

++

# Python 入门

Python Beginner's  
Tutorial



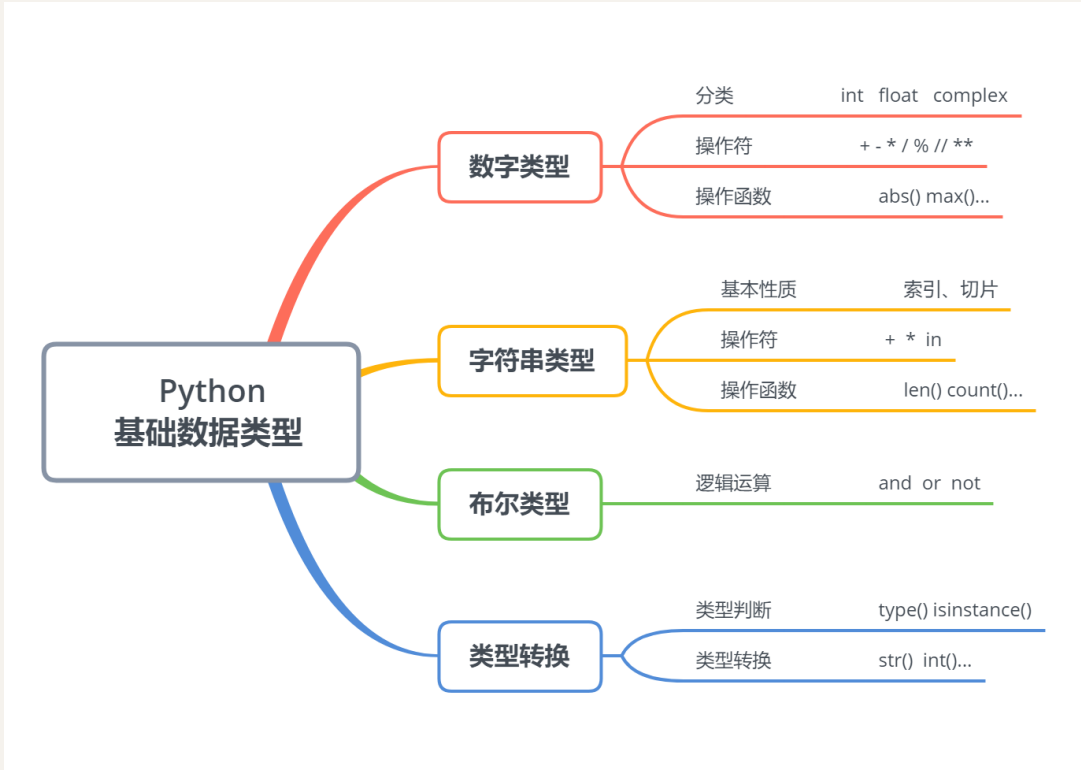
Sidney Zhang

<https://lyzhang.me>

++

# 第三章 基础数据

Python的数据类型包括数字（int、float、complex）、字符串（str）、列表（list）、元组（tuple）、字典（dict）、布尔（bool）和集合（set）。每种类型都有特定的用途和操作方法，适用于不同的编程需求。



