中都包括__init__和__add__方法, 如果不重写__add__去执行两个类加法操作, 程序会抛TypeError异常。

只有重写后, 才能正常执行加法操作。

Python变量命名习惯一般遵守蛇形命名法(snake case):

一般变量命名, book_id, book_store_count;

类名首字符为大写, 如Python内置模块collections.abc中的Iterable类, 我们自定义的Book类等;

类方法名: get_store_count()

其他特殊变量, 全部大写M_PI, MAX_VEHICLE_SPEED

注: 这与Java命名方法不同, Java最典型的命名方法是驼峰命名法(camelcase)。

2.2 缩进原则

通过缩进来标明代码的层次关系。

Python最具特色的地方就是用缩进代替Java, C++中的 {}, 缩进的层级结构表示代码的逻辑层次。

1个缩进等于4个空格, 它是 Python 语言中标明程序框架的唯一手段。

缩进格式, 行间空行数, 变量和等号空格等Python编码规范参考PEP8。

In [6]:

```
''' 题目: 一个整数, 它加上100后是一个完全平方数, 再加上168又是一个完全平方数, 请问该数是多少?
分析: 在10万以内判断, 先将该数加上100后再开方, 再将该数加上268后再开方, 如果开方后的结果满足如下条件
'''
```

```
import math
for i in range(10000):
    x = int(math.sqrt(i + 100)); y = int(math.sqrt(i + 268))           #转化为整型值
    if(x * x == i + 100) and (y * y == i + 268):print(i, end=' ')
```

21 261 1581

: 分隔代码块的头和体。

表示注释, 作用于#之后的内容。

''' ''' 或者 """ """ 表示区间注释, 在三引号之间的所有内容被注释(多行注释)。

换行(\n)是标准的行分隔符(通常一条语句一行)。

; 连接两条语句成一行。

\ 连接下一行(一行代码过长, 中间使用\分隔, 后面部分可以换到下一行)

In [2]:

```
import math
for i in range(10000):
    x = int(math.sqrt(i + 100))
    y = int(math.sqrt(i + 268))
    if(x * x == i + 100) and (y * y == i + 268):
        print(i, end=' ')
```

21 261 1581

In [1]:

```
'''【问题描述】1200000有多少个约数（只计算正约数）。
【答案提交】这是一道结果填空的题，你只需要算出结果后提交即可。
本题的结果为一个整数，在提交答案时只填写这个整数，填写多余的内容将无法得分。'''
a=1200000
n=0
for i in range(1,int(a**0.5)):
    if a%i==0:
        n += 2
print(n)
```

96

2.3 特殊关键字

Python有35个关键字：

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

所有 Python 的关键字只包含小写字母。自定义变量名不能与它们重复。

常用且不同于其他常用语言c++和 Java 的关键字，如：

True和False用于表示值的真假，在Java中true和false；

逻辑反操作Python使用not，而Java是!；

None表示空值，Java中null；

Python两个条件同时满足使用and，而Java是&&；

两者满足其一，Python使用or而Java使用||；

Python使用elif，而Java是else if；

其他比较特殊的关键字，如：

del用于删除可迭代对象中某个元素；def用于定义函数；

带yield用于定义生成器(generator)函数；

global和nonlocal一种应用是Python函数式编程的闭包场景；

pass一种应用是Python中接口定义，也是Python语言特有的一个关键字。

这些关键字的用法会在后面课程中详细介绍，在此先只要对它们形成一个整体上的认识就行。

In [8]:

```
import keyword
s = sorted(keyword.kwlist)
for n, i in enumerate(s):
    print(f' {i:10}', end='')
    j = n+1
    if j%10 ==0:
        print()
        # 每行输出十个关键字后换行
```

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

2.4 Python运算符

Python的运算符包括：

+	-	*	/	**	//	%
@	<<	>>	&		^	~
:=	<	>	<=	>=	==	!=

[Python 运算符 | 菜鸟教程 \(https://www.runoob.com/python/python-operators.html\)](https://www.runoob.com/python/python-operators.html)

2.4.1 算数运算符

加+、减-、乘*、除/、取余%、乘方**、取整//。支持+=、-=、*=、/=、%=、**=、//=。

In [8]:

```
17/4
```

Out[8]:

4.25

In [3]:

```
17%4 # 除法的余数
```

Out[3]:

1

In [4]:

```
17//4 #除法的商
```

Out[4]:

4

In [7]:

```
divmod(17,4)
```

Out[7]:

(4, 1)

In [11]:

```
2**3
```

Out[11]:

8

In [12]:

```
# 三目运算符
a = 2 if 4>5 else 6
print(a)
```

6

2.4.2 赋值运算符

In [9]:

```
x = 10
print(x)
```

10

In [10]:

```
# 增量赋值
x += 1
print(x)
```

11

In [12]:

```
z = 10
y = z
x = y
print(x, y, z)
```

10 10 10

In [13]:

```
# 链式赋值
a = b = c = 7
print(a, b, c)
```

7 7 7

In [14]:

```
# 交换赋值
a = 2
b = 9
print(a, b)
a, b = b, a
print(a, b)
```

2 9

9 2

In [17]:

```
# 解压赋值
nums = [11, 22, 33, 44]
a, b, c, d = nums
print(a, b, c, d)
```

11 22 33 44

In [15]:

```
a = '中华人民共和国'
n = len(a)
if n > 5:
    print(f"{a}，此字符串的长度大于5")
```

