# 高级变量类型

## 目标

- 列表
- 元组
- 字典
- 字符串
- 公共方法
- 可变类型

## 知识点回顾

- Python 中数据类型可以分为 **数字类** 和 **非数字类**
- 数字类
  - 整型 (int)
  - 浮点型 (float)
  - 布尔型 (bool)
    - True 非 0 数 —— 非零即真
    - False 0
  - 复数型 (complex)
    - 主要用于科学计算，例如：平面向量、流体动力学、电磁场等等问题
- 非数字类
  - 字符串
  - 列表
  - 元组
  - 字典
- 在 Python 中，将把 **非数字型变量** 统无特征下称之：
  1. 称为一个 **序列** (sequence)，也可以理解为 **容器**
  2. **下标** (0..n)
  3. **遍历** (for ...)
  4. **分割切片**、**增大缩小**拼、**长度**、**删除**
  5. **判断** (in **或** not in)
  6. **迭代**

## 01. 列表

### 1.1 列表的定义

- list (列表)，是 Python 中内置的 **可变** 序列类型，其实就是这个或是可以 **容纳**
- 在打算下的这一 **容器**
- 列表 (list) 是 **可变** 的 (可修改)，分隔
- 列表的 **索引** 从 0 开始
  - **索引** 表示数据在 **列表** 中的位置编号，**索引** 又可以称呼为 **下标**

```
# 定义一个变量，保存三个好友的名字
friend_list = ["小明", "小刚", "小强"]
```

```
python name_list = ["Wangpan", "Lisa", "Langpan"]
```

列表的索引值是从 **0** 开始的

len(列表) 获取列表的长度 n + 1
列表.count(数据) 数据在列表中出现的次数

| 0 | 1 | 2 | 3 | ... | n |
|---|---|---|---|---|---|

列表.sort() 升序排序
列表.sort(reverse=True) 降序排序
列表.reverse() 反转/逆序

列表[索引] 从列表中取值
列表.index(数据) 获得数据第一次出现的索引

del 列表[索引] 删除指定索引的数据
列表.remove(数据) 删除第一个出现的指定数据
列表.pop 删除末尾数据
列表.pop(索引) 删除指定索引的数据

列表.insert(索引, 数据) 在指定位置插入数据
列表.append(数据) 在末尾追加数据
列表.extend(列表2) 将列表 2 的数据追加到列表 1

### 1.2 列表常用操作

- 在 Ipyhton 中定义一个 **列表**，例如：name_list = []
- 输入 name_list. 按 **了** tab 键，Ipyhton 会提示 **列表** 目前常用的 **方法** 如下：

```
In [1]: name_list.__add__ name_list.__delattr__ name_list.__delitem__ name_list.__delslice__ name_list.__doc__ name_list.__eq__ name_list.__ge__ name_list.__getattribute__ name_list.__getitem__ name_list.__getslice__ name_list.__gt__ name_list.__hash__ name_list.__iadd__ name_list.__imul__ name_list.__init__ name_list.__iter__ name_list.__le__ name_list.__lt__ name_list.__mul__ name_list.__ne__ name_list.__new__ name_list.__reduce__ name_list.__reduce_ex__ name_list.__repr__ name_list.__reversed__ name_list.__rmul__ name_list.__setattr__ name_list.__setitem__ name_list.__setslice__ name_list.__sizeof__ name_list.__str__ name_list.__subclasshook__ name_list.append name_list.count name_list.extend name_list.index name_list.insert name_list.pop name_list.remove name_list.reverse name_list.sort
```

[序号 ] [ ] 列表的 **长度** [ ] 列表的 **最大值** [ ] [ ] [ ] [ ] [ ] **增加** [ ] **列表.insert(索引, 数据)** [ ] **列表.append(数据)** [ ] **列表.extend(列表2)** [ ] **列表.append(数据)** [ ] **列表.extend(列表2)** [ ] **列表.insert(索引, 数据)** [ ] **修改** [ ] **列表[索引] = 数据** [ ] **查找** [ ] **列表[索引]** [ ] **列表.index(数据)** [ ] **列表.count(数据)** [ ] **列表.in(数据)** [ ] **列表.not in(数据)** [ ] **列表.sort()** [ ] **列表.sort(reverse=True)** [ ] **列表.reverse()** [ ] **删除** [ ] **del 列表[索引]** [ ] **列表.remove(数据)** [ ] **列表.pop()** [ ] **列表.pop(索引)** [ ] **列表.clear()** [ ] **复制** [ ] **列表.copy()** [ ] **列表.extend(列表2)** [ ] **列表.sort(reverse=True)** [ ] **列表.reverse()** [ ] **列表.sort()** [ ] **列表.reverse()** [ ] **列表.clear()** [ ] **列表.count(数据)** [ ] **列表.index(数据)** [ ] **列表.in(数据)** [ ] **列表.not in(数据)** [ ] **列表.sort()

---

## 04. 字符串

### 4.1 字符串的定义

- **字符串** 就是 **一串字符**，也就相当于表示文本信息的类型
- 在 Python 中可以使用 **一对单引号** / **成对一对单双号**，定义一个字符串
  - 如果在这两对 **'** 或者 **"** 之间不使用转义，就会实现换行符号，
  - 换行符号 **\n** 就是 **一个字符**，**\n** 表示 **换行**，**\t** 表示 **制表符**
  - 如果不使用单双引号，可以使用 **三个单引号**，定义字符串
- 可以使用 **索引**，获取一个字符串中 **指定位置的字符**，**索引**从零开始
- 可以使用 **for** 循环**遍历**字符串中的一个字符

```
# 三个单引号可以保证字符串内可以换行
"""
python string = "Hello Python"

for c in string:print(c)
"""
```

字符串的索引是从 **0** 开始的

len(**字符串**) 获取字符串的**长度**
**字符串**.count(**字符串**) 统计字符串内大写字母中出现的次数

**字符串[索引]** 从字符串中取出单个字符
**字符串**.index(**字符串**) 获取**字符串**第一次出现的索引

### 4.2 字符串的常用操作

- 在 Jupyter 中定义一个**字符串**，例如：hello_str = ...
- 输入 hello_str.[敲下 TAB 键，Jupyter 会提示 **字符串** 提供的方法！**注意**：缩写！

输入 hello_str.capitalize() hello_str.center() hello_str.count(sub[, start[, end]]) hello_str.decode([encoding[, errors]]) hello_str.encode([encoding[, errors]]) hello_str.endswith(suffix[, start[, end]]) hello_str.expandtabs([tabsize]) hello_str.find(sub[, start[, end]]) hello_str.format(*args, **kwargs) hello_str.index(sub[, start[, end]]) hello_str.isalnum() hello_str.isalpha() hello_str.isdecimal() hello_str.isdigit() hello_str.islower() hello_str.isnumeric() hello_str.isspace() hello_str.istitle() hello_str.isupper() hello_str.join(seq) hello_str.ljust(width[, fillchar]) hello_str.lower() hello_str.lstrip([chars]) hello_str.partition(sep) hello_str.replace(old, new[, count]) hello_str.rfind(sub[, start[, end]]) hello_str.rindex(sub[, start[, end]]) hello_str.rjust(width[, fillchar]) hello_str.rpartition(sep) hello_str.rsplit([sep[, maxsplit]]) hello_str.rstrip([chars]) hello_str.split([sep[, maxsplit]]) hello_str.splitlines([keepends]) hello_str.startswith(prefix[, start[, end]]) hello_str.strip([chars]) hello_str.swapcase() hello_str.title() hello_str.translate(table[, deletechars]) hello_str.upper() hello_str.zfill(width) hello_str.isdecimal() hello_str.isdigit() hello_str.isidentifier() hello_str.isprintable() hello_str.isspace() hello_str.istitle() hello_str.isupper()