# 基础中的基础内容（1）

++

## 变量

用于代表并存储一个数据的标识符。这个可以类比数学公式中的`x`、`y`、`f`这类变量来理解。

# 01

## 数据类型

就是指某一个变量，它所代表的数据是什么类型的，比如说，数字类型或者文本类型（请注意这两个类型不是准确的说法）。

# 02

## 输入输出

和计算机的互动方式，鼠标、键盘都是常用的输入设备，输出设备就是显示器。对于程序而言，输入输出就是字符和输出的文字/图像。

# 03



# 基础中的基础内容（2）

++

## 函数

一段代码组成的程序片段，可以用来实现某一些特定的目标。理解成数学函数也完全没什么问题。

# 04

## 错误处理

代码难免出差错，提前考虑会出现的错误，并针对性的处理就是错误处理。

# 05

## 库/添加包

也就是Package/包，通过包可以扩展程序的使用便利性，很多功能也可以基于此快速完成一些复杂的能力，你可以把它想象成好用的工具箱就可以。

# 06



# 数据类型

++

## 数字类型

Number-Type

数字、整数、小数等。

## 字符类型

String-Type

文字、文本等。

## 混合类型

Fixed-Type

或者叫复合类型，多种类型混合在一起的类型。



Sidney Zhang

<https://lyzhang.me>

# 数字类型

++

整数、有理数、实数这些常见的数学数字。根据实际情况，我们把这些分别归类为，整数类型/整型，以及浮点类型/非整型的两个基本类别。

简单分为3种：



## 整型 (Integer)

用于表示没有小数部分的数，可以是正数、负数或零。整型通常用于计数和索引。



## 浮点型 (Floating-point)

用于表示有小数部分的数。浮点型可以表示非常大或非常小的数值，但精度一般受限。



## 布尔型 (Boolean)

用于表示逻辑值，通常有两个值：真（`True`）和假（`False`）。这个更接近离散数学的逻辑运算所定义的数学类型。

++

# 字符类型

字、词、句，组成了我们通常认知的字符，可能还要包含标点符号、数字、一些特别的符号等等，这些在Python中都可以作为字符来看待。由于计算机发展的历史上，是从英语开始的，所以通常的字符就是英文字符，然后是其他西欧字符。最后才是咱们的中文以及其他亚洲语言字符（日语片假名、平假名，韩语字符等）。

Sidney Zhang

<https://lyzhang.me>

```
>>> "2"  
'2'  
>>> "a"  
'a'  
>>> "%"  
'%'  
>>> "长"  
'长'  
>>> "I"  
'I'  
>>> |
```

## 字符型 (Character)

用于表示单个字符，如字母、数字或特殊符号。

```
import this  
  
The Zen of Python, by Tim Peters  
  
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
Flat is better than nested.  
Sparse is better than dense.  
Readability counts.  
Special cases aren't special enough to break the rules.  
Although practicality beats purity.  
Errors should never pass silently.  
Unless explicitly silenced.  
In the face of ambiguity, refuse the temptation to guess.  
There should be one-- and preferably only one --obvious way  
to do it.  
Although that way may not be obvious at first unless you  
think very hard.  
Now is better than never.  
Although never is often better than *right* now.  
If the implementation is hard to explain, it's a bad idea.  
If the implementation is easy to explain, it may be a good idea.  
Namespaces are one honking great idea -- let's do more of those!
```

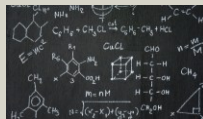
## 字符串型 (String)

用于表示一系列字符的集合，通常用于处理文本。

++

# 混合类型

列表	List	用于构建元素集合，可以用于存储不同类型的数据，且元素可变。
字典	Dict	一个无序的键值对集合（{K:V}），键必须是不可变类型，且每个键值对的键是唯一的。
函数	def	一段可重复使用的代码块（方法，计算逻辑，状态行为），它可以接受输入参数并返回结果。
类型	class	创建对象的蓝图，它定义了对象的属性和方法，实例化后形成具体的对象，也就是数值和方法的集合。



## 重点就是组合

由数字、字符等组合而成的那些类型，就是混合类型。混合类型也是进一步扩充数据类型的基础。

Sidney Zhang

<https://lyzhang.me>



++

# 练习题

1. 尝试理解`"text test spilt".spilt(' ')`的结果。
2. 写出一个数值、一段字符串和一个注释。



Sidney Zhang

<https://lyzhang.me>

```
# 导入库和必要方法
import re
import smtplib
from email.message import EmailMessage
from email.headerregistry import Address

# 发件人邮箱: 请填写公司邮箱
mail_sender = 'your.name@mail.mail'
# 发件人姓名: 可改可不改
mail_name = None
# 邮箱密码: 请填写你的密码或者授权码
mail_passwords = 'passwords'
# 邮箱SMTP: 如果使用QQ企业邮箱, 则无须修改
mail_SMTP = 'smtp.exmail.qq.com'
# 邮箱SMTP端口: 如果使用QQ企业邮箱, 则无须修改
mail_PORT = 465

# 格式化邮件地址
def addrFormat(mailaddr, disname = None):
    name = re.split("@|\\.", mailaddr)[1] if disname is None else disname
    return Address(display_name=name, addr_spec=mailaddr)

# 创建邮件
msg = EmailMessage()
msg['Subject'] = 'Hallo There'
msg['From'] = addrFormat(mail_sender, disname = mail_name)
msg['To'] = addrFormat(mail_sender, disname = mail_name)
msg['Cc'] = addrFormat("zly@lyzhang.me", disname = "张良益")
msg.set_content("""
Hi there!
Python安装成功了! 恭喜!

> From Python email !
""")

# 发送邮件
try:
    with smtplib.SMTP_SSL(host=mail_SMTP, port=mail_PORT) as servers:
        servers.ehlo("touchNow")
        servers.login(mail_sender, mail_passwords)
        servers.send_message(msg)
        print("Hello There! \n\t邮件发送成功啦!")
except smtplib.SMTPException as e:
    print(e)
```

指定需要运行的Python代码。

+ +

Python Beginner's Tutorial  
Part 2 – Basic Data Types

**THANK YOU**

Sidney Zhang

<https://lyzhang.me>