

Python Talk 2



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

▶ 计算下列数值

- ▶ 一百加二十
- ▶ 一天有多少秒
- ▶ 真或假
- ▶ 三的一百次方

▶ 答案

- ▶ $100 + 20$
- ▶ $24 * 60 * 60$
- ▶ `True or False`
- ▶ $3 ** 100$



复杂数据类型

▶ 常用类型

▶ 元组: `tuple`

▶ 列表: `list`

▶ 字符串: `str`

▶ 字典: `dict`

▶ 其他类型

▶ 集合: `set`

▶ 举例

▶ `(1, 2, 3)`

▶ `('h', 'c', 'c')`

▶ `"HCC, I'm. "`

▶ `{1: 'H', 2: 'C'}`

▶ 举例

▶ `{1, 2, 3}`



创建你自己的复杂数据

- ▶ (a, b, c)
- ▶ [a, b, c]
- ▶ 'Some_Letters'
- ▶
{
 key_A: value_A,
 key_B: value_B
}

- ▶ 所有元素都可以换成更复杂的类型
- ▶ 基本类型
 - ▶ int float bool
- ▶ 复杂类型
 - ▶ tuple list str
 - ▶ dict set
- ▶ type



细节

- ▶ 创建一个元素的 `tuple`

```
('HCC')      # 失败
```

```
('HCC',)     # 正确
```

- ▶ `list`和`dict`也支持

- ▶ `str`创建方式

- ▶ 单引号

```
'HCC, I\'m. '
```

- ▶ 双引号

```
"HCC, I'm. "
```

- ▶ 多行字符串

```
'''  
HCC  
'''
```



tuple和list有什么区别

```
a = ()      # tuple
```

```
b = []      # list
```

```
help(a)
```

```
help(b)
```

```
# 你能找到区别吗?
```

▶ insert

▶ pop

▶ remove

▶ reverse

▶ sort



尝试基本运算符

► 加法

```
(1, 2) + (5, 6)
[1, 2] + [5, 6]
'HCC' + ", I'm."
```

► 乘法

```
(1, 2) * 3
'HCC' * 10
```

► 合并dict

```
{1: 2} + {3: 4}           # 失败
a = {1: 2}
a.update({3: 4})
print(a)                 # 正确
```



字符串方法

"HCC, I'm. ".upper()	# 大写
"HCC, I'm. ".lower()	# 小写
"HCC, I'm. ".split()	# 按空白字符分割
len("HCC, I'm. ")	# 得到长度
'&'.join([1, 2])	# 合并
'''Hello from HCC: Welcome!	
'''.split('\n')	# 按换行符分割



分割

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9]
a[4]
a[4: 7]
a[2: -2: 2]
a = 'abcdefghijklmnopqrstuvwxyz'
# 重复上面三行
{'a': 'b', 'b': 'c'}['b']
```



原理

	H	C	C	,	I	'	m	.
0	1	2	3	4	5	6	7	8
-8	-7	-6	-5	-4	-3	-2	-1	



练习

- ▶ 对于这些字符串，截取出用户名（'@'前面）

'HCC@shiyiquan.net'

'lxy@shiyiquan.net'

'mbl@shiyiquan.net'

'shiyiquan@shiyiquan.net'

- ▶ 你能使用两种方式吗？提示：分割和 split
- ▶ 哪种方式还可以正确截取这些字符串？

'HCC@hcc.io'

'lxy@shierquan.tk'



感谢参加此次活动

