



复习

车万翔

哈尔滨工业大学



❖ 高级语言程序设计-I

- 注重培养解决问题的能力，而不是Python语言本身

❖ 主要内容

- 计算机和程序的基础知识
- 数据类型、运算符与表达式、变量赋值与简单I/O
- 控制结构
 - 选择、循环
- 函数与递归函数
- 字符串
- 内建数据结构
 - 列表、元组、字典、集合
- 简单算法
- GUI、数据库、Web编程



计算机和程序的基础知识



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ 谁被誉为“计算机之父”？
- ❖ 程序（Program）的基本概念
- ❖ 高级语言和低级语言的区别是？
- ❖ 解释型和编译型语言的区别是？



- ❖ 基本数据类型有哪些？
- ❖ 为什么要区分数据类型？
- ❖ 类型之间能相互转换吗？
- ❖ 赋值语句的含义是什么？
- ❖ 标示符的命名规则是什么？

```
>>> import keyword
>>> keyword.kwlist
['and', 'as', 'assert', 'break', 'class',
 'continue', 'def', 'del', 'elif', 'else',
 'except', 'exec', 'finally', 'for', 'from',
 'global', 'if', 'import', 'in', 'is',
 'lambda', 'not', 'or', 'pass', 'print',
 'raise', 'return', 'try', 'while', 'with',
 'yield']
```



❖ 算术运算符

- 整数相除的结果仍然是整数

❖ 关系运算符

- 浮点数如何进行相等关系判断？

❖ 逻辑运算符

- 123 and 456 的结果是？
- 123 or 456 的结果是？

❖ 运算优先级

❖ 常用的数学函数



控制结构



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ 顺序、选择和循环结构
 - 所有程序都可由这三种结构实现的
- ❖ 能读会画流程图



- ❖ 函数功能
 - 代码重用与简化
 - 分而治之与信息隐藏
 - 自顶向下、逐步求精的模块化程序设计
- ❖ 函数的定义和调用
 - 形式参数、实际参数、返回值
- ❖ 变量的作用域
- ❖ 递归函数
 - 递归函数的基本模式
 - 递归函数的优缺点



字符串



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ 如何定义字符串？
 - 都学习了哪些转义字符？
- ❖ 字符串索引与切片
 - 索引时注意不要越界
 - 切片下标能越界么？
- ❖ 字符串提供了哪些常用方法？
- ❖ 字符串如何比较大小？
- ❖ 简单正则表达式



❖ 列表

- 与字符串有哪些异同？
- 函数参数传递列表到底发生了什么？

❖ 元组

- 为什么引入元组？

❖ 集合与字典

- 二者的区别和联系
- 键的类型以及无序性

❖ 数据结构的比较与选择

❖ 数据结构的综合利用



❖ 马尔科夫 (Markov) 分析

- 给定一系列单词，下一个单词出现的概率

❖ 如马丁路德金的演讲：I have a dream

- *I have a dream that one day this nation will ...*
- *I have a dream that one day on the red hills ...*
- *I have a dream that one day even the state ...*

- "have a" 后面总是跟着 "dream"

- "one day" 后面可能跟 "this", "on" 或 "even"

❖ 给定任意 n 个单词作为前缀，然后随机生成下一个单词，新的单词和前面 $n-1$ 个单词组成新的前缀，并重复以上过程，生成一段随机文本



❖ 任意 $n-1$ 个连续单词，获得其曾经出现过的所有后缀单词（列表）

```
def create_markov_dict(n):  
    f = open('emma.txt')  
    markov_dict = {}  
    prefix = ('',) * n  
    for line in f:  
        words = line.strip().split()  
        for suffix in words:  
            if prefix in markov_dict:  
                markov_dict[prefix].append(suffix)  
            else:  
                markov_dict[prefix] = [suffix]  
            prefix = prefix[1:] + (suffix,)  
    return markov_dict
```



❖ 根据输入的 n 个单词，生成一段随机文本

```
def markov_analysis(start, markov_dict, word_num):  
    markov_lst = list(start)  
    prefix = start  
    for i in range(word_num):  
        suffix = random.choice(markov_dict[prefix])  
        markov_lst.append(suffix)  
        prefix = prefix[1:] + (suffix,)  
    return markov_lst  
markov_dict = create_markov_dict(2)  
print markov_analysis(('I', 'have'), markov_dict, 100)
```



数据结构对比



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

	string	list	tuple	set	dict
Mutable					
Sequential					
Sortable					
Slicable					
Index/key type					
Item/value type					
Search complexity					