中华人民共和国，此字符串的长度大于5

In [16]:

```
# 如果使用海象运算符，写法上更为精简：
a = '中华人民共和国'
if (n := len(a)) > 5: # := 称为海象运算符。
    print(f"'{a}' 的长度大于5")
```

'中华人民共和国' 的长度大于5

2.4.3 比较运算符

==、>、<、!=、>=、<=等

In [21]:

```
# Python 比较运算符还支持链式比较，应用起来非常方便，比如：
i = 3
print(1 < i < 3)
```

False

In [22]:

```
i = 3
print(1 < i <= 3)
```

True

In [23]:

```
print(4<8<12)
print(5<14<8)
```

True

False

2.4.4 逻辑运算符

逻辑与、逻辑或、逻辑非，它们的优先级为：not>and>or

2.4.5 成员运算符

In [24]:

```
'hello' in 'hello world'
```

Out[24]:

True

虽然下述两种判断可以达到相同的效果，但我们推荐使用第二种格式，因为`not in`语义更加明确。

In [25]:

```
not 'lili' in ['jack', 'tom', 'robin']
```

Out[25]:

True

In [26]:

```
'lili' not in ['jack', 'tom', 'robin']
```

Out[26]:

True

2.4.6 身份运算符

“==”用于判断两个对象的内容是否相同，只关注两个对象的值（value）是否相等。
“is”用于判断两个对象的内存地址是否相同，真正判断两个对象是不是同一个对象。
因此，is的要求更高，不仅要求值一样，同时还要求对象内存地址也是一样。

In [18]:

```
x = 'Info Tony:18'
y = 'Info Tony:18'
print(id(x), id(y))      # x与y的id不同，但是二者的值相同
print(x==y, x is y)
```

161180976 158702448

True False

In [17]:

```
a = b = 6
print(a, id(a), b, id(b), a==b, a is b)
a = 8
print(a, id(a), b, id(b), a==b, a is b)
```

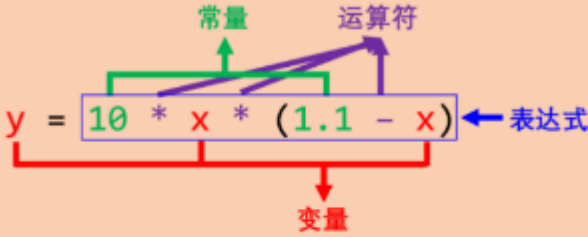
6 8791536707536 6 8791536707536 True True

8 8791536707600 6 8791536707536 False False

3 常量和变量

表达式

3



常量 (constant) 是永远不会改变的值, 如上面的 1.1 和 10。

变量 (variable) 是会变的值, 由赋值符「等号」赋值, 如上面的 x 和 y。

表达式 (expression) 由若干个常量、变量和运算符共同构成。

运算符 (operator) 包括

- 赋值运算符 (=)
- 算术运算符 (+, -, *, /, **, %, -)
- 比较运算符 (==, !=, >, <, >=, <=)
- 按位运算符 (&, |, ^, <<, >>, ~)
- 逻辑运算符 (and, or, not)
- 身份运算符 (is, is not)
- 成员运算符 (in, not in)

常量: 内存中用于保存固定值的单元。在程序中, 常量的值不能发生改变。
Python常量的赋值: 数字、字符串、布尔值、空值、集合。

变量的命名规范:

- (1) 变量名是由大小写字符、数字和下划线(_)组合而成的;
- (2) 第一个字符必须是字母或下划线开始, 不可用数字开头, 不要用中文开头;
- (3) Python中的变量是区分大小写的; 在赋值时单引号、双引号的效果是一样的;
- (4) Python中已经被使用的一些关键词不能用于声明变量。

变量推荐写法:

age_of_alex = 10 # 做到见名知义, 不写拼音、汉字, 全为小写

4 输入和输出

Python的输入函数主要是: input()

Python的输出函数主要是: print()

```
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

将对象以字符串表示的方式格式化输出到流文件对象file里。其中所有非关键字参数都按str()方式进行转换为字符串输出;

关键字参数sep是实现分隔符, 比如多个参数输出时想要输出中间的分隔字符;

关键字参数end是输出结束时的字符, 默认是换行符\n;

关键字参数file是定义流输出的文件, 可以是标准的系统输出sys.stdout, 也可以重定义为别的文件;

关键字参数flush是立即把内容输出到流文件, 不作缓存。

In [31]:

```
str1 = input('请你从键盘上输入一个整数:')
print(f'你输入的整数的平方为: {int(str1)**2}')
```

请你从键盘上输入一个整数:25

你输入的整数的平方为: 625

In [34]:

```
print('Hello World!')  
print() # 没有参数时，每次输出后都会换行。（相当于输出一个空行）  
print("I'm a chinese.")
```

Hello World!

I'm a chinese.

In [35]:

```
print('2020', '9', '10', sep='-')
```

2020-9-10

In [36]:

```
s='静夜思\n作者：李白\n床前明月光，疑是地上霜。举头望明月，低头思故乡。'  
print(s)
```

静夜思
作者：李白
床前明月光，疑是地上霜。
举头望明月，低头思故乡。

In [37]:

```
# Python 还支持格式化输出数据，一种是调用format()函数来实现的。  
print(format(12.3456, '6.2f')) # 输出6位数值的浮点数，小数点后精确到2位
```

12.35

In []: