

数据容器

学习目的

掌握 Python 语言中的基本数据类型

1. 布尔型(bool)
2. 数值型(int, float)
3. 字符串(str)
4. 列表(list)
5. 元组(tuple)
6. 字典(dict)
7. 集合(set)

自学任务

```
text = '''
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 ambxiguity, 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're Dutch.
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!
'''
```

1. 字符串的基本处理

1. 创建一个名为 `d5_exercise_string.py` 的文件
2. 将字符串样本 `text` 里英文单词中包含 `ea` 的英文单词剔除
3. 将 `better` 全部替换成 `worse`
4. 大写字母转成小写，小写字母转成大写
5. 将所有单词按 `a...z` 升序排列
6. 最后输出结果

2. 统计字符串样本中英文单词出现的次数

1. 创建一个名为 `d5_exercise_stats_text.py` 的文件
2. 使用字典 (dict) 统计字符串样本 `text` 中各个英文单词出现的次数。
3. 示例: `{ 'is': 10, 'better': 9, ... }`
4. 按照出现次数从大到小输出所有的单词及出现的次数
5. 只统计英文单词，不包括非英文字符的其他任何符号，如连接符号、空白字符等

3. 数组操作，进制转换

1. 创建一个名为 `d5_exercise_array.py` 的作业文件
2. 将数组 `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]` 翻转
3. 翻转后的数组拼接成字符串
4. 用字符串切片的方式取出第三到第八个字符 (包含第三和第八个字符)
5. 将获得的字符串进行反转
6. 将结果转换为 `int` 类型
7. 分别转换成二进制，八进制，十六进制
8. 最后输出三种进制的结果

4. 提交作业

1. 将本地仓库关于本次作业的变更提交为一个 `commit`
2. 通过 Github 桌面客户端将本地电脑的变更推送到自己账户下的作业仓库
3. 回到 Github 页面，在自己账户下的作业仓库页面向远程公用作业仓库的 `master` 分支发起 Pull Request, 在提交的 Pull Request 中 @自己的教练 提醒他检查作业

参考资料

1. 数据结构, [英文](#), [中文](#)
2. 内置类型, [英文](#), [中文](#)
3. 内置常量, [英文](#), [中文](#)