数据结构对比



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

	string	list	tuple	set	dict
Mutable	No	Yes	No	Yes	Yes
Sequential	Yes	Yes	Yes	No	No
Sortable	No	Yes	No	No	No
Slicable	Yes	Yes	Yes	No	No
Index/key type	Int	Int	Int	Immut	Immut
Item/value type	Char	Any	Any	No	Any
Search complexity	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$



- ❖ 什么是算法？
- ❖ 查找：线性查找、二分查找
- ❖ 排序：选择排序、冒泡排序
- ❖ 不同数据结构内建的查找与排序方法
- ❖ 时间和空间复杂度概念



- ❖ 算法描述如何解决问题
 - 关注通用方法
- ❖ 程序是用特定编程语言实现算法
 - 需要考虑实现的细节
- ❖ 什么是好的算法？
 - 正确及鲁棒 (robustness)
 - 细节描述适当
 - 高效
 - 行为明确
 - 具有一定的通用性



什么是好的程序？



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

❖ 可读性

- *Programs must be written for people to read, and only incidentally for machines to execute.*

---- Abelson & Sussman

❖ 鲁棒性 (Robustness)

- 能够处理异常情况

❖ 正确性

- 很难保证
- 充分的测试



❖ GUI

- Matplotlib进行数据展示
- 基本Tkinter组件
- 事件与事件处理
- 简单动画编程

❖ 数据库

- 基本概念
- 简单SQL操作

❖ 基于Django的简单Web应用开发



- ❖ 集中精力
- ❖ 理解问题，具化问题
- ❖ 编程前先思考
- ❖ 实验部分代码
- ❖ 分而治之 (Divide and Conquer)
 - 自底向上
 - 自顶向下
- ❖ 停下来思考，勇于舍弃
- ❖ 放松一下



- ❖ 法则 1
 - 编程前先思考
- ❖ 法则 2
 - 程序是供人阅读的解决问题的文档
- ❖ 法则 3
 - 提高编程能力的最好途径是“练习”！
- ❖ 法则 4
 - 经常并仔细地测试你的代码！
- ❖ 法则 5
 - 难写的地方也会难读，增加一个注释
- ❖ 法则 6
 - 不要盲从任何法则



❖ 占总成绩50分

❖ 时间

- 2015年12月13日（星期日）
- 10:00 – 12:00

❖ 地点

- 根据各班课表

❖ 形式

- 开卷笔试
- 可以携带任何纸质材料
- 不能携带电子设备



- ❖ 关于赋值语句的作用，正确的描述是：（ ）
 - A. 将变量绑定到对象 B. 每个赋值语句只能给一个变量赋值
 - C. 将变量改写为新的值 D. 变量和对象必须类型相同
- ❖ 关于数据类型，下列说法正确的是：（ ）
 - A. 决定数据存储内存的大小 B. 决定以何种方式输出数据
 - C. 决定数据可以执行的运算 D. Python语言不区分数据类型
- ❖ 执行以下两条语句后，lst的结果是：（ ）
lst = [1, 2, 3]
lst.append(lst)
 - A. [1, 2, 3, lst] B. [1, 2, 3, [1, 2, 3]]
 - C. [1, 2, 3, [...]]，其中“...”表示无穷递归 D. 抛出异常



❖ 改错题

```
x = raw_input('Input: ')

low = 0

high = x

ans = (low + high) / 2

while ans**2 != x:

    if ans**2 < x:

        low = ans + 1

    else:

        high = ans - 1

    ans = (low + high) / 2

print ans
```



❖ 输出结果题

3. 写出下面程序的输出结果

```
def fun(lst):  
    lst = [4, 5, 6]
```

```
lst = [1, 2, 3]  
fun(lst)  
print lst
```

4. 写出下面程序的输出结果

```
d1 = {}  
d1[2] = 10  
d1['2'] = 20
```

```
d2 = {}  
d2[2] = d1  
d2['2'] = d2
```

```
print d2['2']['2']['2']['2'][2][2]
```




- ❖ 假设丢失了单词的最后一个字母，例如单词 “fantasti”，需要从单词表中找到原单词 “fantasic”。请分别使用两种方法（不同的数据结构和算法）查找原单词。注意，在某些情况下，可能会找到多个单词。例如，“bas”，原单词可能为 “bass” 或 “base”，需要找到给定单词表中的全部原单词。其中，单词表存放在 “words.dic” 文件中。最后，请对这两种实现方法的效率加以比较。



敢问路在何方？



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ C/C++、Java、...
- ❖ 面向对象编程 (Object Oriented Programming, OOP)
- ❖ 函数式编程 (Functional Programming)
- ❖ 如何学好编程？
 - Learning language programming by writing program!





谢谢！
预祝大家取得好成绩！