



## 第二章 python编程基础 复习

# 目录

- 变量
  - 赋值
  - 数据类型
    - 数字
    - 字符串
    - 布尔值
    - 列表
    - 元组
    - 字典
- 操作符、语句
  - 算术与比较操作符
  - 条件语句
  - 循环语句

## ◆ 函数

1. 文件操作
2. 函数定义
3. 类定义

## ◆ Numpy

1. 数组
  - 访问
  - 运算
  - 操作函数

2. 矩阵

## ◆ Matplotlib

# 变量

- 赋值

大小写敏感

```
>>> pptName = " Python入门"  
>>> height = 1.71  
>>> n = height  
>>> n *= 100
```

## ◆ 数值（单个）

- 数字: int long bool float complex
- 字符串
  - 转义字符: 需要特殊处理的字符

- 布尔值: and or not 三种运算

**[]**索引操作符  
**[:]**截取操作符  
**+**拼接运算  
**\***重复运算  
**split()**切割  
**find()**查找

# 变量

## ◆ 数值（单个元素）

- 数字
- 字符串
- 布尔值

## • 数值（多个元素）

- 列表(list)
- 元组(tuple)
- 字典(dictionary)

```
>>> A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
>>> B = (' Python' , ' is' , ' cool' )
>>> C= {'name':'LiShuhao', 'height':171, 'age':26}
```

哪个是列表、元组、字典？

# 变量

- 数值 (多个元素)

- 列表(list)

```
创建 word=['a','b','c','d','e','f','g']  
访问 c=word[2]  
切片 ['a','b']=word[:2]  
相加 ['c','f']=word[2]+word[-2]  
扩展  
word1.extend(word2)=['a','b','c',]  
追加append()  
插入insert()  
删除pop() del() remove() clear()
```

- 元组(tuple)

只读

- 字典(dictionary)

由键值对(Key-Value)构成

# 操作符与语句

- 算术操作符
  - `:` `+` `-` `*` `/` `//`取整 `%`取余 `**`乘方
- 比较操作符
  - `<` `<=` `>` `>=` `==` `!=` `<>`

- 条件语句:

`if` `elif` `else`

- 循环语句

- `while`

- `for`

# 条件语句和循环语句

## ◆ if elif else

- if expression1:
- if\_block
- elif expression2:
- elif\_block
- else expression3
- else\_block

## ◆ while

- while expression:
- while\_block

## ◆ for .. in

- for nItem in range(5):
- print(nItem)

如何跳出循环？

break

continue

# 函数

## ◆ 文件操作函数

- `open()` `file()` `readlines()` `close()`

## ◆ 函数的定义

- `def function_name([arguments`
- `function_body`
- `return fun_obj`

## ◆ 类的定义

- `class className(baseClass):`
- `def __init__(self):`
- `self.变量名1 = 值1`
- `self.__变量名2 = 值2`
- `def 方法名1(self):`
- `function_body`

**区分公有、私有？**



# Numpy

## ◆ 数组

- 索引和花式索引
- 运算
  - 四则运算
  - 比较运算 (`==`、`!=`、`<=`、`>=`、`>`、`<`)
  - 布尔运算 (`and`、`or`、`not`)

- 操作函数

```
reshape()  
concatenate()  
where()  
ufunc()
```

```
ravel()  
sort()  
extract()
```

## ◆ 矩阵

- `numpy.matlibs`
- 包含行 (`row`) 和列 (`column`)
- 可以使用线性代数运算

# Matplotlib

- 语句
  - `plt.plot(x, y, format_string, **kwargs)`
  - `plt.bin(...)`
  - `plt.scater(...)`
- `x` : X轴数据, 列表或数组, 可选
- `y` : Y轴数据, 列表或数组
- `format_string`: 控制曲线的格式字符串, 可选