



# python web框架企业实战详解 第1周

**【声明】** 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>

# 第1课：Python基础知识

课程内容：

- python介绍
- python安装
- python基础语法
- python基础数据结构
- 查看python函数帮助

本次所讲的是python的一些基本知识，毕竟后面要使用python来进行相关的项目开发，一些必备的python知识还是需要提前了解和掌握的，课前已经上传了《python简明教程》学习资料，本次课所讲的内容主要就是基于该资料之上，如果之前没有学习过python的，就跟着我一起进入python的世界吧！

## ■ python历史

之所以有python的出现，是因为Python的作者(Guido von Rossum)在学习和使用其他语言（如：C/C++、Pascal、shell等）之后，希望能够得到一个既能够像C语言那样，能够全面调用计算机的功能接口，又可以像shell那样可以轻松的编程的语言。于是1991年，第一个Python编译器(同时也是解释器)诞生。它是用C语言实现的，并能够调用C库(.so文件)。从一出生，Python已经具有了：类(class)，函数(function)，异常处理(exception)，包括表(list)和字典(dictionary)在内的核心数据类型，以及模块(module)为基础的拓展系统。

Python崇尚优美、清晰、简单，是一个优秀并广泛使用的语言；是一个有着自己哲学的语言。有那么一句关于python的座右铭：人生苦短，我用python。可见使用python编程将是一件多么有趣的事情！

## ■ python特性

关于python的特性有很多，这里我们只选取几个比较有特点的来列出来：

- 动态性
- 面向对象
- 可扩展性
- 可嵌入性
- 跨平台性
- 强制类型转换
- 丰富的第三方类库

# 第1节：Python介绍

## ■ Python优点

同python的特性一样样的，它的优点如下：

- 简单易学
- 高效编码
- 应用广泛
- 平台无关

## ■ python版本选择

到目前为止，python有2个大的版本；一个是2.X，一个是3.X。3.X的python对于2.X的语法并不是完全兼容的，所以这2个版本之间的使用还是有一些差别的。

关于2.X的遗留代码转换到3.X版本，python官方也提供了相应的工具；不过目前来看市场上使用2.X的python还是相对较多，很多第三方的类库也是支持的2.X的版本。

因为上述原因，所以我们后面使用到的也将是以2.X的python为学习版本。

## 第2节：Python环境安装

### ■ 安装包下载

Python的基础环境安装有以下文件需要下载：

- Python2.7安装包 ( <https://www.python.org/ftp/python/2.7.9/python-2.7.9.msi> )
- Python SetupTools基础包 ( [https://bootstrap.pypa.io/ez\\_setup.py](https://bootstrap.pypa.io/ez_setup.py) )
- Python pip包管理工具 ( <https://pypi.python.org/packages/source/p/pip/pip-6.0.6.tar.gz> )

**注意：**更多其他python的第三方包，请查看pypi官网。 <https://pypi.python.org>



## 第2节：Python环境安装

### ■ python安装

- 直接双击[python-2.7.9.msi](#)进行正常安装即可
- 在path环境中添加python安装目录
- 启动cmd并运行python测试

### ■ setuptools安装

- 启动cmd并切换到ez\_setup.py所在目录
- 使用命令：`python ez_setup.py install`进行安装

## 第2节：Python环境安装

### ■ pip安装

- 解压[pip-6.0.6.tar.gz](#)包
- 启动cmd并切换到解压目录（有setup.py文件的目录）
- 使用命令：python setup install进行安装
- 在path变量中添加python script目录路径（如：C:\Python27\Scripts）
- 启动cmd并使用pip list命令测试

## 第2节：Python环境安装

### ■ Python编辑器

IDLE

Notepad++

记事本

### ■ 语法格式

Python的语法格式与其它语言有所不同，它不是以花括弧、分号之类的符号来分割代码；而已以TAB或空格缩进来规定语法的。

这样的好处呢！就是保证了代码结构的简明和规整；当然与此同时也增加了一些容易忽视的细节调试；比如空格和TAB混合使用，空格缩进数量不一致，编辑器对TAB的兼容性等等问题

一个简单的python语法：

```
def foo():
```

```
    print "Hello World"
```

同一个缩进块中的内容，就相当于C语言里同一个花括弧中内容的意思

## 第3节：Python基础语法

### ■ 变量定义与赋值

由于python是解释性语言且具有强类型转换的能力，所以python中变量不需要申明变量名与变量类型；而是直接给变量赋值即可，如：

```
a = 4 , b= 'a' , c=True, d=None
```

另外python是大小写区分的，变量a与变量A不是一回事

虽然上面的变量都没有指定类型，但是python已经自动在对其赋值的过程中就已经把它们类型确定了；我们可以通过使用type方法来测试一下上面每个变量的类型。如：

```
type(a) → int
```

```
type(c) → bool
```

## 第3节：Python基础语法

### ■ 分支/判断、

Python的分支只有if...elif...else形式，没有提供switch...case形式的；其主要的条件表达式符号有：

==, !=, >, <, >=, <=, is, not is等等; 如：

```
if a==b:
```

```
    print 'ok'
```

```
else:
```

```
    print 'no equal'
```

