

Python Talk 6



这个指引文档在 [知识共享署名-相同方式共享 3.0 协议](#) 之条款下提供

Available under [Creative Commons Attribution-ShareAlike License](#)

复习

1. 用**for**和**print**打印

▶ 1 2 4 8 16 32 64

2. 使用**input**获取输入字符串并判断其是否含'HCC'

3. 这些函数有什么作用？

▶ all min any id hash

4. 让命令行执行

```
ping 10.60.0.0 -c 1  
ping 10.60.0.1 -c 1  
ping 10.60.0.2 -c 1  
...  
ping 10.60.0.255 -c 1
```



排序

► 如何排序这个数列？

► `a = [3, 1, 4, 2]`

► 答案

► `a.sort()`

► 提高：如何反向排序？

手动实现（选择排序）

```
b = []  
for i in range(len(a)) :  
    m = 0  
    for j in range(len(a)) :  
        if a[m] > a[j] :  
            m = j  
    b.append(a[m])  
    del(a[m])  
a = b
```



函数

- ▶ 函数相当于通过参数计算出返回值
- ▶ 定义

```
def 函数名(参数 1, 参数 2, ...) :  
    语句  
    return 返回值
```

- ▶ 调用
函数名(参数 1, 参数 2, ...)
- ▶ 简便定义形式 (lambda 表达式)
函数名 = **lambda** 参数... : 返回值



特殊排序

- ▶ 如何按照 tuple 的第二位排序?

- ▶ `a = [(1, 2), (4, 9), (6, 7)]`

- ▶ 方法 1

- ▶ `a.sort(key = lambda x: x[1])`

- ▶ 方法 2



```
def f(a) :  
    return a[1]  
a.sort(key=f)
```



字符串编码问题

▶ 方法

```
a = '你好'
```

```
b = a.encode('编码') # 将字符串变成二进制码
```

```
c = b.decode('编码') # 将字节变成字符串
```

▶ 常见编码

- ▶ UTF-8 （多语言通用）

- ▶ GBK （中文）

- ▶ GB2312 （中文）



进制转换

► 从十进制转换

```
bin(1024)      # 二进制
oct(1024)      # 八进制
hex(1024)      # 十六进制
```

► 转换为十进制

```
int('10', 2)   # 二进制的 10
int('0b10', 2) # 二进制的 10
int('13', 7)   # 七进制的 13
```

► 错误用法

```
int('0b10')    # 必须注明进制数
int('0x10', 2) # 输入和注明进制数不匹配
```



时间

► 关于时间的库

```
from datetime import datetime  
a = datetime.now()  
b = datetime(年, 月, 日, ...) # 输入整数  
b = datetime(2000, 8, 24)    # 例  
datetime.fromtimestamp(时间值) # 数值转为时间  
a.timestamp()                # 时间转为数值  
a - b                         # 计算时间差  
a.strftime('%H:%M')          # 输出为特定格式
```



练习

1. 模拟网络攻击检测

- ▶ 假设某个用户上一次访问我方网站的时间已被记录为 a
- ▶ 我们希望判断用户两次访问的间隔是否大于一秒
- ▶ 如果目前访问网站的时间在 a 后一秒之前，则输出“攻击”
- ▶ 否则输出“正常”

2. 将时间输出为中文

- ▶ 例如对于

```
a = datetime(2000, 8, 24, 12, 34, 56)
```

- ▶ 输出“2000 年 8 月 24 日 12 时 34 分 56 秒”
- ▶ 提示：尝试 `a.strftime('%Y %m %d %H:%M:%S')`



感谢参加此次活动