另外你还可以直接放置一个变量作为分支判断的条件，在python中只要是非None类型的都认为是

True；其中None类型有：

None, False, 0, "", (), [], {} 等

## 第3节：Python基础语法

### ■ 循环

Python的语法中，循环有for，while两种；for用于遍历循环，while用于条件循环。其中for循环的语法如下：

```
for i in iterators:
```

```
    ##do something for i
```

也就是说for循环只接收一个迭代器作为输入，并且去遍历这个迭代器，每次返回一个当前迭代内容，而不支持其他样式的遍历；所以如果我们想要遍历数字0..9的话，就得先生成一个0到9的迭代器才行，幸运的是python已经内置了这样一个函数range、xrange。如：

```
for i in range(10): ##生成一个0到9的一个数组作为迭代器传入进去
```

```
    print i    ##break, continue
```

## 第3节：Python基础语法

### ■ 循环

while语法的格式形如：

while condition：

```
##do something you want
```

这个语法和其它语言的基本保持一致，如：

```
a = 1
```

```
while True:
```

```
    print a
```

```
    a += 1
```



## 第3节：Python基础语法

### ■ 函数定义

Python里面进行函数定义的格式如下：

```
def fname(args,...):
```

```
    ##do something
```

调用函数的格式为：

```
fname(arg1,..)
```

关于函数的参数相关的高级使用将在下节课中讲到，这里我们只要掌握基本的函数创建方法和调用即可

### ■ Lambda表达式

之所以叫它表达式是因为它只支持一条返回语句内容，没有换行的概念；其功能其实就是一个简易版的函数，有匿名函数的意思；主要用于那些只接收函数接口但是没有复杂业务逻辑的情景下，这样方便我们快速创建一个临时函数，并在使用之后自动释放。其形式如：

```
lambda x: x*2
```

其功能等同于函数如下：

```
def foo(x):  
    return x*2
```

**区别**：lambda表达式不需要显式的return，没有函数名，只能有一条表达式内容，没有赋值给变量则会自动被回收

## 第3节：Python基础语法

### ■ 类定义

Python里模块通常指的是类，其形式如下：

```
class cname(object):  
    def __init__(self, arg):  
        ##do init  
  
    def __del__(self):  
        ##  
  
    def foo(self, arg):  
        ##
```

其实例形式：cn = cname(arg)

### ■ 元组

Python里面自带了很多的数据类型，这些数据类都是平常工作中经常使用到的，最主要的是python给予它们定义了非常丰富的易用函数支持，这样我们在平常大量的类似工作中一下子就节省了很多的时间；这也是我喜欢python的一个原因之一。

**元组**：用元括弧括起来的一组元素集合。其特点是内容不可变，即一旦定义其长度和内容都是固定的；类似于C语言的数组。如：

```
t = (0, True, 'a' )
```

上面就定义了一个包含3个子元素的元组列表，访问元组成员的格式是使用成员下标，如：

```
print t(1) → True
```

## 第4节：Python基础数据结构

### ■ 列表

Python里面的列表同样是非常会被使用到的数据结构；相比于元组，列表的长度和内容都是可以动态改变的；可以理解为java中的链表数组。

**列表**：由中括弧括起来的包含一组元素的集合；其特点是长度和内容都可以改变。定义如下：

```
L = [0, False, 'l']
```

列表可以进行如下操作：

添加元素：L.append(1)

删除元素：del L[0]

插入元素：L.insert(2, 3)

修改元素：L[1] = True

## 第4节：Python基础数据结构

### ■ 字典

Python中的字典和其它语言的字典是一个意思，是对hashmap的实现；其语法定义为：

```
d = { 'k1' : 'k' , 'k2' : 'k2' }
```

字典可以有如下操作：

增加键值对：d[ 'k3' ] = 3

删除键值对：del d[ 'k' ]

修改键值对：d[ 'k2' ] = True

获取键值：d[ 'k1' ]

## 第5节：查看python帮助

### ■ Python查看帮助的方式

Python还有一个比较好的地方是它自身就提供了很好的帮助系统，除了安装python程序是自带的帮助手册外，在python解释器中也提供了动态查询对象信息的方法。

对于一个给定的对象obj，我们不知道它有哪些方法，怎么查询？

`dir(obj)` ##就会打印出obj对象的所有属性和方法

在了解了对象obj支持的方法后，不知道如何传参，怎么办？

`help(obj.method1)` ##会打印出method1方法的调用参数、功能说明等等

忘记了python有哪些内建函数了，怎么办？

`dir(__builtins__)` ##直接查询python自带的内建函数、常量、对象

`help('modules')` ##查看python的内建模块

# 第1课：Python基础知识

要点回顾：

- Python环境安装：注意版本和环境变量的配置
- python语法学习：熟练掌握基本的python语法知识
- python帮助查看：通过dir、help命令灵活的获取python对象的帮助信息



- Dataguru ( 炼数成金 ) 是专业数据分析网站 , 提供教育 , 媒体 , 内容 , 社区 , 出版 , 数据分析业务等服务。我们的课程采用新兴的互联网教育形式 , 独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围 , 重竞争压力的特点 , 同时又发挥互联网的威力打破时空限制 , 把天南地北志同道合的朋友组织在一起交流学习 , 使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本 , 直线下降至百元范围 , 造福大众。我们的目标是 : 低成本传播高价值知识 , 构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情 , 请看我们的培训网站 <http://edu.dataguru.cn>

# Thanks

**FAQ时间**